

# D-1 100분만에 배워보는 Windows Universal App 개발

A black and white historical photograph showing soldiers during a beach landing. In the foreground, a soldier on the left is looking towards the camera, while another soldier on the right is looking forward. In the background, many soldiers are wading through the water, carrying equipment. A large ship is visible in the distance.

김영욱 Evangelist  
Yowkim@Microsoft.com

# Agenda

1. UWP
2. XAML Controls
3. Networking
4. Linq
5. Adaptive UI
6. Data Binding
7. SQLite Local Database
8. Toast

# Introducing the UWP



Phone



Phablet



Small Tablet



Large Tablet



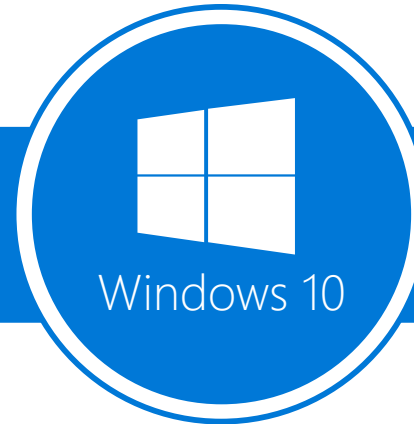
2-in-1s  
(Tablet or Laptop)



Classic  
Laptop



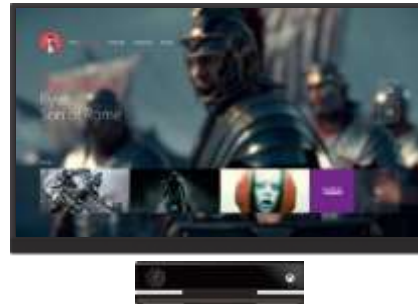
Desktops  
& All-in-Ones



Surface Hub



Xbox



Holographic



IoT

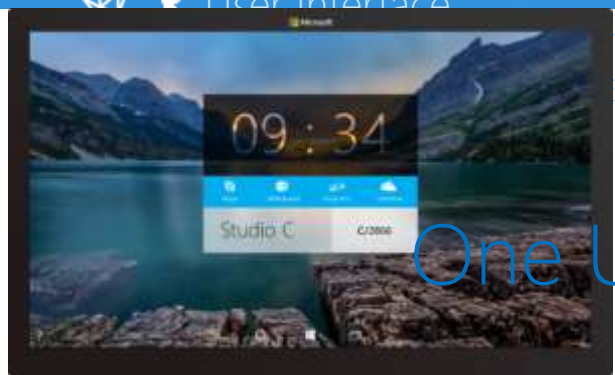






Adaptive  
User Interface

Reuse  
Existing Code



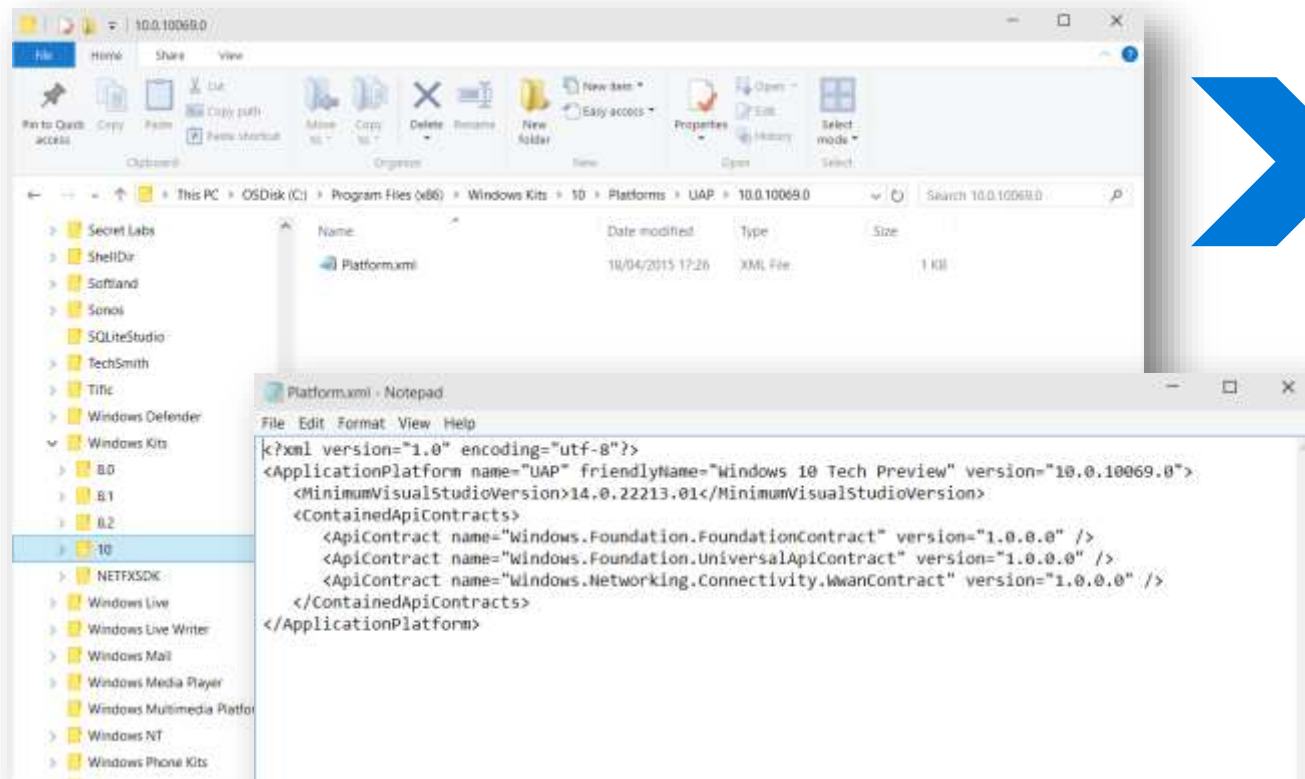
One Universal Windows Platform

# Universal Windows Platform

## A single API surface

A guaranteed API surface

The same on all devices



Universal Windows Platform

Windows Core

Desktop  
Device

Phone  
Device

Xbox  
Device

# Windows app

## A single binary

Running on any device

Testing for capabilities

Adjusting to devices



Windows App

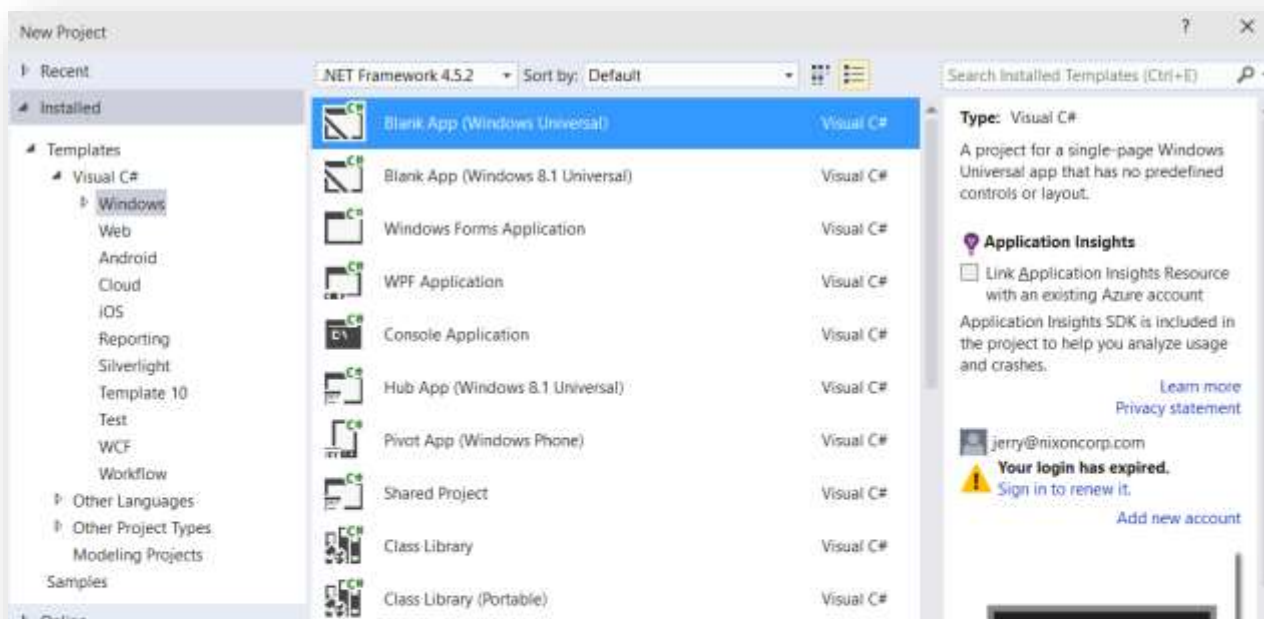
Universal Windows Platform

Windows Core

Desktop  
Device

Phone  
Device

Xbox  
Device



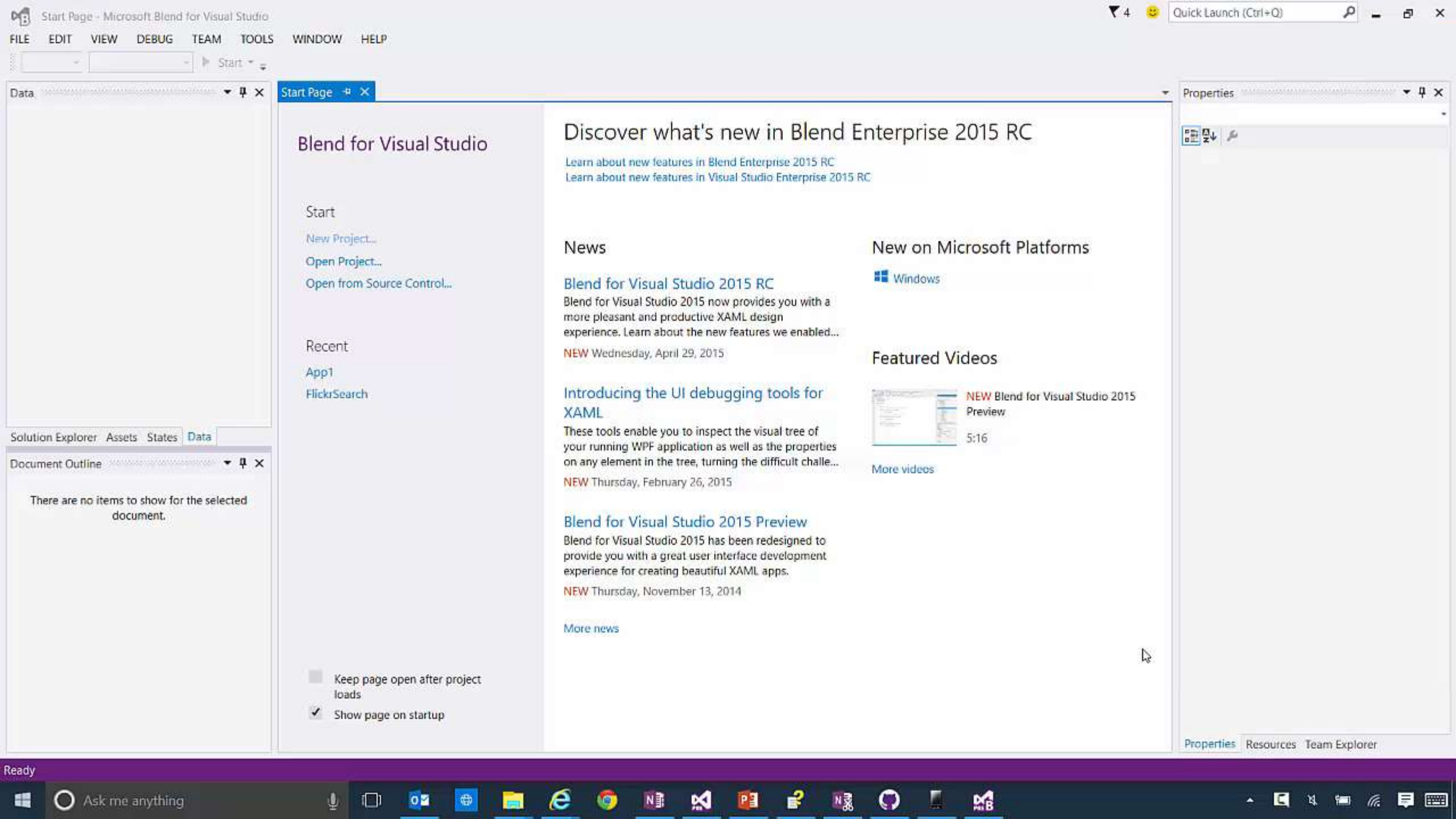
# Hello devices

## DEMO



2:05





# Platform extensions

## Device-specific API

Family-specific capabilities

Compatible across devices

Unique update cadence



Windows App

Desktop  
extension

Phone  
extension

Xbox  
extension

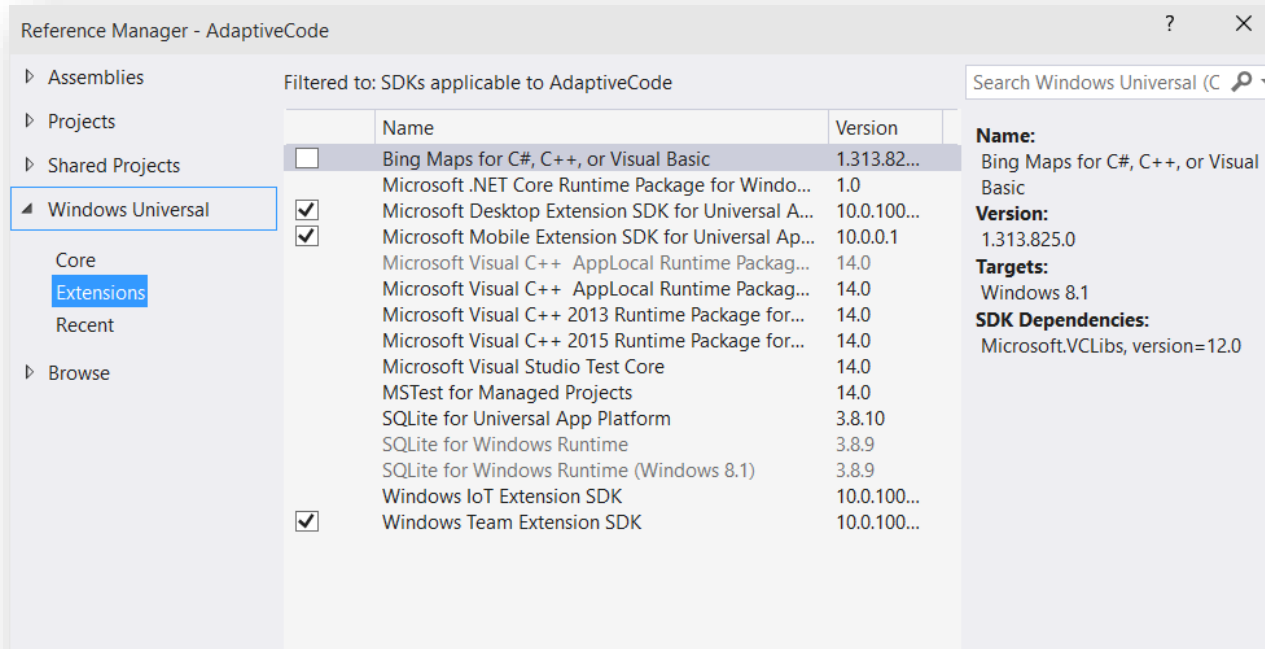
Universal Windows Platform

Windows Core

Desktop  
Device

Phone  
Device

Xbox  
Device



# Universal Windows Platform

## One Operating System

One Windows core for all devices

## One App Platform

Apps run across every family

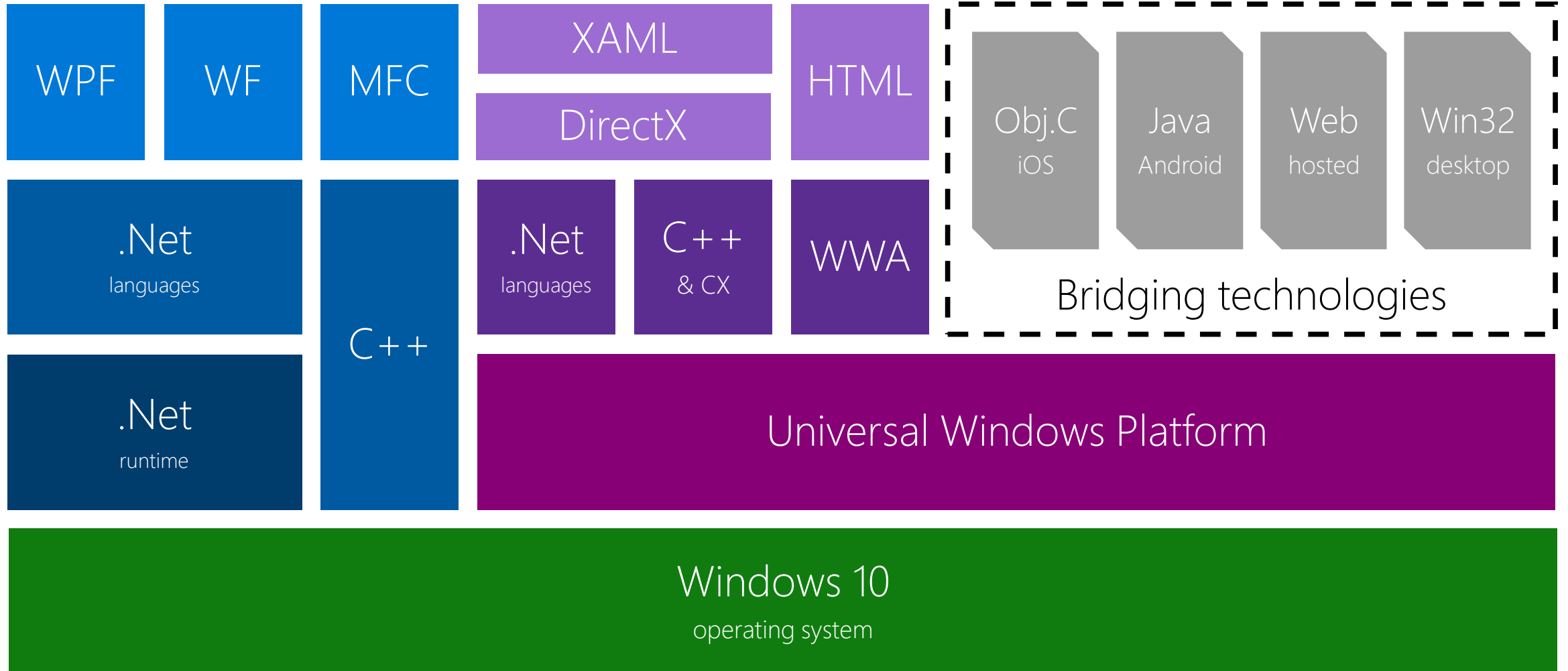
## One Dev Center

Single submission flow and dashboard

## One Store

Global reach, local monetization  
Consumers, Business & Education





# Let's talk about bridge technologies

## Objective-C

At Build we announced iOS code can be reused in a Windows app

## Android

At Build we announced Android code can be reused in a Windows app to run on Windows Phone

