

The `sclayer` interface `sclayer-fancyhdr`*

Das `sclayer`-Interface `sclayer-fancyhdr`*

Markus Kohm

2022-07-08

Contents / Inhalt

I. English User Manual	2
1. The Purpose of this Package	2
2. How it works	2
3. How to use the Package	4
4. Hint	4
5. Known Issues	5
II. Benutzeranleitung auf Deutsch	5
6. Der Sinn dieses Pakets	5
7. Wie es funktioniert	6
8. Wie man das Paket verwendet	7
9. Tipp	8
10. Bekannte Probleme	8
III. Implementation of <code>sclayer-fancyhdr</code>	9
Change History	11

*This is version v0.2.2 of file `sclayer-fancyhdr.dtx`.
Dies ist Version v0.2.2 der Datei `sclayer-fancyhdr.dtx`.

Part I.

English User Manual

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` [3] 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 packages or classes of KOMA-Script [1] 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`.

2. How it works

To combine `fancyhdr` and `sclayer`, `sclayer-scrpage` loads both packages. After loading `fancyhdr` it redefines page style `@fancy` (up to `fancyhdr` version 3.10) resp. `f@nch@fancycore` (from `fancyhdr` version 4) 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.

`@fancy` (*internal*)
`f@nch@fancycore` (*internal*)
`fancyplain` (*style*)
`plain@fancy` (*internal*)

Up to `fancyhdr` version 3.10 page style `@fancy` was an internal page style, users should not select directly. From version 4 page style `f@nch@fancycore` is used internally by package `fancyhdr`. In both cases, the internal page style is used for the user page style `fancy`. The user page style `fancy` is also used for `fancyheadings`’s deprecated 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 an additional deprecated command `\fancyplain{<plain code>}{<fancy code>}`, too. The command uses the `<plain code>` if `\if@fancyplain` is `\iftrue` and `<fancy code>` if `\if@fancyplain` is `\iffalse`. It can be used inside the definition of the page style elements to distinguish settings of `fancy` and `plain` pages.

Page styles defined using `fancyhdr`'s command `\fancypagestyle` also always use page style `fancy` and so the internal basic page style `@fancy` (up to `fancyhdr` version 3.10) resp. `f@nch@fancycore` (from `fancyhdr` version 4).

As a result of the two notes above, every page style of `fancyhdr` always uses the same internal basic page style `@fancy` (up to `fancyhdr` version 3.10) resp. `f@nch@fancycore` (from `fancyhdr` version 4). As a result of redefining page style `@fancy` (up to `fancyhdr` version 3.10) resp. `f@nch@fancycore` (from `fancyhdr` version 4) 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` (up to `fancyhdr` version 3.10) resp. `f@nch@fancycore` (from `fancyhdr` version 4). 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 `scrlayer-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 `scrlayer` 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 a `scrlayer` user interface to a `fancyhdr` user interface is the decision whether or not automatic running heads are used. `scrlayer` 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 `scrlayer-fancyhdr` already using page style `@fancy` (up to `fancyhdr` version 3.10) resp. `f@nch@fancycore` (from `fancyhdr` version 4) 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.

Note: Currently, neither `scrlayer` nor `fancyhdr` nor `scrlayer-fancyhdr` do redefine page styles `headings` or `myheadings` by default. And neither `scrlayer` nor `fancyhdr` nor `scrlayer-fancyhdr` do redefine page style `plain` unless you are activating the deprecated `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 deprecated page style `fancyplain` or redefine page style `plain` either using `\fancypagestyle` as documented in the `fancyhdr` manual or using `\DeclarePageStyleByLayers`, documented in the KOMA-Script manual. However from version 4.0 `fancyhdr` provides options `myheadings` and `headings` to redefine the corresponding page style. `scrlayer-fancyhdr` also provides these options and passes them to `fancyhdr`.

Note: Pagestyle `empty` is somehow special. `scrlayer` redefines it to be a 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 `scrlayer`. Moreover, if `fancyhdr` is loaded before `scrlayer-fancyhdr`, `fancyhdr`'s internal page style `@empty` is not a copy of `scrlayer`'s layer page style `empty` but the original standard page style `empty`. However, with `scrlayer-fancyhdr` package `fancyhdr` does not longer use the internal page style `@empty`. So you would not need to know this.

