

Custom Images and CSS

- [Overview](#)
 - [Custom Directories](#)
- [Images](#)
- [CSS](#)
 - [Default color styles](#)
 - [Browser Developer Tools](#)
 - [Editing & Updating CSS](#)

Overview

[Custom Images & CSS#top](#)

Before you make any changes, remember to create a backup of the entire ROOT directory. Creating a zip of the ROOT directory will mean you can always restore your styling back to Yellowfin's default state. This is especially important if you are new to CSS. **Do not skip this step.**

It is also important to remember to clear your browser cache before attempting to view any changes you've made. This will ensure you don't get a combination of old and new styles and images on your page, only the current correct version.

Custom Directories

Within the ROOT directory in your Yellowfin installation, there are two custom directories used for restyling:

1. customcss (`Yellowfin\appserver\webapps\ROOT\customcss`)
2. customimages (`Yellowfin\appserver\webapps\ROOT\customimages`)

These directories should contain any customised versions of images and css that are to be used in the place of Yellowfin original images. These files are stored in custom directories so that the original files can remain intact. It also means that when Yellowfin is updated they are not touched, as the custom directories are not modified during the update.

Images

[Custom Images & CSS#top](#)

The images used by Yellowfin can be found in the Images directory (`Yellowfin\appserver\webapps\ROOT\images`). There are a wide number of images used throughout Yellowfin to control the look and feel of various menus and results. In order to customise these images you will need to use image editing software. If you do not have image editing software there is a wide range of free tools available online including [GIMP](#) and [Paint.NET](#).

Once you have a tool to use, you need to:

1. Make the required changes to the image file
2. Save your changes to the **custom images directory** (`Yellowfin\appserver\webapps\ROOT\customimages`), ensuring the file maintains the same name and type
3. Clear your browser cache
4. Reload the page that's affected by the updates and review

CSS

[Custom Images & CSS#top](#)

The CSS files used by Yellowfin can be found in the CSS directory (`Yellowfin\appserver\webapps\ROOT\css`). There are a range of stylesheets used by Yellowfin to control the look and feel of various aspects of the product. The main file used when restyling Yellowfin content is `ie.css`, as this is used for the widest range of styles throughout the system, with smaller files used for specific functions or pages. However, this file is a core Yellowfin file and should not be edited. Instead, use your dedicated customcss folder (`Yellowfin\appserver\webapps\ROOT\customcss`) to overwrite the default style details.

Default color styles

In Yellowfin 9.6, we consolidated color name usage within our core CSS files to make it easier to change colors globally.

To customize your color scheme, open your custom file in your `\customcss` folder and at the top of the file, add:

```
:root {  
<variable-overrides-go-here>  
}
```

Make sure you replace <variable-overrides-go-here> with your own styles.

For example,

```
:root {  
  --primary-color: red;  
  --loader-color: green;  
}
```

The table below provides a current list of all style names related to color usage. If you're using an older version of Yellowfin, some of these might not work.

Style name	Details
--primary-color	#009EEC
--loader-color: #009EEC	#009EEC
--leftnav-hover-color	#008CD7
--primary-button-color	#009EEC
--primary-button-hover-color	#008ED4
--grey	#CCCCCC
--background-grey	var(--grey)
--border-grey	var(--grey)
--signal-positive-color	#009EEC
--signal-negative-color	#FF8F00
--dark-grey	#666
--black	#000
--background-black	var(--black)
--white	#fff
--background-white	var(--white)
--primary-text-color	#333740
--light-grey	#F8F6F4

<code>--light-grey-background</code>	<code>var(--light-grey)</code>
--------------------------------------	--------------------------------

Browser Developer Tools

Generally, when heavily restyling Yellowfin, including styles, the use of a browser's developer tools is recommended. This allows you to investigate elements on a page - viewing the styles being used and testing out possible changes before any stylesheets are changed. Most browsers have built in developer tools, as well as some well designed plugins. The following describes how to access developer tools in some common browsers, although it's not an exhaustive list.

Browser	Page Instructions	Menu Instructions
Chrome	Right-click anywhere on the page and select Inspect Element from the menu	Navigate to Settings > Tools > Developer tools
Firefox	Right-click anywhere on the page and select Inspect Element from the menu	Navigate to Firefox Menu > Web Developer > Inspect
Opera	Right-click anywhere on the page and select Inspect Element from the menu	Navigate to Opera > Page > Developer Tools > Opera Dragonfly
Safari	Right-click anywhere on the page and select Inspect Element from the menu	Navigate to Display > Develop > Show Web Inspector Note: This must first be enabled in Settings > Preferences > Advanced > Show Develop menu in menu bar

Note: these instructions were correct for the latest version of each browser when written, if they don't work for you please reference the help of the particular browser you are using.

Editing & Updating CSS

Once you have tried out your changes using a set of developer tools listed above, you will need to make your changes to Yellowfin's CSS files. In order to do this we recommend you make use of a text editor that has syntax highlighting. If you don't have something already, a free tool we like to use is [Notepad ++](#).

All you need to do now is:

1. Make the required changes to the CSS file(s)
2. Save the file(s) to the **custom css directory** (`Yellowfin\appserver\webapps\ROOT\customcss`)
3. Clear your browser cache
4. Reload the page that's affected by the updates and review

Note: your replacement files, stored in the customcss directory do not have to be files of the same names as the original, as long as they have the correct class names Yellowfin will use them. This means that you can condense all your custom styling into one css file if you wish, which only contains items you've customised.

[Custom Images & CSS#top](#)