## Web

At Build we announced web sites can be wrapped to run on Windows

## Win32

At Build we announced that Classic Windows Apps (CWA) can be packaged as an Appx



# Blend for Visual Studio

## The XAML Developer's IDE

Always part of Visual Studio

Uses the Visual Studio shell

Full auto-complete & intellisense

- Validation
- Snippets
- Peek

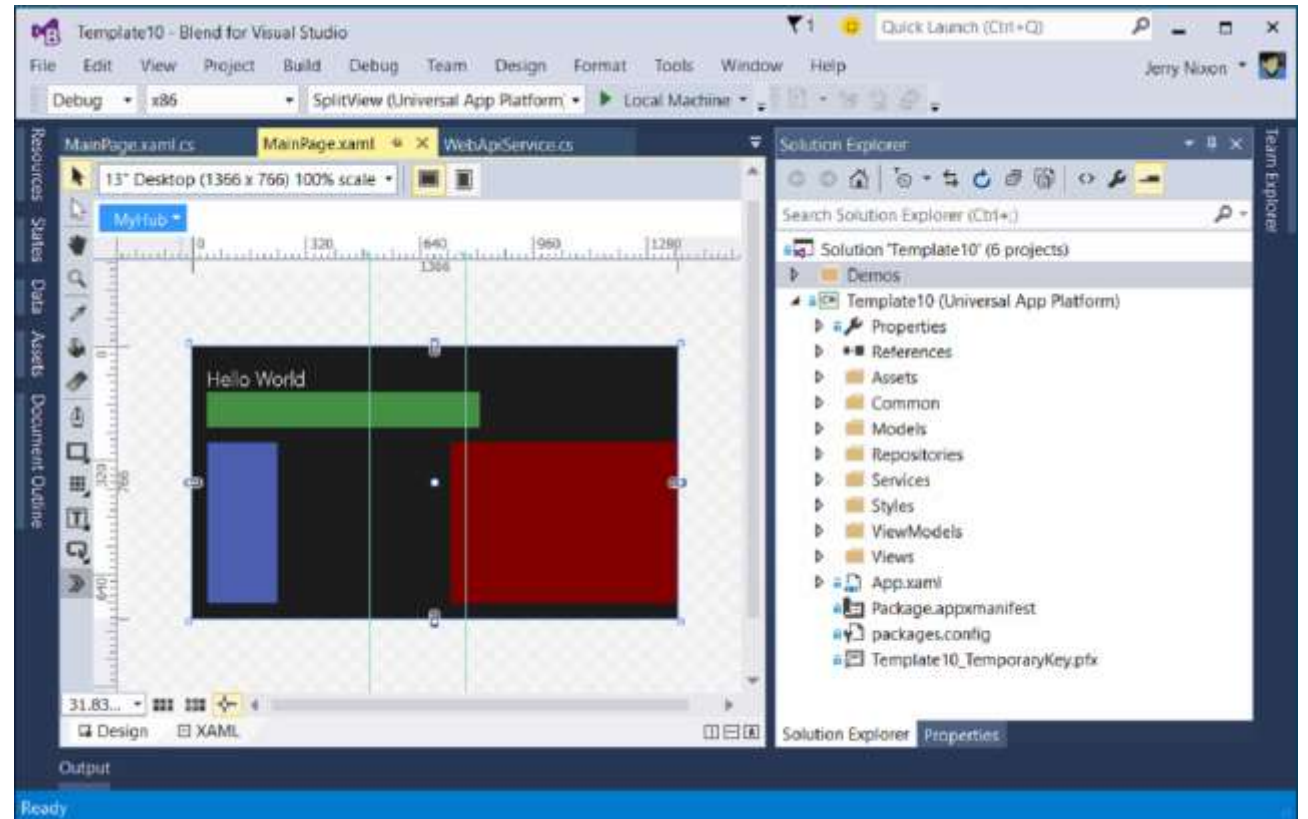
File & solution management

Resource management

Data management

Animation

States



# 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

# Microsoft Developer Network

## Microsoft documentation

<http://msdn.Microsoft.com>

<http://dev.Windows.com>

## Microsoft training

<http://msevents.Microsoft.com>

<http://MicrosoftVirtualAcademy.com>

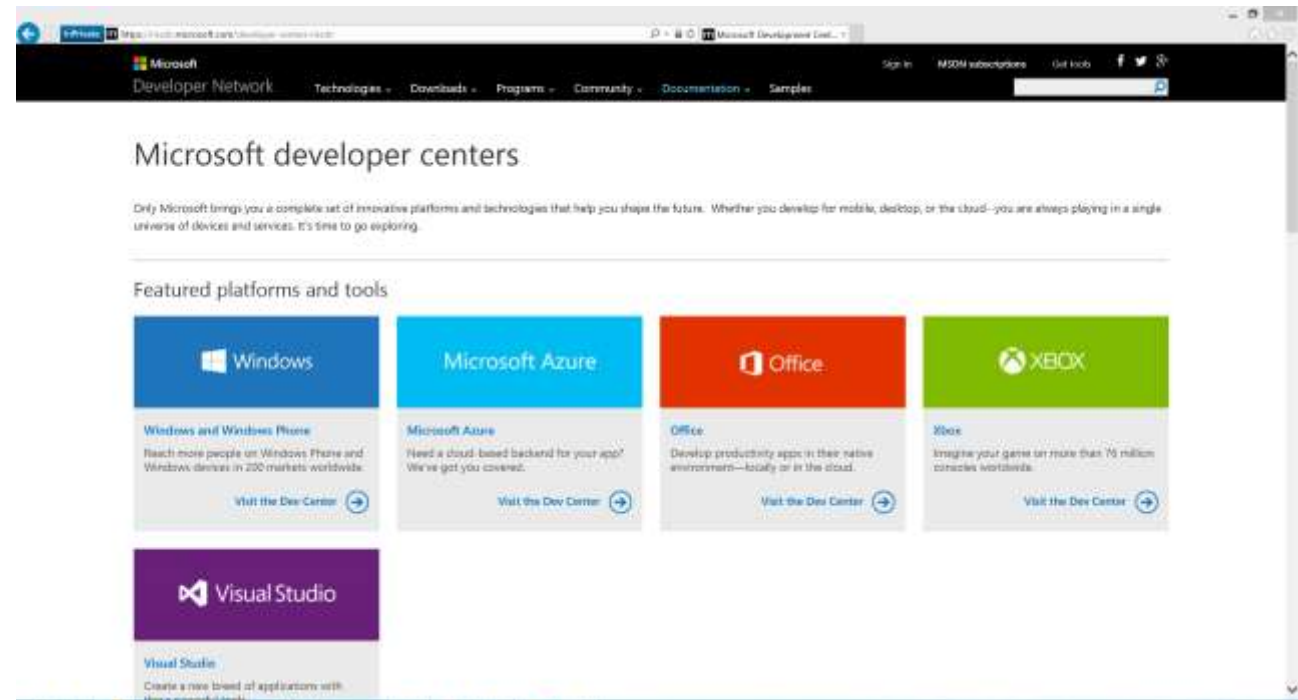
## MSDN Subscription

Operating Systems

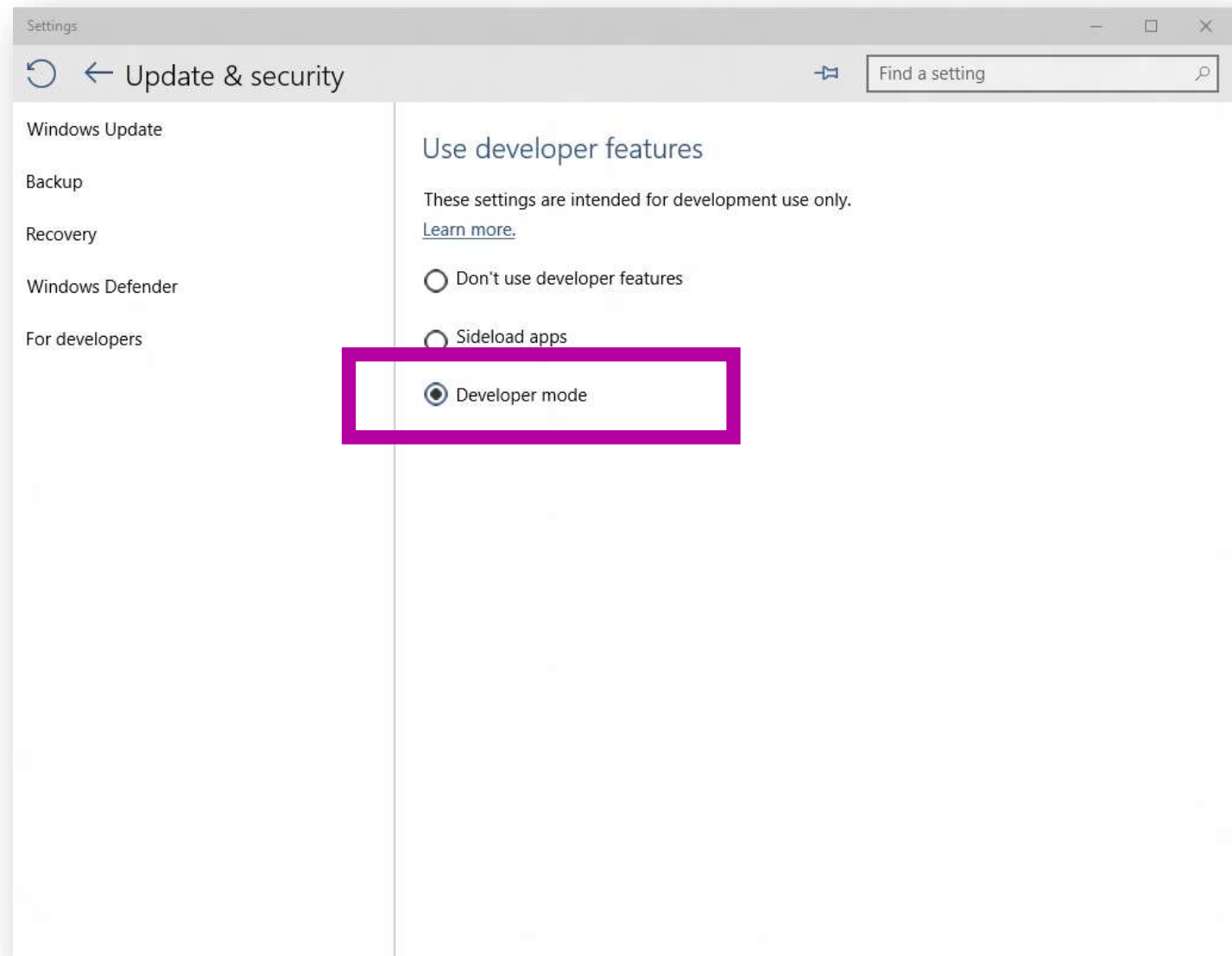
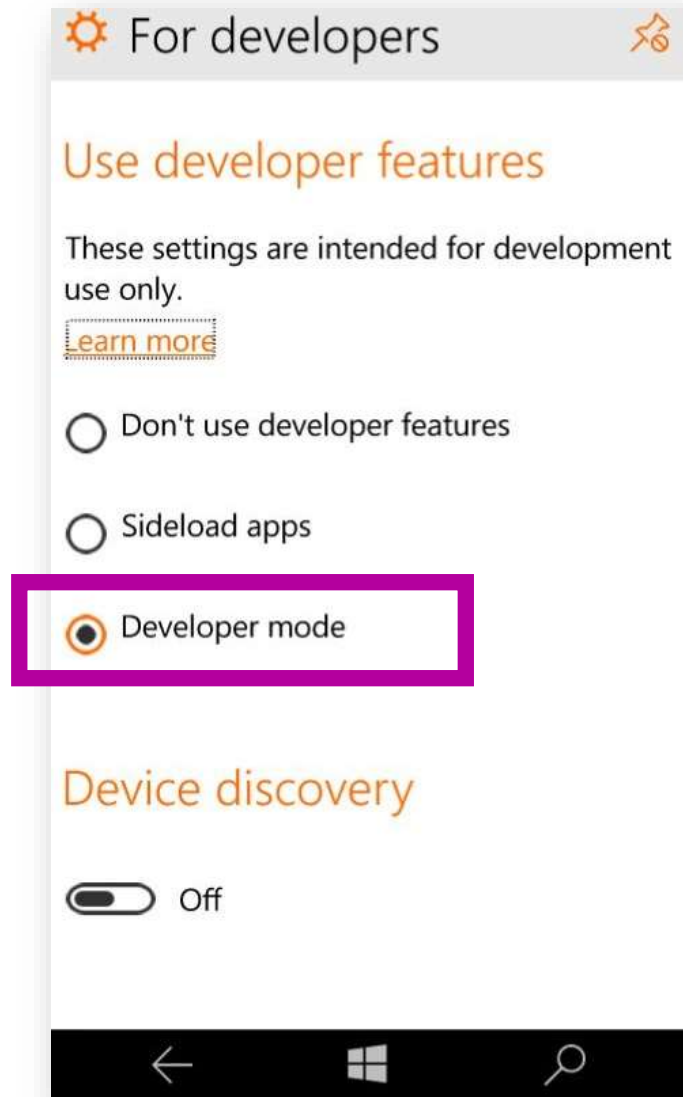
Server/Client Products

Developer Tools

Azure/O365 Credits



# Developer unlock



# XAML controls



# Layout controls

```
