Gradient Image Generator

Release 1.0

Sep 08, 2020

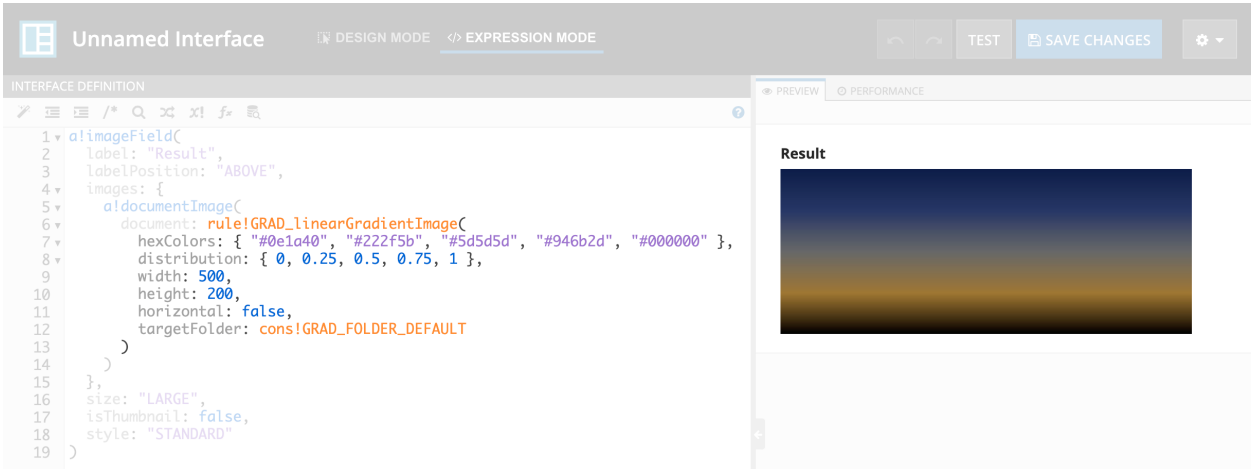
Contents:

This plugin provides an Expression Function called `linearGradientImage()` for generating simple, linear gradient PNG images based on the colors, distribution ratios, and dimensions you specify.

CHAPTER 1

Introduction

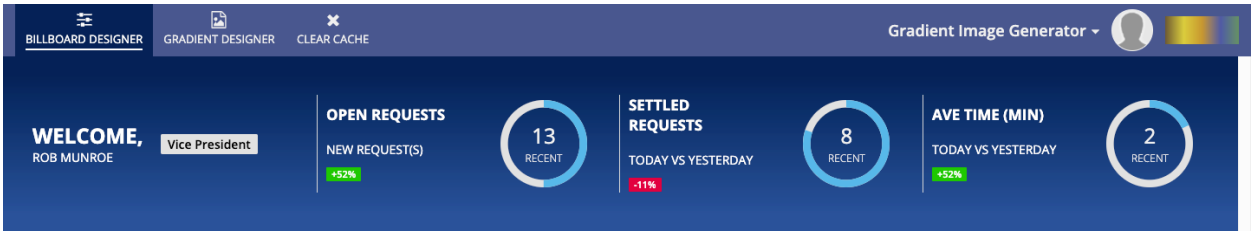
This plugin provides an Expression Function called `linearGradientImage()` for generating simple, linear gradient PNG images based on the colors, distribution ratios, and dimensions you specify.



As seen above, the function returns an Appian `Document` data type which can be used in places such as an `a!documentImage()` or even linked to.

The genesis for this plugin was to provide gradient color backgrounds for a Billboard Layout component's `backgroundMedia` value. A companion Appian Application is distributed with the plugin that includes Expression Rules and Interfaces for easy configuration of Billboard Layouts using this plugin.

Note that this companion application requires the Color Picker Component plugin.



1.1 Performance

To ensure high performance, the function will cache the generated images in Appian's document management system and return existing images on subsequent calls (based on a file naming convention) rather than generating the same image over and over.

Initial image generation usually takes 100 - 200 milliseconds, and subsequent calls that return the existing images usually take less than 10 milliseconds.

