

# Style Sheet Editor

This chapter covers the following topics:

- About the Style Sheet Editor
  - Creating a Folder for Your Own Style Sheets
  - Invoking the Style Sheet Editor
  - Defining the Location for Your Own Style Sheets
  - Creating a New Style Sheet
  - Opening an Existing Style Sheet
  - Changing a Style Sheet
  - Overview of Variables
  - Regenerating Your Own Style Sheet from the Style Sheet Template
  - Using a Version Control Tool
- 

## About the Style Sheet Editor

All controls that are contained inside the control library are rendered using a style sheet. Ajax Developer delivers a variety of predefined styles but also allows to you to create your own styles sheets.

The Style Sheet Editor simplifies the creation of your own style sheets: on the one hand, you can define the very basic style elements (main colors to be used), and on the other hand, you can change a controls' style definition on the lowest level.

For information on how to define a style sheet in the Layout Painter, see *Defining a Style Sheet*.

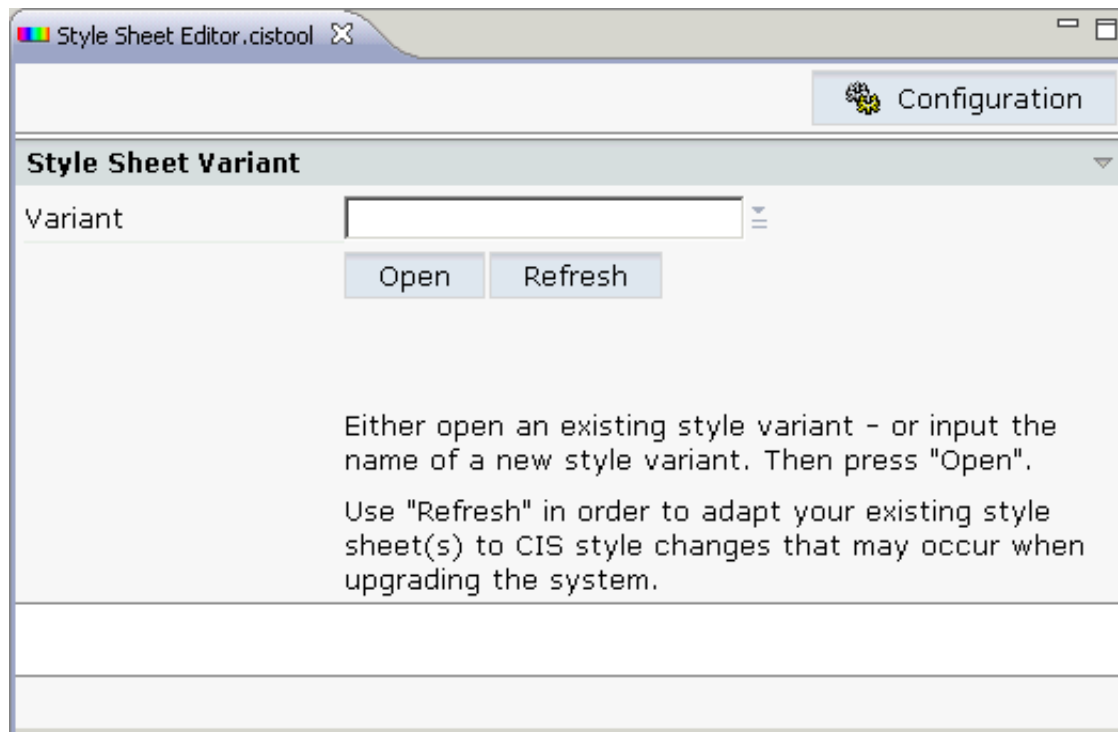
For further information on style sheets, see *Adapting the Look & Feel in Special Development Topics*.

## Creating a Folder for Your Own Style Sheets

Style sheets are part of your application's user interface and should therefore be maintained in a user interface component. It is good practice to maintain your project-specific style sheets in a subfolder of a user interface component, for example, in a subfolder named *styles*.

## Invoking the Style Sheet Editor

When you invoke the Style Sheet Editor, the following is shown:



#### ▶ To invoke the Style Sheet Editor

1. In the **Navigator** view, select the project for which you want to invoke the Style Sheet Editor.
2. Invoke the context menu and from the **Ajax Developer** menu, choose **Style Sheet Editor**.

