Palettes

UDS brand palettes enable all teams with all tech stacks to stay on brand with little or no effort. They provide the lowest level of opt in to UDS with the least amount of friction.

Introduction

Building on the benefits of design tokens, UDS introduces a clear distinction between *design options* and *design decisions*. All the allowable design options for a given brand are collected together into a package called a "brand palette".

Colours **Typography** **Sizes** **Spacing** **Gradients** **Opacity** **Radii** **Icons**

The name brand palette is loosely inspired by the concept of a colour palette, but beyond colours, it typically covers everything from spacing to animation primitives to font sizes and styles – essentially, anything about a brand that can be codified. While it's possible to have multiple palettes per brand in the future, there currently isn't a use case to maintain more than one.

The following snippet illustrates the structure of a brand palette.

```
{
  color: {
    purpleTelus: '#4B286D',
    greenTelus: '#66CC00',
    greenAccessible: '#2B8000',
    // ...
  },
  border: {
    border1: 1,
    border2: 2,
    border4: 4,
    // ...
  },
  // ...
}
```

Brand palettes are the lowest-level design system. Even though they're conceptually quite simple, their power comes from that simplicity. Brand palettes can be packaged up, versioned, and distributed to all teams, independent of the technology used. They enable a low friction opt-in for brand primitives across platforms, making it easy for teams to stay on-brand. Versioning means that consumers of brand palettes can simply “pull” the latest design thinking automatically without needing manual intervention from the brand team. Because brand palettes are built out of design tokens and therefore are machine-readable, we can build both tooling and more complex design systems on top of them.

Source

To facilitate a cross-brand and cross-platform architecture, brand palettes must be portable to as many technologies as possible, but ideally still be human-readable. For this reason, a brand palette source is a JSON file allowing it to be portable and both human and machine-readable. Brand palettes, as with all UDS packages use semantic versioning in order to support evolution over time and disseminate changes in a predictable manner. Refer to versioning documentation for a description of how palettes are versioned.

Builds

The `system-tokens` package implements the tooling that validates and builds palettes into platform specific formats. This includes JavaScript and CSS for JS based platforms (web, react-native, etc.), swift for iOS, and XML for android. Other platforms can be targeted by the UDS team. The built palette files present the lowest level mechanism for all teams to become part of the UDS system of systems. By consuming a built palette, teams (even those not building with JS) can stay on brand and receive brand updates from designers little or no human intervention.

The platform specific built platforms transform the source palette values into the platform appropriate format (rem units vs pixels, RGB vs hex colors, etc.)

Edit this page