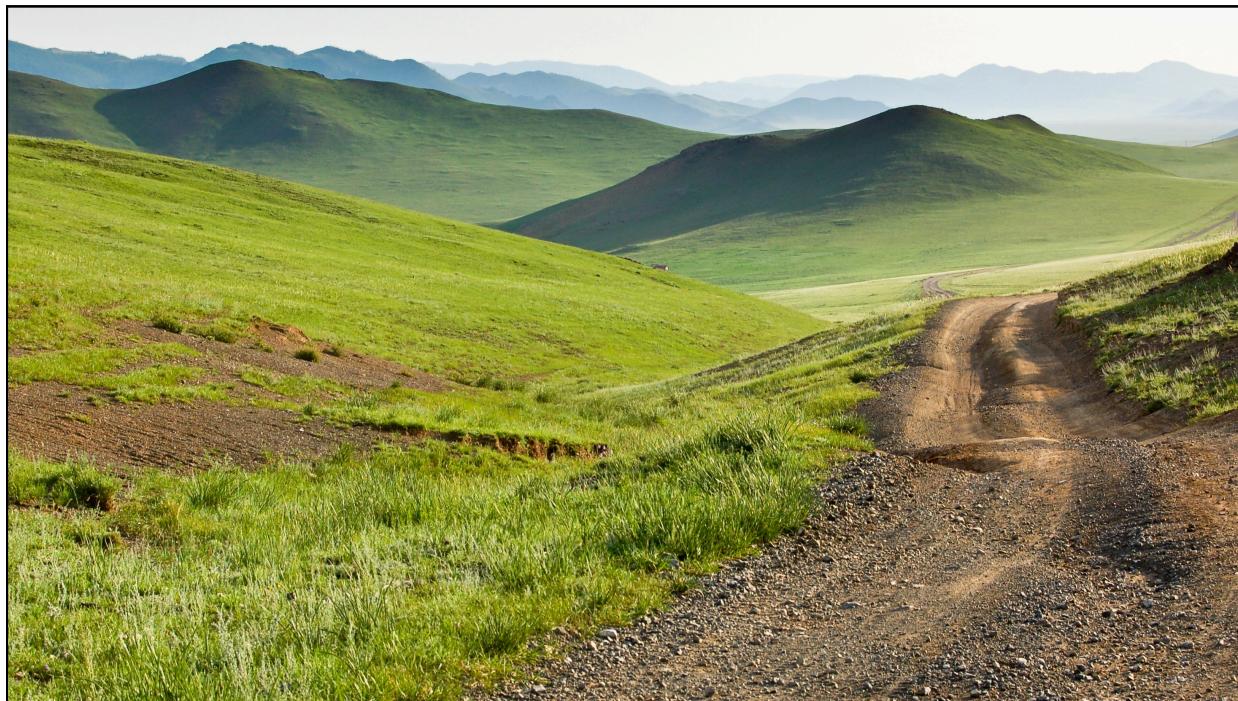


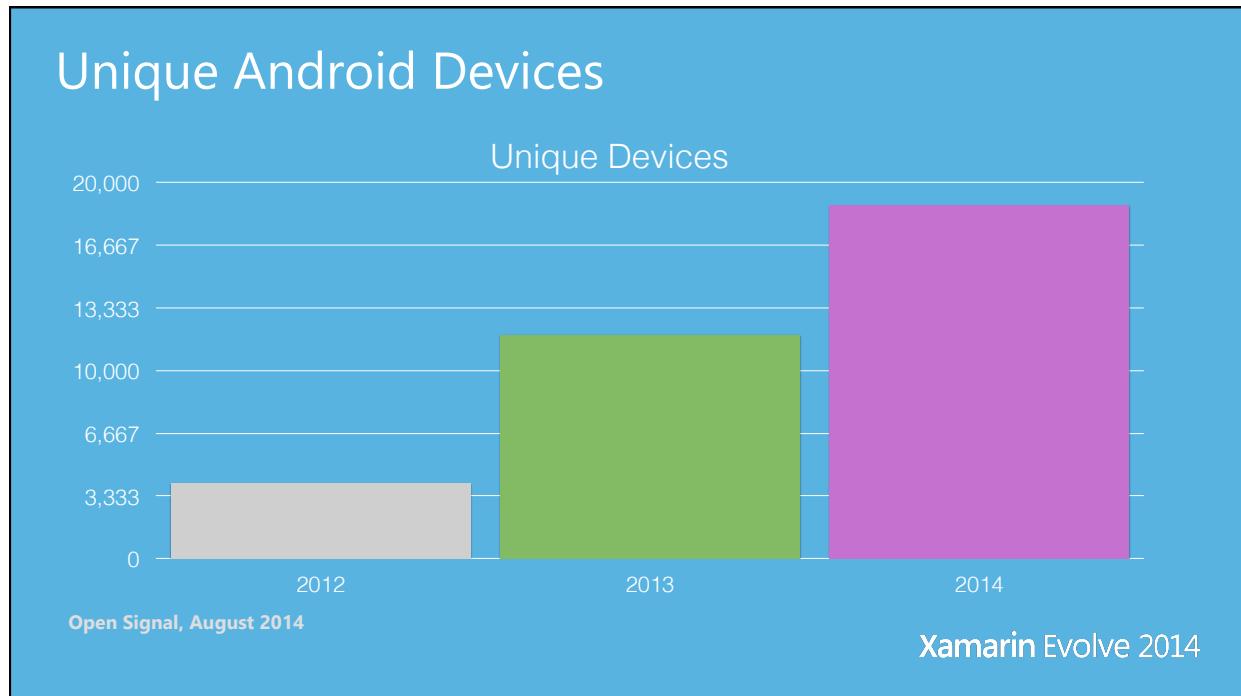
A close-up portrait of a man with short brown hair and blue eyes, wearing thin-framed glasses and a light gray t-shirt. He is smiling warmly at the camera.

