

How to Change the Color of Text in Action Bar for Android



by Dusko Savic
DuskoSavic.com

Up to now, the color of the navbar text was always the same in all Buzztouch apps for Android. In this document, you will see how to change the text to one or more colors.

The version used here is Buzztouch Android 3.0, with fragments.

```
//=====
```

To change the color of text in action bar for Android, find the following line in file BT_activity_host.java:

```
if (actionBar != null) {
```

In my version of BT_activity_host.java it was around line 1256, however, various versions of this file exist and the line numbers given here may vary. The line you are looking for is in function

```
public void configureNavBarAndBackgroundForScreen(BT_item  
theScreenData) {
```

which for me was in line 1226.

Now let us change the color of the navigation bar text. First we shall find the following text, right below, around line 1259:

```
// set title...  
actionBar.setTitle(navBarTitleText);
```

The text we want to color is in variable `navBarTitleText`. There are two ways of entering color:

1) Using an HTML class

The text will be transformed from pure text into HTML text, and we can enter the colors there:

```
actionBar.setTitle(Html.fromHtml("<font color=\"red\">" +  
navBarTitleText + "</font>"));
```

The text will be red when you run the app next. If needed, you can use hex values for color, like this:

```
actionBar.setTitle(Html.fromHtml("<font color=\"'#0000CC\">" +  
navBarTitleText + "</font>"));
```

The text will again be red. Please note that these two lines of Html are not identical character by character, be careful when you paste one or the other into your class.

This approach is idea of forum user [CMCOFFEE](#).

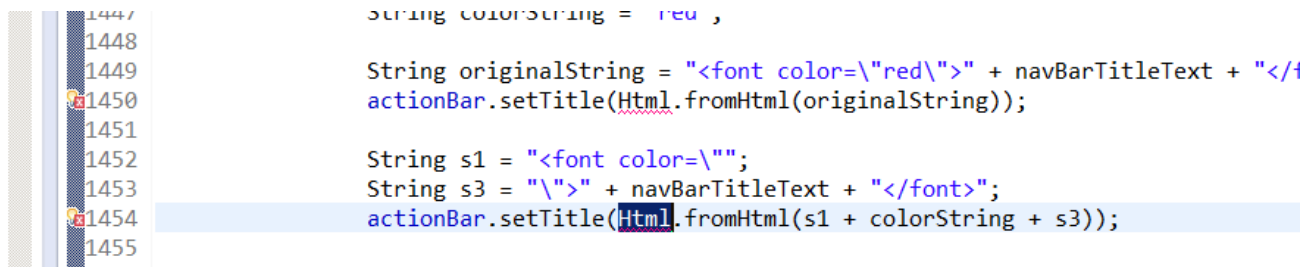
It may well happen that this is the first time your project needs the Html class from Android Java. If there is a red mark on the left of the line Eclipse, you need to import that class into the project. There are several ways of doing it:

a) Enter the following line into the import section of your class:

```
import android.text.Html;
```

b) Or, in Windows, press Ctrl-Shift-O to import it automatically.

c) Or, hover over the red mark with the cursor of the mouse and import the needed class from there.



That's all there is to change the color of the navbar!

2) Using spanned arrays

This is a technical name for using a bit of code in Java. The procedure boils down to two consecutive Copy and Paste routines. Once again we need the line

```
actionBar.setTitle(navBarTitleText);
```

and find it around line 1256. Comment it out and put beneath it the following lines:

```
String taraba = "##";
int color = Color.CYAN;
// color = BT_color.getColorFromHexString("#d3d3d3");
CharSequence s = setSpanBetweenTokens(taraba + navBarTitleText +
taraba, taraba, new ForegroundColorSpan(color));
actionBar.setTitle(s);
```

Then, press Ctrl-Shift-arrow down on Windows and go to the end of the current method, which should be

configureNavBarAndBackgroundForScreen

After that function add the following code, taken from here:

<http://www.androidengineer.com/2010/08/easy-method-for-formatting-android.html>

```
//=====

    public static CharSequence setSpanBetweenTokens(CharSequence
text, String token, CharacterStyle... cs) {
    // Start and end refer to the points where the span will
apply
    int tokenLen = token.length();
    int start = text.toString().indexOf(token) + tokenLen;
    int end = text.toString().indexOf(token, start);

    if (start > -1 && end > -1) {
        // Copy the spannable string to a mutable spannable
string
        SpannableStringBuilder ssb = new
SpannableStringBuilder(text);
        for (CharacterStyle c : cs)
            ssb.setSpan(c, start, end, 0);

        // Delete the tokens before and after the span
        ssb.delete(end, end + tokenLen);
        ssb.delete(start - tokenLen, start);

        text = ssb;
    }

    return text;
}
```

```
//=====
```

(The above function may go anywhere in the class, but it is just convenient to place it below the place it is used in.)

