

# The `sclayer` interface `sclayer-fancyhdr`\*

Markus Kohm

2019/04/04

## Contents

<b>1</b>	<b>The Purpose of this Package</b>	<b>1</b>
<b>2</b>	<b>How it works</b>	<b>2</b>
<b>3</b>	<b>How to use the Package</b>	<b>3</b>
<b>4</b>	<b>Known Issues</b>	<b>4</b>
<b>5</b>	<b>Implementation of <code>sclayer-fancyhdr</code></b>	<b>4</b>

## 1 The Purpose of this Package

This package has been made to give users a chance to combine the features of Piet van Oostrum’s `fancyhdr` [1] with the features of `sclayer` [2]. In other words: It has been made to combine the page layers of `sclayer` with the page styles of `fancyhdr`.

In this combination compatibility with `fancyhdr` is the first aim! Usability and the freedom provided by `sclayer` is only the second one. Compatibility with other parts of KOMA-Script is not a primary aim. Perhaps it will become an optional feature in future. Abolishing any real or virtual limitations of `fancyhdr` other than make it possible to use layers is not an aim and will not be an aim in future.

If you need a better combination of page styles and layers, you should either use the low level interface of `sclayer` to define your page styles or — and this is the recommendation of the author — use `sclayer-scrpage` instead of `sclayer-fancyhdr` or `fancyhdr`. If you need more compatibility with other parts of KOMA-Script, i. e. with the KOMA-Script classes, you should use `sclayer-scrpage` instead of `sclayer-fancyhdr` or `fancyhdr`.

---

\*This is version v0.1.3138 of file `sclayer-fancyhdr.dtx`.

## 2 How it works

To combine `fancyhdr` and `sclayer`, `sclayer-scrpage` loads both packages. After loading `fancyhdr` it redefines page style `@fancy` to make it a layer page style of `sclayer` using the newly defined layers `fancy.head.even`, `fancy.head.odd`, `fancy.foot.even` and `fancy.foot.odd`. The `...head...` layers are background layers like the page head of `fancyhdr`'s page styles (or other usual page styles). The `...foot...` layers are foreground layers like the page footer of `fancyhdr`'s page styles (or other usual page styles). The `...even` layers are restricted to even pages, that means left side pages in two-sided documents. The `...odd` layers are restricted to odd pages, that means right side pages in two-sided documents or all pages in single-sided documents.

`\ps@fancy`  
`\ps@fancyplain`  
`\ps@plain@fancy`

Note: Page style `@fancy` is an internal basic page style of package `fancyhdr`. It is used for the user page style `fancy`. The user page style `fancy` is also used for `fancyhdr`'s undocumented page style `fancyplain`, that also redefines page style `plain` to be `fancyhdr`'s internal page style `plain@fancy`, that also uses `fancy` but with `\if@fancyplain` set to `\iftrue`. There is also an undocumented command `\fancyplain{<plain code>}{<fancy code>}`, that uses the `<plain code>` if `\if@fancyplain` is `\iftrue` and `<fancy code>` if `\if@fancyplain` is `\iffalse`.

Note: Page styles defined using `fancyhdr`'s command `\fancypagestyle` also always use page style `fancy` and so the internal basic page style `@fancy`.

As a result of the two notes above, every page style of `fancyhdr` always uses page style `fancy` and the same internal basic page style `@fancy`. As a result of redefining page style `@fancy` to be a layer page style, users can add layers to or remove layers from all `fancyhdr` page styles by adding layers to or remove layers from page style `@fancy`. You cannot add layers to or remove layers from `fancyhdr`'s single page styles `fancy`, `fancyplain`, `plain@fancy` or the page styles defined using `\fancypagestyle` directly. So using layers is a all or nothing feature with `sclayer-scrpage`. However, you can use the second argument of `\fancypagestyle` to add or remove layers whenever one of the `fancyhdr` page styles is activated. So this is a move from the `sclayer` interface of adding or removing layers to single page styles to the `fancyhdr` interface of defining modifications of page style `fancy`.

Another such movement from an `sclayer` user interface to a `fancyhdr` user interface it the decision whether or not automatic running heads are used. `sclayer` provides the options `automark` and `manualmark` and commands `\automark` and `\manualmark` to do this decision and also to configure commands like `\partmark`, `\chaptermark`, `\sectionmark` etc.

With `sclayer-fancyhdr` also using page style `@fancy` does switch to automatic running heads. The first activation of page style `fancy` still redefines `\chaptermark` and `\sectionmark`, if a class with `\chapter` is used, or `\sectionmark` and `\subsectionmark`, if a class without `\chapter` is used. However, you still can use `\manualmark` and `\automark` after switching to a `fancyhdr` page style to configure the running heads. So this movement is only partial.