Xamarin Evolve 2014

Designing Android UIs for the Ever Changing Device Landscape

Michael Stonis
michael.stonis@xamarin.com





Big Opportunity

Where do we get started?

- Design Fundamentals
- Layout Management
- Android Fragments

Xamarin Evolve 2014

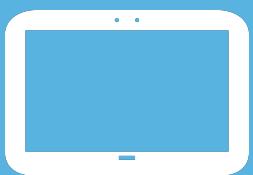
Design Fundamentals

Xamarin Evolve 2014

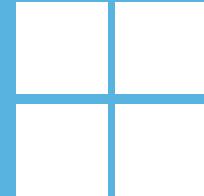
The Big Three



Screen Size



Orientation



Pixel Density

Xamarin Evolve 2014

Density Independent
Pixels

Xamarin Evolve 2014

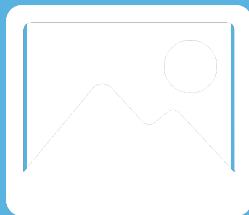
Density Independent Pixel (DIP)

DPI	Pixels Per DIP
160	
320	

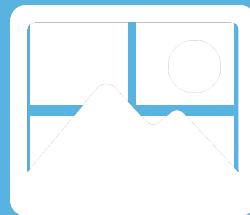
- Represents a single pixel on a 160 dpi screen
- Scales on screens with higher or lower density
- Use as a standard unit of measurement for layouts and drawables
- *Avoid Using Pixels Like the Plague!*