<Border Thickness="" Brush="" />
```

```
<Canvas />
```

```
<Grid />
```

```
<RelativePanel />
```

```
<ScrollViewer />
```

```
<SplitView DisplayMode="" />
```

```
<StackPanel Orientation="" />
```

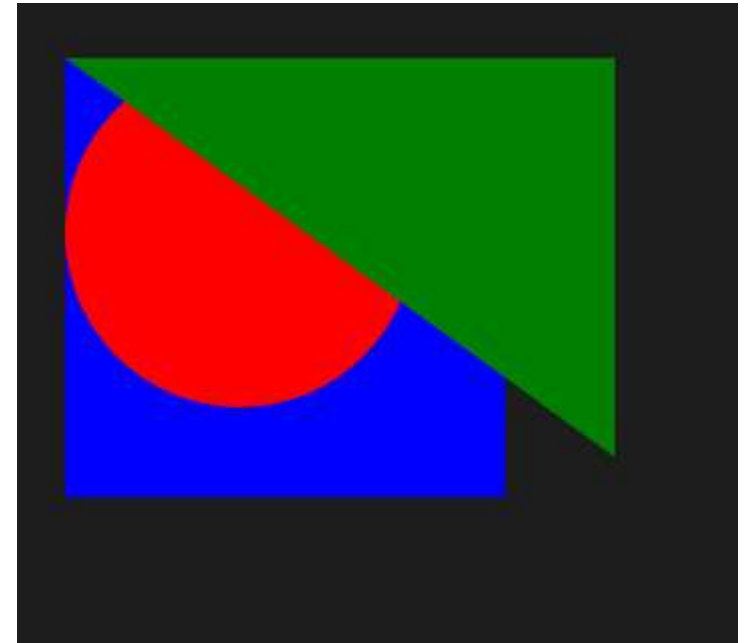
```
<VariableSizedWrapGrid />
```

```
<ViewBox Stretch="" />
```



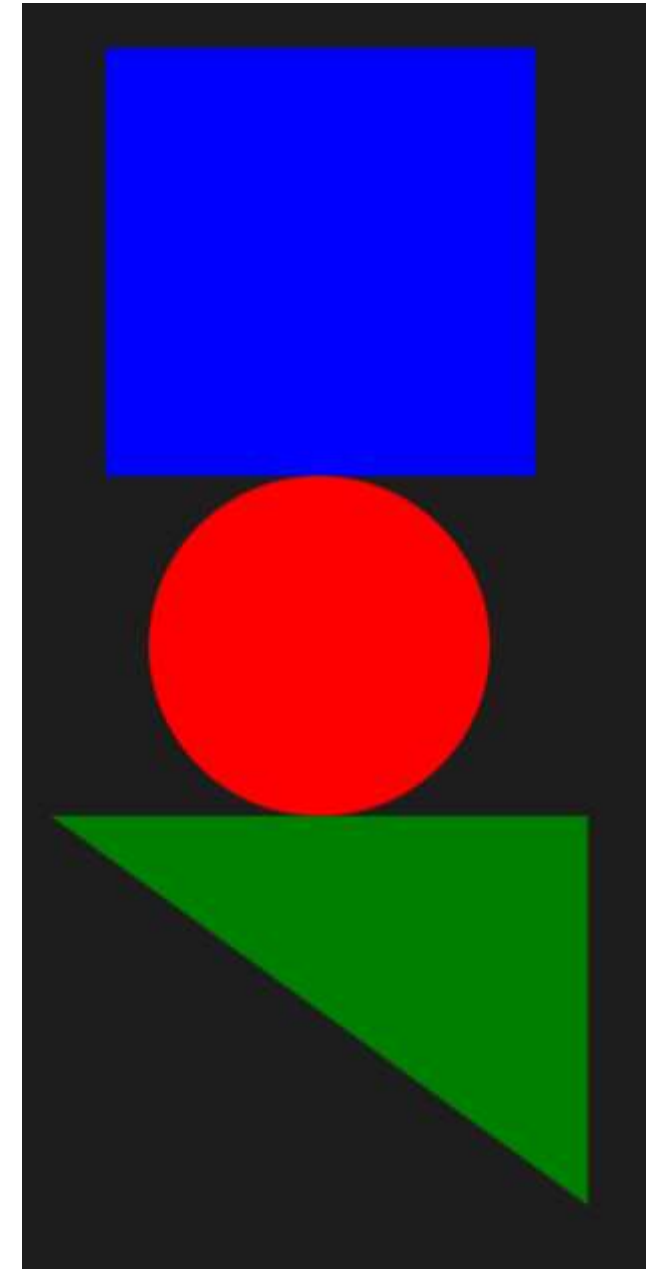
# Canvas

```
<Canvas>  
  <Rectangle Fill="Blue"  
             Width="220"  
             Height="220" />  
  <Ellipse Fill="Red"  
           Width="175"  
           Height="175" />  
  <Polygon Fill="Green"  
           Points="0,0 275,0 275,200"/>  
</Canvas>
```



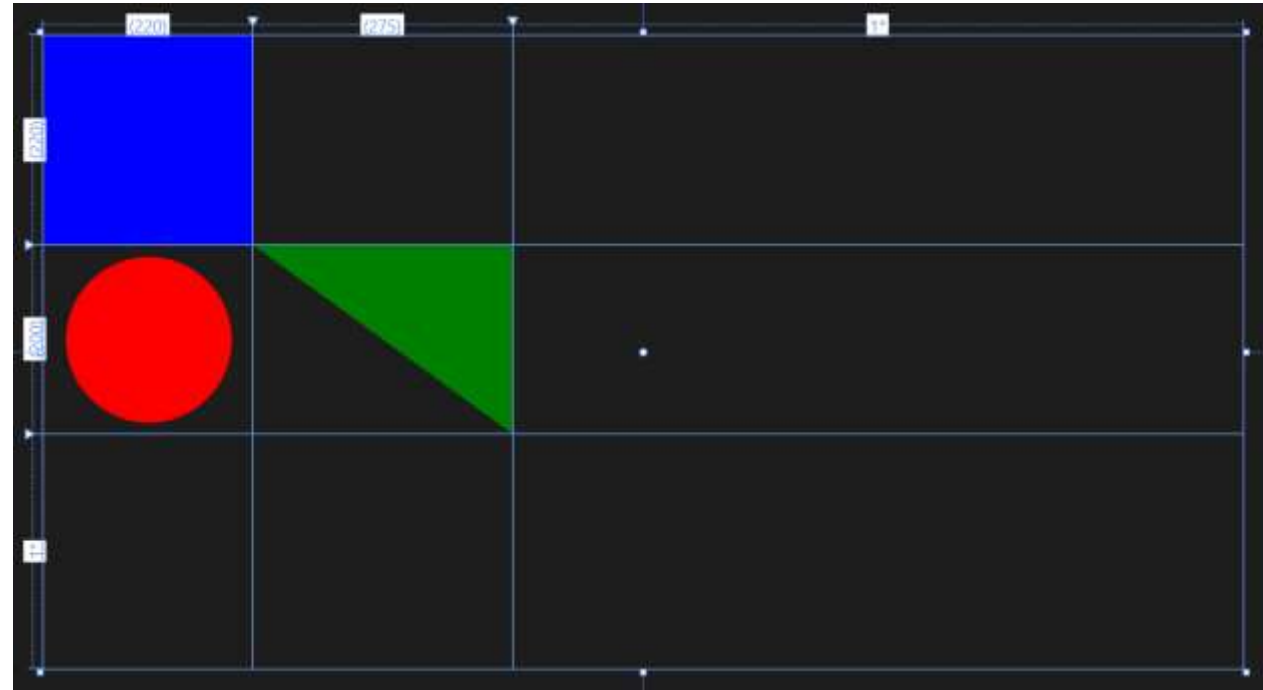
# StackPanel

```
<StackPanel Margin="50"
    HorizontalAlignment="Left"
    VerticalAlignment="Top">
    <Rectangle Fill="Blue"
        Width="220"
        Height="220" />
    <Ellipse Fill="Red"
        Width="175"
        Height="175" />
    <Polygon Fill="Green"
        Points="0,0 275,0 275,200"
        StrokeThickness="2" />
</StackPanel>
```