`headings` (*style*)
`myheadings` (*style*)
`plain` (*style*)

`myheading` (*opt.*)
`heading` (*opt.*)

`empty` (*style*)
`@empty` (*internal*)

3. How to use the Package

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 `fancyhdr` and passes them to `sclayer` resp. `fancyhdr`. Nevertheless sometimes it may be useful to be able to additionally load `sclayer` or `fancyhdr`. In this case you should first load `sclayer` next `fancyhdr` and last `sclayer-fancyhdr`.

<code>@fancy</code> (<i>internal</i>)	After loading the package you should be able to use the page styles and commands of
<code>f@nch@fancycore</code> (<i>internal</i>)	<code>fancyhdr</code> and to add layers to or remove layers from the <code>fancyhdr</code> 's internal basic page style
<code>fancy</code> (<i>style</i>)	<code>@fancy</code> (up to <code>fancyhdr</code> version 3.10) resp. <code>f@nch@fancycore</code> (from <code>fancyhdr</code> version 4).
<code>fancyplain</code> (<i>style</i>)	Note, you are not able to add layers to or remove layers from <code>fancyhdr</code> 's page styles <code>fancy</code> ,
<code>plain@fancy</code> (<i>internal</i>)	<code>fancyplain</code> , <code>plain@fancy</code> or page styles defined by <code>\fancypagestyle</code> . However adding lay-
<code>\fancypagestyle</code>	ers to or removing layers from <code>@fancy</code> (up to <code>fancyhdr</code> version 3.10) resp. <code>f@nch@fancycore</code>
	(from <code>fancyhdr</code> version 4) will always change all these page styles!
<code>plain</code> (<i>style</i>)	Note: Loading <code>sclayer-fancyhdr</code> will not make page style <code>plain</code> nor <code>headings</code> nor
<code>headings</code> (<i>style</i>)	<code>myheadings</code> or any other page style but <code>empty</code> to automatically be a layer page style! But if
<code>myheadings</code> (<i>style</i>)	you are using option <code>headings</code> or <code>myheadings</code> , package <code>fancyhdr</code> redefines the corresponding
<code>empty</code> (<i>style</i>)	page styles to be <code>fancyhdr</code> page styles — and so relate to the same internal layer page style,
	either <code>@fancy</code> (up to <code>fancyhdr</code> version 3.10) or <code>f@nch@fancycore</code> (from <code>fancyhdr</code> version 4.0).

4. Hint

To become independent from the used version of `fancyhdr` you can use:

```
\IfPackageAtLeastTF{fancyhdr}{2019/03/21}{%  
  \DeclarePageStyleAlias{@fancy}{f@nch@fancycore}%  
}%  
\DeclarePageStyleAlias{f@nch@fancycore}{@fancy}%  
}
```

after loading `sclayer-fancyhdr` and before adding a layer to the internal layer page style. This code defines page style `@fancy` to be an alias of `f@nch@fancycore` if `fancyhdr` from version 4.0 is used or vice versa `f@nch@fancycore` as an alias of `@fancy` if `fancyhdr` before version 4.0 is used. So after this code using either

```
\AddLayersToPageStyle{@fancy}{...}
```

or

```
\AddLayersToPageStyle{f@nch@fancycore}{...}
```

would both result in adding the layers to the internal layer page style.

5. 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.

`fancy` (*style*)
`f@nch@fancycore` (*internal*)
`fancyplain` (*style*)
`\fancypagestyle`
`@fancy` (*internal*)

`@empty` (*internal*)

`@fancy` (*internal*)
`f@nch@fancycore` (*internal*)

- You are not able to add layers to the user level page style `fancy` or `fancyplain` or any page style defined by `\fancypagestyle` but only to the internal page style `@fancy` (if you are using `fancyhdr` up to version 3.10) resp. `f@nch@fancycore` (if you are using `fancyhdr` from version 4). This is intended as explained in this manual (see also section 4).
- Using `fancyhdr`'s internal page style `@empty` could have strange results. However, there is a simple solution for this: Don't use the internal `@empty` but always the user level page style `empty`!
- The vertical position of the page header does differ a little bit, if `scrlayer-fancyhdr` is used instead of `fancyhdr`. This could be fixed by a modification of the layers `fancy.head.odd` and `fancy.head.even` using `addvoffset`. More tests are needed.

