# Creating the App and First View



Gill Cleeren
ARCHITECT

@gillcleeren www.snowball.be

# Agenda



Xamarin. Android fundamentals

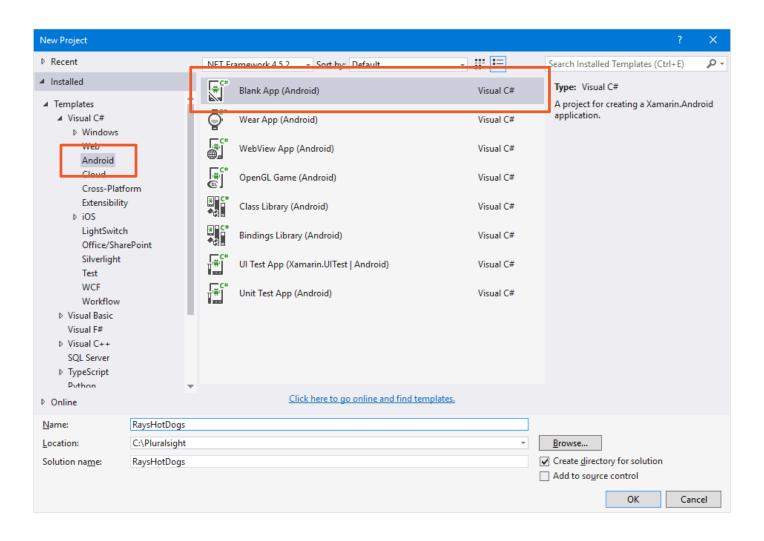
**Creating our first view** 



#### Xamarin. Android Fundamentals

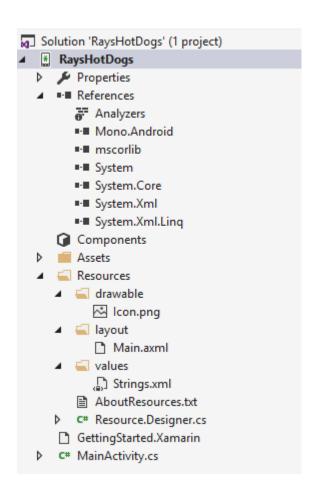


# Available Project Templates





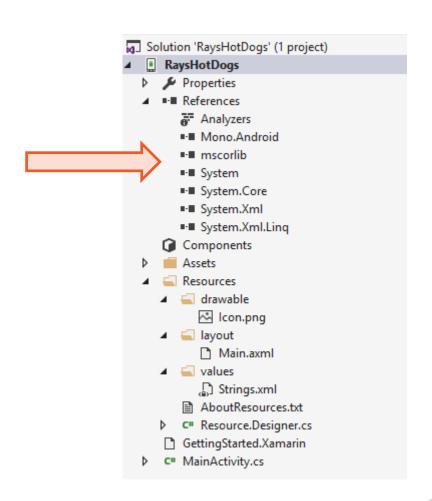
## File > New Project





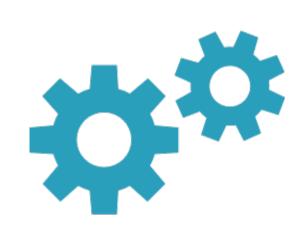
#### References

References are created to some System assemblies as well as to Mono.Android