# Grid

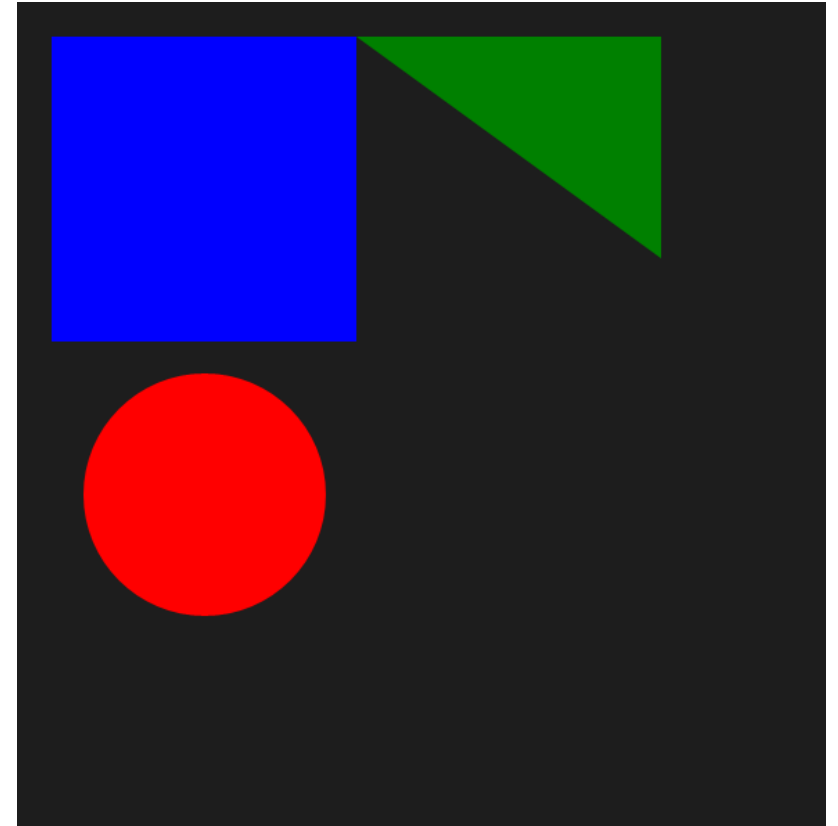
```
<Grid Margin="50">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="*/>
  </Grid.ColumnDefinitions>
  <Rectangle Fill="Blue" Width="220" Height="220"
    Grid.Row="0" Grid.Column="0" />
  <Ellipse Fill="Red" Width="175" Height="175"
    Grid.Row="1" Grid.Column="0" />
  <Polygon Fill="Green" Points="0,0 275,0 275,200"
    StrokeThickness="2" Grid.Row="1" Grid.Column="1" />
</Grid>
```



# WrapGrid

```
<ItemsControl Margin="50">
  <ItemsControl.ItemsPanel>
    <ItemsPanelTemplate>
      <WrapGrid MaximumRowsOrColumns="2"/>
    </ItemsPanelTemplate>
  </ItemsControl.ItemsPanel>

  <ItemsControl.Items>
    <Rectangle Fill="Blue"
      Width="220"
      Height="220" />
    <Ellipse Fill="Red"
      Width="175"
      Height="175" />
    <Polygon Fill="Green"
      Points="0,0 275,0 275,200"/>
  </ItemsControl.Items>
</ItemsControl>
```





# ScrollView

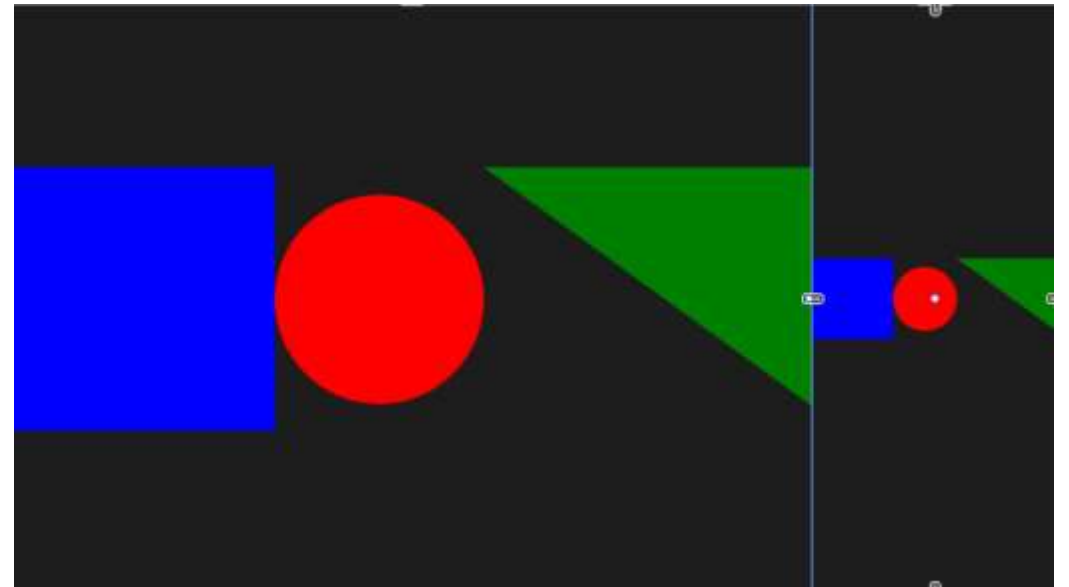
```
<ScrollView Margin="50">
  <StackPanel Margin="50"
    HorizontalAlignment="Left"
    VerticalAlignment="Top">
    <Rectangle Fill="Blue" Width="220" Height="220" />
    <Ellipse Fill="Red" Width="175" Height="175" />
    <Polygon Fill="Green" Points="0,0 275,0 275,200"/>
    <Rectangle Fill="Blue" Width="220" Height="220" />
    <Ellipse Fill="Red" Width="175" Height="175" />
    <Polygon Fill="Green" Points="0,0 275,0 275,200" />
  </StackPanel>
</ScrollView>
```



# Viewbox

```
<Viewbox Grid.Column="0">
  <StackPanel Orientation="Horizontal">
    <Rectangle Fill="Blue" Width="220" Height="220" />
    <Ellipse Fill="Red" Width="175" Height="175" />
    <Polygon Fill="Green" Points="0,0 275,0 275,200"
      StrokeThickness="2" />
  </StackPanel>
</Viewbox>

<Viewbox Grid.Column="1" Stretch="Uniform">
  <StackPanel Orientation="Horizontal">
    <Rectangle Fill="Blue" Width="220" Height="220" />
    <Ellipse Fill="Red" Width="175" Height="175" />
    <Polygon Fill="Green" Points="0,0 275,0 275,200"
      StrokeThickness="2" />
  </StackPanel>
</Viewbox>
```

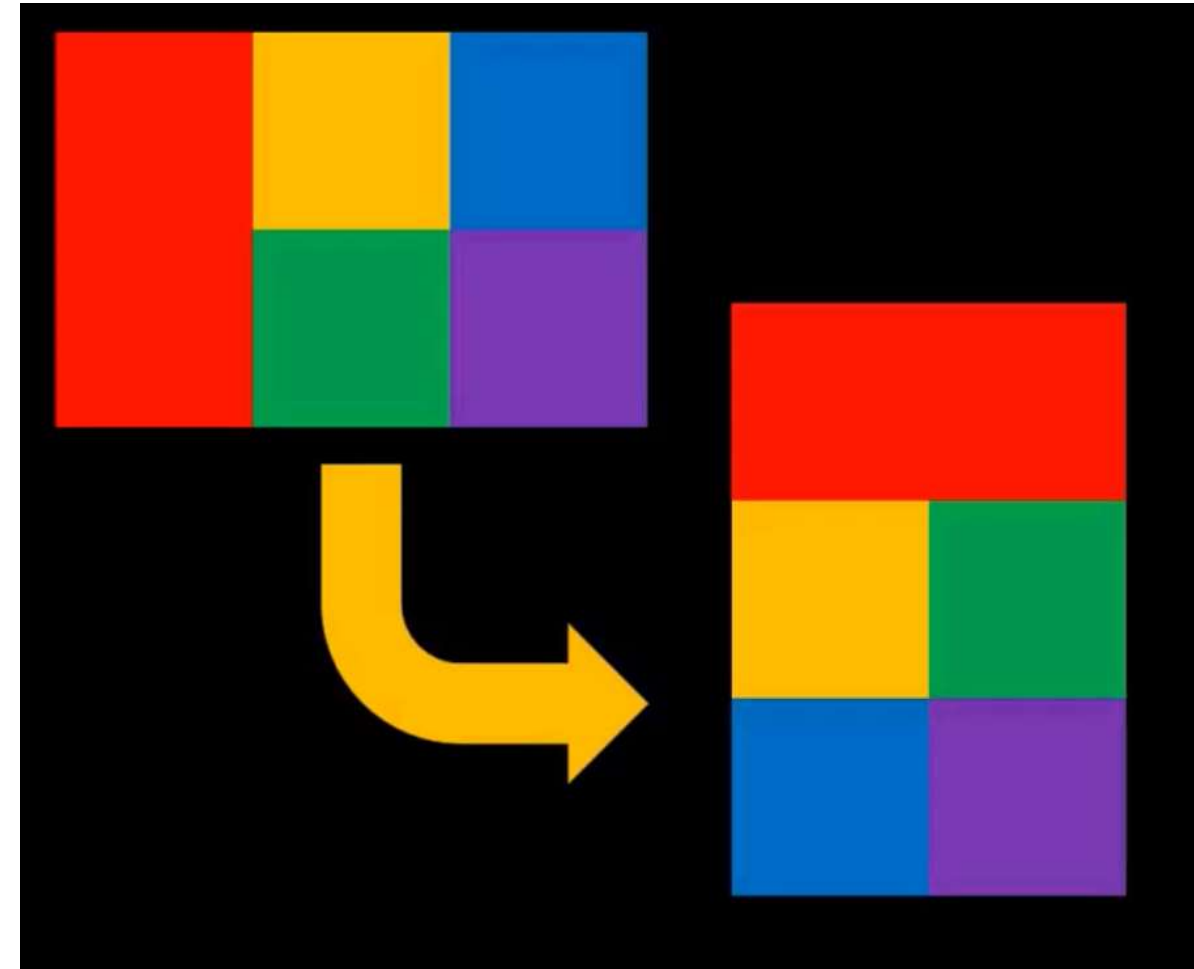


# RelativePanel

**Some child elements  
act as anchors**

**Most child elements  
relate to others**

**It's a layout technique  
friendly with States**



# Buttons

```
<Button Content="" />
```

```
<HyperlinkButton />
```

```
<RepeatButton />
```

```
<ToggleButton IsChecked="" />
```

## Primitive States

Rest Unselected

Button

Rest Selected

Button

Hover Unselected

Button

Hover Selected

Button

Pressed Unselected

Button

Pressed Selected

Button

Disabled Unselected

Button

Disabled Selected

Button

Rest Unselected  
Keyboard Focus

Button

Rest Selected  
Keyboard Focus

Button

Hover Unselected  
Keyboard Focus

Button

Hover Selected  
Keyboard Focus

Button

Pressed Unselected  
Keyboard Focus

Button

Pressed Selected  
Keyboard Focus

Button

Rest  
Maximum Size

XX

# Text controls

```
<TextBox Text="" />
```

```
<PasswordBox Text="" />
```

```
<TextBlock Text="" />
```

```
<RichEditBox Content="" />
```

```
<RichTextBlock Content="" />
```