See <https://github.com/komascript/scrlayer-fancyhdr/issues> for more issues.

Teil II. Benutzeranleitung auf Deutsch

6. Der Sinn dieses Pakets

Dieses Paket wurde entwickelt, um den Anwendern die Möglichkeit zu geben, die Fähigkeiten von Piet van Oostrums `fancyhdr` [3] mit den Fähigkeiten von `scrlayer` [2] zu kombinieren. In anderen Worten: Es wurde geschaffen, um die Seiten-Ebenen von `scrlayer` mit den Seiten-Stilen von `fancyhdr` zu kombinieren.

In dieser Kombination ist Kompatibilität mit `fancyhdr` das oberste Ziel! Verwendbarkeit und die Freiheiten, die von `scrlayer` geboten werden, ist das zweite Ziel. Kompatibilität mit anderen Paketen oder Klassen von KOMA-Script [1] ist kein primäres Ziel. Vielleicht wird dies in Zukunft eine optionale Fähigkeit werden. Die Aufhebung irgendwelcher realen oder eingebildeter Beschränkungen von `fancyhdr`, die darüber hinaus gehen, die Verwendung von Ebenen zu ermöglichen, ist kein Ziel und wird auch in Zukunft kein Ziel werden.

Wenn Sie eine bessere Kombination von Seitenstilen und Ebenen benötigen, so sollten Sie entweder die Low-Level-Schnittstelle von `scrlayer` zur Definition von Seitenstilen oder — und das ist die Empfehlung des Autors — das Paket `scrlayer-scrpage` anstelle von `scrlayer-fancyhdr` oder `fancyhdr` verwenden. Wenn Sie bessere Kompatibilität mit anderen Teilen von KOMA-Script, beispielsweise mit den KOMA-Script-Klassen. benötigen, sollten Sie `scrlayer-scrpage` anstelle von `scrlayer-fancyhdr` oder `fancyhdr` verwenden.

7. Wie es funktioniert

Zur Kombination von `fancyhdr` mit `sclayer` lädt `sclayer-scrpage` beide Pakete. Nach dem Laden von `fancyhdr` definiert es den Seitenstil `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4.0) in einen Ebenen-Seitenstil von `sclayer` um, bestehend aus den neu definierten Ebenen `fancy.head.even`, `fancy.head.odd`, `fancy.foot.even` und `fancy.foot.odd`. Die `...head...`-Ebenen sind Hintergrund-Ebenen entsprechend dem Seitenkopf von `fancyhdr`-Seitenstilen (oder anderen gebräuchlichen Seitenstilen). Die `...foot...`-Ebenen sind Vordergrund-Ebenen entsprechend dem Seitenfuß von `fancyhdr`-Seitenstilen (oder anderen gebräuchlichen Seitenstilen). Die `...even`-Ebenen sind auf gerade Seiten, also linke Seiten von doppelseitigen Dokumenten, beschränkt. Die `...odd`-Ebenen sind auf ungerade Seiten, also rechte Seiten bei doppelseitigen Dokumenten oder alle Seiten bei einseitigen Dokumenten, beschränkt.

`@fancy` (*internal*)
`f@nch@fancycore` (*internal*)
`fancyplain` (*style*)
`plain@fancy` (*internal*)