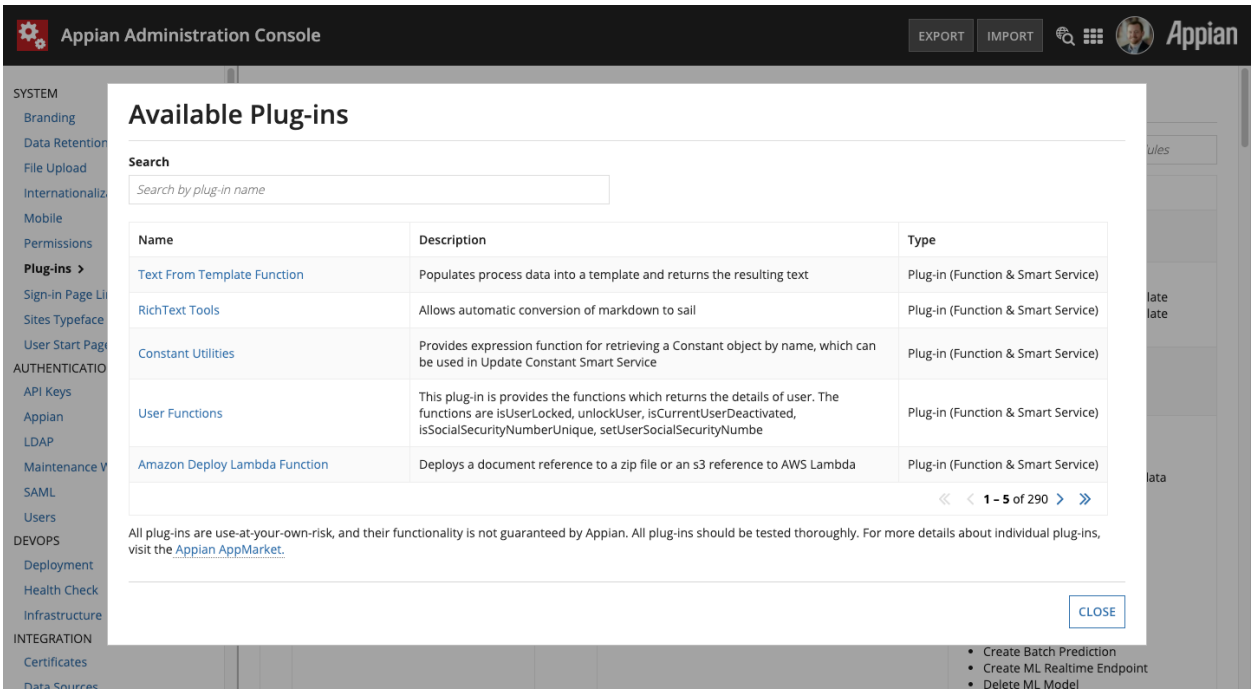
You may wish to generate the image in a lower environment (e.g. DEV) and copy/move the image into your application as well, however the speed at which the plugin returns existing images is comparable to directly referencing an image using a Constant.

1.2 Compatibility

This plugin and its companion application were built and tested on Appian version 20.2.

1.3 Installation

If installing to a fully-managed Appian Cloud instance, install using the Plugins panel of the Administration Console.



If installing to a self-managed Appian instance, copy the `GradientImageGenerator-1.0.jar` plugin JAR file to the `<APPIAN_HOME>/_admin/plugins` directory.

Import the companion `Generate Gradient Image Companion App 1.0.zip` application as normal in your Appian instance.

CHAPTER 2

Expression Function

2.1 linearGradientImage()

This is the expression function that generates the PNG gradient image. It allows specification of many values to produce gradients that match your color schemes.

Returns: `Document`

hexColors *(List of Text String)*: The list of colors in the gradient, as hexadecimal, e.g. `{ "#FFAA00", "#EE5500", "#EEEEEE" }`

distribution *(List of Number (Floating Point))*: The distribution positions of the colors in the gradient change as a percentage (value between 0 and 1), e.g. `{ 0.0, 0.2, 1.0 }`

width *(Number (Integer))*: The width of the image in pixels

height *(Number (Integer))*: The height of the image in pixels

horizontal *(Boolean)*: If true, the gradient will be oriented horizontally instead of vertically

targetFolder *(Folder)*: The Appian `Folder` to store (and cache) the image

2.1.1 Example

```
a!imageField(
  label: "Result",
  labelPosition: "ABOVE",
  images: {
    a!documentImage(
      document: linearGradientImage(
        hexColors: { "#cc3300", "#ff9966", "#ffcc00", "#99cc33", "#339900" },
        distribution: { 0, 0.25, 0.5, 0.75, 1 },
        width: 500,
        height: 200,
        horizontal: false,
```

(continues on next page)

(continued from previous page)

```
        targetFolder: cons!GRAD_FOLDER_DEFAULT,
      )
    )
  },
  size: "LARGE",
  isThumbnail: false,
  style: "STANDARD"
)
```

This would produce an image named: `linear_cc3300_ff9966_ffcc00_99cc33_339900_0.0_0.25_0.5_0.75_1.0_500_200_vertical.png`

Note: The filename is a concatenation of the arguments to the expression function, producing a consistently named image for the same arguments. This is how the plugin finds and returns existing images instead of regenerating them each time. As such, if you move, rename, or delete the Document, the plugin will regenerate the image using this long filename again.

