Archived:Stylesheet in WidSets

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article is believed to be still valid for the original topic scope.

Style HowTo

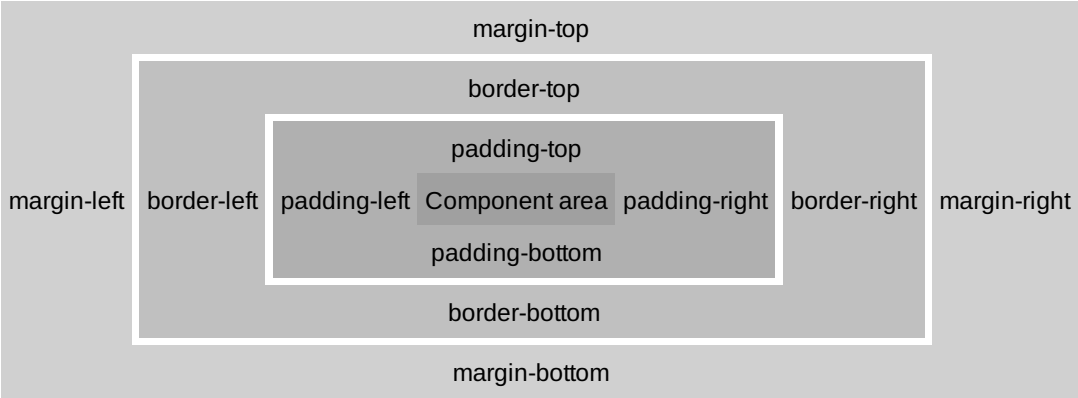
This article covers the use of the stylesheet format. Widget configuration 2.0 redefines some of our old styles, as it introduces a CSS-like syntax.

Styles are used to decorate the area surrounding a component, and to customize the contents of a component. Each style may have an optional focused variant that is used automatically when a component flagged as FOCUSABLE is focused. Focused variant when defined must be the last expression in the style.

It is important to ensure that the total area consumed by margin, border, and padding is equal in the normal style and the focused variant. Within the consumed area, the dimensions of margin, border, and padding may change as long as the total areas remains equal. Graphical glitches may occur if this constraint is violated.

The diagram below illustrates the margin, border and padding. Top, right, bottom and left dimensions are individually adjustable. It is not required to have any of them, and by default a style does not contain a margin, a border, or a padding. A style may also specify a background which is used to fill the padding and component area.

A style can also specify up to four fonts, colors, images, and arguments. They are used to customize the contents of the components.



Format

Styles are written with CSS-like notation. If the name of a style contains a reserved word (for example, background) it can be escaped with a dot.

```
menu
{
  width: 100%;
  align: left bottom;
  background: solid #163458;
  border: 7 5 0 5;
  border-type: image "/menu.png";
  padding: 0 2 5 2;
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
menu.item
{
  padding: 2 10 2 10;
  font: proportional medium bold;
  color: white;
  image: "/menu2.png";
  label-icon-orientation: right;

  // focused variant inherits all the properties of it’s parent
  focused
  {
    background: grid9 "/menuitem.png" 3 3 3 3;
  }
}

.background
{
  background: solid blue;
}
```

Margins

Margins define the empty area outside the given element.

```
margin: <top-dim> <right-dim> <bottom-dim> <left-dim>;
margin-top: <dim>;
margin-right: <dim>;
margin-bottom: <dim>;
margin-left: <dim>;
```

Border

Border defines the weight of the border element. You can specify a separate border weight for each side of a given border.

```
border: <top-dim> <right-dim> <bottom-dim> <left-dim>;
border-top: <dim>;
border-right: <dim>;
border-bottom: <dim>;
border-left: <dim>;
```

Border-type

Border-type specifies the type of border used within the style. You can define a line border by using *rectangle*, or specify an image as a border.

```
border-type: none;
```

- No border is drawn, even bounds specified in border-element are skipped.

```
border-type: transparent;
```


(C) Copyright Nokia 2014. All rights reserved.

- No border is drawn, but the space is reserved. Useful (in non-focused style) if you want to surround focused component with border. *(Since version 1.0)*

```
border-type: rectangle <color>;
```

- A border is drawn with solid color.

```
border-type: rectangle <top-color> <right-color> <bottom-color> <left-color>;
```

- A border is drawn like classic beweled button.

```
border-type: image <image>;
```

- The given image is split into nine regions according to border dimensions, as suggested by the diagram, and the regions are used when filling up the border area.

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

Regions 1, 3, 7, and 9 are used to fill corners of border area. Regions 2 and 8 are used to horizontally fill the upper and lower border areas, regions are repeated/clipped as needed. Regions 4 and 6 are used to vertically to fill the left and right borders areas, regions are repeated/clipped as needed. Region 5 is ignored.