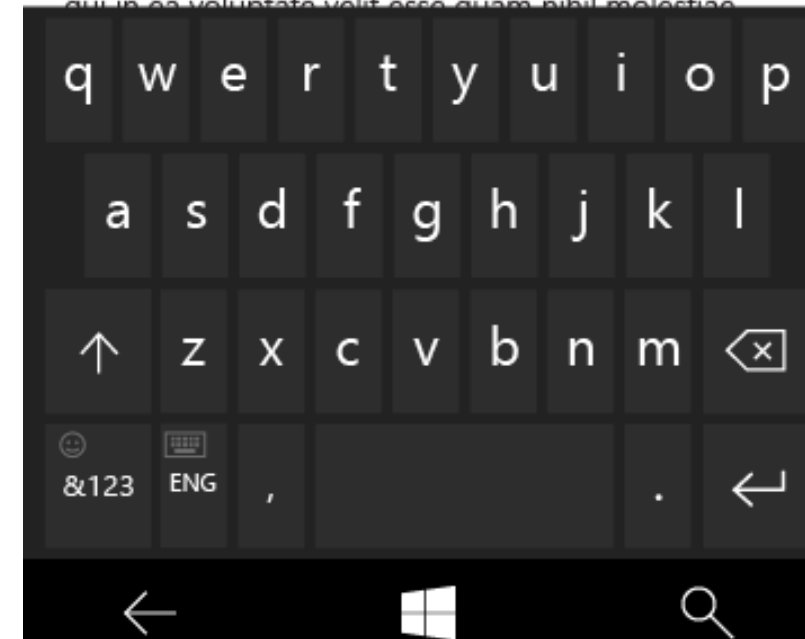
```
<BitmapIcon UriSource="" />
```

```
<FontIcon Glyph="" />
```

```
<SymbolIcon Icon="" />
```

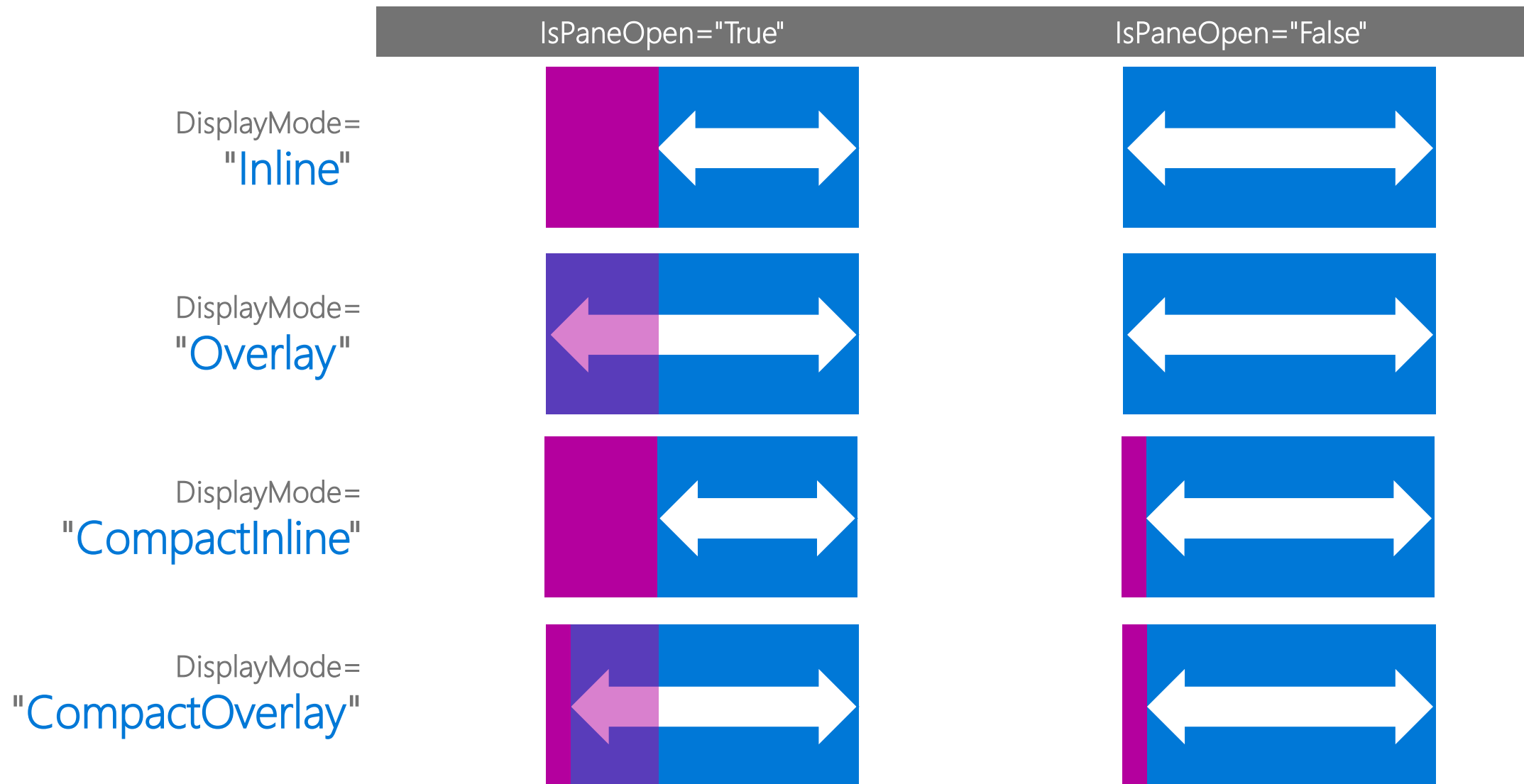
```
<PathIcon Data="" />
```

totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae





# SplitView



# SplitView

DEMO

# App bars and commands

<AppBar />

<CommandBar />

<AppBarButton Label="" Icon="" />

<AppBarToggleButton IsChecked="" />

<AppBarSeparator />

Compact Mode Pg 1

Rest



Icon Hover



Icon Pressed



Icon Toggled



Icon Toggled & Hover



Icon Toggled & Pressed



# AppBar

## DEMO

# Networking

# Daum 검색 API

<http://developers.daum.net/>

# Daum Open APIs

Daum Developers

콘솔

서비스

블로그

카페

서비스 winkey

어디서든 접근 가능한  
새로운 콘솔



팀원 등록, API Quota 설정, 사용량 조회, API 상태 조회 등  
다양한 기능을 어디서나 사용 가능합니다.



# Daum Open APIs

```
<?xml version="1.0" encoding="UTF-8"?>
<channel>
  <title>Search Daum Open API</title>
  <description>Daum Open API search result</description>
  <generator>Daum Open API</generator>
  <link>http://dna.daum.net/apis</link>
  <lastBuildDate>Tue, 28 Jul 2015 15:56:24 +0900</lastBuildDate>
  <totalCount>192905</totalCount>
  <result>10</result>
- <item>
  <title>시빌 워</title>
  <link>http://book.daum.net/detail/book.do?bookid=KOR9788952756107</link>
  <cover_s_url>http://t1.daumcdn.net/thumb/R72x100/?fname=http%3A%2F%2Ft1.daumcdn.net%2Fbook%2FKOR9788952756107%3I
  <cover_l_url>https://t1.search.daumcdn.net/thumb/R110x160/?fname=http%3A%2F%2Ft1.daumcdn.net%2Fbook%2FKOR978895275
  <description>당신은 어느 편인가? 그래픽노블 『시빌 워』는 2006년 7월부터 2007년 1월까지 총 7부작으로 연재되었던 미니 시리즈이다. 마블의 2006년 후반
  <author>마크 밀러</author>
  <translator>최원서</translator>
  <pub_nm>시공사</pub_nm>
  <pub_date>20091201</pub_date>
  <category>만화</category>
  <isbn>895275610X</isbn>
  <sale_yn>Y</sale_yn>
  <list_price>14000</list_price>
  <sale_price>12600</sale_price>
  <status_des>정상판매</status_des>
  <barcode>KOR9788952756107</barcode>
  <ebook_barcode/>
  <isbn13>9788952756107</isbn13>
  <etc_author>최원서 스티브 맥니븐 스티브맥니븐</etc_author>
  <author_t>마크 밀러</author_t>
</item>
- <item>
  <title>도서관 전쟁. 13</title>
  <link>http://book.daum.net/detail/book.do?bookid=KOR9791133400140</link>
  <cover_s_url>http://t1.daumcdn.net/thumb/R72x100/?fname=http%3A%2F%2Ft1.daumcdn.net%2Fbook%2FKOR9791133400140%3I
```

# System.Net.Http.HttpClient

Namespace: System.Net.Http.HttpClient

Assembly: System.Net.Http (in System.Net.Http.dll)

## System.Net.Http.HttpClient

```
private async void GetData(Data.SampleDataGroup group)
{
    System.Net.Http.HttpClient client
        = new System.Net.Http.HttpClient();

    System.Net.Http.HttpResponseMessage response
        = await client.GetAsync("http://apis.daum.net/search/book");

    response.EnsureSuccessStatusCode();
    string responseBody
        = await response.Content.ReadAsStringAsync();
}
```

# **System.Net.Http.HttpClient**

DEMO

# Entity Class

```
class BookItem
```

```
{  
    public Int64 Id { get; set; }  
    public string Title { get; set; }  
    public string Category { get; set; }  
    public string ImageUrl { get; set; }  
    public string Description { get; set; }  
}
```

# Linq

# System.Net.Http.HttpClient

## LINQ를 통한 XAML의 파싱

```
System.Xml.Linq.XDocument xDoc =  
    System.Xml.Linq.XDocument.Parse(responseBody);
```

```
var books = from bookItem in xDoc.Descendants("item")  
            select new BookItem
```

```
{
```

```
    Title = (string)bookItem.Element("title"),  
    Category = (string)bookItem.Element("category"),  
    ImageUrl = (string)bookItem.Element("cover_l_url"),  
    Description = (string)bookItem.Element("description");
```

```
};
```