Xamarin Evolve 2014

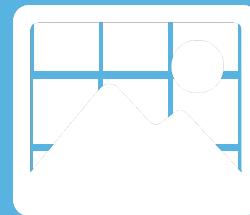
Density Independent Pixel



160 dpi
MDPI
1x

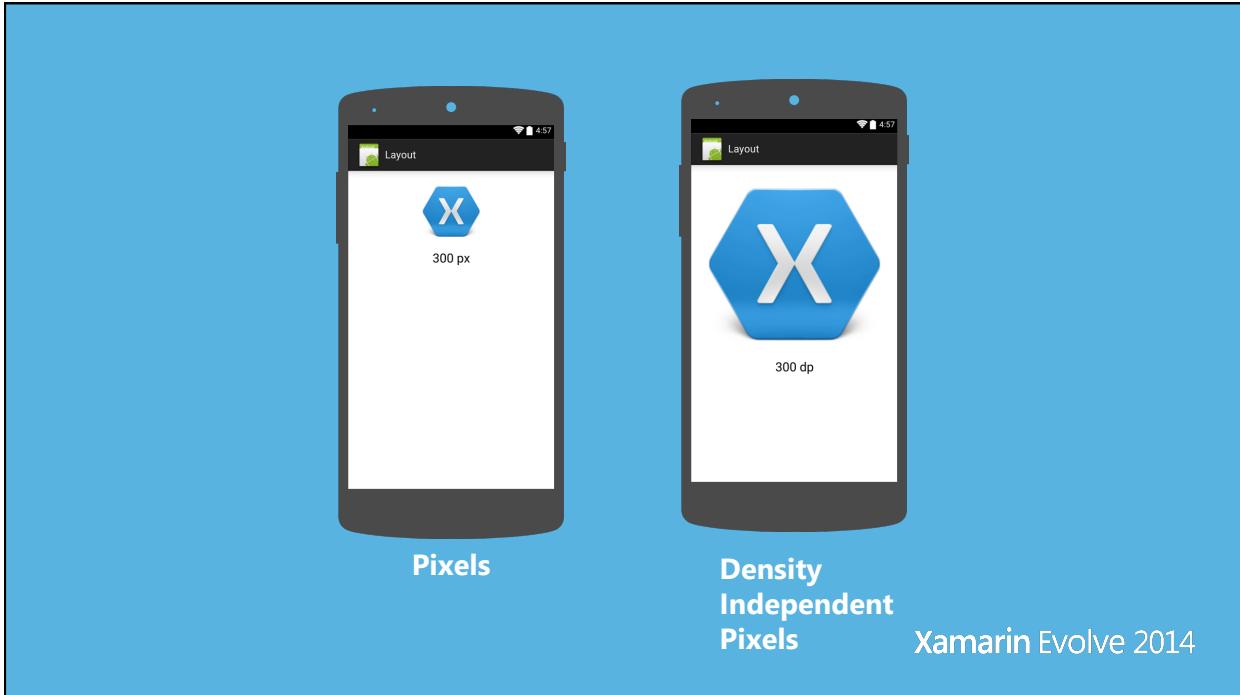


320 dpi
XHDPI
2x

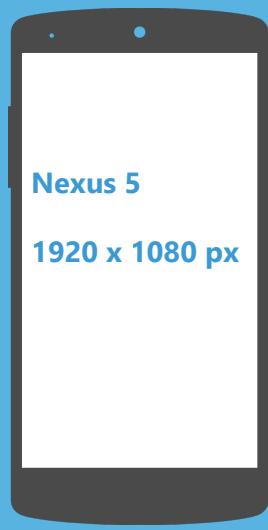


480 dpi
XXHDPI
3x

Xamarin Evolve 2014



Density Independent Pixels



Xamarin Evolve 2014

Density Independent Pixels



1 inch

445 pixels

Xamarin Evolve 2014

Warning

Math Ahead...Approach
with Caution

Xamarin Evolve 2014

Calculating DIPs

$$\text{DIP} = \frac{\text{Resolution (px)} \times 160}{\text{dpi}}$$

Xamarin Evolve 2014

Density Independent Pixels



$$690 \text{ Vertical} = \frac{1920 \text{ px} \times 160}{445 \text{ dpi}}$$

$$388 \text{ Horizontal} = \frac{1080 \text{ px} \times 160}{445 \text{ dpi}}$$

Xamarin Evolve 2014

Density Independent Pixel



160 dpi
MDPI
1x

240 dpi
HDPI
1.5x

320 dpi
XHDPI
2x

480 dpi
XXHDPI
3x

Xamarin Evolve 2014

Density Independent Pixels



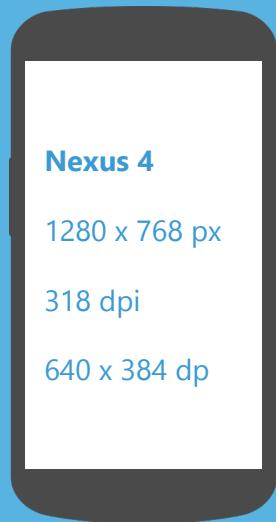
Nexus 5
1920 x 1080 px
445 dpi
640 x 360 dp

640 Vertical = $\frac{1920 \text{ px} \times 160}{480 \text{ dpi}}$

360 Horizontal = $\frac{1080 \text{ px} \times 160}{480 \text{ dpi}}$

Xamarin Evolve 2014

Device Comparison

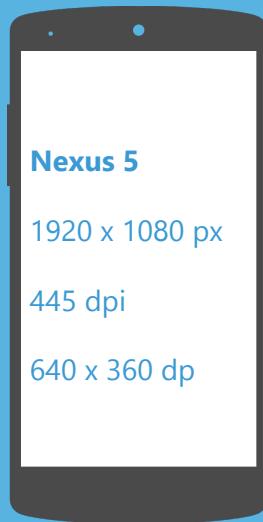


Nexus 4

1280 x 768 px

318 dpi

640 x 384 dp



Nexus 5

1920 x 1080 px

445 dpi

640 x 360 dp

Xamarin Evolve 2014

Device Comparison

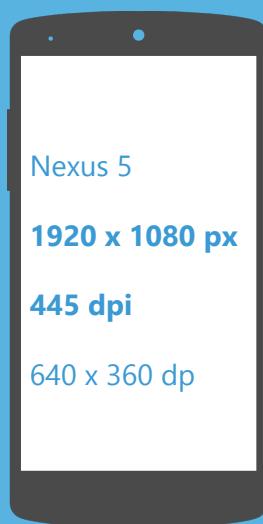


Nexus 4

1280 x 768 px

318 dpi

640 x 384 dp



Nexus 5

1920 x 1080 px

445 dpi

640 x 360 dp

Xamarin Evolve 2014

Device Comparison



Nexus 4

1280 x 768 px

318 dpi

640 x 384 dp



Nexus 5

1920 x 1080 px

445 dpi

640 x 360 dp

Xamarin Evolve 2014

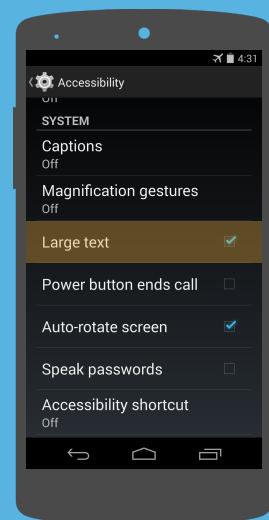


Xamarin Evolve 2014

Scale Independent Pixels (SIP)

Scaling for Fonts

- Use this when specifying sizes for fonts
- Will scale depending on the device's pixel density similar to a DIP
- Scales appropriately to the user's font size preferences



