

# Styling with CSS

## Table of Contents

- Table of Contents
- Overview
- Creating a Style
- Working with a Style
  - Structure of the Style Sheet
- Using Styles in Forms
  - Setting the style in the form
  - Setting styles to form elements

## Overview

Servoy supports the use of HTML *cascading style sheets* (CSS) for styling applications. These text files are stored in the resource project. A resource project can have many CSS files, and more than one CSS can be used in a solution.

The styles in a CSS apply to a form and the elements on a form. Although the solution can utilize multiple style sheets, only one style sheet at a time can be used on a form.

Benefits of using CSS in Servoy include:

- Adds to the consistency of the user interface; form elements across the application can use the same styles.
- Gives the developer to make style changes over many/all forms by changing a single value in the CSS.
- Style sheets can be used over multiple applications, adding consistency to all the company's applications.
- Adds consistency to an application in team development, making it easier for development teams to use the same styling on all forms.
- Style sheets can be changed programmatically, allowing a developer to change styles to different users' taste or to have periodic style changes in their applications. See [overrideStyle](#) function for more details.

## Creating a Style

To create a style in Servoy:

1. Do one of the following
  1. Select File>New>New Project from the menu in Servoy Developer. A window will appear; select the Servoy node and select New Style, click Next.
  2. Select the New Button in the top toolbar. A window will appear; select the Servoy node and select New Style, click Next.
  3. Right click on the **Styles** node under the Resources node. Select **Create new style** from the menu. A window will appear.
  4. With the Styles node selected, right click on any existing style and select **Create new style** from the menu. A window will appear.
  5. With the Styles node selected, select the Create new style button in the Solution Explorer list view toolbar. A window will appear.
2. Enter the name of the new style. Click Next.
3. Select to add sample content. Click Finish.
4. The new css file will appear in an editor view in the center of the Workbench.

### TIP

It is normally easier to work with an existing style. Most of the sample solutions have a style associated with them and these styles are imported when the solution is imported into Servoy Developer. You can also create a new style and copy/paste different entries from one style to another.

## Working with a Style

To open an existing style sheet:

1. Click on **Styles** under the Resources project in the Solution Explorer and do one of the following:
  1. Select the style desired (shown in the Solution Explorer list view) and click on the **Open style** button in the Solution Explorer list view toolbar.
  2. Double click the style desired.
  3. Right click on the style desired and select **Open style** from the menu.
2. The style will open in the center of the Workspace in an editor view.

## Structure of the Style Sheet

A style sheet for Servoy has basic style definitions and style definition classes.

The style definitions for Servoy are as follows:

- form
- label
- button
- field
- check
- radio
- combobox
- tabpanel
- portal

Under any of these style definitions, the developer can create many style definition classes. For example, the *label* style definition could have the style definition classes *label.title*, *label.small*, and *label.bold*.

Each definition and definition class can have one or many properties associated with it. Properties specified within the style definition are inherited (cascaded) to any style definition class under it. Study the example below

```
label
{
color: #ffffff;
border-style: solid;
font: bold 10pt Verdana;
}

label.mytext
{
color: red;
vertical-align: middle;
border-width: 1px 1px 1px 1px;
border-color: #111111 #111111 #111111 #111111;
margin: 2px 2px 2px 2px;
}
```

Notice that the border style and font are not modified in the *mytext* class. This means if a label were specified to use *mytext* for its style, it would be bold, 10pt, Verdana because that is what is specified in the parent style definition. The color would be red (not black) because that was overridden by the *mytext* definition class.

## Using Styles in Forms

In order to use a style sheet, the style sheet must be applied to a form. Solutions do not have a style sheet applied, only forms. Styles for the elements on a form are dependent on the style sheet applied to the form.

### *Setting the style in the form*

A style can be applied to a form in one of two ways.

- During form creation - The option to set a style for a form is available in the New Form Wizard. See [Creating Forms](#) for more details.
- On the Form Properties view - Change the styleName property to the desired style sheet name.

#### **TIP**

If you are creating forms, the style sheet selected for the first form will be automatically selected in the New Form Wizard for subsequent forms. If you happen to not enter a style sheet in the wizard, but would like to apply a style sheet to many forms, just multiple select the forms in the Solution Explorer and edit the Properties view. This will change the property on all the forms you have selected.

### *Setting styles to form elements*

Any elements on a form with a style applied AND have a style definition entered for the element type will take the styling from the style definition entry. For example, if I am using the style *test*, and there is a *label* entry in the *test* style sheet, then ANY label placed on the form will take the *label* style from the style sheet.

To use something other than the default style for any given element, change the styleClass entry in the Properties view for that element. To do this:

1. Select the desired element.
2. In the Properties view, find the property styleClass.
3. When clicking in the field, a drop down list will appear with all the available style classes for the element. This list is contextual for the element (meaning you will only see label style classes if the element is a label, field classes if the element is a field, and so on.)
4. Select the desired class from the list.
5. If you want to go back to the default style definition, select DEFAULT.