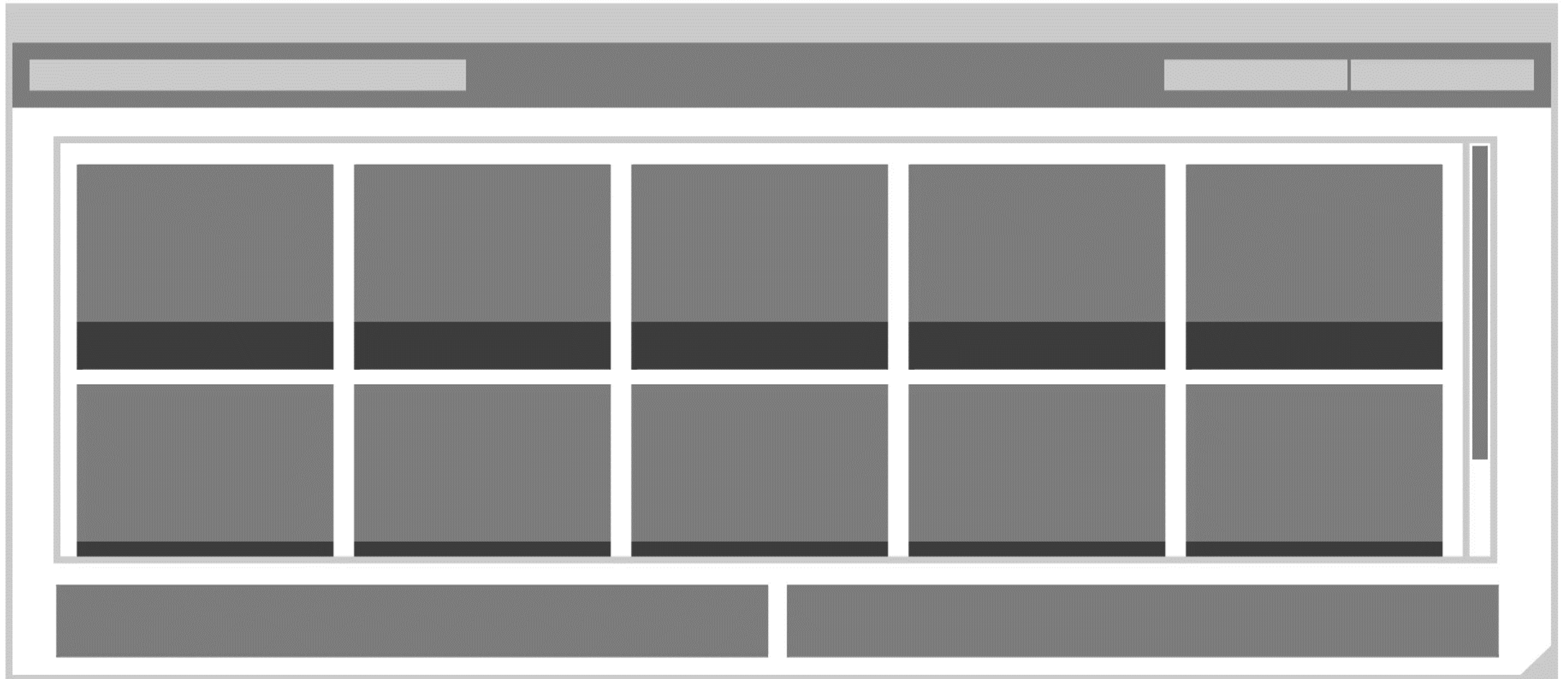
# Linq

# DEMO

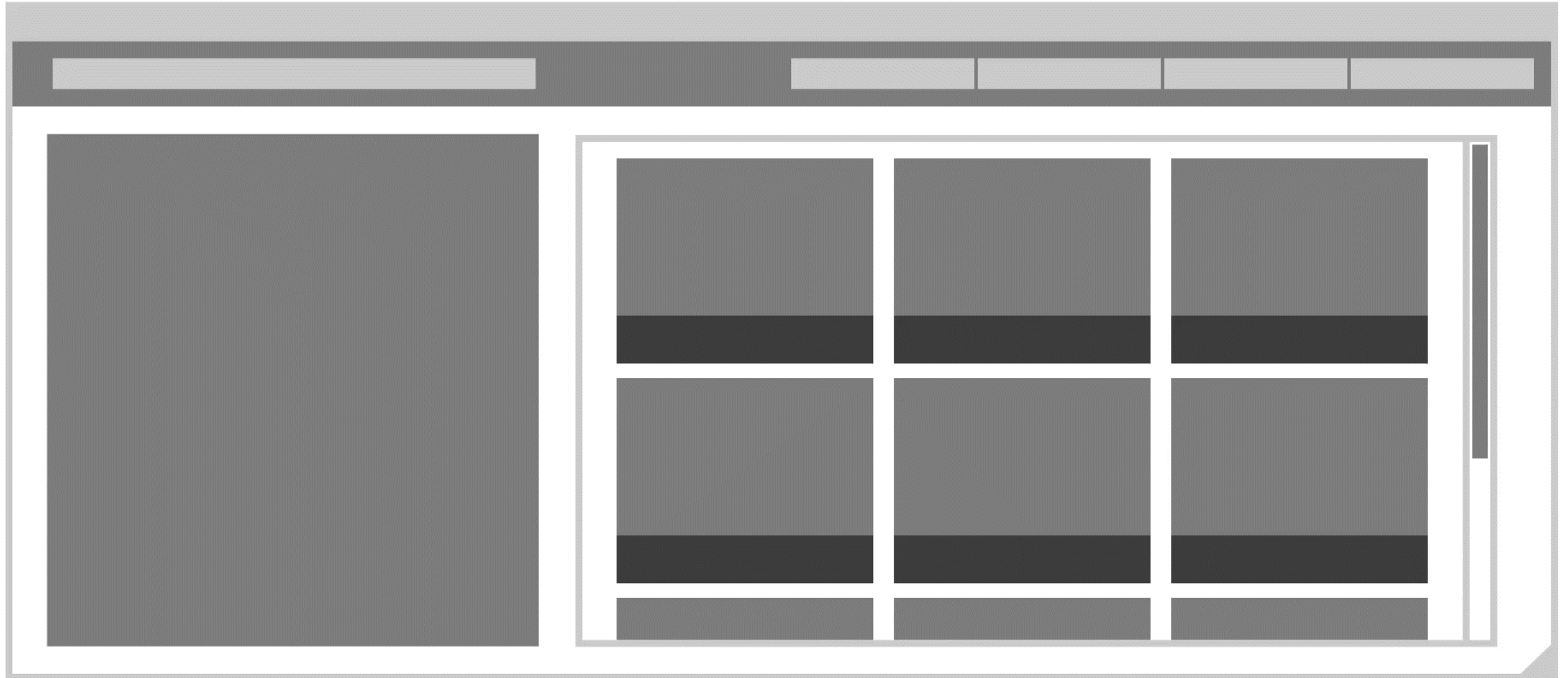


# Adaptive UI

# Responsive

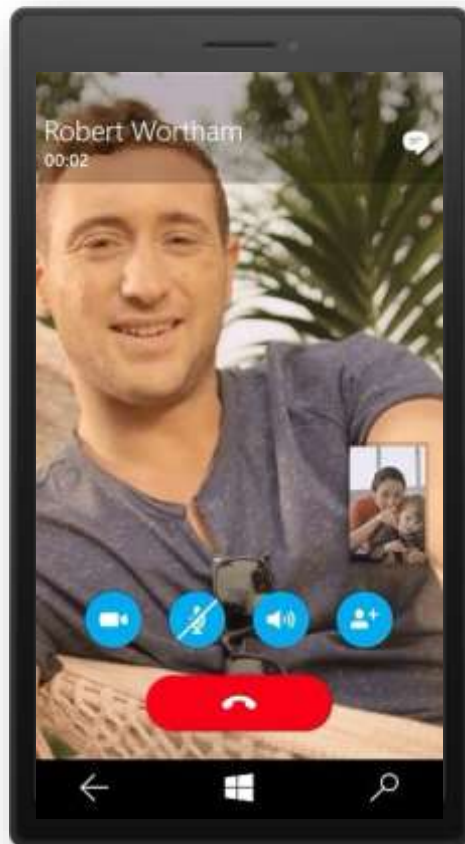


# Adaptive

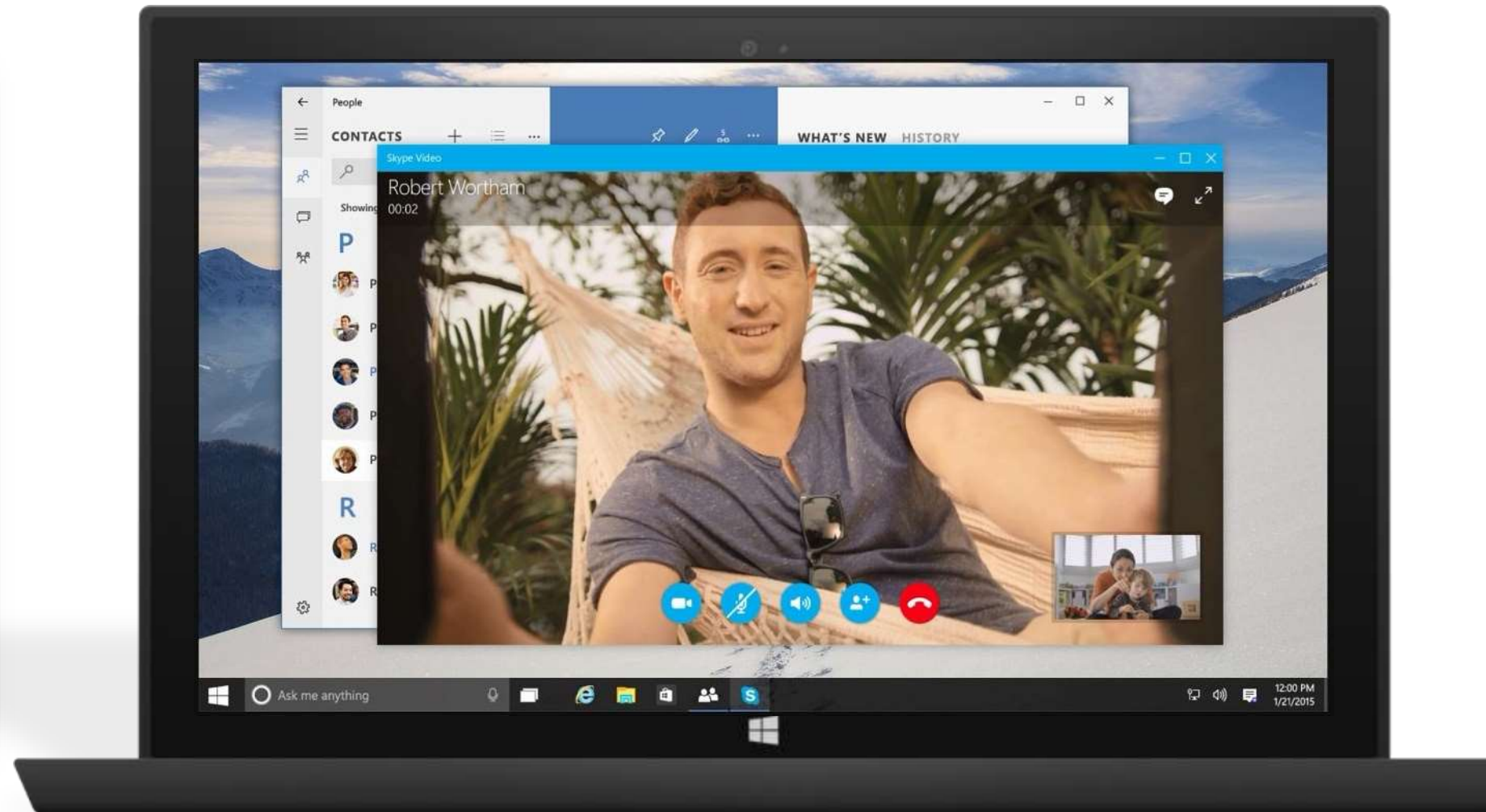


# Adaptive design

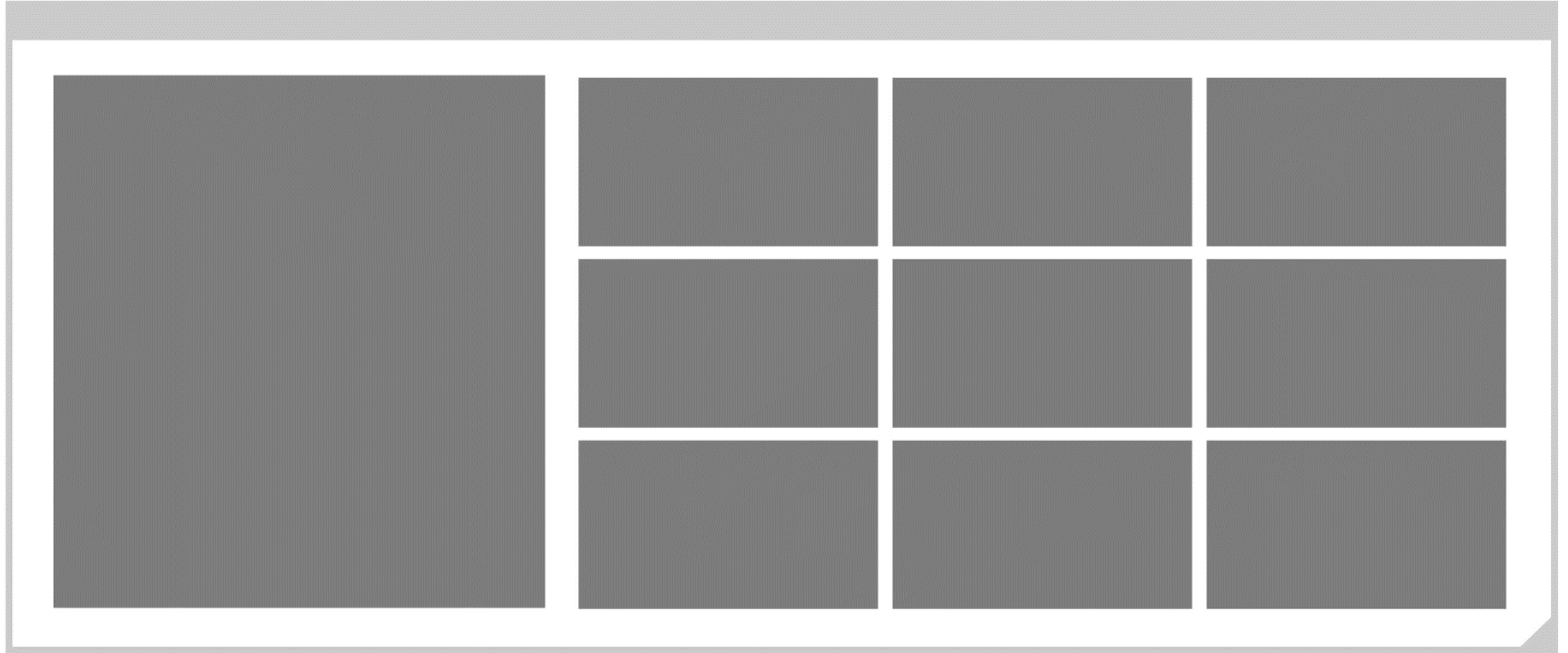
Phone (portrait)



