How to modify the color of generated corner images

This How-To describes how to modify the corner images that appear in the menu and tabs of the skins.

Table of contents

Copyright © 2002 The Apache Software Foundation. Licensed under Apache License 2.0 All rights reserved.

1 Intended Audience

Users who want to change the colors of the corner images in the output html documents.

This explanation is also useful for skin developers to understand the corner image generation process.

2 Purpose

Forrest renders the corner images through Scalable Vector Graphics (SVG). It may be necessary to change the color of the corner images to be suitable for your own skin colors.

3 Prerequisites

- Understand how to use the skinconf.xml file to change the appearance of the output documents.
- Understand how to create or modify skins by following the Using Forrest document (topic "Forrest skins").

4 Steps

The procedure outlined below provides an understanding of how corner images are named (the contract) and then shows how to define new colors for these images by modifying the `src/documentation/skinconf.xml` of a project.

4.1 Understand how corner images are named (the contract)

The corner images are referenced in some .css files of the above-named skins; for example, in screen.css of the pelt skin:

```
/*Example from screen.css of pelt*/
...
/**
* Round corner
*/
#roundtop {
    background-image: url(images/rc-t-r-15-1body-2menu-3menu.png);
    background-repeat: no-repeat;
    background-position: top right;
}
...
```

The naming follows a contract which is described below. In general, the naming looks like:

```
images/{$name}-{$v-orientation}-{$h-orientation}-
{$size}-1{$backgroundColor}-2{$strokeColor}-3{$foregroundColor}
```

The first part (`images/{$name}-{$v-orientation}-{$h-orientation}-{$size}`) identifies which images is used and how big (width x height) it should be. The second part\ (`-1{$backgroundColor}-2{$strokeColor}-3{$foregroundColor}`) identifies the coloring of each portion of the image. The input parameter for the second part comes from the color profile of `src/documentation/skinconf.xml`. The second part is easily identifiable through the numbering 1-2-3.

Let us get into details:

`images`

Copyright © 2002 The Apache Software Foundation. Licensed under Apache License 2.0 All rights reserved.

Path to the xslt that creates the corner.
`images/ = {$FORREST_HOME}/main/webapp/skins/common/images/`
`{$name}`
In the common skin there are two XSLT files ready for use:

- `rc.svg.xslt`: handles rounded corners
- `dc.svg.xslt`: handles diagonal 45-degree corners

`name = [rc|dc]`
e.g.`rc`
`{$v-orientation}`
Vertical orientation of the corner images (top or bottom).
`v-orientation = [t|b]`
e.g.`t`
`{$h-orientation}`
Horizontal orientation of the corner images (left or right).
`h-orientation = [l|r]`
e.g.`r`
`{$size}`
Pixels size of the width **and** height of the corner image.
`size=x`
e.g.`5`
`{$backgroundColor}`
Any `<color name=""/>` element in the `skinconf.xml` (the `value="{$color}"` attribute will be applied).
e.g.`header`
`{$strokeColor}`
Any `<color name=""/>` element in the `skinconf.xml` (the `value="{$color}"` attribute will be applied).
e.g.`searchbox`
`{$foregroundColor}`
Any `<color name=""/>` element in the `skinconf.xml` (the `value="{$color}"` attribute will be applied).
e.g.`searchbox`

The corner images are made by generating a dymanic svg image to add the colors and size. Then this svg is serialize to the png image via the org.apache.cocoon.serialization.SVGSerializer (see docs).

FIXME (thorsten):
The following link is for pure debugging reason. test png image - this image is taken from the svg pipe instead of directly generating it.

4.2 Modifying the skinconf.xml of your project

modifying the `skinconf.xml` of your project (by default you find it at `[project-dir]/src/documentation/`).

Starting about line 155 you find a `<colors>` ... `</colors>` element with content commented-out:

```
<colors>
<!-- These values are used for the generated CSS files. -->

<!-- Krysalis -->
<!--
```

Copyright © 2002 The Apache Software Foundation. Licensed under Apache License 2.0 All rights reserved.

```
  <color name="header"    value="#FFFFFF"/>

  <color name="tab-selected" value="#a5b6c6" link="#000000" vlink="#000000" hlink="#000000"/>
  <color name="tab-unselected" value="#F7F7F7" link="#000000" vlink="#000000" hlink="#000000"/>
  <color name="subtab-selected" value="#a5b6c6" link="#000000" vlink="#000000" hlink="#000000"/>
  <color name="subtab-unselected" value="#a5b6c6" link="#000000" vlink="#000000" hlink="#000000"/>

...

-->

</colors>
```

To modify the colors of the corner images, you can either define your own `<color name=.../>` elements or uncomment one of the existing `<color name=.../>` elements and adjust the color value to your needs.

e.g.

```
<color name="tab-selected" value="#FF0000"/>
```

This affects all corner images whose `{$backgroundColor}`, `{$strokeColor}` or `{$foregroundColor}` is set to `tab-selected`.
For example, in `screen.css` (of the "pelt" skin) you find:

```
#roundbottomsmall {
    background-image: url(images/rc-b-r-5-1header-2tab-selected-3tab-selected.png);
    background-repeat: no-repeat;
    background-position: top right;
}
```

Now the stroke color (`-2tab-selected`) and the foreground color (`-3tab-selected`) are set to red (remember: we defined `#FF0000` as the "color" value of `tab-selected`).

4.3 Modifying .css files

In addition to the modification of `skinconf.xml` you can also modify the respective .css file of your skin.

Here's another example:

```
/*your .css file*/
...
#roundbottomsmall {
    background-image: url(images/rc-b-r-5-1foo-2secondfoo-3thirdfoo.png);
    background-repeat: no-repeat;
    background-position: top right;
}
...



<!-- your skinconf.xml -->
...
<colors>
  <color name="foo" value="#FF0000"/>
  <color name="secondfoo" value="#00FF00"/>
  <color name="thirdfoo" value="##00FF00"/>
</colors>
```

Copyright © 2002 The Apache Software Foundation. Licensed under Apache License 2.0 All rights reserved.

Here we have created our own color tags (in the .css file) and defined the respective values for them (in `skinconf.xml`). Now you have color images with a red background and a green foreground. Horrible, isn't it?

5 Feedback

Please provide feedback about this document via the mailing lists.

Copyright © 2002 The Apache Software Foundation. Licensed under Apache License 2.0 All rights reserved.