CHAPTER 3

Companion Application

The companion application features 2 Expression Rules and a Site that are of most interest.

3.1 Expression Rules

3.1.1 `GRAD_linearGradientImage()`

This Expression Rule simply wraps around the `linearGradientImage()` function, accepting the same arguments and passing them to the plugin.

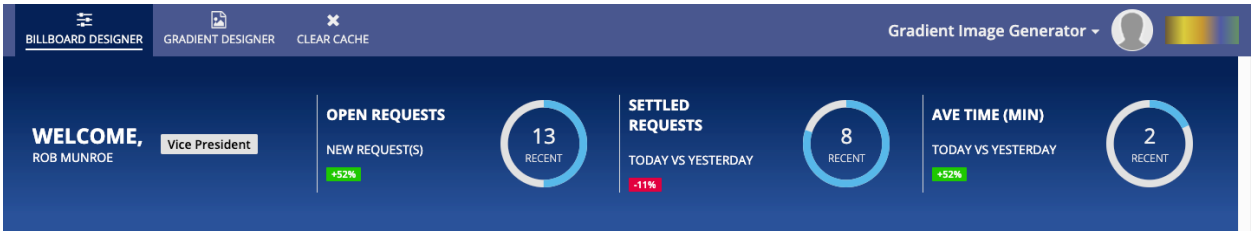
The reason it exists is that the Appian Interface Designer's Design Mode will throw an error for any Interface that directly references `linearGradientImage()`. *This only occurs in Design Mode*, so this rule can be bypassed if you plan to only use Expression Mode in the Interface Designer.

It is recommended that you use this Expression Rule in any interface referencing the plugin as future enhancements to the plugin and companion application can gracefully handle changes.

3.1.2 `GRAD_billboardGradientImage()`

This Expression Rule provides simplified arguments that are suitable for a Billboard Layout component's `backgroundMedia` value, providing sensible and aesthetically pleasing defaults.

The gradients produced will always be vertically oriented and will include a 10% top color buffer before beginning the gradient. This looks best when using a Site with a Selected Tab Background Color the same as the `hexTopColor` value, as seen below.



hexTopColor *(Text)*: The top color of the gradient

hexBottomColor *(Text)*: The bottom color of the gradient

billboardHeight *(Text)*: The same value set for the Billboard Layout's height property, e.g. `SHORT` or `AUTO`

customWidth *(Number (Integer))*: If `billboardHeight` is set to `AUTO`, this is the width in pixels of the generated image

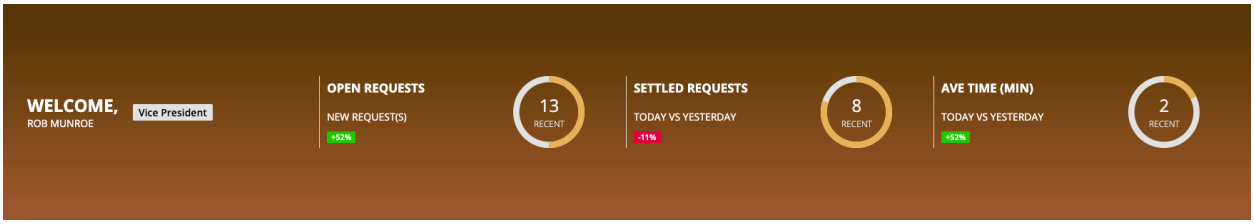
customHeight *(Number (Integer))*: If `billboardHeight` is set to `AUTO`, this is the height in pixels of the generated image

targetFolder *(Folder)*: An Appian `Folder` where the generated image is to be stored. If left blank, defaults to `cons!GRAD_FOLDER_DEFAULT`.