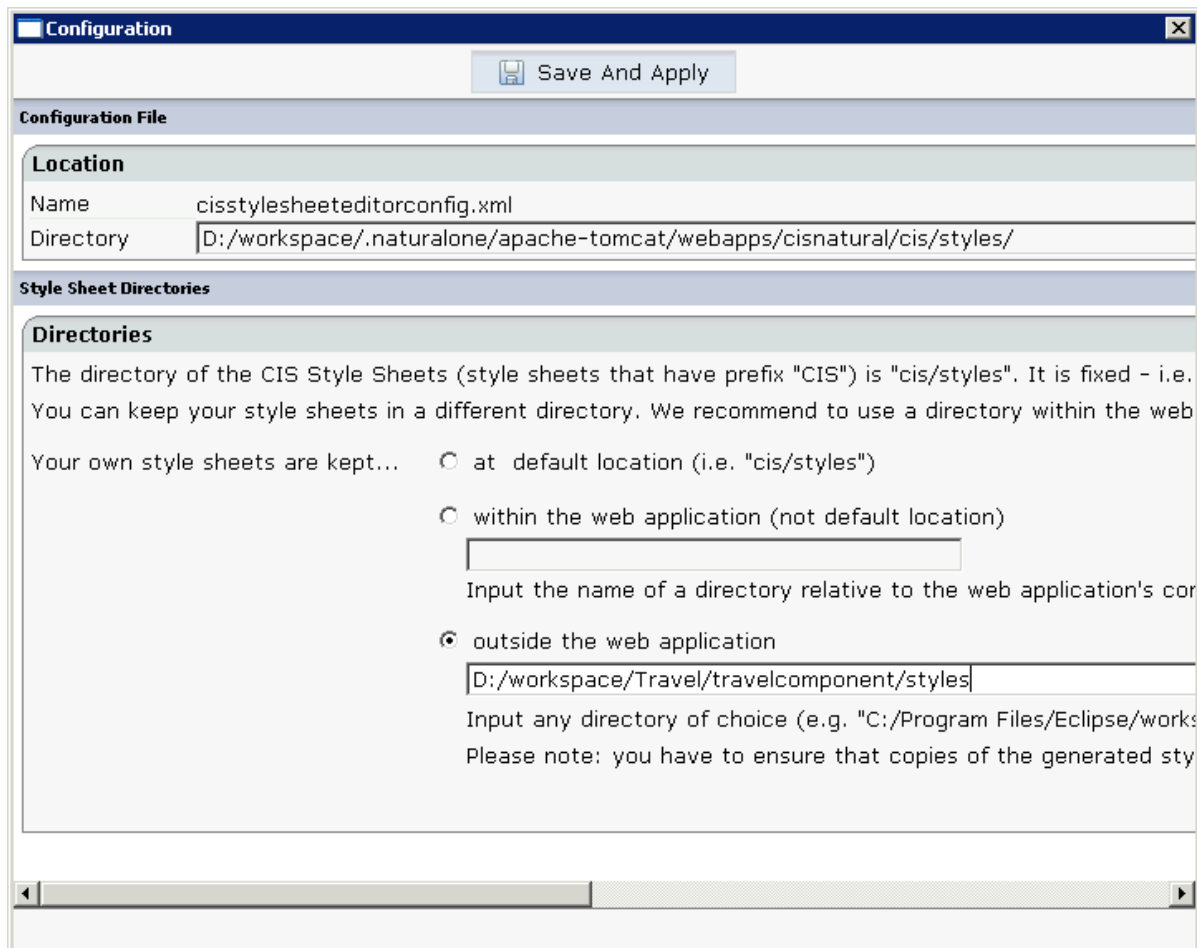
## Defining the Location for Your Own Style Sheets

Ajax Developer provides a set of predefined style sheets. In a NaturalONE installation, these style sheets are available in the directory `<eclipse-workspace>/naturalone/apache-tomcat/webapps/cisnatural/cis/styles`. Otherwise, these style sheets are located in `<your-webapplication>/cis/styles`. The names of the predefined style sheets have the prefix "CIS\_".

When you create your own style sheets, first copy the predefined style sheets into the style sheet folder of your user interface component.

#### ▶ To define the location for your own style sheets

1. Choose the **Configuration** button which is shown in top right of the Style Sheet Editor.
2. In the resulting dialog box, choose the option button **outside the web application**.
3. Specify the absolute path to your style sheet folder, for example, `<eclipse-workspace>/Travel/travelcomponent/styles`.



4. Choose the **Save and Apply** button.

## Creating a New Style Sheet

When you create and save a new style sheet, it is written to the directory that has been defined in the **Configuration** dialog.

You can determine that your new style sheet is immediately applied to the current session.