Xamarin Evolve 2014

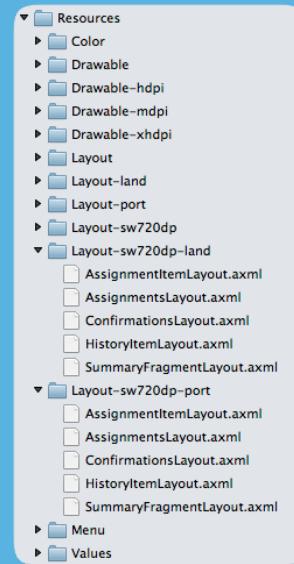
Android Resource Qualifiers

Xamarin Evolve 2014

Android Resource Qualifiers

Different Strokes for Different Folks

- Provides us the ability to use different resources for different configurations
- Qualifiers Apply to Resource Folders
 - drawable
 - layout
 - etc.
- *Do not apply to resources directly*



Xamarin Evolve 2014

Android Resource Qualifiers

Qualifier Format

<resource type>-<qualifier>

Xamarin Evolve 2014

Android Resource Qualifiers

Example

drawable-large-xhdpi

A drawable resource on a large screen size with an Extra-high-density screen

Xamarin Evolve 2014

Android Resource Qualifiers

Example

layout-zh-land

A layout resource for the Chinese language presented in landscape orientation

Xamarin Evolve 2014

Android Resource Qualifiers

Resource Qualifier Types

- Region & Language
 - Mobile Country Code (MCC)
 - Language and Language Region
- Device Configuration
 - Portrait
 - Landscape
- Device Size Resolution
 - Screen Size
 - Pixel Density
- Device Features
 - UI Mode
 - Hardware Software Keyboard

Xamarin Evolve 2014

Android Resource Qualifiers

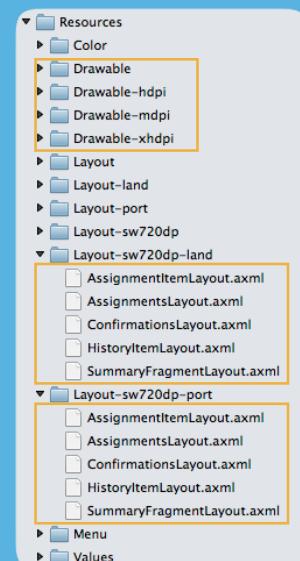
bit.ly/1toKVDF

Xamarin Evolve 2014

Android Resource Qualifiers

Different Strokes for Different Folks

- Provides us the ability to use different resources for different configurations
- Qualifiers Apply to Resource Folders
 - drawable
 - layout
 - etc.
- *Do not apply to resources directly*



Xamarin Evolve 2014

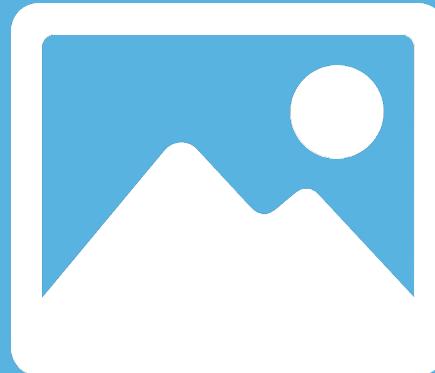
Resolving Resources



160 dpi
MDPI
1x



240 dpi
HDPI
1.5x



480 dpi
XXHDPI
3x

Xamarin Evolve 2014

Layout Management

Xamarin Evolve 2014

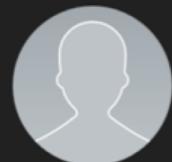
Layout Management

Xamarin Android Designer

- Full WYSIWYG Editor for Android
 - Control Management
 - Document Explorer
 - XML Editor
- Layouts are standard Android
- Supports Editing *Multiple* Layouts