### Base Concepts in Android

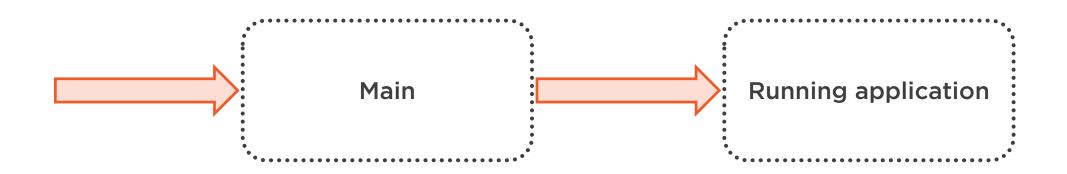


**Activities** 



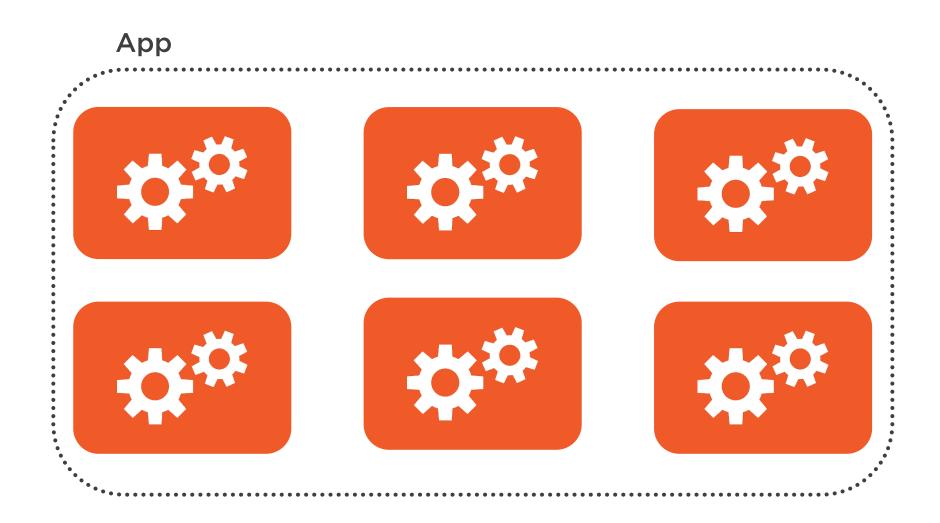


# The Typical "Main" Method

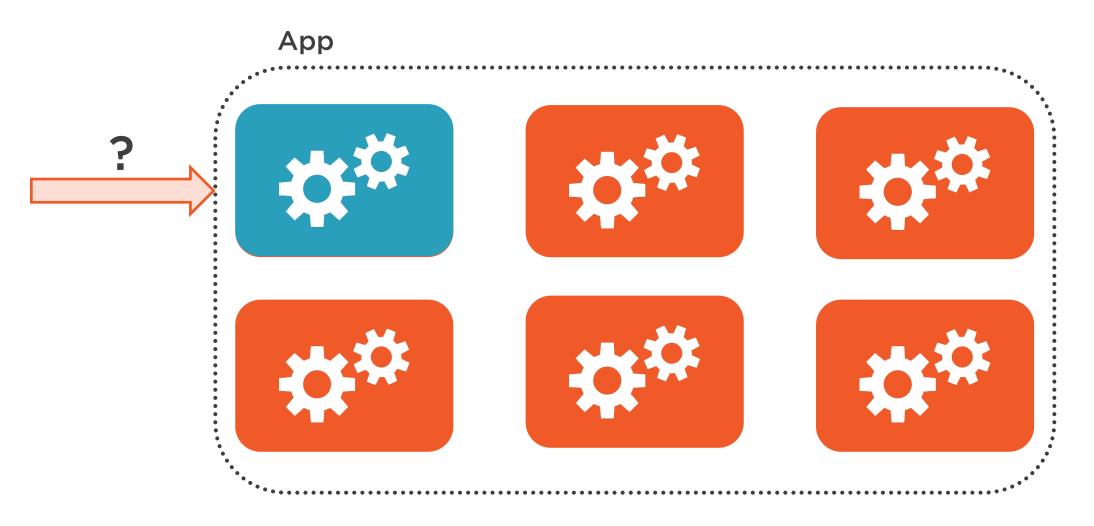




### Activities



# Starting an App



#### Activities

All activities inherit from base Activity

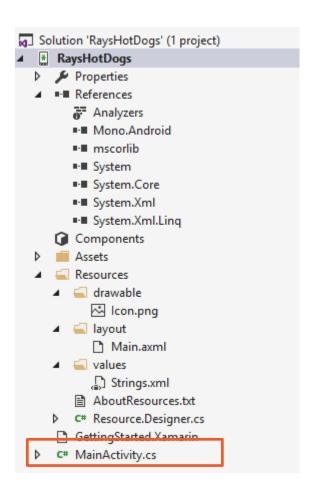
Functionality for a single task

Corresponds to one screen

Specific lifecycle

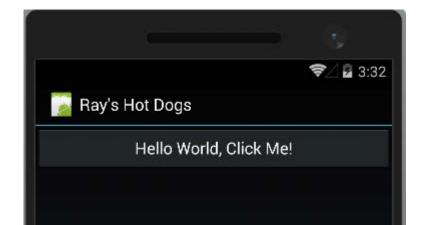


## The Default Activity



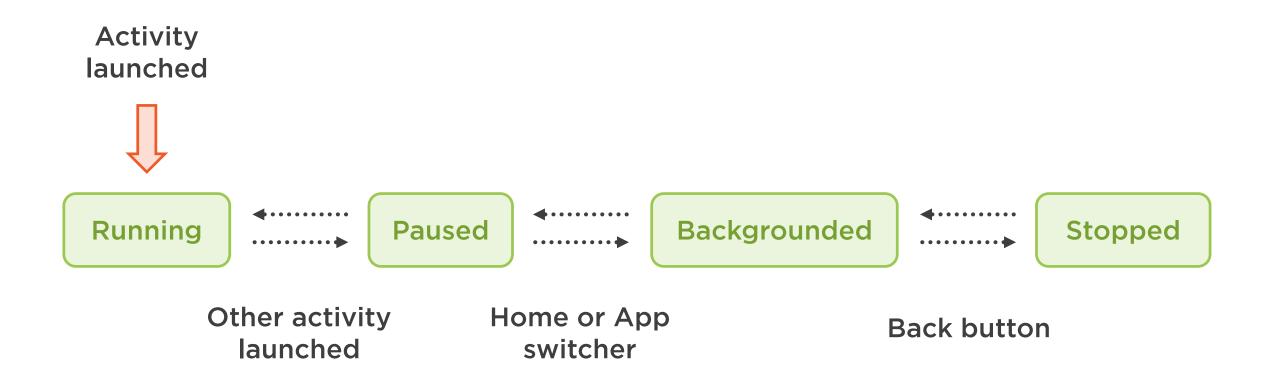


