

Stylesheets

- [Styling cheatsheet](#)
- [Using Google fonts](#)
- [What are Stylesheets](#)

Styling cheatsheet

Shared page styling

All pages are structured in the following CSS class structure

- .TempusServaPage (root)
 - .menuList
 - .menuItem
 - ... content ...

Content can easily be wrapped around the "TempusServaPage" by use of wrappers

Command bases styling

Menu mode

Simple / Standard

Advanced / Accordion

List mode

- .TempusServaPage
 - .tableListHeader
 - .viewList
 - .tableList
 - tr
 - td (field names)
 - **tr [repeat]**
 - td (field values)
 - .pager
 - .pagerLink
 - .pagerPagesize

type	class	content
tr	customStyle_[item status name]	Container for a single record line

Variant views

Most list related operations will have a structure like this.

- .TempusServaPage
 - .tableListHeader
 - [**.viewHeat** | **.tableStatistics** | ...]
 - .tableList

Note the inner table "tableList", that is present in all list operations. Specific dialogue styling should use the outer tag - example: "tableStatistics".

Item mode

- .TempusServaPage
 - *.tableForm*
 - .PAGE_[field PageID]
 - .fieldLabel
 - .fieldValue
 - .fieldNotes

Although the ".tableForm" is present in the default template, there is no guarantee that the element can be found within the page.

type	class	id	content	template
tbody	PAGE_[field PageID]		Container for a single field	{default_start} / {default_end}
div	FieldLabel	NB_[system fieldname]	Label for field	{default_name}
div	FieldValue	VB_[system fieldname]	Value of field / Input for field	{default_value}
div	FieldNotes	HB_[system fieldname]	Optional help text for the field	{default_help}

Further information on: [Form templates](#)

Other element styling

ITEM page selector

The page selector may be placed anywhere within the template

- formpageselector
 - tablePageSelector
 - <node>
 - tablePageSelectorElementLeftActive
 - tablePageSelectorElementActive
 - tablePageSelectorElementRightActive
 - <node>
 - tablePageSelectorElementLeft
 - tablePageSelectorElement
 - tablePageSelectorElementRight

Targeting mobile devices

The following example shows a **script** that dynamically assigns a red background to the menu, but only while using mobile devices.

```
if (/Android|webOS|iPhone|iPad|iPod|BlackBerry/i.test(navigator.userAgent)) {  
    $('#menuList').css('background','red');  
}
```

Hiding form update elements

As of build 2345 there is no easy way to refer ALL the update elements (some browsers are not able to collapse cells with hidden contents).

```
<script>  
    $('#NB_DATA_StatusID').parent().parent().hide();  
    $('#NB_DATA_Revision').parent().parent().hide();  
    $('#updateSubmit').parent().parent().parent().hide();  
</script>
```

Using Google fonts

Find a font you like

<http://www.google.com/webfonts>

Click "Quick use" on the font you want to use and follow the directions.

Change your stylesheet like this example

```
...  
@import url(http://fonts.googleapis.com/css?family=Molle:400italic);  
...  
.TempusSeraPage h3 { font-family: 'Molle'; }  
...
```

The import method is needed, but embedding by a link inside a wrapper is also possible.

What are Stylesheets

Stylesheets can be applied in three ways

- Attached to solution
- Attached to solution interface
- Default application stylesheet

Stylesheets are defined in: Designer > Ressources > Stylesheet

Stylesheets contain CCS code to markup the page elements. The stylesheets may inherit content from each other, which is specified by the *Parent stylesheet*. Parent stylesheets are automatically prepended to the stylesheet in question, so definitions may be overloaded if needed.

A good ressources for learning CSS can be found here: <http://www.w3schools.com/css/>

The default stylesheet for the application is defined in the policy: **defaultStylesheetID**