Android L (v21) | (All languages) | Mode

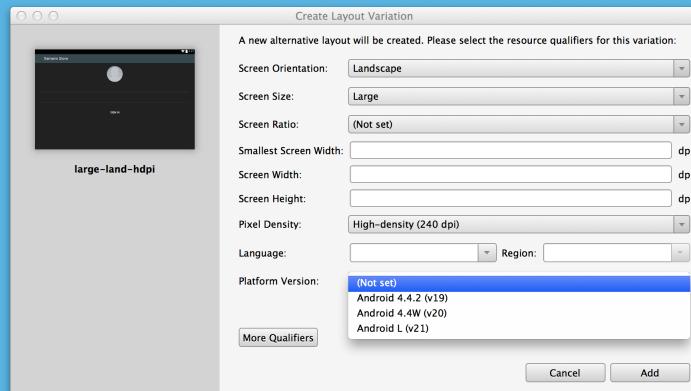
Xamarin Store



Xamarin Evolve 2014

Layout Management

Multiple Layouts



Xamarin Evolve 2014

Demo

Xamarin Evolve 2014

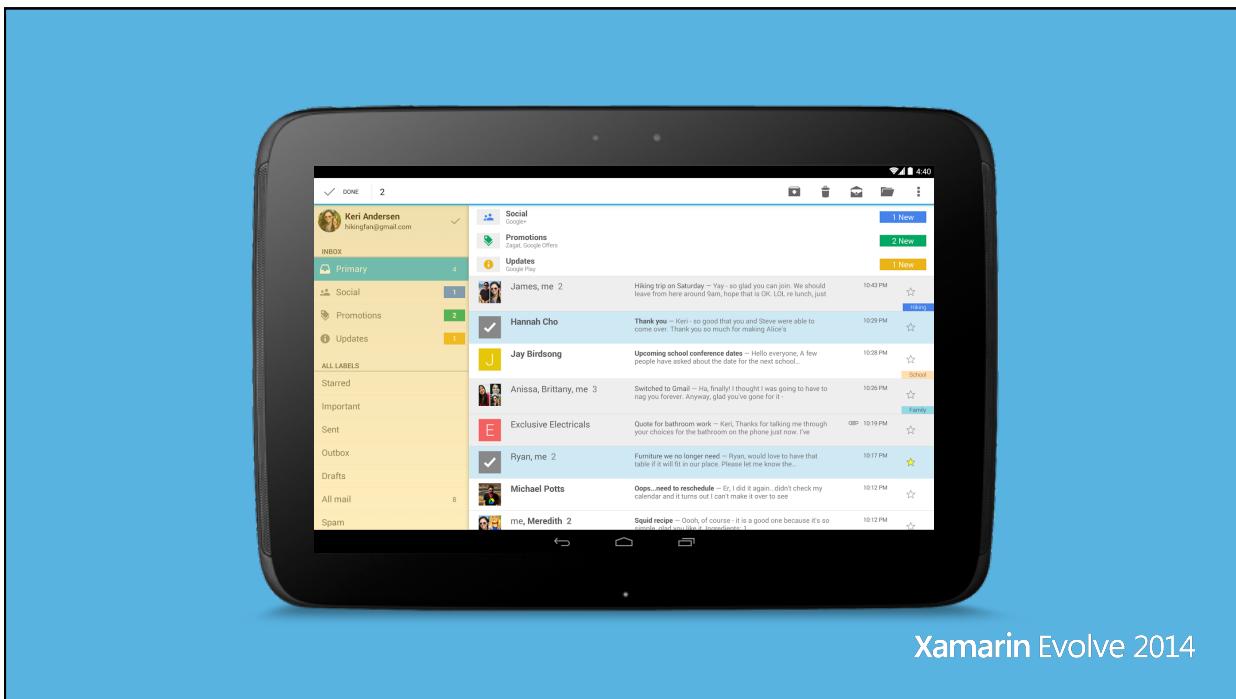
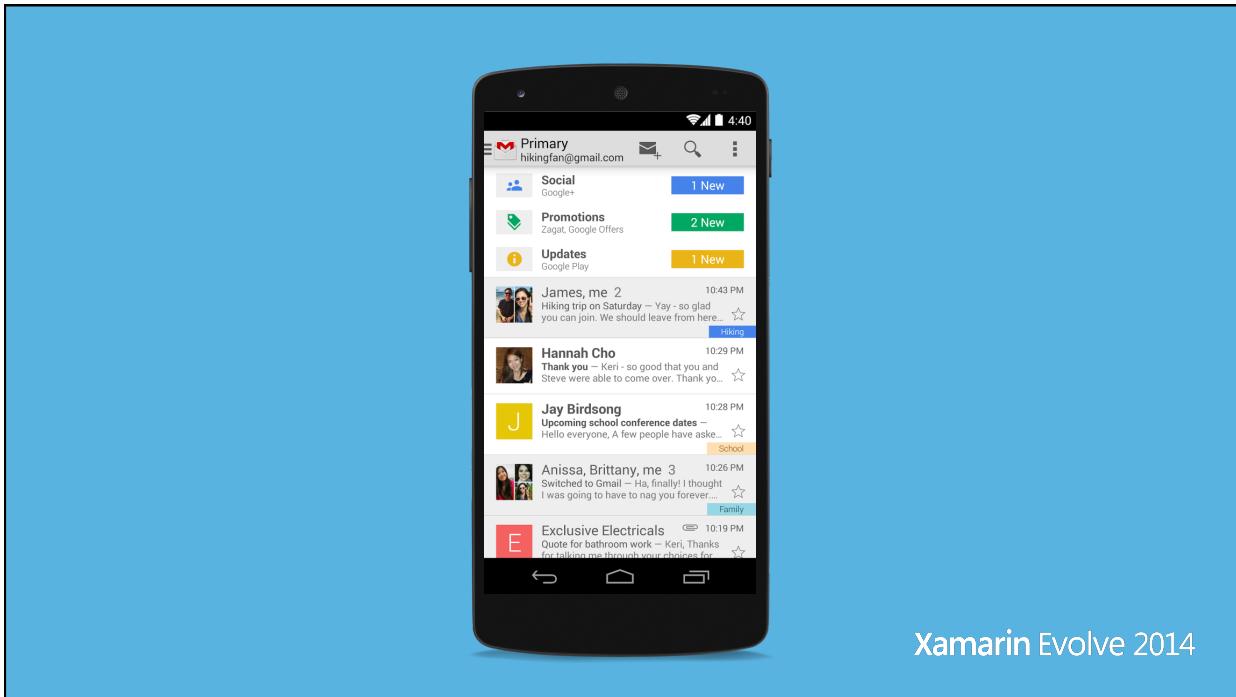
Layout Management

