**ANDROID TUTORIAL** – Transparent Status and Navigation Bars.

**Notes and Information -**

I have prepared a simple guide to help you achieve the new translucent feature that was introduced in the latest version of Android. (Android 4.4.x Kit Kat).

The reason for this tutorial is because it’s a tedious task to reply to each and every message I receive (Gmail and xda-developers). And it’s annoying to reply to all over these emails which in fact have the same answer. So hopefully this guide will help you. I will also be doing a video demonstration. Which should be in progress if you’re watching it right now.

**How to achieve this -**

There are a few ways to achieve this, I will be explaining both ways. One way demonstrates how you can achieve this whether you’re building from an existing project (from source) or a standalone project (from scratch). And the other way is when you’re simply using a decompiler to modify an existing application. This tutorial will explain both methods, of course the second method (decompile one) will be done in another video tutorial.

**Pre-requisites –**

* Some basic knowledge in app development/modding.
* Eclipse Studio or any app developing IDE.
* Notepad++ or any code editing/text editing software.
* A tool to decompile .apk’s.
* Image editing software.
* Patience and a brain, if you don’t have a brain, please refer to this [**link**](http://24.media.tumblr.com/tumblr_lvei4fcBQD1qmw84bo1_500.jpg).

Tutorial One – IDE Method

**Setup -**

Let’s start by creating a new project, make the max, minimum and target (If prompted) sdk version to 19. If you want to build the application with a lower sdk as you’ll need to refer to the second tutorial as it’s based around that. Leave everything as default, change the icon if you want?

**Adding colours -**

Next, you’ll have to navigate to the following folder - res/values, this folder contains all the necessary components, such as them layouts, colours, texts, integers and more. We will work with this in another tutorial. In the values folder we’re going to have to create a new colour xml.

If you’re not working from scratch you probably have this .xml made so just keep on reading. Make sure you add **.xml** as an extension. Once you’ve created the new .xml. You can use the inbuilt resource maker so simply code it yourself. Now we’re going to have to create a new colour because we’re going to apply it to our translucent layouts. So simply create a new colour and add the rgb colour in the textbox. If you open the actual **.xml** it should look something like this - <color name="ui">#f0f0f0</color>

**Explanation –**

The reason we’re adding this colour code as said earlier is because we’re going to apply this to our windowBackground code, so the colours of the transparent status and navigation bars go according to the applied rgb colour. This is really important as this will be used for all points.

**Adding colours 2 –**

Now you’re probably wondering which colour you should apply? Well it depends on what colour your actionbar is, so let’s say your actionbar colour is orange. We want our layouts to match this colour so it looks appealing. Now using an image editor, take a screenshot and check the hex colour of your actionbar.

This is important as we’re going to apply it to our colour, if you already have the proper colour of the actionbar simply copy the code and make a new colour in colors.xml.For example like this - #ffff6a00 (That’s orange). So it’ll look like this - <color name="orange">#ffff6a00</color>. Please note you don’t actually have to add the “ff” so you can simply make it <color name="orange">#ff6a00</color>. It really depends what format you want to use.

**Adding translucent bars –**

You will need the following code, I will explain what the code does.

<item name="android:windowTranslucentStatus">true</item>

The code above adds transparency to the status bar, the true statement lets the device know that it has to be applied.

<item name="android:windowTranslucentNavigation">true</item>

The code above adds transparency to the navigation bar, in some cases we don’t need this, but this is entirely up to you. If you wish to disable it, simply remove it or change true to false.

Add both of these code to the styles.xml, in the main theme base (AppThemeBase – AppBaseTheme) or pretty much whatever you called your base theme? Now you will have to apply the colour, to do so you’ll need to add this line to styles.xml under the two code above –

<item name="android:windowBackground">@color/orange</item>

Replace “orange” with whatever you called your colour. Now we’re nearly done, we just have to add a new code so that the layout doesn’t go all crazy. Why? Because trust me, the content will overlap. To prevent this from occurring simply add this code to your main theme –

<item name="android:fitSystemWindows">true</item>

Now nothing should overlap, you can go ahead and compile your app and try it. If it doesn’t work you’ll have to switch layout code, so simply copy the code you added to the main code and paste it to the second theme section. This should fix it.

Tutorial Two – Decompile Method

**Setup -**

First, you’ll need to find the desired app you want to modify, decompile the application using a decompiler. There are plenty tools available over at xda-developers.

**Adding colours –**

This part is the same as the first tutorial, so to avoid me typing out loads. Please refer back to the first tutorial as it explains everything in detail. (Adding colours).

**Adding layouts 1 –**

There are two ways to do this, the reason I’m saying this is because the application initially is available for quite a bit of Android platforms. This first section will explain how you can apply the layouts, but it will not work on any platform version besides 4.4.X +.

**Starting point – Adding layouts 1/2 –**

You will need the following code –

<item name="android:windowTranslucentStatus">true</item>

<item name="android:windowTranslucentNavigation">true</item>

<item name="android:windowBackground">@color/orange</item>

<item name="android:fitSystemWindows">true</item>

Again, you can change the navigation one, to false if you wish not to have transparency on the navigation bars. Also you have to change the windowBackground code to whatever you named your colour.

But before you add the code to your styles.xml, you’re going to have to search for the main theme section, so here is a few ways to find it. Using a hex/code editor, open styles.xml from res/values. Now there should be a shortcut key for finding text. Default key is Ctrl + F, this opens a mini text input box. The main keywords to look for when finding the main theme section is main, theme and the applications name. So you will have to apply the above code to the main section, compile and see if it works. If it doesn’t, copy it and find another section and do this over and over again till you get it correct.

**Adding layouts 2 –**

This method allows you to make the application work on the available platforms it was developed for but adds the translucent bars if you’re on kitkat. What you’ll need to do is create a new folder in /res called values-v19. Go to res/values and copy styles.xml and paste it in the res/values-v19 folder. Open the styles.xml. Now refer back to the **Starting point – Adding layouts1/2** section to see how you can add the code.

**Credits –**

**Ashish Dubey –** For teaching me how to do this.

**Me –** For passing on the tutorial to you guys.

Project Ascend Apps