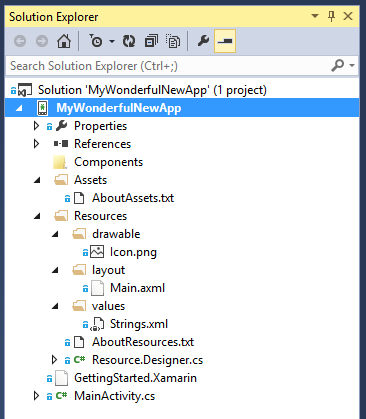
Day 4: Android Project Structure Walk Through

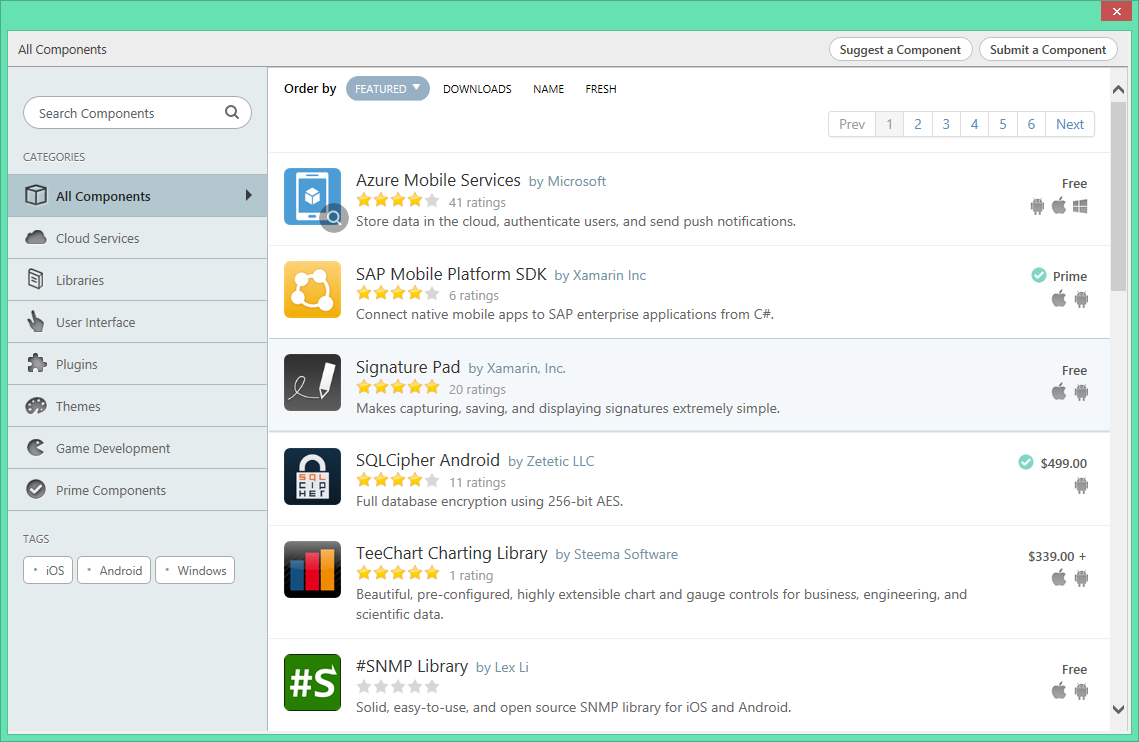
Today, we are going to take a look at what each and every folder in a Xamarin.Android application means.

Let’s go back and look at the project we created in Day 3 and look at its structure in Visual Studio’s Solution Explorer.



## Components

This folder is more specific to the Xamarin platform than Android. This folder contains all the components that you can download and install from [Xamarin Component Store](https://components.xamarin.com/). The component store has lot of useful libraries, user interface widgets, plugins, themes and a lot more.



If you are in the .NET world and know of Nuget, think of Component Store as an extension to Nuget and serving Android / iOS and Windows Phone libraries.

## Assets

Assets folder is where you would place assets that don’t need to be referenced in code. These are generally files like text files, fonts, audio and video files. Android will not process files under the Assets folder at all so if you need raw access to any resource in your application placing them in Assets is the way to go.

To read an Asset you can use the following snippet,

|  |
| --- |
| using (StreamReader sr = new StreamReader (Assets.Open ("samplefile.mp3")))  {  content = sr.ReadToEnd ();  } |

For a gist, please go to <https://gist.github.com/vkoppaka/2104ad37d27550556f6f>

In my experience, I haven’t found myself using assets folder much.

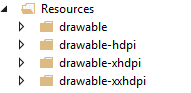
## Resources

Resources folder is where, well, Android Resources go. In Android Studio, or other Java based IDEs this folder is generally called as “res” and in Xamarin.Android the folder is referred to as “Resources”. Resources folder is again categorized into three main groups

1. Drawable
2. Layout
3. Values

### Drawable

Drawable folder contains the application’s image resources like png, jpg etc. The Drawable folder is again categorized into multiple “resolution” specific folders. In a typical Android application you will end up seeing Drawable-ldpi, Drawable-mdpi, Drawable-hdpi, Drawable-xhdpi, Drawable-xxhdpi etc

  
All these folder have a suffix of DPI (Dots per Inch) and represent low, medium, high, extra-high DPI devices. Android, as well as Xamarin.Android, smartly serves the file from a respective folder based on the device it’s running on. For example if the device your Android application is running on is a XHDPI device: Android serves all the images from Drawable-xhdpi folder and if as a fallback it looks in the Drawable folder.

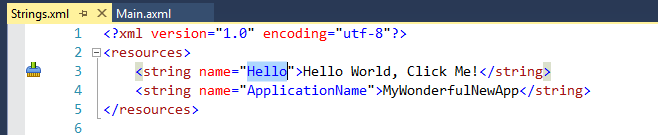
### Layout

The Layout folder contains layout of your Android Application controlled by AXML (Android XML) files. This folder can also be categorized into multiple device specific folders although in DP (Density-Independent Pixel) scale.

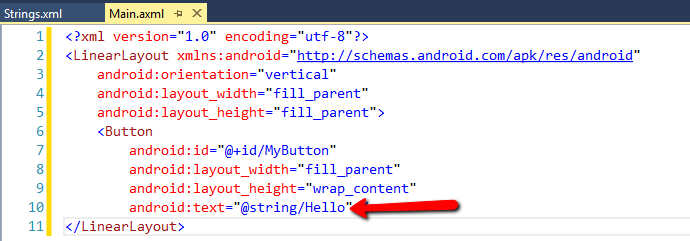
### Values

Values folder holds string resources of the various labels and other string messages in your application. This is a key-value pair based XML file and can be used by defining a string in the file and later referencing the string in our Layout files.

In the screenshot below, we added a new value with key “Hello” and value “Hello World, Click Me!”



And later in our layout file, we refer to this string value using the “@” syntax: “@string/Hello”



And Android automatically display “Hello World, Click Me!” as the text for the button when the application is run.

That’s it for today. Tomorrow, we will start learning how to design your Android Application.

Venkata