Using Style Sheets (J2EE only)

The font, the color and the representation of the PF keys is controlled by a style sheet (CSS file).

The Natural Web I/O Interface client is delivered with the style sheet *3270.css*.

Natural for Ajax is delivered with a number of predefined style sheets. The default style sheet is *natural.css*.

This chapter covers the following topics:

- Location of the Style Sheets

- Editing the Style Sheets

- Switching to Another Style Sheet During the Session

- Modifying the Position of the Main Output and of the PF Keys

- Modifying the Font Size

- Modifying the Font Type

- Defining Underlined and Blinking Text

- Defining Italic Text

- Defining Bold Text

- Defining Different Styles for Output Fields

- Modifying the Natural Windows

- Modifying the Message Line

- Modifying the Background Color

- Modifying the Color Attributes

- Modifying the Style of the PF Key Buttons

For more information on style sheets, see *http://www.w3.org/Style/CSS/*.

Location of the Style Sheets

The location of the style sheets depends on the application server and type of client that you are using.

- **JBoss Application Server**
  Natural Web I/O Interface client:

../natuniapp.ear/natuniweb.war/resources

Natural for Ajax:

../njx<nn>.ear/cisnatural.war/resources

- **Sun Java System Application Server**
  Natural Web I/O Interface client:

  ../j2ee-apps/natuniapp/natuniweb_war/resources

  Natural for Ajax:

  ../j2ee-apps/njx<nn>.ear/cisnatural_war/resources

where *<nn>* is the current Natural for Ajax version.

Editing the Style Sheets

It is recommended that you have a basic understanding of CSS files.

You can edit the predefined style sheets or create your own style sheets.

It is recommended that you work with backup copies. When a problem occurs with your style sheet, you can thus always revert to the original state.

To see your changes in the browser, you have to

1. delete the browser's cache, and

2. restart the session.

Switching to Another Style Sheet During the Session

This feature is available with Natural for Ajax only.

If enabled in the configuration file for the session, a user can switch to another style sheet during a running session. In this case, the user can open the **Style Sheet** control in the output screen.

To switch to another style sheet, the user has to select it from the drop-down list box and then choose the **Apply** button.

Modifying the Position of the Main Output and of the PF Keys

Applies when only the named PF keys are displayed. This feature cannot be used when all PF keys are displayed, since they are always displayed at the same position. See also *Overview of Session Options*.

The following elements are available:

Element Name	Description
`#mainlayer`	Controls the position of the main output in the output window.
`#pfkeydiv`	Controls the position of the PF keys in the output window.

If the PF keys are to appear at the bottom, define these two elements as shown in the following example:

```
#mainlayer {
    top: 0px;
    left: 0px;
    height: 600px;
}

#pfkeydiv {
    top: 610px;
    left: 0px;
    width: 100%;
    height: 100px;
}
```

If the PF keys are to appear at the left, define these two elements as shown in the following example:

```
#mainlayer {
    top: 0px;
    left: 100px;
    height: 600px;
}

#pfkeydiv {
    top: 0px;
    left: 0px;
    width: 100px;
    height: 600px;
}
```

Predefined sample CSS files are also provided in which the PF keys are defined to appears at the right or at the top.

Modifying the Font Size

Depending on the screen resolution, one of the following style sheets for defining the font size is used in addition to the default style sheet:

- *model2.css*

- *model3.css*

- *model4.css*

- *model5.css*

These style sheets are located in the *tmodels* subdirectory of the *resources* directory in which all style sheets are located.

Depending on what comes closest to the standard 3270 screen model, the corresponding style sheet from the *tmodels* subdirectory is automatically used. It is selected according to the following criteria:

Standard 3270 Screen Model	Criteria	Style Sheet
Model 2 (80x24)	30 rows or less.	*model2.css*
Model 3 (80x32)	Between 31 and 40 rows.	*model3.css*
Model 4 (80x43)	41 rows or more.	*model4.css*
Model 5 (132x27)	30 rows or less, and more than 100 columns.	*model5.css*

The font sizes in the above style sheets can be adjusted. Example for *model4.css*:

```
body {
    font-size: 10px;
}
```

The default font sizes for the above 3270 screen models are:

Standard 3270 Screen Model	Default Font Size
Model 2	18px
Model 3	14px
Model 4	10px
Model 5	12px

When you modify the font size in one of the above style sheets, it is recommended that you also adjust the font width multiplication factor in the following JavaScript files:

- *natunicscript-ie.js* (for Internet Explorer)

- *natunicscript-ff.js* (for Firefox)

These JavaScript files are located in the *scripts* directory which can be found in the directory which also contains the *resources* directory.

You enter the font width multiplication factors in the definitions of the variables `FONT_WIDTH_MULT_MOD2` to `FONT_WIDTH_MULT_MOD5`. The following example shows the default values:

```
var FONT_WIDTH_MULT_MOD2 = 0.612;
var FONT_WIDTH_MULT_MOD3 = 0.573;
var FONT_WIDTH_MULT_MOD4 = 0.600;
var FONT_WIDTH_MULT_MOD5 = 0.586;
```

The following table shows the multiplication factors that should be used for the default font "Courier New":

Font Size	Multiplication Factor
9px	0.557
10px	0.60
11px	0.635
12px	0.586
13px	0.615
14px	0.572
15px	0.60
16px	0.625
17px	0.589
18px	0.612
19px	0.579
20px	0.600
21px	0.619
22px	0.591
23px	0.608
24px	0.584

Modifying the Font Type

As a rule, you should only use monospace fonts such as Courier New or Lucida Console. With these fonts, all characters have the same width. Otherwise, when using variable-width fonts, the output will appear deformed.