Tablet (landscape) / Desktop

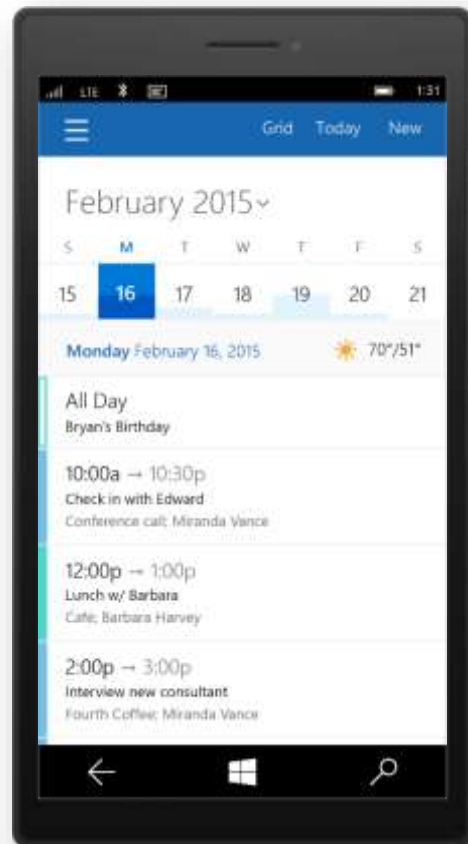


# Scaling

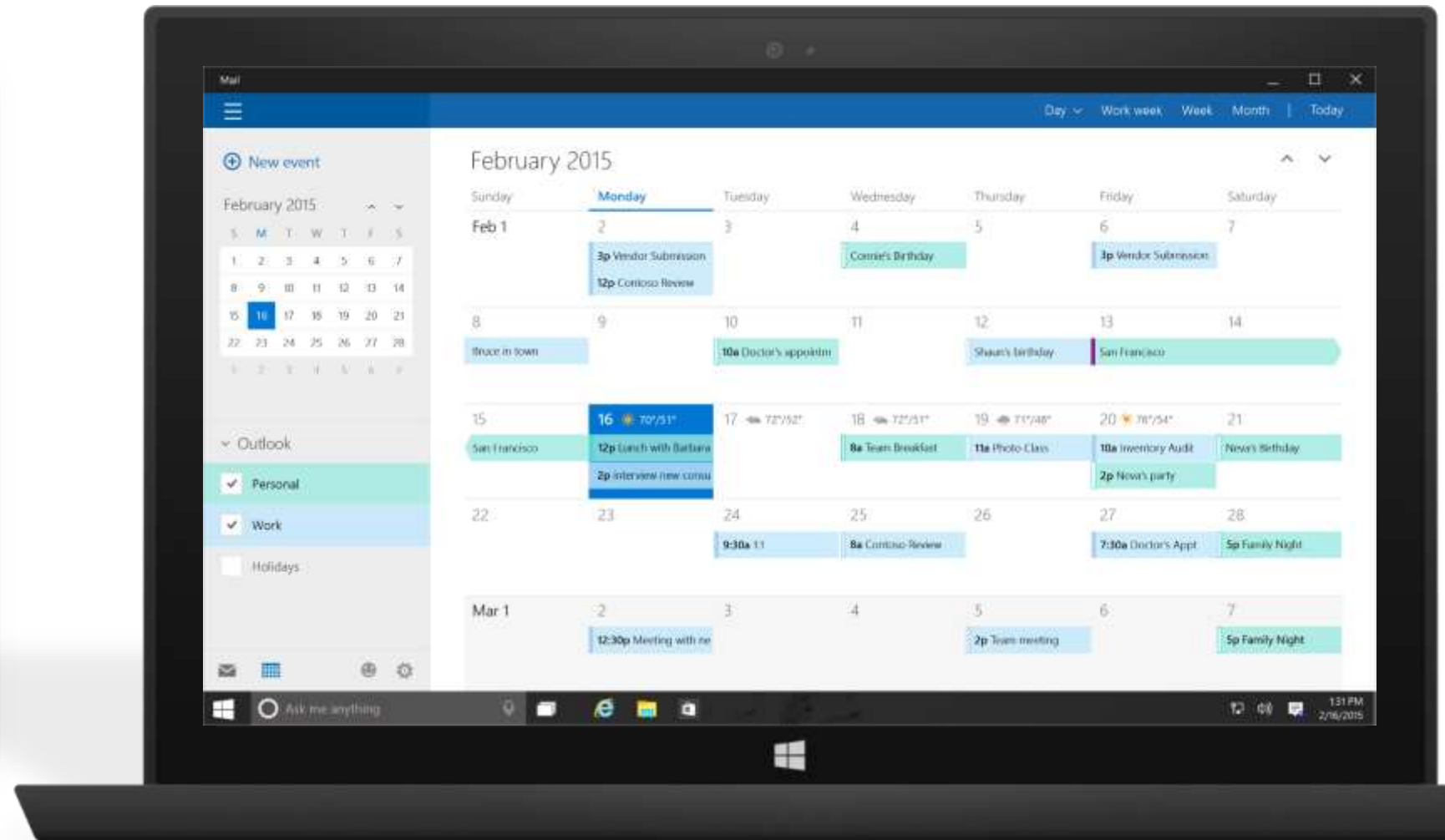


# Tailored design

Phone (portrait)



Tablet (landscape) / Desktop



# Tailored design

## Build pages/code for individual families

Use MRT in App.xaml.cs to determine the family

## One-handed interface?

Typically phone or small tablets

Test diagonal screen size (<7")

```
if (physical_diagonal_size <= 7)
```

```
// optimized for one-handed operation
```

```
    rootFrame.Navigate(typeof(MainPage_OneHanded), e.Arguments);
```

```
else
```

```
    rootFrame.Navigate(typeof(MainPage), e.Arguments);
```

# Visual States

## Define XAML views

Unique layout for distinct states

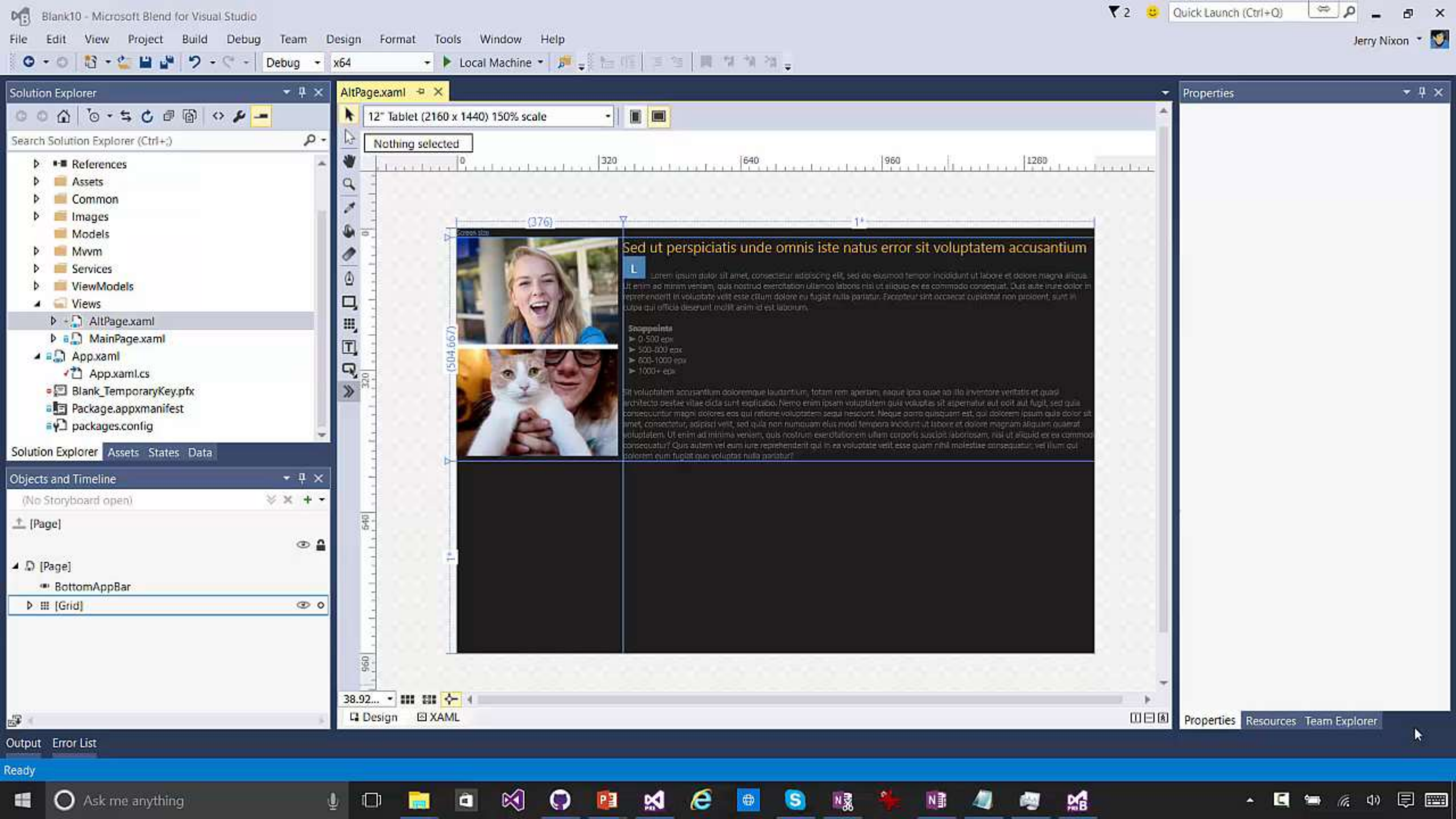
## Simplify animation

Automatically implement state transitions

## Build in Blend

Design and preview states and transitions





# Visual states

DEMO

# 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);
    };
}
```

# Adaptive triggers

DEMO

# Data binding



A young man and woman are sitting on a grassy field. The man is wearing a grey t-shirt and dark pants, and the woman is wearing a light blue long-sleeved shirt and patterned pants. They are both smiling and looking towards the right. A yellow and black soccer ball is on the grass in front of them. The background is a blurred green field with trees in the distance.

# Data binding basics

# Dynamic Data

## Use data binding to connect to a data source

Typical data source would be a view model

```
<Page.Resources>
    <local:ViewModel x:Key="Vm"/>
</Page.Resources>

<Grid Background="{StaticResource ApplicationPageBackgroundThemeBrush}">
    <StackPanel DataContext="{StaticResource Vm}">
        <TextBlock Text="{Binding Message}"/>
        <Button Content="{Binding ButtonText}"/>
    </StackPanel>
</Grid>
```

Dummy Message

Click Me

# Data Binding

DEMO



# SQLite Local Database

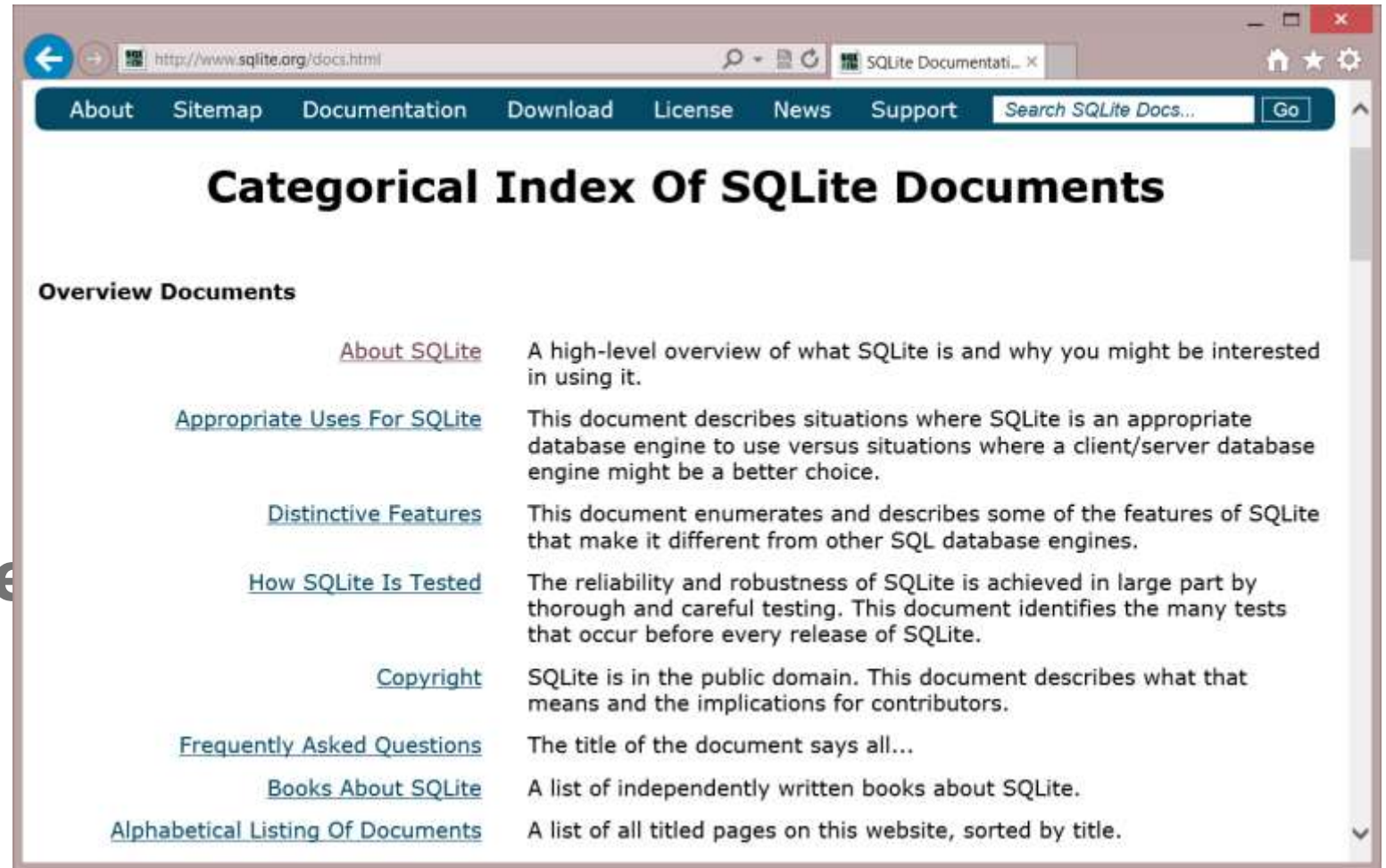
# SQLite.org

## Documentation

## SQL Syntax

## C/C++ API Reference

## Source and tools download



# Choice of .NET APIs

## SQLite-NET

LINQ syntax  
Lightweight ORM