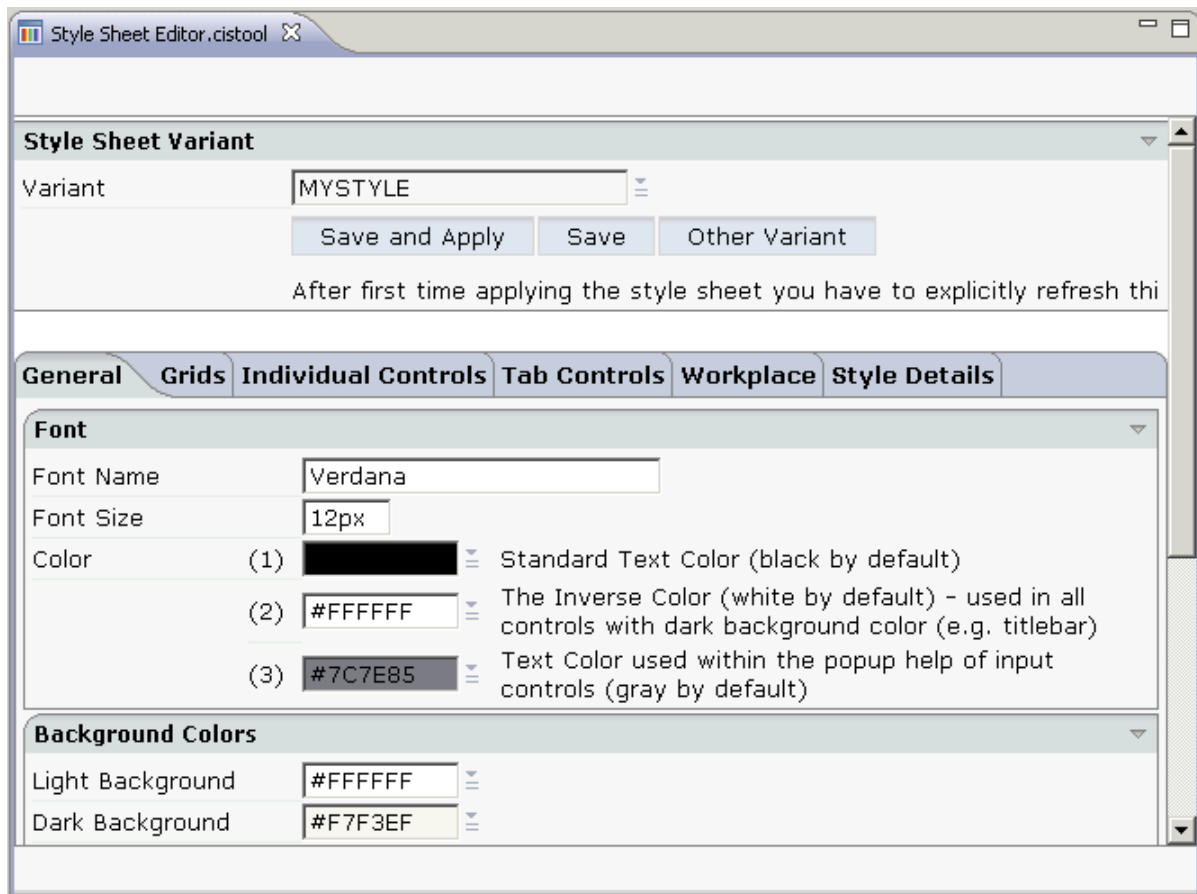
### Note:

By default, your pages are rendered with the *CIS\_DEFAULT* style sheet which is one of Ajax Developers' predefined style sheets.

### ▶ To create a new style sheet

1. In the **Variant** drop-down list box, enter the name for your new style sheet.
2. Choose the **Open** button.

The following is shown:



3. Modify the values on the different tabs according to your requirements. See *Changing a Style Sheet* for information on the **Style Details** tab.

**Note:**

When you choose the **Other Variant** button, the dialog is shown again which appears when you invoke the Style Sheet Editor. You can then open another style sheet or create a new style sheet. If you have not saved your input previously, it is lost.

4. To save your changes, choose either the **Save** button or the **Save and Apply** button.

When you save a new style sheet for the first time, it is created in your style sheet directory with the extension *css*.

When you choose the **Save and Apply** button, the style sheet file is immediately applied to the current session.

## Opening an Existing Style Sheet

When you open an existing style sheet, you can modify it according to your requirements and/or immediately apply the style sheet to the current session.

▶ **To open an existing style sheet**

1. From the **Variant** drop-down list box, choose the name of the style sheet that you want to open.

**Note:**

In addition to the style sheet names, the drop-down list box also shows the paths to the corresponding files in the file system.

2. Choose the **Open** button.

The dialog with the style sheet definitions appears.

3. Modify the values on the different tabs according to your requirements. See *Changing a Style Sheet* for information on the **Style Details** tab.

**Note:**

When you choose the **Other Variant** button, the dialog is shown again which appears when you invoke the Style Sheet Editor. You can then open another style sheet or create a new style sheet. Any changes that you have applied after the last save are lost.

4. To save your changes, choose either the **Save** button or the **Save and Apply** button.

When you choose the **Save and Apply** button, the style sheet file is immediately applied to the current session.