If you want to define a different font type, you should define the same font type for the body, the output fields and the input fields as shown in the following example:

```
body {
    background-color: #F3F5F0;
    font-family: Lucida Console;
    }

.OutputField {
    white-space:pre;
    border-width:0;
    font-family: Lucida Console;
    font-size: 100%;
}

.InputField {
    background-color: white;
    font-family: Lucida Console;
```

```
    border-width: 1px;
    font-size: 100%;
    border-color: #A7A9AB;
}
```

Defining Underlined and Blinking Text

The following elements are available:

Element Name	Description
.natTextDecoUnderline	Defines underlined text.
.natTextDecoBlinking	Defines blinking text.
.natTextDecoNormal	Defines normal text (no underline, no blinking).

Example:

```
/* Text decoration */
.natTextDecoUnderline { text-decoration:underline; }
.natTextDecoBlinking {text-decoration:blink; }
.natTextDecoNormal {text-decoration:normal;}
```

Blinking text is not supported by the Internet Explorer.

Defining Italic Text

The following elements are available:

Element Name	Description
.natFontStyleItalic	Defines italic text.
.natFontStyleNormal	Defines normal text (no italics).

Example:

```
/* font style */
.natFontStyleItalic {font-style:italic;}
.natFontStyleNormal {font-style:normal;}
```

Defining Bold Text

The following elements are available:

Element Name	Description
.natFontWeightBold	Defines bold text.
.natFontWeightNormal	Defines normal text (not bold).

```
/* Font weight */
.natFontWeightBold {font-weight:bolder}
.natFontWeightNormal {font-weight:normal;}
```

Defining Different Styles for Output Fields

The following elements are available:

Element Name	Description
.FieldVariableBased	Defines the style for output fields that are based on a variable.
.FieldLiteralBased	Defines the style for output fields that are based on a literal.

Example:

```
.FieldVariableBased {
   /* font-style:italic; */
}

.FieldLiteralBased {
    /* font-style:normal; */
}
```

Note:
In the above example, as well as in the standard CSS files delivered by Software AG, the variable-based output fields are defined as italic, but are commented out.

Modifying the Natural Windows

The following elements are available:

Element Name	Description
.naturalwindow	Controls the rendering of the Natural windows.
.wintitle	Controls the rendering of the titles of the Natural windows.

Example:

```
.naturalwindow {
    border-style: solid;
    border-width: 1px;
    border-color: white;
    background-color: black;
}

.wintitle {
    left: 0px;
    top: 1px;
    height: 17px;
    width: 100%;
    color: black;
```

```
    font-size: 100%;
    font-weight: bold;
    background-color: white;
    text-align: center;
    font-family: Verdana;
    border-bottom-style: solid;
    border-bottom-width: 2px;
}
```

Note:
In a mainframe environment, you have to set the Natural profile parameter WEBIO accordingly to enable this feature. See *WEBIO - Web I/O Interface Screen Rendering* in the *Parameter Reference* which is provided with Natural for Mainframes.

Modifying the Message Line

The rendering of the message line is controlled by the .MessageLine element.

Example:

```
.MessageLine {
    color: blue;
}
```

Note:
In a mainframe environment, you have to set the Natural profile parameter WEBIO accordingly to enable this feature. See *WEBIO - Web I/O Interface Screen Rendering* in the *Parameter Reference* which is provided with Natural for Mainframes.

Modifying the Background Color

The background color is defined in the body element.

Example:

```
body {
      background-color: #F3F5F0;
      font-family: Lucida Console;
}
```

Modifying the Color Attributes

You can define different colors for all Natural color attributes. These are:

Red
Green
Blue
Yellow
White
Black
Pink
Turquoise
Transparent

You can define these color attributes for input fields and output fields, and for normal output and reverse video.

The following examples show how to define the color attribute "Red".

Define the color for a normal output field:

```
.natOutputRed {color: darkred;}
```

Define the foreground and background colors for an output field with reverse video:

```
.reverseOutputRed {background-color: darkred; color:#F3F5F0;}
```

Define the color for a normal input field:

```
.natInputRed {color: darkred;}
```

Define the foreground and background colors for an input field with reverse video:

```
.reverseInputRed {background-color: darkred; color:#F3F5F0;}
```

Modifying the Style of the PF Key Buttons

The following elements are available:

Element Name	Description
.PFButton	Controls the style for normal rendering.
.PFButton:hover	Controls the style that is used when the mouse hovers over a PF key button.

Example:

```
.PFButton {
    text-align: center;
    width: 90px;
    border-style: ridge;
    border-width: 3px;
    padding: 2px;
    text-decoration: none;
    font-family: Verdana;
    font-size: 12px;
    height: 22px;
}

.PFButton:hover {
    color: #FFFF00;
    background-color: #222222;
}
```

Note:
In a mainframe environment, you have to set the Natural profile parameter WEBIO accordingly to enable this feature. See *WEBIO - Web I/O Interface Screen Rendering* in the *Parameter Reference* which is provided with Natural for Mainframes.