### A Sample Activity

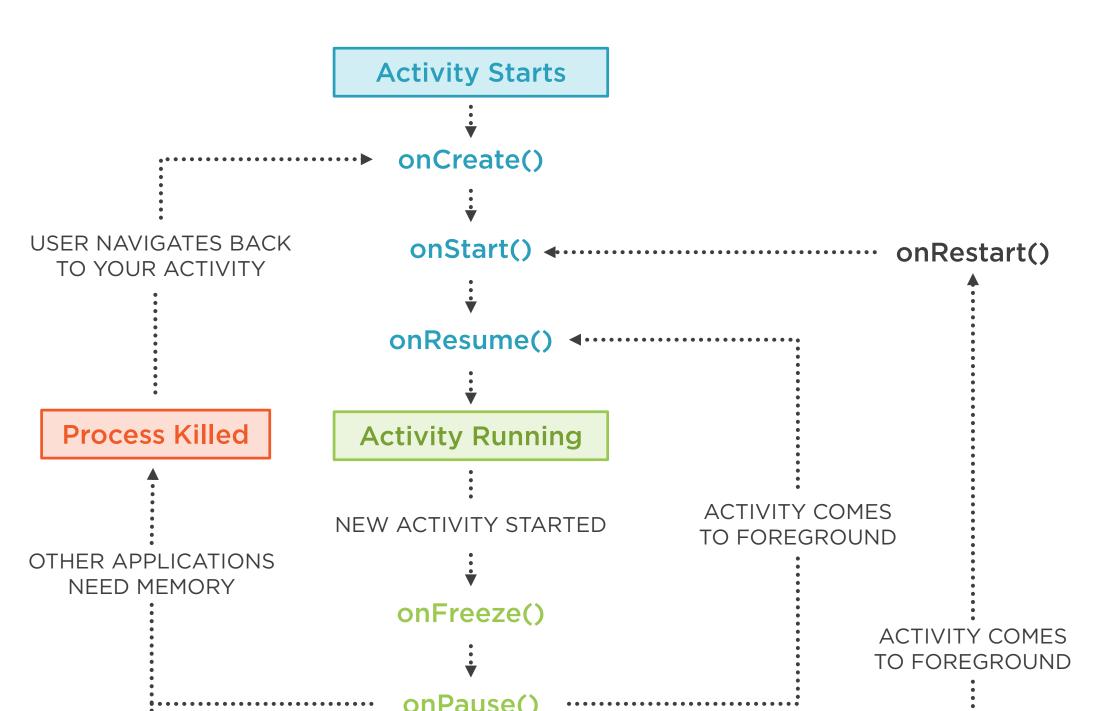




# Activity States

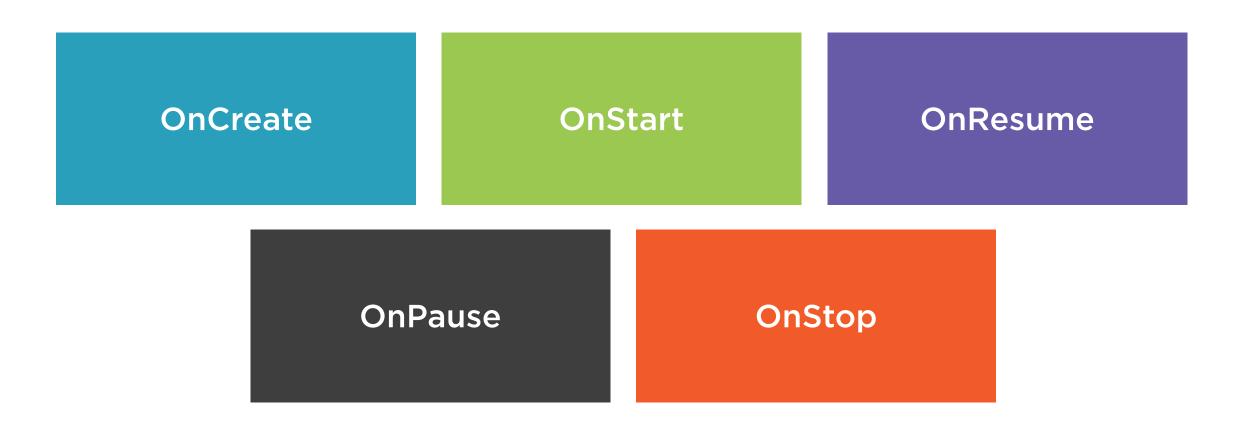








# Important Lifecycle Methods





#### OnCreate

```
[Activity(Label = "RaysHotDogs", MainLauncher = true, Icon = "@drawable/icon")]
public class MainActivity : Activity
    protected override void OnCreate(Bundle bundle)
        base.OnCreate(bundle);
        // Set our view from the "main" layout resource
        SetContentView(Resource.Layout.Main);
```



