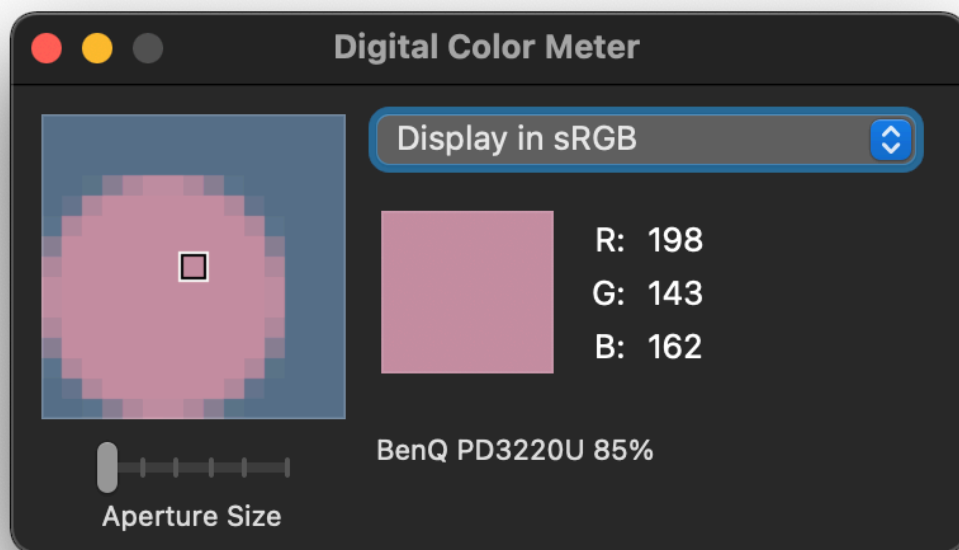


# Inspecting Colors on macOS

Inspecting color values presented on screen, either in a screenshot or in the Android emulator, is a relatively common task for developers dealing with UI and graphics. macOS offers a few tools out of the box to help in this process but they can lead to misleading conclusions unless used carefully. This document explains how to use those tools correctly and what to look out for to get meaningful results.

## Digital Color Meter

*Digital Color Meter* is a default application in macOS available in `/Applications/Utilities/`. This tool can be used to quickly read the RGB value of any pixel present on your display:



## What to do with it

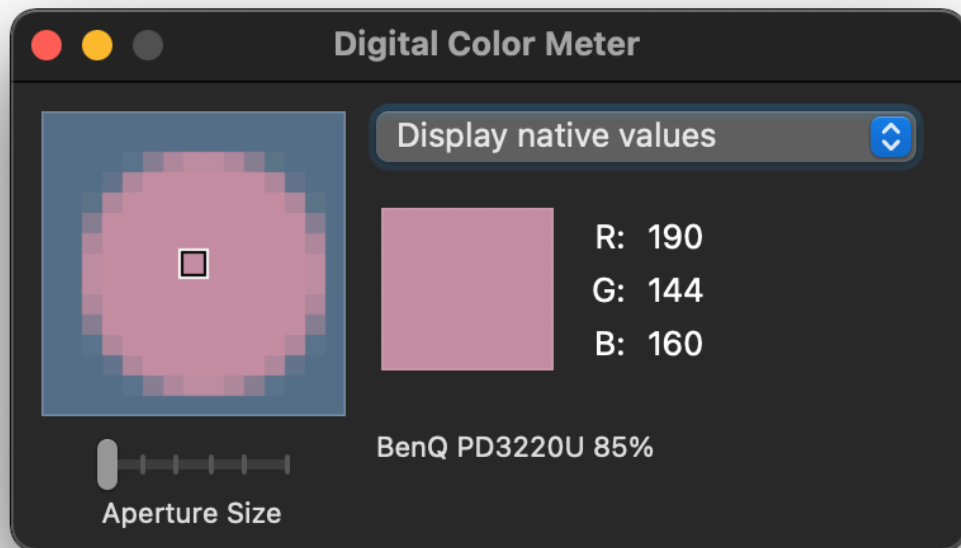
Use the menu *View > Display Values* to change the representation to one of the following:

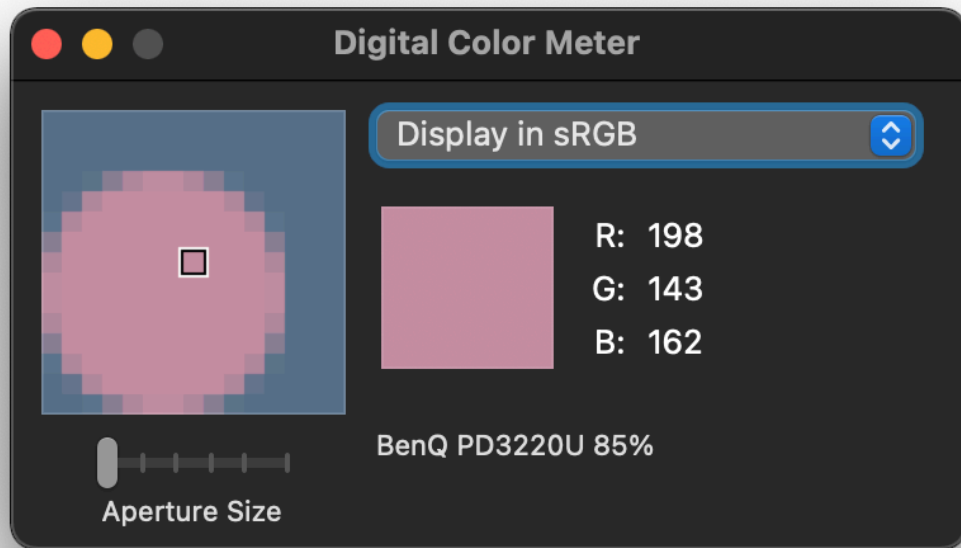
- **As Decimal**  
Display the RGB color as integer components from 0 to 255. Useful when using APIs such as `Color.argb()` on Android.

- **As Hexadecimal**  
Gives you a packed integer representation of the form 0xRRGGBB. This is what all `@ColorInt` APIs on Android expect.
- **As Percentage**  
Display the RGB color components as values from 0% to 100%. This is useful when using APIs that expect colors as floats between 0.0 and 1.0.

## What to look out for

Make *sure* to select *Display in sRGB* in the dropdown menu. Otherwise the color you read will be presented in a different color space (your display's color space by default). For instance, on my machine at home, my display uses a calibrated color profile called "BenQ PD3230U 85%". Here is how the same color on screen is presented when using the default option *Display native values* and *Display in sRGB*:

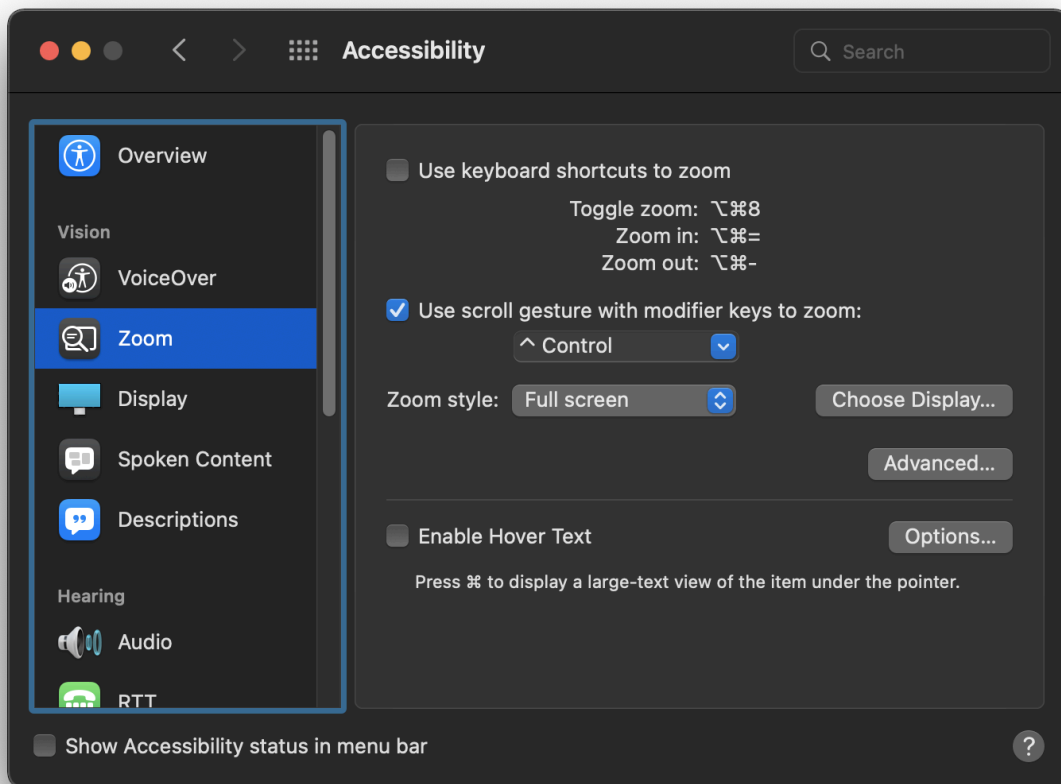




Unless you know exactly what you are doing, always use *Display in sRGB*: you can expect most (if not all) design tools and APIs to expect sRGB colors by default. Android APIs are all in sRGB unless you explicitly specify a color space.