```
var db =  
    new SQLite.SQLiteAsyncConnection(App.DBPath);  
  
var _customer = await  
    (from c in db.Table<Customer>()  
     where c.Id == customerId  
     select c).FirstOrDefaultAsync();  
  
if (customer != null)  
{  
    var Id = _customer.Id;  
    var Name = _customer.Name;  
}
```

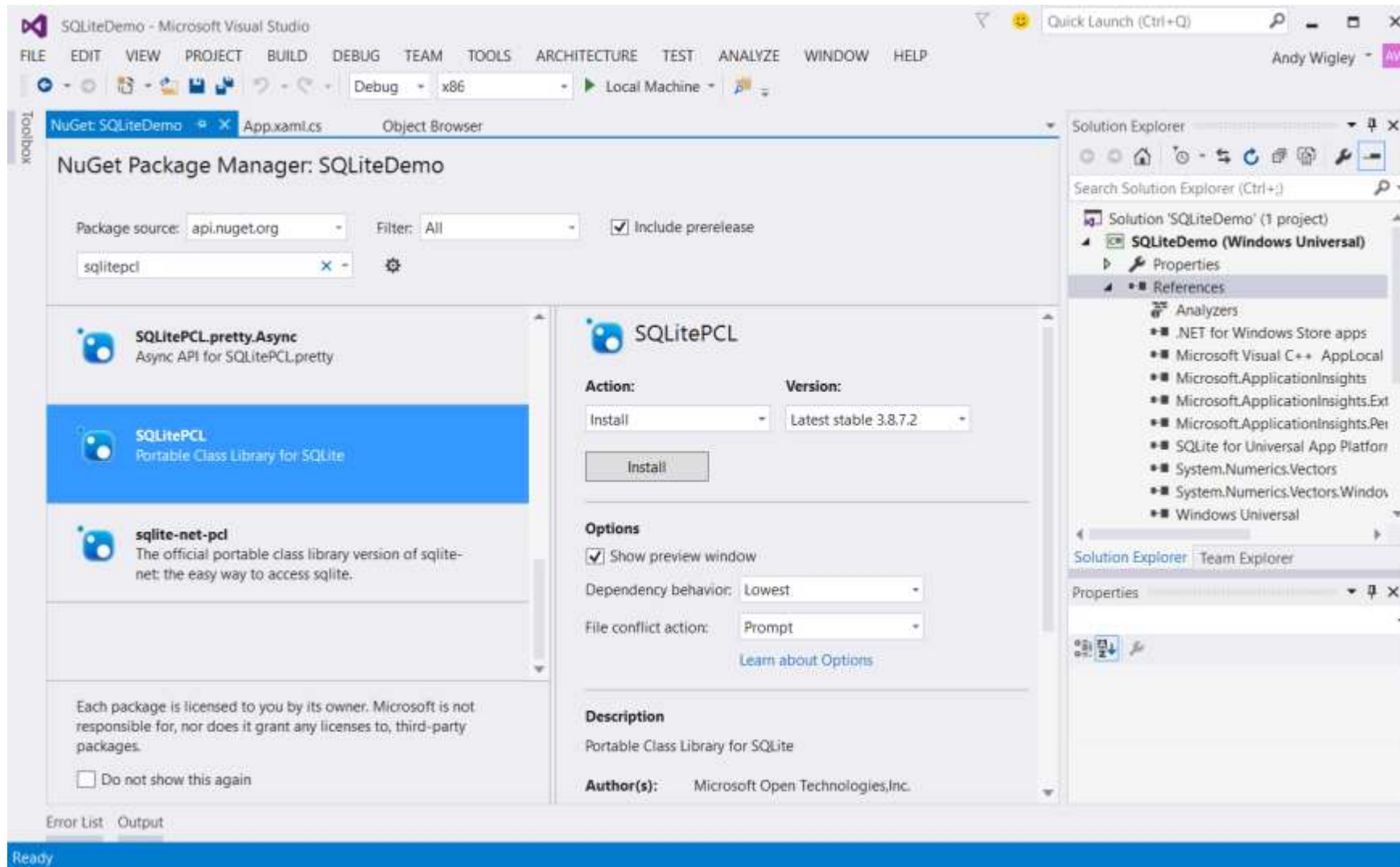
## SQLitePCL

SQL statements  
Thin wrapper around the SQLite C API

```
using (var conn = new SQLiteConnection("demo.db"))  
{  
    Customer customer = null;  
    using (var statement = conn.Prepare(  
        "SELECT Id, Name FROM Customer WHERE Id = ?"))  
    {  
        statement.Bind(1, customerId);  
        if (SQLiteResult.DONE == statement.Step()) {  
            customer = new Customer() {  
                Id = (long)statement[0],  
                Name = (string)statement[1] };  
        }  
    }  
}
```

...and others!

# Installing SQLitePCL to your Solution



# SQLitePCL

## DEMO



# Create database and tables

```
private void LoadDatabase()
{
    // Get a reference to the SQLite database
    conn = new SQLiteConnection("sqlitepcldemo.db");
    string sql = @"CREATE TABLE IF NOT EXISTS
                    Customer (Id      INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
                             Name    VARCHAR( 140 ),
                             City    VARCHAR( 140 ),
                             Contact VARCHAR( 140 )
                    );";
    using (var statement = conn.Prepare(sql))
    {
        statement.Step();
    }
}
```

# Insert

```
// SqlConnection was opened in App.xaml.cs and exposed through property conn
var db = App.conn;
try
{
    using (var custstmt = db.Prepare("INSERT INTO Customer (Name, City, Contact) VALUES (?, ?, ?)"))
    {
        custstmt.Bind(1, customerName);
        custstmt.Bind(2, customerCity);
        custstmt.Bind(3, customerContact);
        custstmt.Step();
    }
}
catch (Exception ex)
{
    // TODO: Handle error
}
```

# Select

```
public Customer GetCustomer(int customerId)
{
    Customer customer = null;
    using (var statement = dbconn.Prepare("SELECT Id, Name, City, Contact FROM Customer WHERE Id = ?"))
    {
        statement.Bind(1, customerId);
        if (SQLiteResult.DONE == statement.Step())
        {
            customer = new Customer()
            {
                Id = (long)statement[0],
                Name = (string)statement[1],
                City = (string)statement[2],
                Contact = (string)statement[3]
            };
        }
    }
    return customer;
}
```



# Update

```
// See if the customer already exists
var existingCustomer = GetCustomer(customer.Id);
if (existingCustomer != null)
{
    using (var custstmt =
dbconn.Prepare("UPDATE Customer SET Name = ?, City = ?, Contact = ? WHERE Id=?"))
    {
        // NOTE when using anonymous parameters the first has an index of 1, not 0.
        custstmt.Bind(1, customer.Name);
        custstmt.Bind(2, customer.City);
        custstmt.Bind(3, customer.Contact);
        custstmt.Bind(4, customer.Id);
        custstmt.Step();
    }
}
```

# Delete

```
public void DeleteCustomer(int customerId)
{
    using (var statement = dbconn.Prepare("DELETE FROM Customer WHERE Id = ?"))
    {
        statement.Bind(1, customerId);
        statement.Step();
    }
}
```

# SQLite using SQLitePCL

DEMO

# Toast

# Toasts

## Glance (consume)

See new information from your apps.

## Act (chase, or take actions)

Toasts invite you to begin or complete a task.

The toast is the app's door by chasing (clicking) it.

Additional actions enable users to perform simple tasks without context switching.



**Torrance Shum**

This is a mockup showing 6 lines of content assuming that the developers have chosen a content-heavy toast template. This means there are no controls in this mockup. It still clips on the second line. You can see the exa



12:00 AM  
7/31/2014



**Torrance Shum**



This is a mockup showing 4 lines of content and 1 type of control. The content for this template clips at the fourth line – shown her

Like

Dislike



12:00 AM  
7/31/2014

# Toast templates

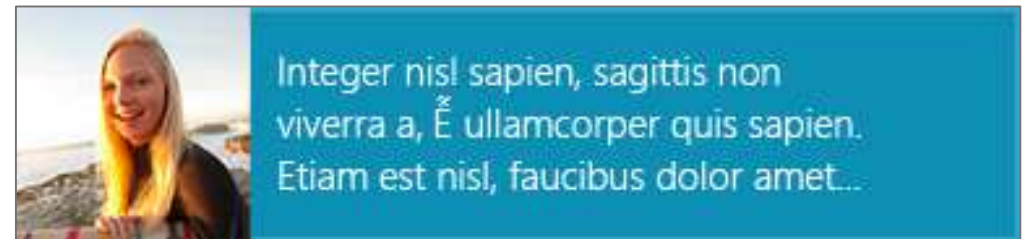
If a template meets your needs,  
go ahead and use it.

## Previous templates remain

Phone and Windows templates have been merged

## Adaptive template

Same XML syntax as tiles



Lorem ipsum dolor sit amet consectetur  
Integer nisl sapien, sagittis non viverra a, Æ  
ullamcorper quis sapien. Etiam est nisl, faucibus

# **Sending toast**

## **Scheduled**

Set template and time with "ScheduledToastNotification"  
Toast can also be set to be recurring.

## **Local**

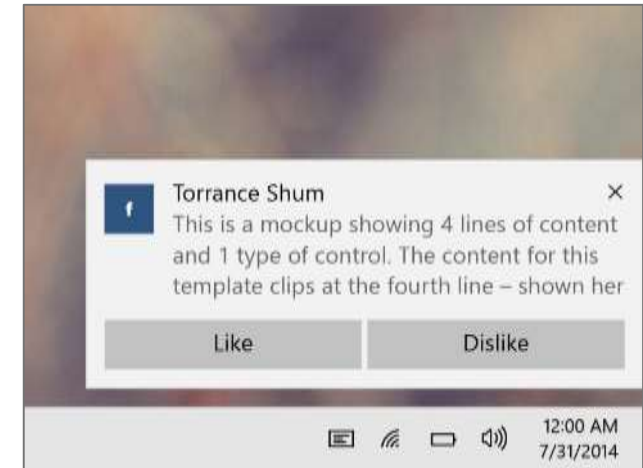
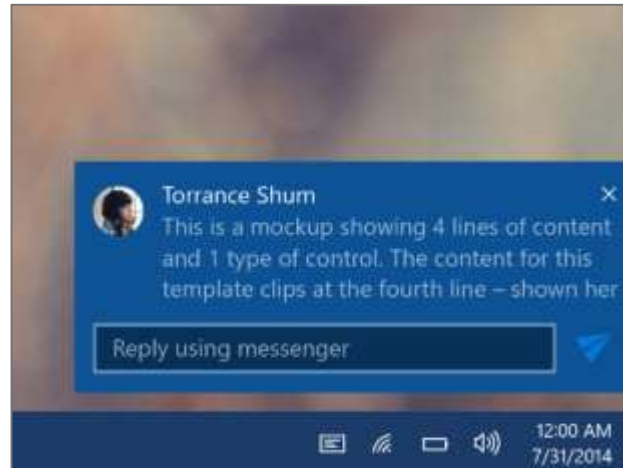
Send from (foreground/background) app  
This includes desktop apps with "AppUserModelID"

## **Push**

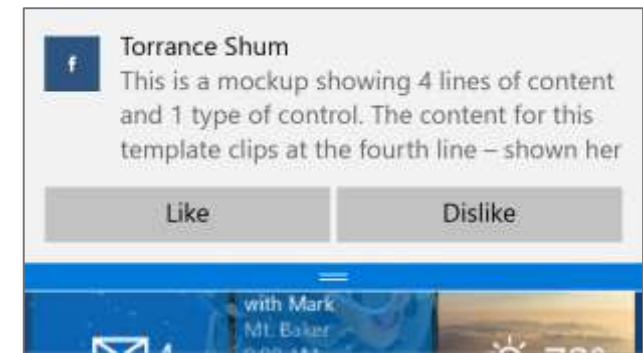
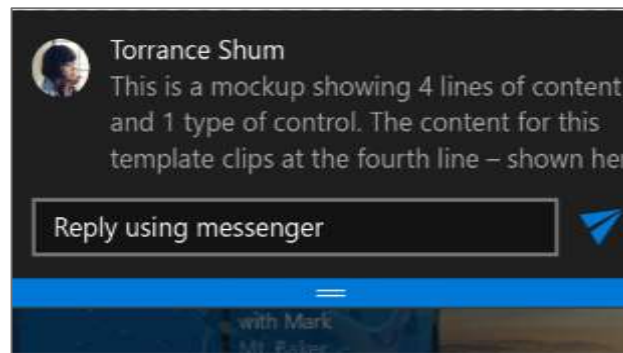
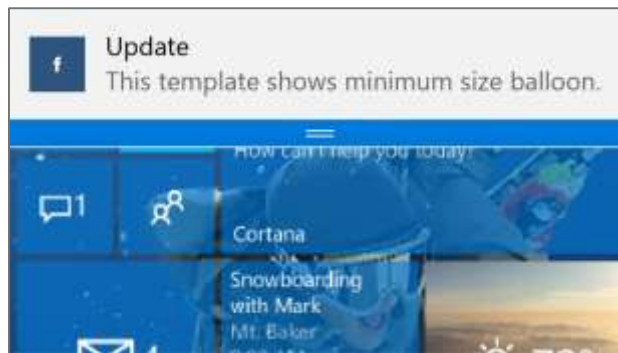
Use push services

# Interactive toast

Desktop



Mobile





# A Developer's Guide to Windows 10 Preview

<http://www.microsoftvirtualacademy.com/training-courses/a-developers-guide-to-windows-10-preview>