# Views



Visual part



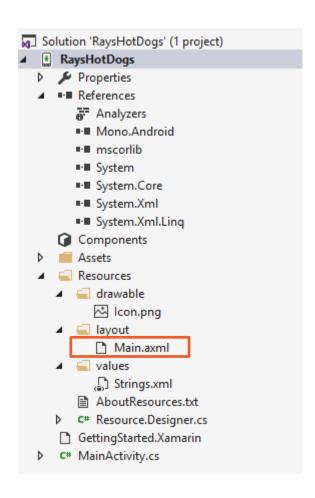
\*.axml



Live in Resources\Layout

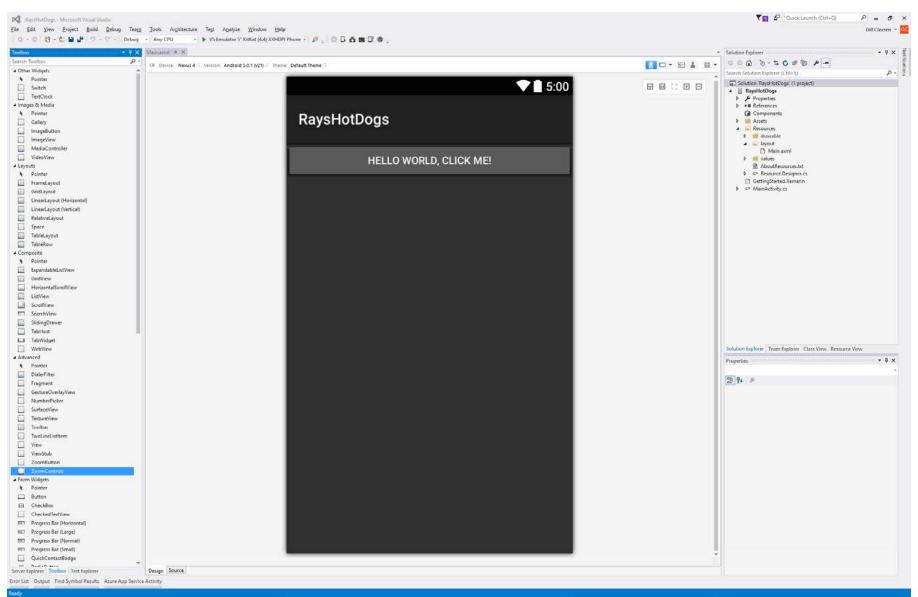


#### The Default View



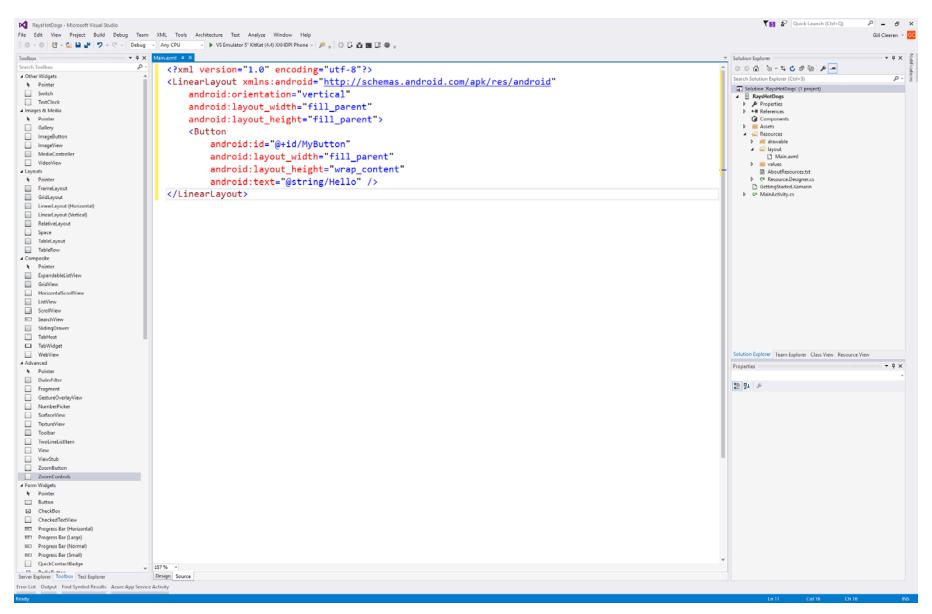


## The Designer in Visual Studio





#### The Source Code Editor





## Sample View Code

```
<?xml version="1.0" encoding="utf-8"?>
                                                               <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
                                                                   android:orientation="vertical"
                                          6:00
                                                                   android:layout width="fill parent"
                                                                   android:layout height="fill parent">
   RaysHotDogs
                                                                   <TextView
                                                                       android:text="How many hot dogs do you want to order?"
                                                                       android:textAppearance="?android:attr/textAppearanceLarge"
How many hot dogs do you want to
                                                                       android:layout width="match parent"
order?
                                                                       android:layout height="wrap content"
                                                                       android:id="@+id/AmountTextView" />
                                                                   <EditText ■
                                                                       android:layout width="match parent"
                                                                       android:layout height="wrap content"
                    ORDER NOW!
                                                                       android:id="@+id/amountEditText" />
                                                                   <Button
                                                                       android:id="@+id/OrderButton"
                                                                       android:layout_width="fill_parent"
                                                                       android:layout height="wrap content"
                                                                       android:text="Order now!" />
                                                               </LinearLayout>
```



## Linking the View and the Activity

```
Solution 'RaysHotDogs' (1 project)
                                                                                                   RaysHotDogs
                                                                                                     Properties

■ ■ References

                                                                                                       Analyzers