## Accessibility Zoom

Pixel-peeping is easier when zooming in directly on live pixels, without having to take a screenshot and relying on an image viewer's magnifying tools. macOS lets you zoom in on content either with the keyboard or mouse, options you can enable in *System Preferences > Accessibility > Zoom*:



My favorite method is to press Control and use the mouse wheel to zoom in and out.

## What to look out for

Under the *Advanced* menu make sure to disable the option *Smooth images*. With this option off, pixels and colors will be preserved as-is.

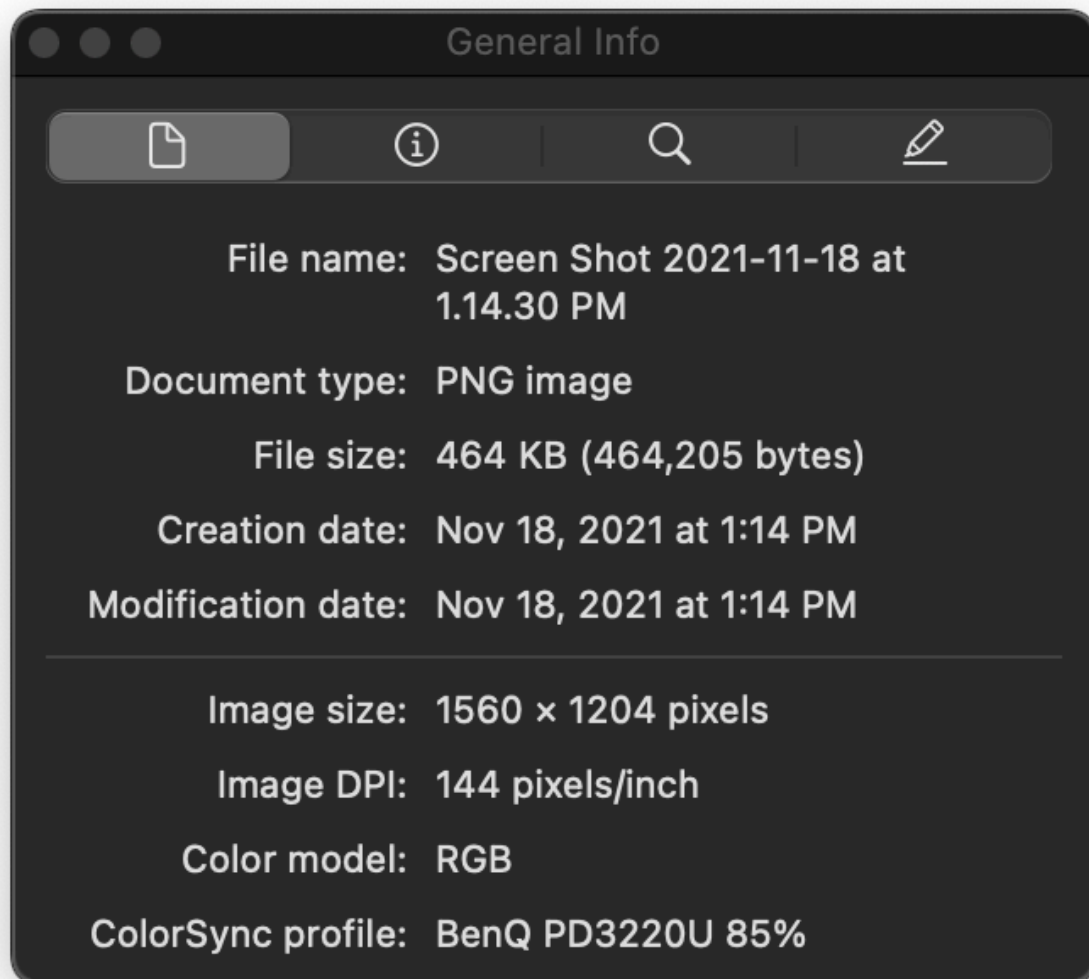
## Screenshot

You can capture screenshots in two way on macOS:

- By using the *Screenshot* app in */Applications/Utilities/*. It offers access to a number of options, including annotating screenshots.
- By using keyboard shortcuts: Cmd-Shift-3 will take a fullscreen capture, and Cmd-Shift-4 will capture a region. After pressing Cmd-Shift-4 you can then press Space to capture a window easily.