## Changing a Style Sheet

Ajax Developer provides an internal style template in which all the default style information is kept. A newly created style sheet is identical to this template. All style updates that you apply to your style sheet are stored within a separate file which has the extension *info*. For example, if you named your style sheet "MyStyle", your style sheet folder contains the file *MyStyle.info*.

**Note:**

The style sheet folder also contains GIF files. They are generated according to your specifications for background, font, border color, etc. Each style sheet has thus individual GIF files. When you modify, for example, a color setting, new GIF files are generated which overwrite the previous GIF files.

When you edit a style sheet, you define the standard settings such as the font size on the **General** tab. All of the settings defined on the **General** tab are then assigned to individual controls on the **Style Details** tab via the corresponding variable (for example, @@FONTSIZE@@). See *Overview of Variables*.



*	Attribute	Value
<input type="checkbox"/>	<b>color</b>	<b>#FFFF00</b>
<input type="checkbox"/>	font-family	@@FONT@@
<input type="checkbox"/>	font-size	@@FONTSIZE@@
<input type="checkbox"/>	color	@@FONTCOLORDARK@@

**Notes:**

1. If you do not want to take over a property from the style sheet template, specify "DELETE" as the attribute value and choose the **Update** button. All attributes that are marked with "DELETE" are not included in your style sheet.
2. If you want to take back a style update (that is: a change which is indicated by a bold line), delete the corresponding attribute value (that is: set it to blank) and choose the **Update** button. The bold line will disappear.

## Overview of Variables

The following variables can be defined on the **Style Details** tab:

```

@@FONTCOLORDARK@@
@@FONTSIZE@@
@@FONT@@
@@LIGHTBACKGROUND@@
@@HEADLINEBACKGROUND@@
@@BORDERCOLOR@@
@@DARKBACKGROUND@@
@@TITLEBARBACKGROUND@@
@@FONTCOLORLIGHT@@
@@BUTTONHEIGHT@@
@@BUTTONIMAGE@@
@@BUTTONCOLOR@@
@@FONTCOLORINACTIVE@@
@@CONTROLERRORBACKGROUND@@
@@CONTROLEDITBACKGROUND@@
@@CONTROLDISPLAYBACKGROUND@@
@@LIGHTTITLEBARBACKGROUND@@
@@VARIANT@@
@@FIELDHEIGHT@@
@@FIELDBORDERCOLOR@@
@@CONTROLPOPUPINPUTONLYBACKGROUND@@
@@SHADED DARKBACKGROUND@@
@@SELECTEDCELLBACKGROUND@@
@@SELECTEDBACKGROUND@@
@@EVENCELLBACKGROUND@@
@@ODDCELLBACKGROUND@@
@@TABAREALEFTPADDINGFIRST@@
@@TABAREALEFTPADDINGSECOND@@
@@EMPTYCELLBACKGROUND@@
@@SHADEDSELECTEDBACKGROUND@@

```

```
@@ODDCELLBACKGROUNDVAR1@@  
@@EVENCELLBACKGROUNDVAR1@@  
@@SELECTEDCELLBACKGROUNDVAR1@@  
@@EMPTYCELLBACKGROUNDVAR1@@  
@@ODDCELLBACKGROUNDVAR2@@  
@@EVENCELLBACKGROUNDVAR2@@  
@@SELECTEDCELLBACKGROUNDVAR2@@  
@@EMPTYCELLBACKGROUNDVAR2@@  
@@TITLEBARHEIGHT@@  
@@COLORTOPIC1@@  
@@COLORTOPIC2@@  
@@COLORTOPIC3@@  
@@COLORTOPIC4@@  
@@COLORTOPIC5@@  
@@COLORTOPIC6@@  
@@COLORTOPIC7@@  
@@COLORTOPIC8@@  
@@COLORTOPIC9@@  
@@COLORTOPIC10@@
```

## Regenerating Your Own Style Sheet from the Style Sheet Template

From release to release, new controls are added to the control library. As a consequence, the style sheet template is typically enhanced with every new control (for example, new style classes are added). As your style changes are kept in a separate file which has the extension *info*, your style sheet can easily include the enhancements. You just have to regenerate your own style sheet file.

### Important:

After the installation of a new Ajax Developer build, you have to regenerate your own style sheet(s).

#### To regenerate your own style sheet

1. From the **Variant** drop-down list box, select the name of your style sheet.
2. Choose the **Refresh** button.

The following command buttons are now shown: **Selected Variant** and **All Variants**.

3. Choose the **Selected Variant** button.

### Note:

Choose the **All Variants** button if you want to regenerate all existing style sheets.

## Using a Version Control Tool

If you want to secure your style sheets with a version control tool, it is sufficient to keep the files with the extension *info* under version control. The actual style sheet file (with the extension *css*) is just a generation result out of the INFO file.