                                                                                                       ■- ■ Mono.Android
                                                                                                       ■ mscorlib
                                                                                                       ■-■ System
                                                                                                       ■·■ System.Core
                                                                                                       ■·■ System.Xml
protected override void OnCreate(Bundle bundle)
                                                                                                       ■·■ System.Xml.Ling

    Components

                                                                                                     Assets
       base.OnCreate(bundle);
                                                                                                     Resources
                                                                                                     drawable

    Icon.png

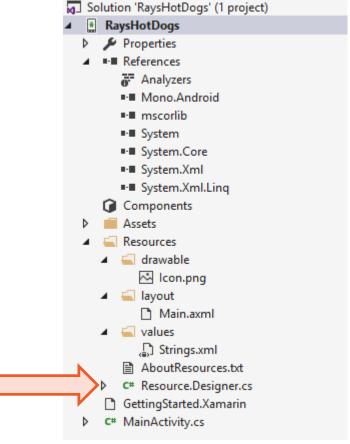
                                                                                                     layout
       SetContentView(Resource.Layout.Main);
                                                                                                         Main.axml
                                                                                                       values
                                                                                                         Strings.xml
                                                                                                       ■ AboutResources.txt
                                                                                                     ▶ C# Resource.Designer.cs
                                                                                                     ☐ GettingStarted.Xamarin
                                                                                                   ▶ C# MainActivity.cs
```



## The Magic: Resource.Designer.cs

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("Xamarin.Android.Build.Tasks", "1.0.0.0")]
2 references
public partial class Resource
    0 references
    static Resource()...
    0 references
    public static void UpdateIdValues()...
    public partial class Attribute...
    public partial class Drawable...
    2 references
    public partial class Id...
                                                                                                                            Assets
    3 references
    public partial class Layout
        // aapt resource value: 0x7f030000
        public const int Main = 2130903040;
        static Layout()...
        0 references
         private Layout()
```

public partial class String...