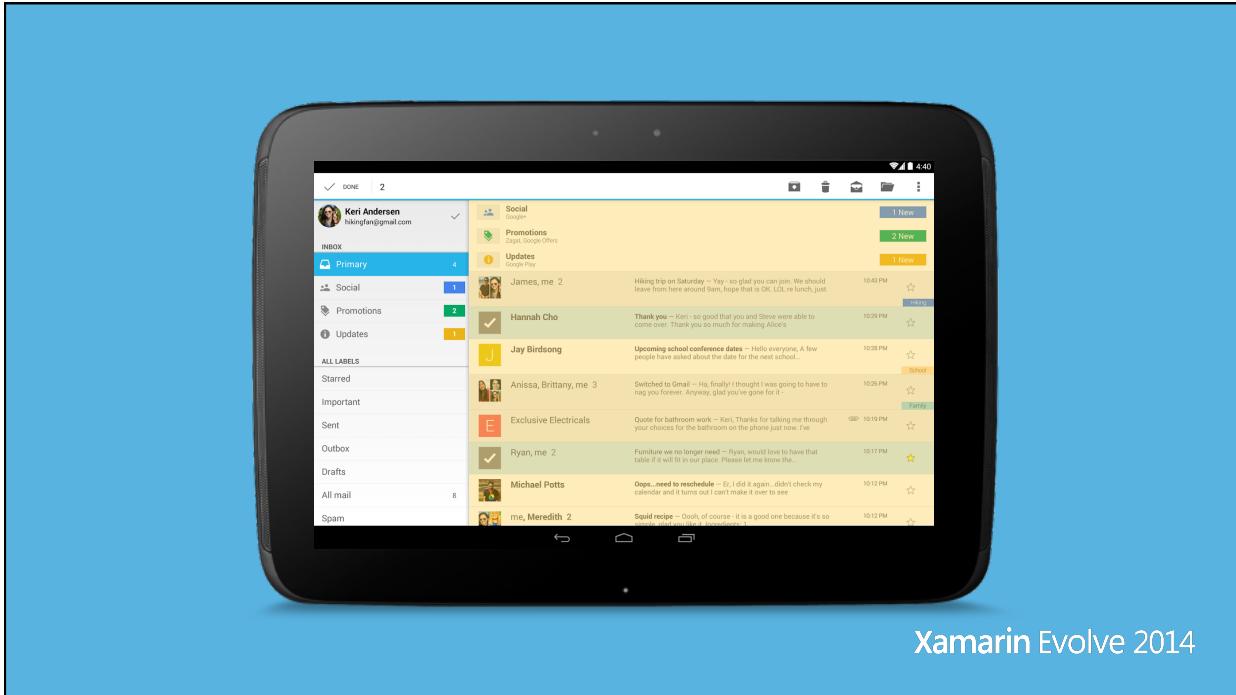
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1" android:versionName="1.0"
    package="com.xamarin.university.mobilenav.android">
    <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="14" />
    <application android:icon="@drawable/ic_launcher" android:label="ActionBar Tabs">
        <supports-screens android:compatibleWidthLimitDp="integer"/>
    </application>
</manifest>
```

Xamarin Evolve 2014

// Fragments //

Xamarin Evolve 2014





Android Fragments

Application Building Blocks

- Reusable User-Interface components
- Conceptually similar to an Activity with its own lifecycle and layout
- Create in XML or programmatically

Xamarin Evolve 2014

Android Fragment Creation

```
public class FragmentToAdd : Fragment
{
    public override void OnCreate (Bundle savedInstanceState)
    {
    }

    public override View OnCreateView (LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState)
    {
    }