`\ps@headings` Note: Currently, neither `sclayer` nor `fancyhdr` nor `sclayer-fancyhdr` do  
`\ps@myheadings` redefine page styles `headings` or `myheadings`. And neither `sclayer` nor  
`\ps@plain` `fancyhdr` nor `sclayer-fancyhdr` do redefine page style `plain` unless you are activating the undocumented `fancyhdr` page style `fancyplain`. So if you like to use layers on `plain` pages, i. e. usually the first page of a chapter or part or the page with a title head, you have to either use `fancyhdr`'s undocumented page style `fancyplain` or redefine page style `plain` either using `\fancypagestyle` as documented in the `fancyhdr` manual or using `\DeclareNewPageStyleByLayers`, documented in the KOMA-Script manual.

`\ps@empty` Note: Pagestyle `empty` is somehow special. `sclayer` redefines it to be a  
`\ps@@empty` layer page style. And `fancyhdr`'s internal page style `@empty` is the same like `empty`. So `@empty` also uses the layers of `empty` but you should not try to modify it directly using the interface of `sclayer`. Moreover, if `fancyhdr` is loaded before `sclayer-scrpage`, `fancyhdr`'s internal page style `@empty` is not a copy of `sclayer`'s layer page style `empty` but the original standard page style `empty`. However, with `sclayer-fancyhdr` package `fancyhdr` does not longer use the internal page style `@empty`. So you would not need to know this.

### 3 How to use the Package

First of all, this package is experimental and should not be used for anything else but testing whether or not it works as documented! This package even is not an official part of KOMA-Script!

To use the package you have to load it, e. g., using:

```
\usepackage{sclayer-fancyhdr}
```

instead of loading `sclayer` and `fancyhdr` or before or after loading one of these packages. However it is recommended to replace loading `sclayer` and `fancyhdr` by loading `sclayer-fancyhdr` because this avoids option clashes. `sclayer-fancyhdr` provides all options of `sclayer` and passes them to `sclayer`. Nevertheless sometimes it may be useful to be able to additionally load `sclayer` or `fancyhdr`.

<code>\ps@fancy</code> <code>\ps@fancy</code> <code>\ps@fancyplain</code> <code>\ps@plain@fancy</code> <code>\fancypagestyle</code>	<p>After this you should be able to use the page styles and commands of <code>fancyhdr</code> and to add layers to or remove layers from the <code>fancyhdr</code>'s internal basic page style <code>@fancy</code>. Note, you are not able to add layers to or remove layers from <code>fancyhdr</code>'s page styles <code>fancy</code>, <code>fancyplain</code>, <code>plain@fancy</code> or page styles defined by <code>\fancypagestyle</code>. However adding layers to or removing layers from <code>@fancy</code> will always change all these page styles!</p>
<code>\ps@plain</code> <code>\ps@headings</code> <code>\ps@myheadings</code> <code>\ps@empty</code>	<p>Note: Loading <code>sclayer-fancyhdr</code> will not make page style <code>plain</code> nor <code>headings</code> nor <code>myheadings</code> or any other page style but <code>empty</code> to automatically be a layer page style!</p>

## 4 Known Issues

Please note, the follow issues are either notes to the package author or notes to the user to avoid them reporting the same issues again and again. Listing these issues does not say they are bugs or features.

- |  |  |
|--|--|
| <code>\ps@fancy</code><br><code>\ps@fancyplain</code><br><code>\fancypagestyle</code><br><code>\ps@@fancy</code> | <ul style="list-style-type: none"> <li>• You are not able to add layers to the user level page style <code>fancy</code> or <code>fancyplain</code> or any page style defined by <code>\fancypagestyle</code> but only to the internal page style <code>@fancy</code>. This is intended as explained in this manual.</li> </ul>   |
| <code>\ps@@empty</code>  | <ul style="list-style-type: none"> <li>• Using <code>fancyhdr</code>'s internal page style <code>@empty</code> could have strange results. However, there is a simple solution for this: Don't use the internal <code>@empty</code> but always the user level page style <code>empty</code>!</li> </ul>  |
| <code>\ps@@fancy</code>  | <ul style="list-style-type: none"> <li>• The vertical position of the page header does differ a little bit, if <code>sclayer-fancyhdr</code> is used instead of <code>fancyhdr</code>. This could be fixed by a modification of the layers <code>fancy.head.odd</code> and <code>fancy.head.even</code> using <code>addvoffset</code>. More tests are needed.</li> </ul> |

## 5 Implementation of `sclayer-fancyhdr`

This section if for developers only.