```
public partial class Layout
{
    // aapt resource value: 0x7f030000
    public const int Main = 2130903040;

    static Layout()
    {
        global::Android.Runtime.ResourceIdManager.UpdateIdValues();
    }
}
```

The "Magic": Resource.Designer.cs



## Accessing Controls from Code

```
<Button
    android:id="@+id/MyButton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/Hello"
    />
```

```
public partial class Resource
{
   public partial class Id
   {
     public const int MyButton = 2131034112;

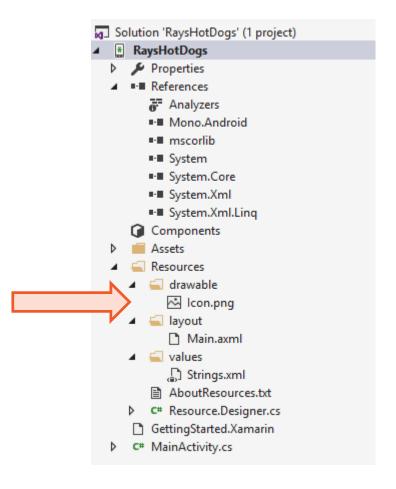
     static Id()
     {
        global::Android.Runtime.ResourceIdManager.UpdateIdValues();
     }
   }
}
```



### Accessing Controls from Code

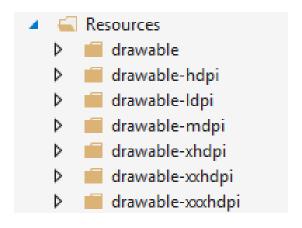
```
protected override void OnCreate(Bundle bundle)
    base.OnCreate(bundle);
    // Set our view from the "main" layout resource
    SetContentView(Resource.Layout.Main);
    // Get our button from the layout resource,
    // and attach an event to it
    Button button = FindViewById<Button>(Resource.Id.MyButton);
    button.Click +=
      delegate { button.Text = string.Format("{0} clicks!", count++); };
```

#### Drawables



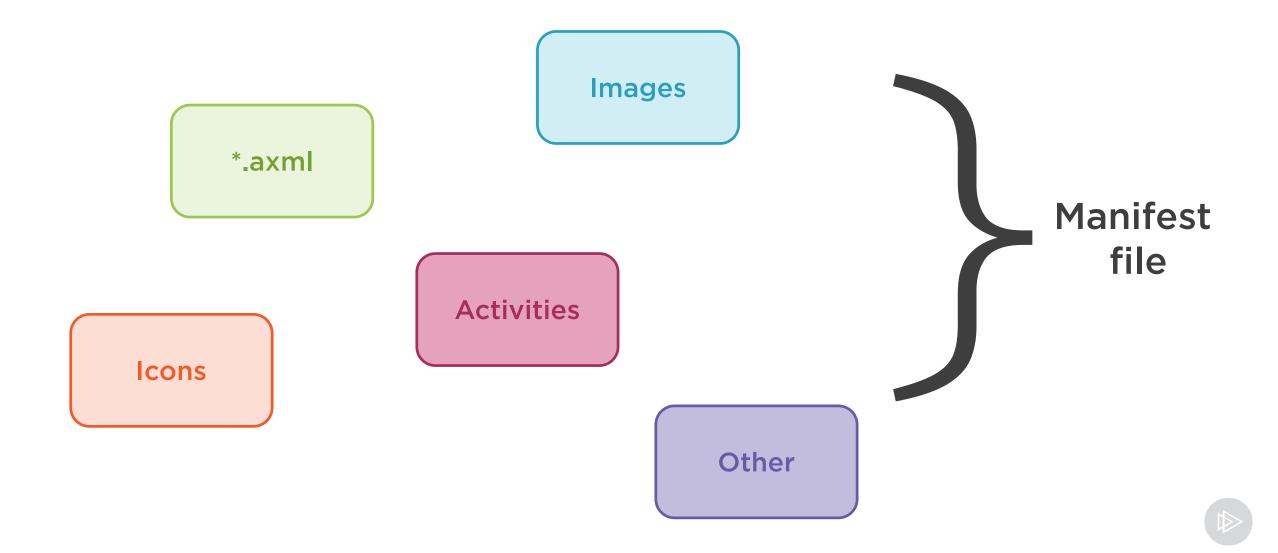


## Supporting Many Resolutions

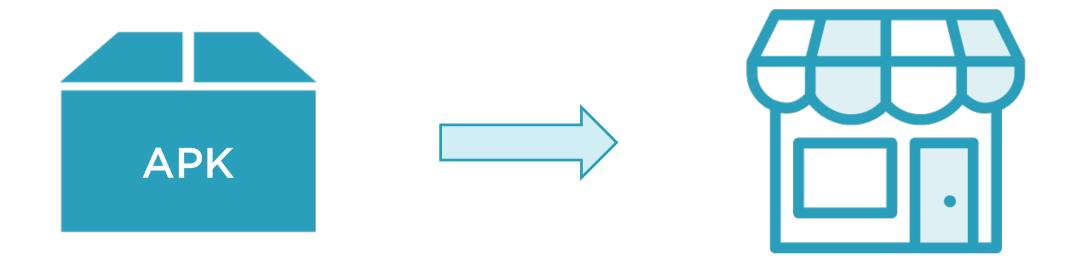




# Application Manifest



# Compilation & Deployment



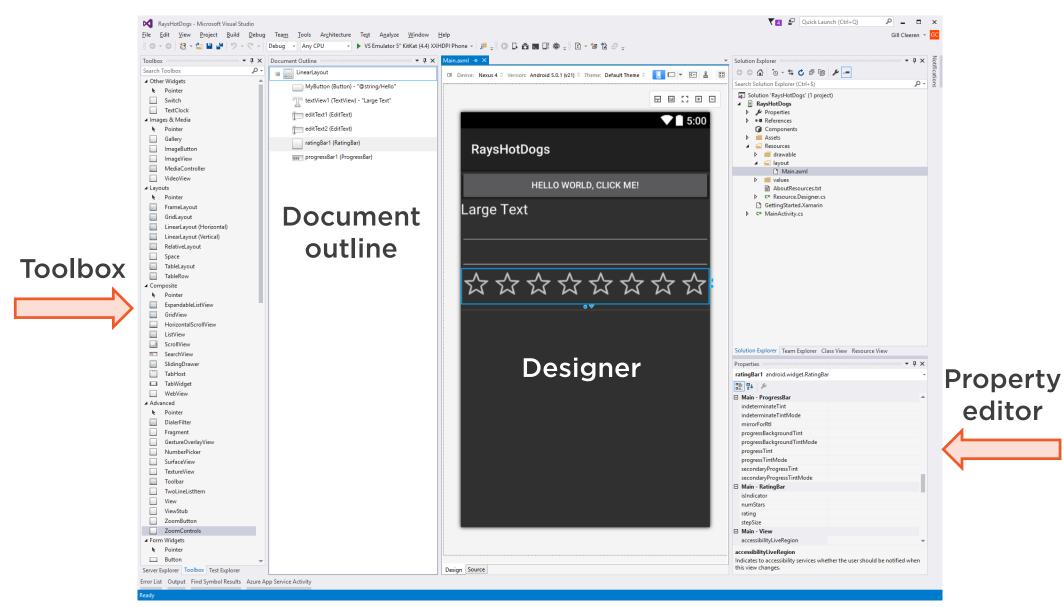




# Creating Our First View



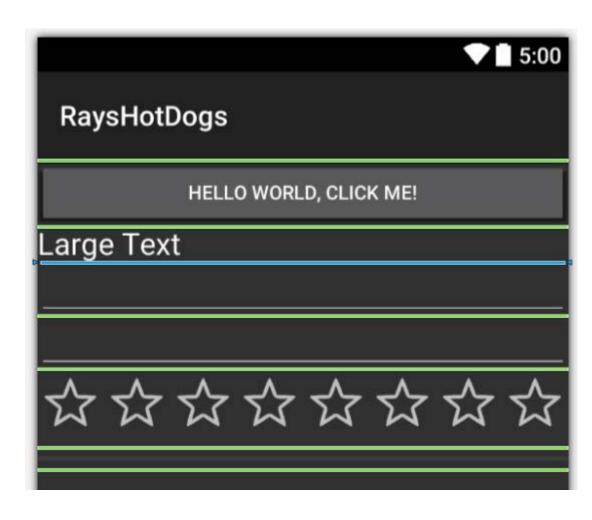
## Designer Features





editor

# Drag and Drop





#### **Demo** Taking a Look at the Designer



### Layout Views

LinearLayout RelativeLayout