3.1.3 Example

```
a!headerContentLayout(
  header: {
    a!billboardLayout(
      backgroundMedia: a!documentImage(
        document: rule!GRAD_billboardGradientImage(
          hexTopColor: "#4c3006",
          hexBottomColor: "#8f4b24",
          billboardHeight: "SHORT",
          targetFolder: cons!GRAD_FOLDER_DEFAULT
        )
      ),
      backgroundColor: "#000000",
      marginBelow: "NONE",
      overlay: a!fullOverlay(
        contents: {
          rule!GRAD_Interface_DefaultBillboardOverlayContent(
            gaugeFillColor: "#e6a84c"
          )
        },
        alignVertical: "MIDDLE",
        style: "NONE"
      )
    )
  },
  contents: {}
)
```

This would produce the following Billboard:

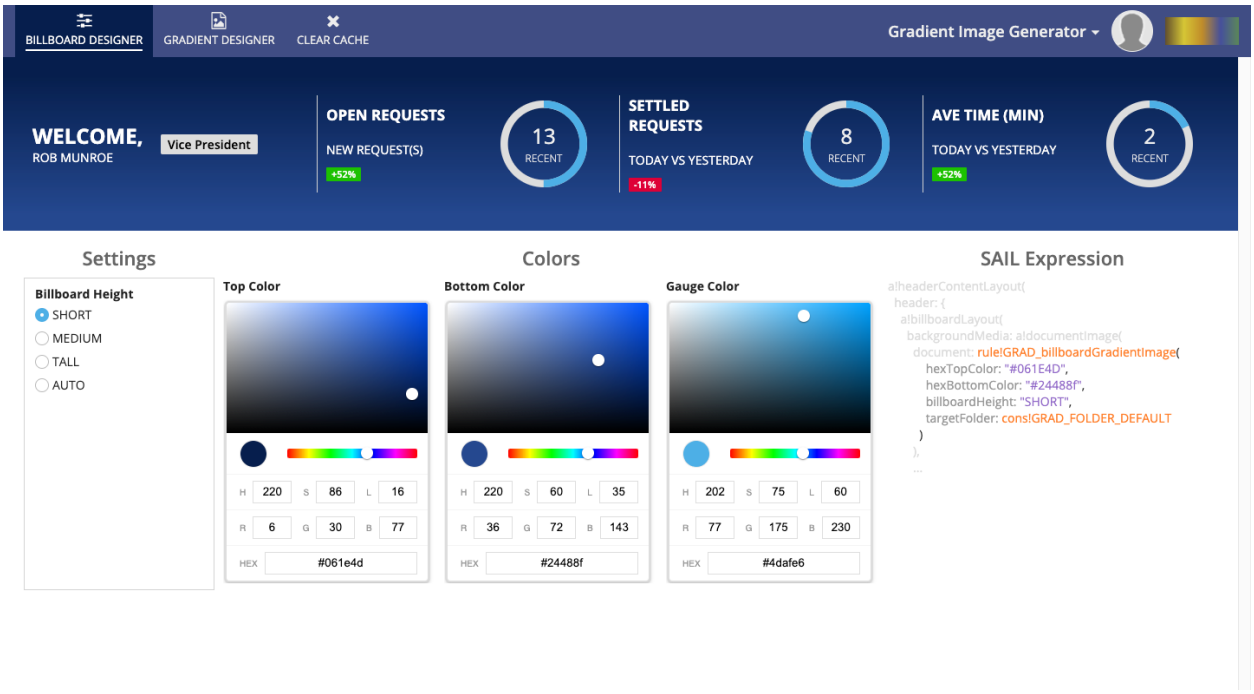


3.2 Gradient Image Generator Site

The companion application includes a single Site to give further examples and help design gradients images as well as Billboard backgrounds.