## What to look out for

Screenshots are always captured in your display's color space, and the appropriate color profile is embedded inside the screenshot. You can verify which color profile the screenshot is using by opening the file in *Preview* and pressing **Cmd-I** (*Tools > Show Inspector*). The information you are looking for is the last line, called *ColorSync profile*. Here is an example of a screenshot taken on my home display that uses a custom color calibrated color profile:



This should generally not be an issue but you may open the screenshot inside an application without color management or, worse, send it to someone else who will be using such an application. If you observe color shifts after opening the screenshot in another application, or if you want to be perfectly safe, refer to “Image color matching” in the next section on *ColorSync Utility*.

# ColorSync Utility

This application, found once again in */Applications/Utilities/*, is macOS' swiss-army knife when it comes to color management and color profiles.

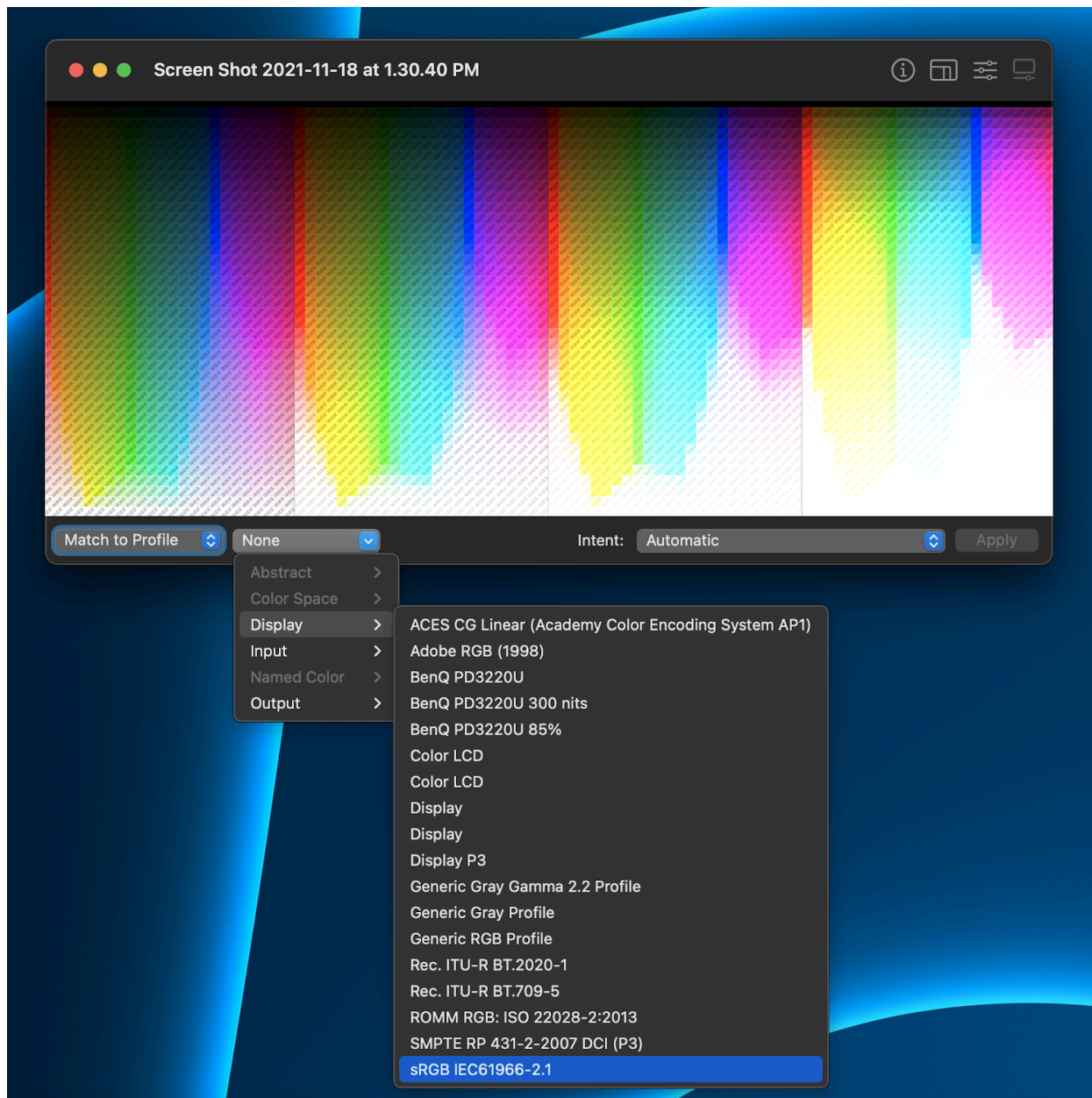
## What to do with it

There are two main use cases for this application: image color matching and color conversion.

### Image color matching

This feature can be used to convert an image from one color space (or profile) to another. This is particularly useful to convert screenshots from your display's native profile to the ubiquitous sRGB profile. To do so, first open the image you wish to convert using *File > Open* or by just dropping it on the app icon.