TableLayout GridView



## LinearLayout



#### Hot Dog Name

#### **Short Description**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque ut nisi egestas, auctor arcu quis, tristique urna. Nunc et rutrum nunc, commodo feugiat nisi. Fusce pharetra sollicitudin sapien, nec facilisis risus consectetur vel. Interdum et malesuada fames ac ante ipsum primis in faucibus. Praesent dignissim tincidunt felis, eget tempus ligula egestas vel. Donec massa est, luctus eget facilisis eu, faucibus eu elit. Suspendisse potenti. Cras aliquam lectus ut elit varius consectetur. Proin lacinia ante at euismod elementum. Fusce dictum mauris eu sagittis facilisis. Vivamus a dignissim enim, id luctus leo.

**Price: 1111** 

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    android:orientation="vertical"
    android:layout width="fill parent"
    android:layout height="fill parent"
    android:id="@+id/mainLinearLayout">
    <TextView
        android:text="Hot Dog Name"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:id="@+id/hotDogNameTextView" />
    <TextView
        android:text="Short Description"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:id="@+id/shortDescriptionTextView" />
    <TextView
        android:layout width="match parent"
        android:layout height="wrap content"
        android:id="@+id/descriptionTextView"
        android:text="Lorem ipsum dolor sit amet"/>
    <TextView
        android:text="Price: 1111"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:id="@+id/priceTextView" />
</LinearLayout>
```



