# GLUtils::texImage2D() with internal format GL_SRGB8_ALPHA8 causes GL_INVALID_ENUM on Android 8.0.0 device



WAI Bug P2 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION be...@gmail.com created issue #1

Device in question: Sony G3121 Android version: 8.0.0

Scenario: I load an image into a bitmap, generate and bind a texture to GL_TEXTURE_2D, and make this function call:

```
GLUtils.texImage2D(GL_TEXTURE_2D, 0, GL_SRGB8_ALPHA8, bitmap, GL_UNSIGNED_BYTE, 0)
```

This gives me a *GL_INVALID_ENUM* error on the mentioned device. The same call works fine though on another device which has Android version 10.0.

At first I thought that this might be a driver bug. But when I load the bitmap data into a buffer (bitmapBuffer) and then directly call glTexImage2D() like so:

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_SRGB8_ALPHA8, bitmap.width, bitmap.height, 0, GL_RGBA, type, bitmapBytes)
```

then it works just fine.

I'm not sure which branch matches my Android version, so I just checked the latest 8.0.0 branch: https://cs.android.com/android/platform/superproject/+/android-8.0.0_r51:frameworks/base/co

Line 766 has this: `glTexImage2D(target, level, internalformat, w, h, border, internalformat, type, p);`

It uses the *internalFormat* for the *format* as well, so essentially the call looks like this:

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_SRGB8_ALPHA8, bitmap.width, bitmap.height, 0, GL_SRGB8_ALPHA8, type, bitmapBytes)
```

`GL_SRGB8_ALPHA8` is not a valid *format* specifier.

It seems that this has been fixed with version 9.0: https://cs.android.com/android/platform/superproject/+/android-cts-9.0_r1:frameworks/base/core/jni/android/opengl/util.cpp (see the `getPi`

Not sure if bug fixes are still applied to version 8.0.0. If not, perhaps mention this somewhere in the docs?

✓ Links (3)

**Links (3)**

"I'm not sure which branch matches my Android version, so I just checked the latest 8.0.0 branch: https://cs.android.com/android/platform/superproject/+/android-8.0.0_r51:frameworks/base/core/j
"It seems that this has been fixed with version 9.0: https://cs.android.com/android/platform/superproject/+/android-cts-9.0_r1:frameworks/base/core/jni/android/opengl/util.cpp (see the getPixelFo
" … Version 9 does not fully fix it, it only accounts for GL_RGBA16F. Version 10 introduces a check for GL_SRGB8_ALPHA8: https://cs.android.com/android/platform/superproject/+/android-10.0.0_r1:

COMMENTS

**ra...@google.com** <ra...@google.com>

*Assigned to an...@google.com.*

**be...@gmail.com** <be...@gmail.com> #2

I have to correct myself. Version 9 does not fully fix it, it only accounts for `GL_RGBA16F`. Version 10 introduces a check for `GL_SRGB8_ALPHA8`: https://cs.android.com/android/platform/supe

**ch...@google.com** <ch...@google.com> #3

*Reassigned to lp...@google.com.*

Over toe Peiyong, related to code you added to util.cpp.
We can't fix this in the previous releases, but wondered (per filter's question) if it needed to be doc'd?

**lp...@google.com** <lp...@google.com> #4

*Status: Won't Fix (Intended Behavior)*

Returning `GL_INVALID_ENUM` is not a wrong behaviour, pragmatically it's probably better to fall back to GL calls instead of relying on the documentation specifying which format works in a ve