Overriding the output of Macros

The output of some Confluence macros are not optimal for display in a viewport. For example, if a theme uses Twitter Bootstrap ([http://getbootstrap.com/](http://getbootstrap.com/)) as HTML5/CSS framework, the output of the warning macro is not compatible with the HTML required by Twitter Bootstrap.

Therefore Scroll Viewport provides the functionality to override the output of macros. To do so there are a couple alternatives with different complexities (sorted by complexity):

1. Create a *Velocity Macro Override* to override the output of an arbitrary macro.
2. Use a *Macro Replacement* provided by Scroll Viewport and style the output with a velocity template.
3. Create your own macro to override an existing macro (planned).

**Velocity Macro Override**

The Velocity Macro Override is a simple way to define the output of a macro, based on the parameters of the macro.

To define and register a macro override simply create a velocity template in the theme using the naming scheme `overrides/<macro-name>.vm`. When Scroll Viewport renders a page that contains a macro with that `macro-name` it renders the template `overrides/<macro-name>.vm` and passes the macro parameters and the body (if existent) to the macro.

```
<%@ include file="overrides/panel.vm" %>
```

**Creating a simple Macro Override Template**

For example, to override the `panel` macro, create a file in the `overrides/panel.vm`:

```html
<div class="panel panel-default">
    #if($params.title)
        <div class="panel-heading">
            <h3 class="panel-title">$params.title</h3>
        </div>
    #end
    <div class="panel-body">
        $body
    </div>
</div>
```

**Available Placeholders**

In the macro override template the following placeholders are available:

- `$macroName` – the name of the macro
- `$params` – a map of parameters of the macro
- `$body` – the body of the macro (null for macros with no body)
- `$stringUtils` – `org.apache.commons.lang.StringUtils`
- `$stringEscapeUtils` – `org.apache.commons.lang.StringEscapeUtils`

Since Viewport 2.1 also all placeholders as described in the Page Context are available as well.

**Using a Different Template Location**

It’s also possible to configure where with templates should be used for which template as follows:

```javascript
$page.renderContent("include/include-content.vm", {
    "panel" : "overrides/special-panel.vm"
})
```
This was the standard approach in Scroll Viewport <2.0, however, we recommend using the naming convention `/overrides/<macro-name>.vm` as described above.

**Macro-Replacements Provided by Scroll Viewport**

Scroll Viewport provides some macros that replace built-in macros. To do so, Scroll Viewport replaces the macros before rendering, and also allows to override the output with a velocity template.

The advantage of this approach is that users can just use the macros they are used to, and Scroll Viewport replaces them on the fly.

**Styling the Macro Replacements**

Just create a velocity template in `/overrides/<macro-name>.vm` as described in Velocity Macro Overrides (above).

**List of Macro Replacements**

Currently the following Macro Overrides are available:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Macro Name</th>
<th>Compatibility</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attachments</td>
<td>attachments</td>
<td>PARTIAL</td>
<td>Currently, the Attachments macro replacement is limited.</td>
</tr>
</tbody>
</table>

**Creating your own Macro (planned)**

See here: [VPRT-202 - A Theme Developer can override the output of complex Macros](#)