Bis Version 3.10 von `fancyhdr` war Seitenstil `@fancy` ein interner Seitenstil, den Anwender selbst nicht direkt verwenden sollten. Ab Version 4.0 wird `f@nch@fancycore` als interner Seitenstil von Paket `fancyhdr` genutzt. In beiden Fällen wird der interne Seitenstil für den Benutzerseitenstil `fancy` verwendet. Der Benutzerseitenstil `fancy` wird ebenfalls für den veralteten Seitenstil `fancyplain` des Pakets `fancyheadings` verwendet, der zusätzlich den Seitenstil `plain` in `fancyhdrs` internen Seitenstil `plain@fancy` ändert, der ebenfalls `fancy` verwendet, wobei allerdings `\if@fancyplain` auf `\iftrue` gesetzt ist. Es gibt eine weitere überholte Anweisung `\fancyplain{<plain-Code>}{<fancy-Code>}`. Diese Anweisung verwendet `<plain-Code>`, wenn `\if@fancyplain` `\iftrue` ist, aber `<fancy-Code>`, wenn `\if@fancyplain` `\iffalse` ist. Sie kann innerhalb der Definition der Seitenstilelemente verwendet werden um zwischen den Einstellung für `fancy` und `plain` zu unterscheiden.

Seitenstile, die mit der `fancyhdr`-Anweisung `\fancypagestyle` definiert wurden, verwenden ebenfalls immer den Seitenstil `fancy` und damit den internen grundlegenden Seitenstil `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4).

Als Ergebnis der beiden obigen Anmerkungen, verwenden alle Seitenstile von `fancyhdr` immer denselben internen grundlegenden Seitenstil `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4). Als Ergebnis daraus resultiert aus der Umdefinierung des Seitenstils `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4) in einen Ebenen-Seitenstil, dass Anwender Ebenen zu allen Seitenstilen von `fancyhdr` hinzufügen oder daraus löschen können, indem sie Ebenen zu `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4) hinzufügen oder daraus löschen. Sie können keine Ebenen unmittelbar zu einzelnen Seitenstilen von `fancyhdr` wie `fancy`, `fancyplain`, `plain@fancy` oder anderen mit `\fancypagestyle` definierten Seitenstilen hinzufügen oder daraus löschen. Aber natürlich können Sie das zweite Argument von `\fancypagestyle` nutzen, um Ebenen zum Zeitpunkt der Aktivierung eines `fancyhdr`-Seitenstils hinzuzufügen oder zu entfernen. Dies stellt eine Verschiebung dar weg von der `sclayer`-Schnittstelle zum Hinzufügen oder Entfernen von Ebenen hin zur `fancyhdr`-Schnittstelle zur Definition von Änderungen des Seitenstils `fancy`.

Ein weitere solche Verschiebung von der Anwenderschnittstelle von `sclayer` zur Anwenderschnittstelle von `fancyhdr` ist die Entscheidung, ob automatische, lebende Kolumnentitel verwendet werden sollen oder nicht. `sclayer` bietet dafür die Optionen `automark` und `manualmark` und die Anweisungen `\automark` und `\manualmark`. Diese konfigurieren gleichzeitig Anweisungen wie `\partmark`, `\chaptermark`, `\sectionmark` etc. Mit `sclayer-fancyhdr` wird bereits durch Verwendung von Seitenstil `@fancy` (bis einschließlich `fancyhdr` Version

3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4) auf automatische, lebende Kolumnentitel umgeschaltet. Die erste Aktivierung von `fancy` definiert auch weiterhin `\chaptermark` und `\sectionmark`, wenn eine Klasse mit `\chapter` verwendet wird, oder `\sectionmark` und `\subsectionmark`, wenn eine Klasse ohne `\chapter` verwendet wird. Dennoch können Sie weiterhin `\manualmark` und `\automark` nach der Umschaltung zu einem `fancyhdr`-Seitenstil verwenden, um die lebenden Kolumnentitel zu konfigurieren. Diese Verschiebung ist also nur partiell.

`headings (style)`
`myheadings (style)`
`plain (style)`

Hinweis: Derzeit definieren weder `sclayer` noch `fancyhdr` noch `sclayer-fancyhdr` die Seitenstile `headings` oder `myheadings` in der Voreinstellung um. Und weder `sclayer` noch `fancyhdr` oder `sclayer-fancyhdr` definieren den Seitenstil `plain` um, solange nicht der veraltete `fancyhdr`-Seitenstil `fancyplain` aktiviert wird. Wenn Sie also einen Ebeneseitenstil für `plain`-Seiten wünschen, insbesondere für die erste Seite eines Kapitels oder Teils oder die Seiten mit dem Titelkopf, so müssen Sie entweder den veralteten `fancyhdr`-Seitenstil `fancyplain` verwenden oder den Seitenstil `plain` entweder per `\fancypagestyle` umdefinieren, wie dies in der `fancyhdr`-Anleitung beschrieben ist, oder per `\DeclarePageStyleByLayers`, wie dies in der KOMA-Script-Anleitung dokumentiert ist. Darüber hinaus bietet `fancyhdr` ab Version 4.0 die Optionen `myheadings` und `headings`, um die entsprechenden Seitenstile umzudefinieren. `sclayer-fancyhdr` bietet diese Optionen ebenfalls und leitet sie an `fancyhdr` weiter.