#### RelativeLayout

Cancel 1 Order now!

```
<RelativeLayout</pre>
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout height="wrap content">
   <Button
        android:text="Cancel"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:id="@+id/cancelButton"
        android:layout alignParentLeft="true"
        android:layout centerInParent="true"
        android:layout marginTop="20dp" />
   <EditText
        android:inputType="number"
        android:text="1"
        android:layout_width="30dp"
        android:layout height="40dp"
        android:id="@+id/amountEditText"
        android:layout centerHorizontal="true"
        android:layout centerVertical="true" />
    < Button
        android:text="Order now!"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:id="@+id/orderButton"
        android:layout alignParentRight="true"
        android:layout centerInParent="true" />
</RelativeLayout>
```



## TableLayout

Cancel 1 Order now!

```
≺TableLayout
    android:layout width="fill parent"
    android:layout_height="wrap_content"
    android:stretchColumns="1">
    <TableRow>
        <Button
            android:layout_column="0"
            android:text="Cancel"
            android:layout width="wrap content"
            android:layout height="wrap content"
            android:id="@+id/cancelButton" />
        <EditText
            android:layout column="2"
            android:inputType="number"
            android:text="1"
            android:layout width="30dp"
            android:layout height="40dp"
            android:id="@+id/amountEditText" />
        <Button
            android:layout column="3"
            android:text="Order now!"
            android:layout_width="wrap_content"
            android:layout height="wrap content"
            android:id="@+id/orderButton" />
    </TableRow>
</TableLayout>
```

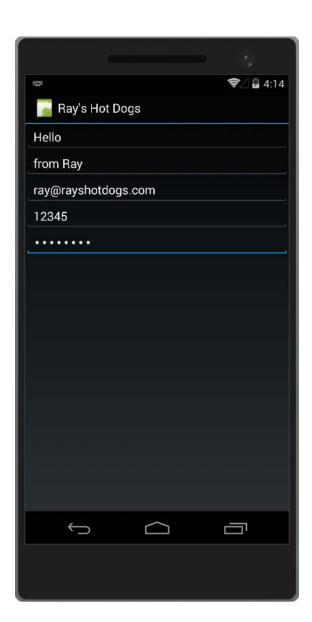


#### Base Views

**Button TextView** EditText CheckBox RadioButton **ImageView** 



#### EditText



```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText1" />
<EditText
    android:inputType="textMultiLine"
    android:layout width="match parent"
    android:layout height="wrap content"
    android:id="@+id/editText2" />
<EditText
    android:inputType="textEmailAddress"
    android:layout width="match parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText3" />
<EditText
    android:inputType="number"
    android:layout width="match parent"
    android:layout height="wrap content"
    android:id="@+id/editText5" />
<EditText ■
    android:inputType="textPassword"
    android:layout width="match parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText6" />
```



#### DatePicker and TimePicker



```
<DatePicker
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/datePicker1" />
<TimePicker
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/timePicker1" />
```



#### Other Controls



```
<Switch
    android:layout width="match parent"
    android:layout_height="wrap_content"
    android:id="@+id/switch1"
    android:text="Add ketchup? " />
<ProgressBar</pre>
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
   android:id="@+id/progressBar1"
    android:progress="55" />
<RatingBar</p>
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/ratingBar1"
    android:rating="3" />
```



```
protected override void OnCreate(Bundle bundle)

base.OnCreate(bundle);

SetContentView(Resource.Layout.Main);

Button button = FindViewById<Button>(Resource.Id.MyButton);

button.Click += delegate
{
    //Event handling code goes here
};
}
```

### Handing UI Events





**Demo** Creating the Detail View

#### Summary



Activities and Views are the heart of any Xamarin. Android app

Visual Studio offers a good editing experience to create Views