Background

By defining a background for a style, you can create various effects.

```
background: none;
```

- No background is drawn.

```
background: solid <color>;
```

- A background is drawn with solid color.

```
background: hgradient <left-color> <right-color>;
```

- A background is drawn as horizontal gradient.

```
background: vgradient <top-color> <bottom-color>;
```

- A background is drawn as vertical gradient.

```
background: image <image> (transparent | <color>) <alignment> (repeat-x, repeat-y | no-repeat);
```

- A background is drawn using the given image. The image is aligned to the given position on the target area and repeated as

(C) Copyright Nokia 2014. All rights reserved.

specified. The remaining area is either left transparent or filled with a specified color.

```
background: grid8 <image> transparent <top-dim> <right-dim> <bottom-dim> <left-dim>;
```

- The given image is split into nine regions according to the given dimensions, and drawn like an image border. Padding and component areas are not drawn.

```
background: grid8 <image> automatic <top-dim> <right-dim> <bottom-dim> <left-dim>;
```

- The given image is split into nine regions according to the given dimensions, and drawn like an image border. Padding and component areas are filled with color picked from first (top left) pixel of region 5.

```
background: grid8 <image> <color> <top-dim> <right-dim> <bottom-dim> <left-dim>;
```

- The given image is split into nine regions according to the given dimensions, and drawn like an image border. Padding and component areas are filled with a specified color.

```
background: grid9 <image> <top-dim> <right-dim> <bottom-dim> <left-dim>;
```

- The given image is split into nine regions by given dimensions and drawn like an image border. Padding and component areas are filled using the region 5 of image, repeating and clipping as needed.

```
background: button <corner-size> <border-top-color> <border-bottom-color> <fill-top-
color> <fill-bottom-color>;
```

- Creates background with rounded corners (size of corner-size pixels). Separate border colors can be given for top and bottom parts. Filling is a vertical gradient between given 2 colors. Usable for "Button" styles. *(Since version 3.5)*

Possible flags are:

```
top
vcenter
bottom
left
hcenter
right
repeat-x
repeat-y
no-fill [indicates that color filling is not applied]
```

Padding

Padding allows you to define spacing in your content area.

```
padding: <top-dim> <right-dim> <bottom-dim> <left-dim>;
padding-top: <dim>;
padding-right: <dim>;
padding-bottom: <dim>;
padding-left: <dim>;
```

Color

(C) Copyright Nokia 2014. All rights reserved.

Colors can be defined within styles in multiple ways.

```
#rrggbb – HTML hexadecimal color
0xrrggbb
rgb(40, 80, #A0) – RGB tuple
Symbolic (LightSkyBlue) – HTML color name
```

Color is component specific; see an individual component for description. Note that there is overlap between color slots; style compiler will issue warning if it detects a conflict.

```
color: <color>;
color-1: <color>;
color-2: <color>;
color-3: <color>;
color-4: <color>;
cursor-color: <color>;
```

- See Text
- See HTML colors

Images

Image usage is component specific.

```
image-1: <image>;
image-2: <image>;
image-3: <image>;
image-4: <image>;
```

Fonts

Font is a free list of flags which are:

```
face:
  proportinal (default)
  monospace
  system

size:
  large
  medium (default)
  small

style:
  plain (default)
  bold
  italic
  underlined
```

Examples of font usage:

```
font-1: mono large bold
font-1: prop medium italic underlined
```


(C) Copyright Nokia 2014. All rights reserved.

- Font usage is component specific. For example, in normal feed syndication widgets, the first font defines the font that is used in the widget.

```
font-1: <font>;
font-2: <font>;
font-3: <font>;
font-4: <font>;
```

Arguments

Arguments define some component specific customizations. See an individual component for description. Note that there is overlap between argument slots; style compiler will issue warning if it detects a conflict.

```
argument: <integer>;
argument-1: <integer>;
argument-2: <integer>;
argument-3: <integer>;
argument-4: <integer>;
```

```
hspacing: <dim>;
```

- See Flow

```
vspacing: <dim>;
```

- See Flow

```
line-spacing: <dim>;
```

- See Text

```
line-alignment: left | hcenter | right;
```

- See Text

```
label-spacing: <dim>;
```

- See Label

```
label-icon-orientation: left | right;
```

- See Label

```
indicator-background: <color>;
```

- See Progress

```
indicator-foreground: <color>;
```


(C) Copyright Nokia 2014. All rights reserved.

- See Progress

```
indicator-height: <dim>;
```

- See Progress

```
ticker-speed: <integer>;
```

- See Ticker

```
ticker-mode: loop | bounce;
```

- See Ticker

Alignment

If the Java code has not defined any width and/or height attributes for Component, the width and height defined in the style sheet are used instead.

```
width: <width>;
height: <height>;
```

For relative dimensions, use the following:

```
width: -33;
width: 67%;
```

Some components also support alignment control:

```
align = (left|hcenter|right) (top|vcenter|bottom);
```

Default values

```
margin: 0 0 0 0;
border: 0 0 0 0;
border-type: none;
padding: 0 0 0 0;
background: none;
font-1: proportional medium plain;
font-2: proportional medium plain;
font-3: proportional medium plain;
font-4: proportional medium plain;
color-1: black;
color-2: black;
color-3: black;
color-4: black;
argument-1: 0;
argument-2: 0;
argument-3: 0;
argument-4: 0;
```


(C) Copyright Nokia 2014. All rights reserved.

```
width: 0;
height: 0;
align: top left;
```

Inheritance

Styles can be extended from existing styles.

```
baseText
{
  align: left bottom;
  background: solid #163458;
  padding: 0 2 5 2;
  color: black;
  font: medium plain;
}

redText : baseText
{
  color: red;
}
```

Note. Possible focused-variants are not inherited.

See also

- Archived:Introduction to developing WidSets widgets
- Archived:Widget files
- Archived:Mobile developing and design guidelines in WidSets
- Archived:Publishing your own WidSet widgets
- Archived:Widget Configuration 2.1

(C) Copyright Nokia 2014. All rights reserved.