

Styling with LESS (using default shipped servoy less file)

When creating a Servoy NG Client solution or module, it is possible to enable the usage of a Servoy default [less](#) theme, that styles the basic components and configures the colors, borders or fonts used in the application.

New solution

Create a new solution

- a new Servoy solution project will be created

Solution name:

Solution type:

☒ Use default location

Location:

Please choose the resources project this solution will reference (for styles, column/sequence info, security)

☐ Create new resource project

☒ Use existing resource project

☐ No resources project

☒ Add default servoy .less theme (configurable by a less properties file)

If 'Add default .less theme' is checked in the New solution wizard, the `custom_servoy_theme_properties.less` and `<solution_name>.less` files are generated in the media node of the solution:

The **solution .less** file may contain your own css/less styles and using variables from the `properties.less` file is allowed. Here you can also add your own .less variables which can be used in your own less style.

The **custom_servoy_theme_properties.less** file can be used to configure the theme properties, which are defined as [less variables](#). The file should not be edited directly, it can be opened with the Servoy Theme Properties editor (on double click).



Styling/theming can be a bit different in Titanium NGClient compared to NGClient1 because we are now build on top of Bootstrap 5 but also because some components are now a bit differently build up in the browser so existing selectors could not match on them. Some components are also now quite different because of the underlying library change, like the Datepicker.

With 2021.03 release we added a special Servoy theme for NGClient2, the best way to start using this for testing NGClient2 but not disrupting also NGClient1 is to make a copy of your solution stylesheet, if you stylesheet is named "mysolution.less", make a copy and name that copy "mysolution_ng2.less". If you startup a NGClient2 then when serving out the stylesheet, Servoy will check if there is a "_ng2" variant of the given solution stylesheets name. In that "mysolution_ng2.less" stylesheet you can then point to another copy: `@import 'custom_servoy_theme_properties_ng2.less';` you can make, and that properties less file then points to the new "2021.3.0_ng2" Servoy theme file. This way if you start the same solution in NG1 you will just be served out the normal less/css file including our ng1 servoy stylesheet through 'custom_servoy_theme_properties.less' and starting in in NG2 we will serve out another set of stylesheets and a different Servoy theme file. After this you can then start tweaking the NG2 look in the "_ng2" solution stylesheet.

Using this editor, you can select the Servoy theme version:

For example you can change the value of the `@main-color` variable to use a different primary color applied to your theme. You can use any applicable CSS or LESS value as: `#E9720B`, orange, `lighten(#E9720B, 10%)`... You can also reference other less variables applicable to the same property, like `@text-color-inverse`: `@text-color`. If the property value is different than the default, the label color of the property is blue:

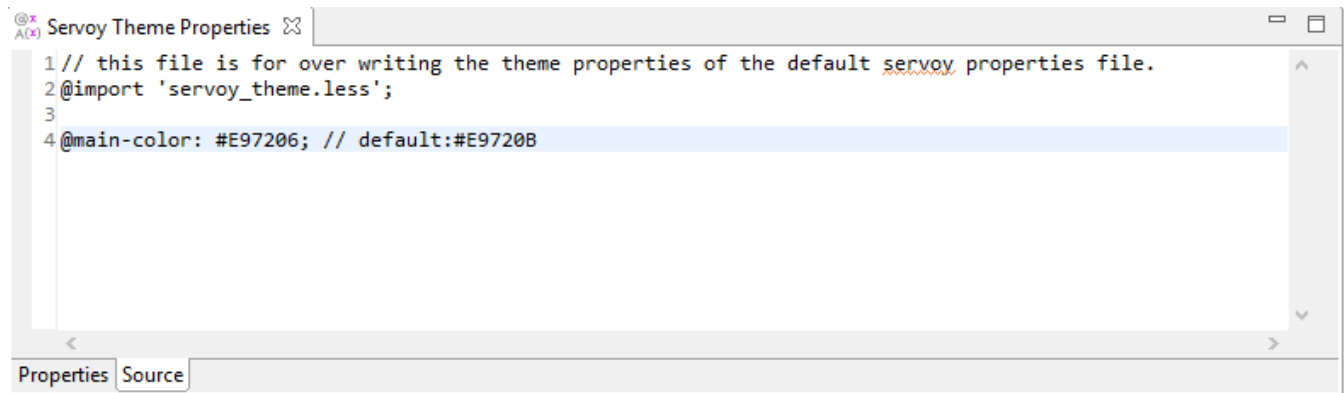
To reset the value to the default, right click the property name.

In case you upgrade the theme to a newer version, and there are conflicts between properties values (so a value that was changed in the editor was also changed in the theme),

the labels will be colored in red:

You can also view the actual content of the **custom_servoy_theme_properties.less** file, in the Source tab of the editor.

NOTE CAREFULLY: In case you choose to adjust the file manually, you should not remove any less variable or add any new less variable.



```
1 // this file is for over writing the theme properties of the default servoy properties file.  
2 @import 'servoy_theme.less';  
3  
4 @main-color: #E97206; // default:#E9720B
```

The screenshot shows a code editor window titled 'Servoy Theme Properties'. The code is written in LESS and includes a comment, an import statement for 'servoy_theme.less', and a definition for the @main-color variable. The variable is set to #E97206, with a default value of #E9720B in a comment. The editor has a 'Properties' tab selected at the bottom.

CSS Theme classes

The Servoy themes provides a set of utilities classes which can be applied to Servoy elements. The full list of classes which can be used when the LESS theme is enabled is available [here](#).