Note: The main problem of this interface is, that it tries to implement the user interface of package `fancyhdr` by Piet van Oostrum, that is completely different from `sclayer` and not really compatible with `sclayer`, using `sclayer`. This means, that `sclayer-fancyhdr` never can be a drop-in replacement of `fancyhdr`. Nevertheless it can help to let `sclayer` and `fancyhdr` coexist. To do so

- the lowest level of `fancyhdr` should not be the page style but a layer

- the page styles of fancyhdr should be layer page styles
- init code of the page styles of fancyhdr should use the layer page init code

Currently it is unsure whether it would be best to do a new implementation or to load original fancyhdr and to only modify some things. First I'll try the second method and therefore:

```
1 <*body>
2 \RequirePackage{fancyhdr}
3 </body>
```

We need at least one new layer for the new layer page style fancy. However, it could be useful to have not only one but four layers (even side head, odd side head, even side foot, odd side foot).

```
4 <*body>
5 \DeclareNewLayer[%
6   background,oddpage,
7   head,
8   contents={\hb@xt@ \layerwidth{%
9     \f@nch@head\f@nch@0olh\f@nch@olh\f@nch@och\f@nch@orh\f@nch@0orh}}
10 ]{fancy.head.odd}
11 \DeclareNewLayer[%
12   background,evenpage,
13   head,
14   contents={\hb@xt@ \layerwidth{%
15     \f@nch@head\f@nch@0elh\f@nch@elh\f@nch@ech\f@nch@erh\f@nch@0erh}}
16 ]{fancy.head.even}
17 \DeclareNewLayer[%
18   foreground,oddpage,
19   foot,
20   contents={\hb@xt@ \layerwidth{%
21     \f@nch@foot\f@nch@0olf\f@nch@olf\f@nch@ocf\f@nch@orf\f@nch@0orf}}
22 ]{fancy.foot.odd}
23 \DeclareNewLayer[%
24   foreground,evenpage,
25   foot,
26   contents={\hb@xt@ \layerwidth{%
27     \f@nch@foot\f@nch@0elf\f@nch@elf\f@nch@ecf\f@nch@erf\f@nch@0erf}}
28 ]{fancy.foot.even}
```

\ps@fancy And have to create a layer page style from this new layers, but we do not redefine page style fancy but the low level page style @fancy.

```
29 \DeclarePageStyleByLayers[
```

fancyhdr does some initialization at the very first call of \pagestylefancy. To do so fancyhdr first uses a different page style definition, that does the initialization and redefines the page style afterwards. This is still active with sclayer-fancyhdr. Additionally,

fancyhdr redefines `\@mkboth` at every selection of the internal page style `@fancy`. This can be adapted using the `onselect` feature of the new layer page style `@fancy`. In my opinion, the `\let\@mkboth\markboth` used by page style `headings` of, e.g., the standard classes or the KOMA-Script classes would be best here. However, fancyhdr uses the uncommon `\def\@mkboth{\protect\markboth}`, which would fail if a class or package tests `\@mkboth` using `\ifx\@mkboth\markboth`. However, KOMA-Script's `\IfActiveMkBoth` (see the `scrbase` chapter in the KOMA-Script manual) does also recognize the fancyhdr definition and copying this is more compatible with fancyhdr.

```

30 onselect={\def\@mkboth{\protect\markboth}},
31 ]{@fancy}{%
32 fancy.head.odd,fancy.head.even,fancy.foot.odd,fancy.foot.even
33 }
34 </body>

```

Note: Redefining page style `@fancy` instead of `fancy` does also mean, that features like options `automark` and `manualmark` resp. `\automark` and `\manualmark` are not fully supported by `sclayer-fancyhdr`. Also currently the font features of the KOMA-Script classes are not supported by `sclayer-fancyhdr`. However you are now able to combine other features of `sclayer` with features of `fancyhdr` and you can, e.g., use `sclayer-notecolumn` with `sclayer-fancyhdr`.

A future release of `sclayer-fancyhdr` may even provide the font features of the KOMA-Script classes and a working `markcase`. However, in this case I would have to redefine the initial page style `fancy` and the layers above.

## References

- [1] Piet van Oostrum: *Page layout in L<sup>A</sup>T<sub>E</sub>X*, release 3.9a 2017-06-30, <https://ctan.org/tex-archive/macros/latex/contrib/fancyhdr>, last visited: 2018-09-06.
- [2] Markus Kohm: *KOMA-Script*, release 3.25 2018-03-30, <https://ctan.org/tex-archive/macros/latex/contrib/koma-script>, last visited: 2018-09-06.

## Change History

v0.0	documentation . . . . .	1
General: some user	start of interface . . . . .	1

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

	<b>Symbols</b>		
<code>\@mkboth</code>	.....	<u>29</u>	
	<b>F</b>		
<code>\fancypagestyle</code>	.....	<i>4</i>	
	<b>P</b>		
<code>\ps@@empty</code>	.....	<i>3, 4</i>	
			<code>\ps@@fancy</code> .....
			<i>2, 4, <u>29</u></i>
			<code>\ps@empty</code> .....
			<i>3, 4</i>
			<code>\ps@fancy</code> .....
			<i>4</i>
			<code>\ps@fancyplain</code> .....
			<i>2, 4</i>
			<code>\ps@headings</code> .....
			<i>3, 4</i>
			<code>\ps@myheadings</code> .....
			<i>3, 4</i>
			<code>\ps@plain</code> .....
			<i>3, 4</i>
			<code>\ps@plain@fancy</code> .....
			<i>2, 4</i>