

Style Sheets 3.x

Overview

This article explains which style sheets are loaded by Theme Builder 3.x, under what circumstances and in what order.

Theme Builder 3 loads several style sheets, most of which are optional or editable, as shown in the diagram to the right.

The style sheets are loaded as resources rather than embedding them directly in the HTML to enable browser caching.

Each subsequent style sheet can override any styles defined in previous style sheets, for example the "Combined CSS" can override any settings in both the "Menu Style Sheets" and "Confluence CSS".

Confluence CSS

Theme Builder 3.0 and above import Confluence's own style sheet by default. For this reason we strongly recommend upgrading to Confluence 2.6.1 or above which has a much cleaner style sheet than earlier versions of Confluence.

This style sheet can be enabled or disabled via the [CSS Tab](#). Note, however, that if you disable it you'll need to provide your own custom styles to make Confluence pages work properly.

Caching: [Cache Expire](#) (1 year)
Filter: `/s` (Confluence's Web Resource Manager)
Resource: `styles/main-action.css`
Source: Confluence - the same version of this style sheet is used for all layouts.

Menu Style Sheets

These read-only style sheets are included for all pages in your layout.

The actual style sheets loaded (if any) depend on the "look and feel" selected for the layout in the [Menus Tab](#).

The following table shows which CSS file is loaded for each "look and feel" and whether it then imports any additional CSS file(s).

Look and Feel	CSS Files
aqua	skin-aqua.css > hmenu.css
beos	skin-beos.css > hmenu-3d.css > hmenu.css
dark	skin-hmenu-dark.css > hmenu.css
longhorn	skin-longhorn.css > hmenu-extended.css > hmenu.css
modern	skin-modern.css > hmenu-3d.css > hmenu.css
sample	skin-sample.css > hmenu.css
system	skin-system.css > hmenu-3d.css > hmenu.css
win2k	skin-win2k.css > hmenu-3d.css > hmenu.css
win98	skin-win98.css > hmenu-3d.css > hmenu.css
xp-apps	skin-xp-apps.css > hmenu.css
xp-apps2	skin-xp-apps2.css > hmenu.css
xp-extended	skin-xp-extended.css > hmenu-extended.css > hmenu.css
xp	skin-xp.css > hmenu.css
yp	skin-yp.css > hmenu.css



custom	<i>hmenu.css</i>
custom 2D	<i>hmenu-extended.css > hmenu.css</i>
custom 3D	<i>hmenu-3d.css > hmenu.css</i>
none	

You can prevent these style sheets from being included in a layout by selecting the "none" look and feel on the [Menus Tab](#).

Caching: [Cache Expire](#) (1 year)
Filter: /s (Confluence's Web Resource Manager)
Resource(s): :sitebuilder/hmenu/src/*.css
Source: Plugin - the same version of these style sheets is used for all layouts.

Combined CSS

This style sheet is customised in the [Layout Manager](#) and is actually a combination of several style sheets loaded in this order:

- **Imports**, edited via the [CSS Tab](#)
- **Builder CSS**, in-built styles required by the theme
- **Panel CSS**, configured via [Panel Editor](#)
- **Custom CSS**, edited via the [CSS Tab](#)

Each layout has it's own version of the Combined CSS which is intelligently cached for all pages viewed using that layout.

The web browser will generally cache the Combined CSS for up to a year, unless you make alterations in which case the browser will download a new version.

Note: Read important notes on [Style Sheet Inheritance](#).

Caching: [Cache Expire](#) (1 year)
Servlet: /plugins/builder/layout-resources.action
Resource: combinedcss
Source: Layout - each layout has it's own version of this style sheet.

More specific details about the components of the Combined CSS resource are listed below.

Imports

This part of the style sheet takes styles from the "Imports" setting on the [CSS Tab](#) and is designed to import external style sheets, a process which needs to take place at the very start of a CSS file.

If the layout does not have any imports, enter `/* empty */` in the Imports section of the CSS Tab.

To remove the Imports component, de-select the "Merge this layout's CSS with it's parent's" and leave the Imports text box empty in the [CSS Tab](#).

Builder CSS

This read-only style sheet component is included on all pages for all layouts.

It contains a small number of styles required for correct operation of pages customised with Theme Builder layouts.

You can prevent this style sheet from being included in a layout by de-selecting the "Use inbuilt CSS for this layout" option on the [CSS Tab](#).

Panel CSS

This component is derived from the [Panel Editor](#) settings on the [Layout Tab](#) and is therefore hierarchical in nature.

To remove the Panel CSS component, de-select the "Use panel CSS for this layout" option on the [CSS Tab](#).

Custom CSS

This style sheet component is edited via the Custom CSS setting on the [CSS Tab](#).

If the layout does not have any custom CSS, enter `/* empty */` in the Custom CSS section of the CSS Tab.

To remove the Custom CSS component, de-select the "Merge this layout's CSS with it's parent's" and leave the Custom CSS text box empty in the [CSS Tab](#).

IE CSS

This style sheet is customised via the [CSS Tab](#) and is unique in that it only gets loaded if the end-user is browsing your wiki with Internet Explorer 5.5 or above by default. You can customise the version of Internet Explorer for which the style sheet by editing the [conditional IF statement](#) on the CSS tab.

Each layout has it's own version of the IE CSS which is intelligently cached for all pages viewed using that layout.

The web browser will generally cache the IE CSS for up to a year, unless you make alterations in which case the browser will download a new version.

Note: Read important notes on [Style Sheet Inheritance](#).

To remove the IE CSS component, de-select the "Merge this layout's CSS with it's parent's" and leave the IE CSS text box empty in the [CSS Tab](#).

Caching: [Cache Expire](#) (1 year)

Servlet: /plugins/builder/layout-resources.action

Resources: iecss

Source: Layout - each layout has it's own version of this style sheet.

FAQs

We don't recommend doing this as it will prevent them from being cached and will also increase the size of every HTML page generated by Confluence.

However, if you want to embed styles in this way, use Confluence's Custom HTML feature (which can be found in the [Administration Console](#)) and add your markup to the <head> section.

There are two methods for adding styles to specific pages, the first one being the most "technically correct"...

- 1. Create a new layout containing the desired CSS alterations and then apply it to specific pages using the [use-layout macro](#), or...*
- 2. Simply insert additional styles in to the page body using the [style macro](#).*

While using the style macro is probably the easiest and generally works just fine, it will result in slower page loads because it's adding <style> tags inside the HTML <body> region. [Why?](#)

When you use a page-level layout (via the [use-layout macro](#)), you're ensuring that your style sheets are output at the top of the HTML in the <head> section.

We started looking in to what would be required to remove the <link> tag that includes the Combined CSS and came to the conclusion that the amount of extra work required, both in terms of our development time and also the extra processing time required on the server for every page request, was just not worth it. We'd hit the point of "diminishing returns" where the amount of work could not be justified by what would be achieved.

As such, even if all components are disabled the Combined CSS resource will still be included, it will just be empty.

It should be noted that due to the caching used both by the browser and on the server, this will have very little effect on performance. The first time a user views a page for such a layout, their browser will download a very small (just a few bytes) CSS file which will then be cached for up to a year by their browser.

Furthermore, we envisage that this will be a very rare scenario because most layouts will generally have at least one of the components enabled.

See Also

- [CSS Tab](#) - the bulk of CSS editing and configuration happens on this tab
- [Layout Tab](#) - this configures the Panel CSS, specifically via the [Panel Editor](#)
- [Menus Tab](#) - determines which, if any, menu style sheets are loaded