`myheading (opt.)`
`headings (opt.)`

`empty (style)`
`@empty (internal)`

Hinweis: Der Seitenstil `empty` ist etwas speziell. `sclayer` definiert dieses in einen Ebenenseitenstil um. Und `fancyhdrs` interner Seitenstil `@empty` ist ebenfalls identisch mit `empty`. Daher verwendet auch `@empty` die Ebenen von `empty`, aber Sie sollten nicht versuchen, diese direkt mit der Schnittstelle von `sclayer` zu verändern. Mehr noch, wenn `fancyhdr` vor `sclayer-fancyhdr` geladen wird, ist der interne Seitenstil `@empty` von `fancyhdr` nicht länger eine Kopie des Ebenen-Seitenstils `empty` von `sclayer`, sondern weiterhin des Standard-Seitenstils `empty`. Allerdings verwendet `fancyhdr` mit `sclayer-fancyhdr` selbst den internen Seitenstil `@empty` gar nicht mehr. Daher brauchen Sie das eigentlich auch nicht zu wissen.

8. Wie man das Paket verwendet

Um das Paket zu verwenden, muss man es laden, beispielsweise per:

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

an Stelle von `sclayer` oder `fancyhdr` oder vor dem Laden eines dieser Pakete. Empfohlen wird das komplette Ersetzen von `sclayer` und `fancyhdr` durch das Laden von `sclayer-fancyhdr`, weil dadurch Optionskonflikte (engl. *option clash*) vermieden wird. `sclayer-fancyhdr` stellt alle Optionen von `sclayer` und `fancyhdr` bereit und leitet diese an `sclayer` bzw. `fancyhdr` weiter. Nichtsdestotrotz kann es in seltenen Fällen nützlich sein, dass man das Paket auch zusätzlich nach `sclayer` und `fancyhdr` laden kann. In diesem Fall sollte `sclayer` vor `fancyhdr` und zuletzt `sclayer-fancyhdr` geladen werden.

`@fancy (internal)`
`f@nch@fancycore (internal)`
`fancy (style)`
`fancyplain (style)`
`plain@fancy (internal)`
`\fancypagestyle`

Nach dem Laden des Pakets sollten Sie in der Lage sein, die Seitenstile und Befehle von `fancyhdr` zu verwenden und Ebenen zum internen `fancyhdr`-Seitenstil `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4) hinzuzufügen oder davon zu entfernen. Hinweis: Es ist Ihnen nicht möglich, Ebenen zu den `fancyhdr`-Seitenstilen `fancy`, `fancyplain`, `plain@fancy` oder jedem anderen mit `\fancypagestyle` definierten Seitenstil hinzuzufügen. Jedoch wird das Hinzufügen oder Entfernen von Ebenen zu bzw. von `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4) immer alle diese Seitenstile mit ändern!

`plain` (*style*) Hinweis: Das Laden von Paket `sclayer-fancyhdr` wird weder den Seitenstil `plain` noch
`headings` (*style*) `headings` noch `myheadings` noch irgend einen anderen Seitenstil außer `empty` automatisch
`myheadings` (*style*) in einen Ebenen-Seitenstil umwandeln! Wenn Sie aber Option `headings` oder `myheadings`
`empty` (*style*) angeben, definiert `fancyhdr` die entsprechenden Seitenstile in `fancyhdr`-Seitenstil um – somit
hängen diese dann von demselben internen Seitenstil, entweder `@fancy` (bis einschließlich
`fancyhdr` Version 3.10) oder `f@nch@fancycore` (ab `fancyhdr` Version 4), ab.

