

Style Sheet Inheritance

Style Sheet Inheritance

Theme Builder 3.x and above allows you to inherit style sheet properties from parent layouts, however this process is different to that described in [Layout Hierarchy](#) for certain CSS resources.

This document only discusses CSS resources that have some form of inheritance. For a general overview of all CSS resources, please see [Style Sheets 3.x](#).

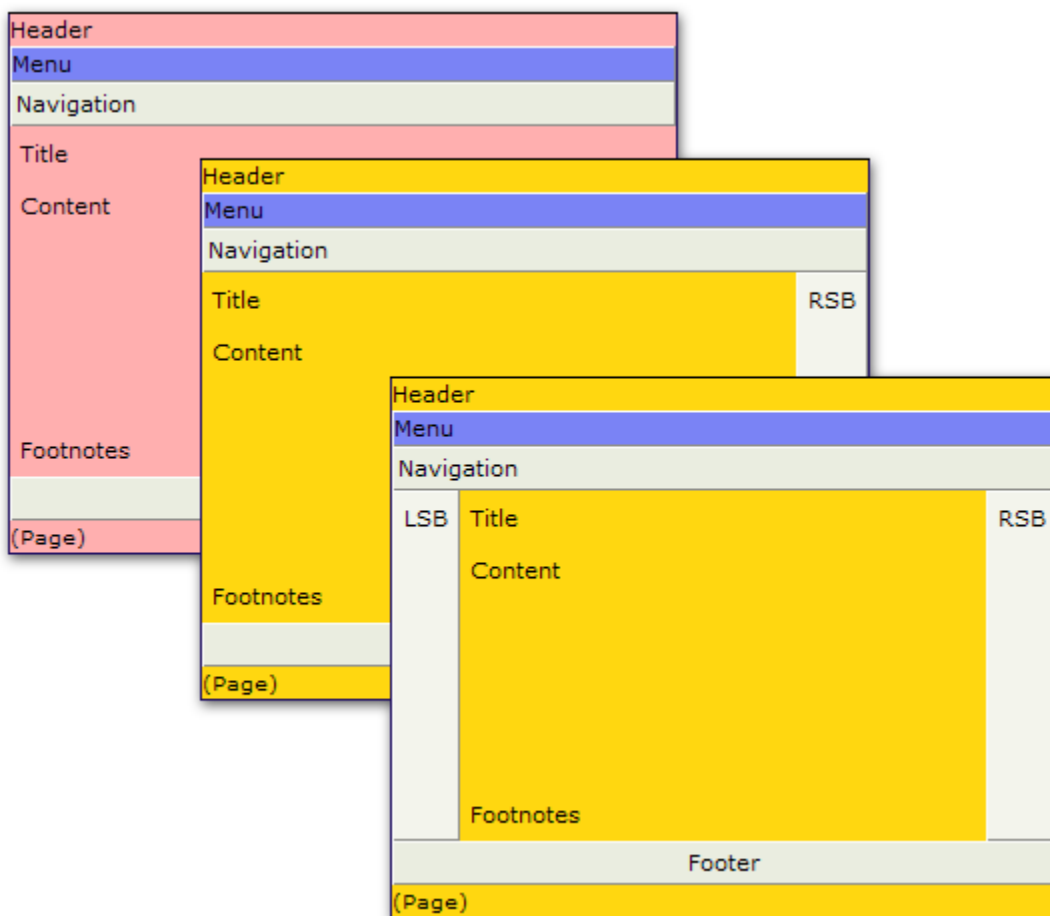
Panel CSS

The Panel CSS resource uses the same form of inheritance as described in [Layout Hierarchy](#).

Settings such as [Background Settings](#), [Border Settings](#) and [Text Settings](#), etc., can quickly be configured using the visual editor in the [Panel Editor](#).

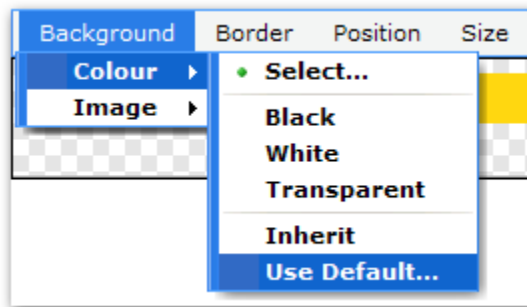
These settings are used as the basis of the "Panel CSS" style sheet, which is output as a component within the [Combined CSS](#) resource.

By default, all panel properties will be inherited from the parent layout unless specifically set in the current layout as illustrated below:



When you override a specific property, such as background colours, your layout will use that setting rather than the setting found in the parent layout.

You can reset any panel CSS setting back to default (ie. inherit from the parent layout) using the "Use Default..." link associated with the setting:



For a full description of this process, please see [Layout Hierarchy](#).

De-select the "Use panel CSS for this layout" option on the [CSS Tab](#).

Imports, Custom CSS and IE CSS

These two CSS resources are defined manually using the dialog which is launched from the buttons in the [CSS Tab](#).

The Imports and Custom CSS settings are components which are output as part of the [Combined CSS](#).

The IE CSS component is a separately loaded resource which is only loaded if the end-user is using Internet Explorer 5.5 or above.

You can see how and where these components get loaded on the [Style Sheets 3.x](#) page.

All three of these resources allow inheritance from the parent layout, but in a different way to everything else due to, well, they just do. It's a long story and beyond the scope of this document.

By default, the parent settings for these three components are combined with the settings in the current layout.

For example, if the parent layout Custom CSS contained this:

```
.atb-page h1 {  
  color: red;  
}
```

And the current layout Custom CSS contained this:

```
.atb-page h1 {  
  color: green;  
  border: 1px solid red;  
}
```

The actual CSS that would be output by the current will look something like this:

```
.atb-page h1 {  
  color: red;  
  background-color: yellow;  
}  
.atb-page h1 {  
  color: green;  
  border: 1px solid red;  
}
```

Because the CSS in the current layout is output after the CSS from the parent layout, it will supersede (i.e. override) any identical settings. The result is a heading that looks a bit like this:

This is what your heading 1 text would look like.

The background colour has been retained from the parent layout, because the current layout doesn't override it. However, the red text color specified by the parent layout has been overridden by the green text colour specified in the current layout. In addition, the current layout also adds a border style to the heading.

One of the really neat things about this is that you can then change CSS in the parent and all the child layouts will get updated (unless they supersede those bits of the CSS).

Yes, it does.

When the parent styles are merged with the styles from the current layout (as shown above) the resulting file size of the CSS resource will indeed be bigger because there will likely be some duplication of items (eg. the colour in our example) and there will also be additional styles (eg. the border in our example).

It should be noted however that we've ensured that these CSS resources are heavily cached by the browser - for up to 1 year or until the layout is edited, whichever is first.

The main benefit to merging CSS in this way is one of convenience - it makes it really quick and easy for you to add or supersede styles in the current layout.

Another benefit is server load - because we're doing very simple string concatenations (ie. joining the two chunks of CSS together) it's very fast and doesn't waste resources on the server. The web browser is left to do the complex work of deciding which style settings should be used and in which order.

De-select the "Merge this layout's CSS with it's parent's" option on the [CSS Tab](#).

Tick the "Merge this layout's CSS with it's parent's" option on the [CSS Tab](#) to include the parent CSS, then delete any CSS that's present in the current layout as applicable.

First, check that the "Merge this layout's CSS with it's parent's" option on the [CSS Tab](#) is ticked.

If styles defined in the parent layout are still not being applied then the most common cause will be styles in the current layout superseding them.