3.2.1 Billboard Designer

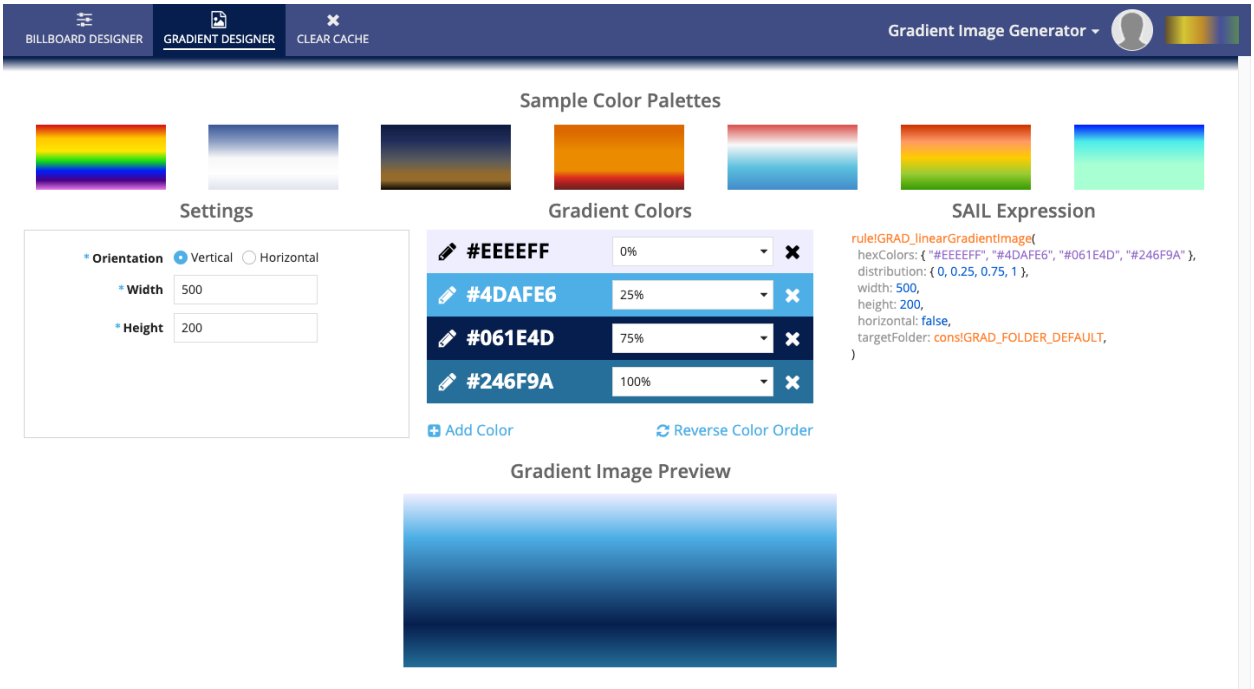
This page of the Site allows you to experiment with Billboard background colors and produces a SAIL Expression snippet that you can copy and paste into your Interfaces.



Simply change the values of the Billboard **Height**, **Top Color**, and **Bottom Color** to alter the Billboard's background.

3.2.2 Gradient Designer

This page of the Site allows you much finer-grained control over gradient image generation.



To begin click on a gradient in the Sample Color Palettes, or simply modify the default gradient provided. As you make changes to the Settings and Gradient Colors, the resulting image will appear below, and the SAIL Expression snippet will update, allowing you to copy and paste into your Interfaces.