9. Tipp

Um unabhängig von der verwendeten Version von `fancyhdr` zu werden, kann man:

```

\IfPackageAtLeastTF{fancyhdr}{2019/03/21}{%
  \DeclarePageStyleAlias{@fancy}{f@nch@fancycore}%
}%
\DeclarePageStyleAlias{f@nch@fancycore}{@fancy}%
}

```

nach dem Laden von `sclayer-fancyhdr` verwenden, bevor man irgendwelche Ebenen zu dem internen Seitenstil hinzufügt oder davon entfernt. Diese Codezeilen definieren bei Verwendung von `fancyhdr` aber Version 4.0 Seitenstil `@fancy` als Alias für Seitenstil `f@nch@fancycore` und umgekehrt für `fancyhdr` bis einschließlich Version 3.10 `f@ch@fancycore` als Alias für Seitenstil `@fancy`. Damit führt dann sowohl:

```
\AddLayersToPageStyle{@fancy}{...}
```

als auch

```
\AddLayersToPageStyle{f@nch@fancycore}{...}
```

dazu, dass die Ebenen zum internen Ebenen-Seitenstil hinzugefügt werden.

10. Bekannte Probleme

Bitte beachten Sie, dass die folgenden Hinweise entweder Notizen für den Paketautor oder Hinweise für die Anwender darstellen und verhindern sollen, dass dieselben Probleme immer wieder gemeldet werden. Die Auflistung sagt nichts darüber aus, ob es sich dabei um Fehler oder Eigenschaften handelt:

- `fancy` (*style*)
`f@nch@fancycore` (*internal*)
`fancyplain` (*style*)
`\fancypagestyle`
`@fancy` (*internal*)

`@empty` (*internal*)

 - Sie können keine Ebenen zu den Seitenstilen auf Benutzerebene `fancy` oder `fancyplain` oder irgendwelchen anderen Seitenstilen, die mit `\fancypagestyle` definiert wurden, hinzufügen. Dies ist nur für den internen Seitenstil `@fancy` (bis einschließlich `fancyhdr` Version 3.10) bzw. `f@nch@fancycore` (ab `fancyhdr` Version 4) möglich (see auch Abschnitt 9).
 - Die Verwendung des internen `fancyhdr`-Seitenstils `@empty` kann zu unerwarteten Ergebnissen führen. Es gibt jedoch eine einfache Lösung für dieses Problem: Verwenden Sie nicht den internen Seitenstile `@empty`, sondern immer den Seitenstile `empty` der Benutzerebene.

`@fancy` (*internal*)
`f@nch@fancycore` (*internal*)

- Die vertikale Position des Seitenkopfes weicht ein wenig ab, wenn `sclayer-fancyhdr` an Stelle von `fancyhdr` verwendet wird. Dies kann durch leichte Modifikation der Ebenen `fancy.head.odd` und `fancy.head.even` mit Hilfe von `addvoffset` behoben werden. Weitere Tests sind hier erforderlich.

Siehe <https://github.com/komascript/sclayer-fancyhdr/issues> für weitere bekannte Probleme.

Part III.

Implementation of `sclayer-fancyhdr`

This section is for developers only.

We need `sclayer` at least version 2021/02/15,

```
1 <*init>
2 \RequirePackage{sclayer}[2021/02/15]
3 </init>
```

Because before `\sclayer@do@inherited@options` would not be defined. And we need it, to inherit all options of `sclayer`.

```
4 <*options>
5 \sclayer@do@inherited@options{\sclayer@inherit@option}
6 </options>
```

And the very first thing at the body, after definition of the options is to process the options.

```
7 <*body>
8 \KOMAProcessOptions
9 </body>
```

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. Currently we explicitly provide the options of `fancyhdr`.

```
10 <*options>
11 \DeclareOption{nocheck}{\PassOptionsToPackage{nocheck}{fancyhdr}}
```

```

12 \DeclareOption{compatV3}{\PassOptionsToPackage{compatV3}{fancyhdr}}
13 \DeclareOption{myheadings}{\PassOptionsToPackage{myheadings}{fancyhdr}}
14 \DeclareOption{headings}{\PassOptionsToPackage{headings}{fancyhdr}}
15 \}
```