    public override void OnPause ()
    {
    }
}
```

Xamarin Evolve 2014

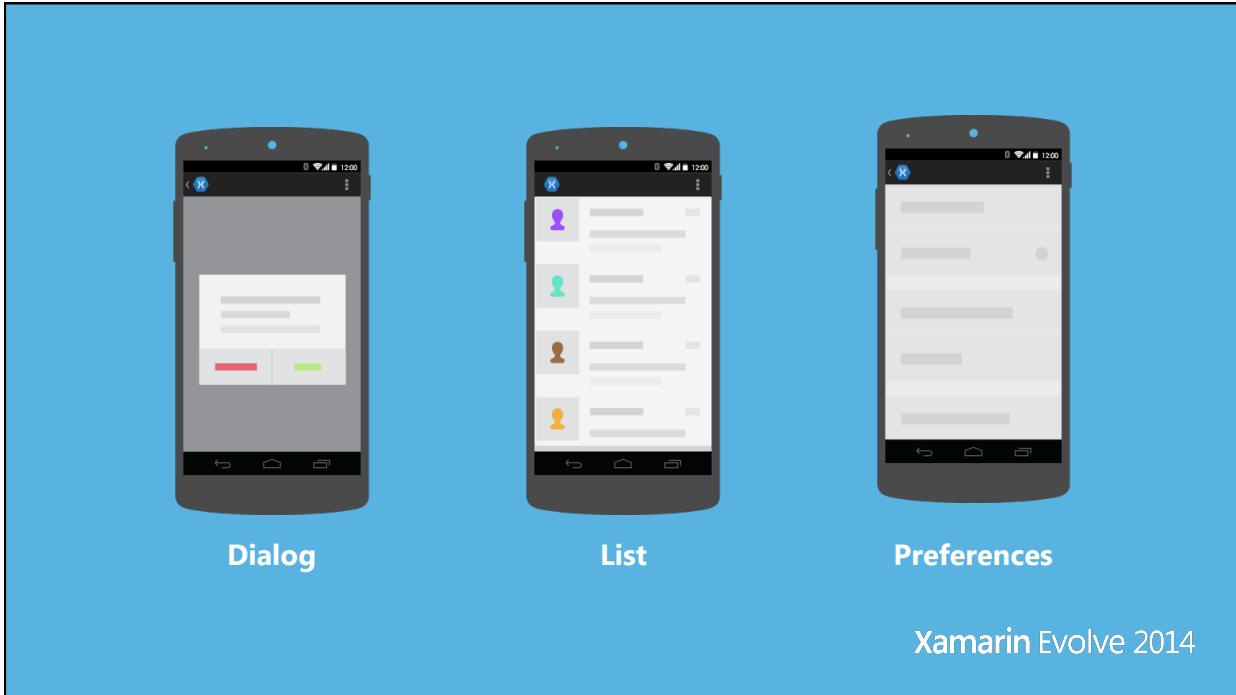
Android Fragment Key Lifecycle Methods

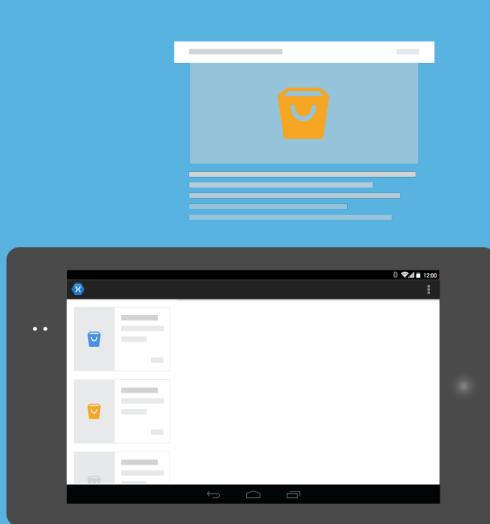
```
public class FragmentToAdd : Fragment
{
    public override void OnCreate (Bundle savedInstanceState)
    {
    }

    public override View OnCreateView (LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState)
    {
    }

    public override void OnPause ()
    {
    }
}
```

Xamarin Evolve 2014





Xamarin Evolve 2014

Adding Android Fragments via XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:minWidth="25px"
    android:minHeight="25px">
    <fragment
        class="fragments.NavigationFragment"
        android:id="@+id/navigation"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</LinearLayout>
```

Xamarin Evolve 2014

Adding Android Fragments via Code

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.Add(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014

Fragment Manager

- Provides Fragment Discovery
 - FindFragmentById and FindFragmentByTag
- Manage Fragment Navigation History
- Creation of Transactions

Xamarin Evolve 2014

Adding Android Fragments via Code

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.Add(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014

Fragment Transactions

- Facilitates one-to-many fragment changes
- Conceptually similar to database transactions
- Can Create, Replace and Remove Fragments

Xamarin Evolve 2014

Adding Android Fragments via Code

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.Add(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014

Adding Android Fragments via Code

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.Add(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014

Adding Android Fragments via Code

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.Add(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014

Adding Android Fragments via Code

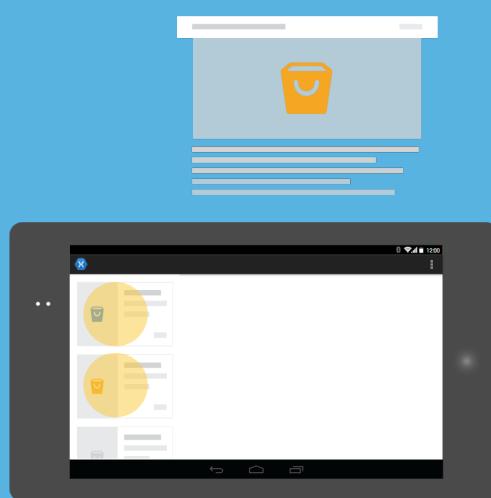
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <FrameLayout
        android:minWidth="25px"
        android:minHeight="25px"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:id="@+id/container"
        android:layout_weight="1" />
```

Xamarin Evolve 2014

Adding Android Fragments via Code

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.Add(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014



Xamarin Evolve 2014

Replacing Android Fragments