With the image loaded, make sure the dropdown in the bottom left is set to *Match to Profile*, then in the second dropdown from the left, select *Display > sRGB*. Finally, click *Apply* in the bottom right and save your converted image.



Note: color matching can also be done in all good image editing tools, such as Affinity Photo, Pixelmator, Adobe Photoshop, etc.

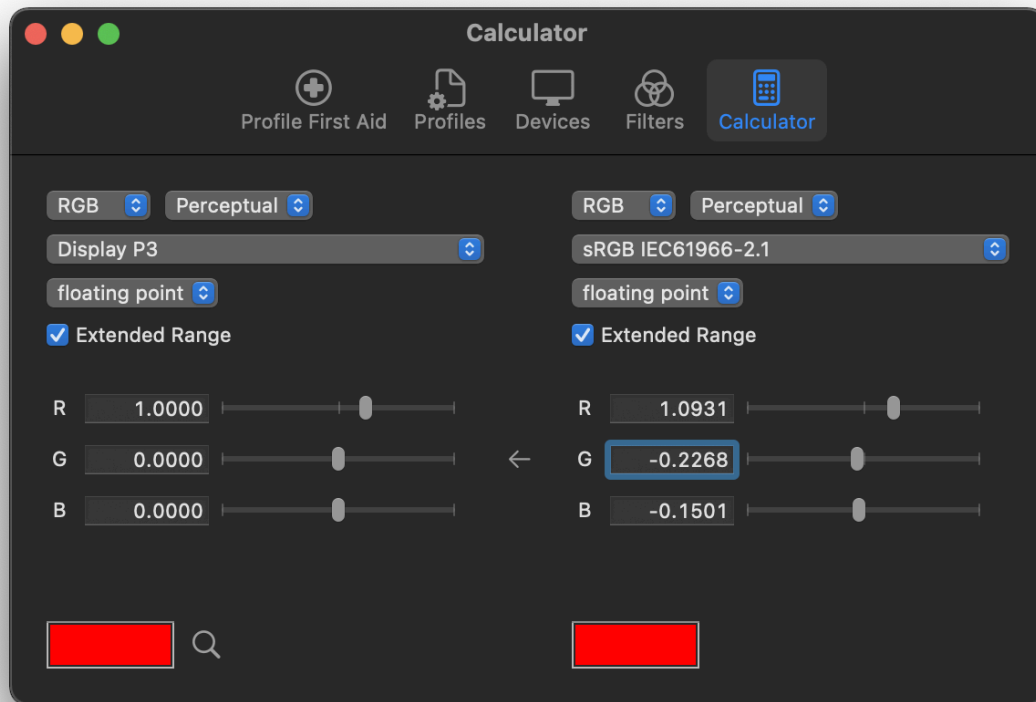
## Color conversions

In the main window, select the *Calculator* tab to convert colors from one color space to another. The color model is set to RGB by default but you can change it to HSV, HLS or Lab, which are supported by some of Android's color APIs, and by most design tools.

In RGB mode you can select the source and destination color profiles. I recommend using floating point representations and checking *Extended range* in both the source and destination.



This is important as values  $< 0$  or  $> 1$  indicate out-of-gamut colors in the destination, i.e. colors that cannot be represented exactly. Here is for instance the conversion of pure red (RGB 1,0,0 or 0xFF0000) from Display P3 to sRGB:



The closest representable color in sRGB is also 1,0,0 but it won't be exact because of out-of-gamut values.