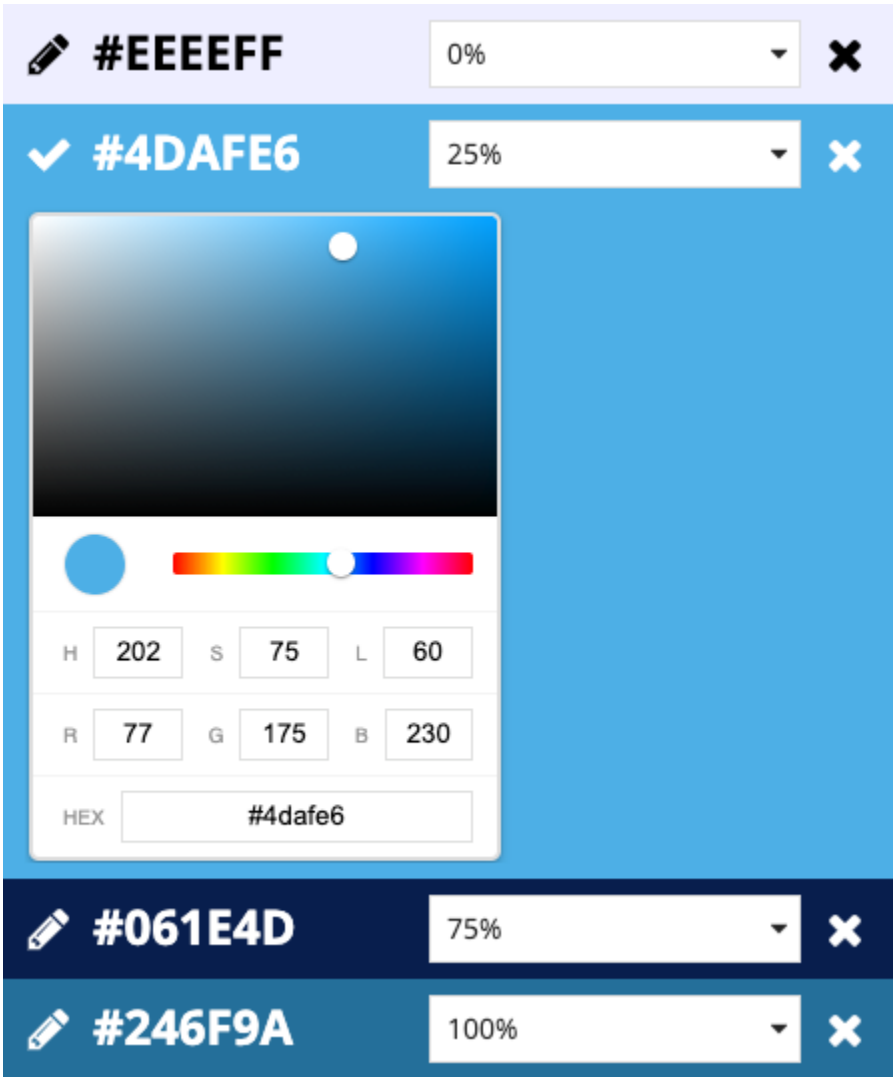
Settings

These values will alter the orientation and size in pixels of the generated image.

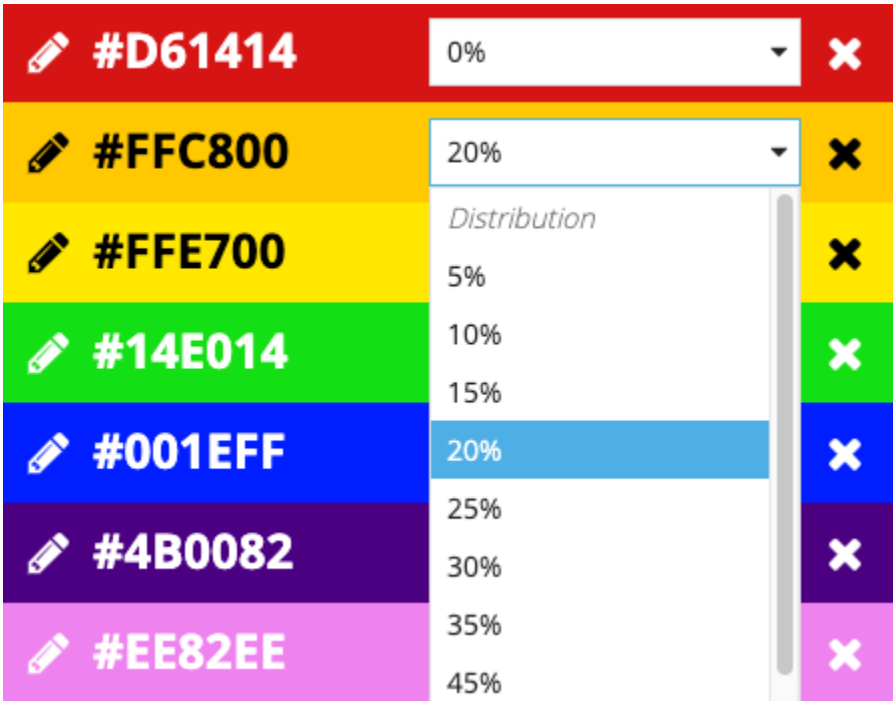
Gradient Colors

This is the list of colors, either top to bottom if the orientation is Vertical, or left-to-right if Horizontal.

To edit a color, click the **pencil icon** next to the color hex code. This will display a color picker allowing you to alter the color. When done, click the checkmark to hide the color picker.



To adjust where the color is positioned in the gradient, select a value from the **Distribution dropdown**. Note that you can only select values that differ from the existing colors, so you may need to alter another color's to position first.



To remove the color from the gradient, click the **X icon**.

To add a new color, click the **Add Color** link.

To reverse the order of the colors in the gradients (but keeping the distribution percentages the same) click the **Reverse Color Order** link.

SAIL Expression

This section produces a snippet of SAIL that can be pasted anywhere in your application that accepts a `Document`, including `a!documentImage()`.

Gradient Image Preview

This image is the result of the selected values in the designer. To download the PNG file, simply click on the image.

3.2.3 Clear Cache

This page is the start form for a process that will delete all images in the **Default Image Folder**, which is where the designer UIs store the generated images.

Note: If you delete an image created by this application, reloading Interfaces that still reference the plugin (e.g. reloading the Site and designer pages) will regenerate the gradient images.

CHAPTER 4

Changelog

Date	Version	Description
2020-09-04	1.0	Initial version