```
using (var transaction = FragmentManager.BeginTransaction())
{
    transaction.SetCustomAnimations(
        Android.Resource.Animator.FadeIn,
        Android.Resource.Animator.FadeOut);

    transaction.Replace(Resource.Id.container, new FragmentToAdd());
    transaction.Commit();
}
```

Xamarin Evolve 2014

Communicating with Fragments

Xamarin Evolve 2014

Build for Modularity

Think like Building Blocks

- Assume that the Activity knows nothing about your Fragment and vice-versa
- Build knowing that it will be reused and used in varying different contexts
- Make it easy for your Fragment to discover host Activity Functionality



Xamarin Evolve 2014

Providing Data to a Fragment

```
var fragmentToAdd = new FragmentToAdd {  
    Arguments = new Bundle ()  
};  
  
fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorRed, backgroundColor.R);  
fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorGreen, backgroundColor.G);  
fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorBlue, backgroundColor.B);
```

Xamarin Evolve 2014

Guaranteeing Fragment Receives Data

```
public static FragmentToAdd BuildFragmentToAdd (Color backgroundColor)
{
    var fragmentToAdd = new FragmentToAdd {
        Arguments = new Bundle ()
    } ;

    fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorRed, backgroundColor.R);
    fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorGreen, backgroundColor.G);
    fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorBlue, backgroundColor.B);

    return fragmentToAdd;
}
```

Xamarin Evolve 2014

Guaranteeing Fragment Receives Data

```
public static FragmentToAdd BuildFragmentToAdd (Color backgroundColor)
{
    var fragmentToAdd = new FragmentToAdd {
        Arguments = new Bundle ()
    } ;

    fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorRed, backgroundColor.R);
    fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorGreen, backgroundColor.G);
    fragmentToAdd.Arguments.PutInt (ArgumentBackgroundColorBlue, backgroundColor.B);

    return fragmentToAdd;
}
```

Xamarin Evolve 2014

Messaging with the Host Activity

Passing notes in class is okay

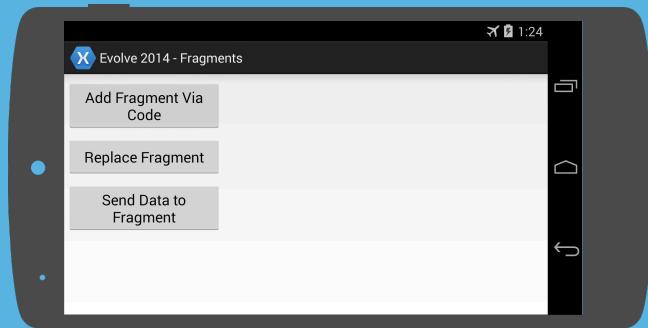
- Create an Interface that defines any external functionality your fragment needs
- Implement the Interface in your Activity
- In the fragment, check if the Activity has implemented the Interface and make use of any defined functionality

Xamarin Evolve 2014



Demo

Xamarin Evolve 2014



Xamarin Evolve 2014

Communicating with the Host Activity

```
public interface IDisplayDetail
{
    bool CanDisplayDetail { get; }

    void DisplayDetail(Fragment fragmentToDisplay);
}
```

Xamarin Evolve 2014

Communicating with the Host Activity

```
var displayDetailActivity = Activity as IDisplayDetail;  
  
if (displayDetailActivity != null && displayDetailActivity.CanDisplayDetail )  
    displayDetailActivity.DisplayDetail (new FragmentToAdd());  
else  
    StartActivity(new Intent(this.Activity, typeof(AddFragmentActivity)));
```

Xamarin Evolve 2014

Communicating with the Host Activity

```
var displayDetailActivity = Activity as IDisplayDetail;  
  
if (displayDetailActivity != null && displayDetailActivity.CanDisplayDetail)  
    displayDetailActivity.DisplayDetail (new FragmentToAdd());  
else  
    StartActivity(new Intent(this.Activity, typeof(AddFragmentActivity)));
```

Xamarin Evolve 2014

Communicating with the Host Activity

```
var displayDetailActivity = Activity as IDisplayDetail;  
  
if (displayDetailActivity != null && displayDetailActivity.CanDisplayDetail )  
    displayDetailActivity.DisplayDetail (new FragmentToAdd ());  
else  
    StartActivity(new Intent(this.Activity, typeof(AddFragmentActivity)));
```

Xamarin Evolve 2014

Messaging Alternatives

When an Interface is Not Enough

- Consider using Events
- Create an event in your fragment that you can subscribe to in an activity or another fragment
- When using events, make sure that you unsubscribe properly to avoid any potential memory or subscription issues

Xamarin Evolve 2014

Tying It All Together

Work together in harmony

- Fragments are individual modules, but they can work together
- Use an Activity to orchestrate communications between multiple fragments
- Mix-and-Match Fragments when space allows, such as when in landscape mode or on a tablet



Xamarin Evolve 2014

Xamarin Fragment Docs

bit.ly/1nUploU

Xamarin Evolve 2014

One Last Thing...

Xamarin Evolve 2014

Testing. Testing Never Changes

- Device manufacturers implement Android Differently
 - Camera
 - Storage
 - Even ListViews!
- Emulators Lie



// Questions? //

Xamarin Evolve 2014

Xamarin Evolve 2014

Designing Android UIs
for the Ever Changing
Device Landscape

Michael Stonis
[@michaelstonis](https://twitter.com/michaelstonis)