The function will hunt for the delimiter set in variable `taraba` and will insert the color codes into the string.

That's it. Then next time you run your app, its action bar text will be in **CYAN**. Change the line

```
int color = Color.CYAN;
```

to whichever the color you want, including hex codes. You can use the function

```
getColorFromHexString(String hexString)
```

from `BT_color.java`, using it like this:

```
int color = BT_color.getColorFromHexString("#d3d3d3");
```

and so on.

Pros and Cons

This file, `BT_activity_host.java`, is one of the fundamental classes for Buzztouch app development system for Android. ***Change it at your own risk.*** If you only apply changes described here, everything will be OK, however, bear in mind that if you apply this hack in one app, it will not be transferred to the next app you are making. Make a list of such changes so that you know which files to visit routinely when you start a new app for Android.

This hack will change the color of the navbar text globally for the entire app. This is what you usually want, however, if you wanted different colors in different plugins, this solution will not fit the bill. Ideally, the Buzztouch core system should be changed so that in each plugin you have (or may be able to install) particular color of the navbar text. This may happen in the future, but currently, this hack should be sufficient.

How to Complicate Your Life Further -- Using Variables

You may want to create a special variable that will hold the value of color. Later it would be easier to change it, document it and maintain it.

Here is how you could do it even if you do not know or like Java.

```
String colorString = "red";
```

```
String originalString = "<font color=\"red\">" + navBarTitleText +  
"</font>";  
actionBar.setTitle(Html.fromHtml(originalString));  
  
String s1 = "<font color=\"";  
String s3 = "\">" + navBarTitleText + "</font>";  
actionBar.setTitle(Html.fromHtml(s1 + colorString + s3));
```

The first line introduces a new local variable *colorString*. The next two lines introduce another string, called *originalString*. These two lines do nothing on the outside, I include them here only to show the genesis of code. The last three lines tear *originalString* into three pieces, variables *s1* and *s3* are constant, and variable *colorString* is in the middle. In this way, changing the value of *colorString* will also change the color of the navbar.

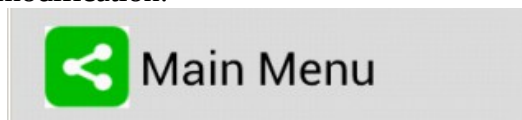
Several Colors in the NavBar

If *navBarTitleText* contained several words, it would be possible to give each of them their own colors. Here is the code:

```
String colorString1 = "red";  
String colorString2 = "blue";  
  
String s4 = "<font color=\"";  
String s5 = "\">" + "Main " + "</font>";  
  
String s6 = "<font color=\"";  
String s7 = "\">" + "Menu " + "</font>";  
  
String temp2 = s4 + colorString1 + s5;  
String temp3 = s6 + colorString2 + s7;  
String tempAll = temp2 + temp3;  
actionBar.setTitle(Html.fromHtml(tempAll));
```

Example:

This is a navbar without color modification:



This is the *red* variation:



And this is an example with two colors:



The sky is the limit here!

How to Contact the Author of This Buzztouch How-To Document



Dusko Savic (DuskoSavic.com) has authored 25 books about programming and software. One of his sites was visited more than 600000 times since 2005. With Buzztouch.com since 2011, Dusko is equally at home with Java for Android and Objective-C for iOS. In 2014, he has created about a **two dozen new plugins** which he uses in his own apps and/or writes directly for clients.

[You can see his Android apps here](#). His flagship app is [Opera Music](#), download it and enjoy the best opera arias from the best performers in history. [See his AdMob plugin in action there](#), an elegant way of making passive money with Android apps.

If you want your apps to make money, [let me know](#). If you want me to

-- **code a plugin** or an

-- **entire app for you**, either for Android or iOS,

-- **click** on the link below and **send** an email:

<http://duskosavic.com/blog/contact-dusko-savic/>