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).

```

19 (*body)
20 \DeclareNewLayer[%
21   background,oddpaper,
22   head,
23   contents={\hb@xt@ \layerwidth{%
24     \f@nch@head\f@nch@Oolh\f@nch@olh\f@nch@och\f@nch@orh\f@nch@Oorh}}
25 ]{fancy.head.odd}
26 \DeclareNewLayer[%
27   background,evenpage,
28   head,
29   contents={\hb@xt@ \layerwidth{%
30     \f@nch@head\f@nch@Oelh\f@nch@elh\f@nch@ech\f@nch@erh\f@nch@Oerh}}
31 ]{fancy.head.even}
32 \DeclareNewLayer[%
33   foreground,oddpaper,
34   foot,
35   contents={\hb@xt@ \layerwidth{%
36     \f@nch@foot\f@nch@Oolf\f@nch@olf\f@nch@ocf\f@nch@orf\f@nch@Oorf}}
37 ]{fancy.foot.odd}
38 \DeclareNewLayer[%
39   foreground,evenpage,
40   foot,
41   contents={\hb@xt@ \layerwidth{%
42     \f@nch@foot\f@nch@Oelf\f@nch@elf\f@nch@ecf\f@nch@erf\f@nch@Oerf}}
43 ]{fancy.foot.even}
```

`f@nch@fancycore` (*internal*) And have to create a layer page style from this new layers, but we do not redefine `@fancy` (*internal*) page style `fancy` but the low level page style `@fancy` (up to `fancyhdr` version 3.10) resp. `\mkboth` `f@nch@fancycore` (from `fancyhdr` version 4.0).

`fancyhdr` does some initialization at the very first call of `\pagestyle{fancy}`. 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` resp. `f@nch@fancycore`. This can be adapted using the `onselect` feature of the new layer page style `@fancy` resp. `f@nch@fancycore`. 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`.

```

44 \ifpackageatleast{fancyhdr}{2019/03/21}{%
45   \DeclarePageStyleByLayers[
46     onselect={\def\mkboth{\protect\markboth}},
47   ]{f@nch@fancycore}{%
48     fancy.head.odd,fancy.head.even,fancy.foot.odd,fancy.foot.even
49   }%
50 }{%
51   \DeclarePageStyleByLayers[
52     onselect={\def\mkboth{\protect\markboth}},
53   ]{@fancy}{%
54     fancy.head.odd,fancy.head.even,fancy.foot.odd,fancy.foot.even
55   }%
56 }
57 \end{body}

```

Note: Redefining page style `@fancy` resp. `f@nch@fancycore` 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 / Literatur

- [1] Markus Kohm. *koma-script* — *A bundle of versatile classes and packages*. July 4, 2022. URL: <https://www.ctan.org/pkg/koma-script> (visited on 07/08/2022).
- [2] Markus Kohm. *sclayer* — *Manage text ‘layers’ within KOMA-Script*. July 4, 2022. URL: <https://www.ctan.org/pkg/sclayer> (visited on 07/08/2022).
- [3] Piet van Oostrum. *fancyhdr* — *Extensive control of page headers and footers in L^AT_EX 2_ε*. May 18, 2022. URL: <https://www.ctan.org/pkg/fancyhdr> (visited on 07/08/2022).

Change History

0.1.3558 – 2021/02/15	KOMA-Script	2
<code>f@nch@fancycore</code> : support for <code>fancyhdr</code> 4 10	v0.2.1 – 2021/03/30	
v0.0 – 2018/09/01	General: German user manual	2
General: start of interface	improved user manual	2
v0.0 – 2018/09/05	v0.2.2 – 2022/07/08	
General: some user documentation	General: using class	
v0.2 – 2021/02/18	<code>koma-script-source-doc</code>	2
General: package is no longer part of		

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.

Symbols		H		
<code>@empty</code> (<i>internal</i>)	<i>3, 5, 7, 8</i>	<code>heading</code> (<i>opt.</i>)	<i>3</i>	<code>@fancy</code>
<code>@fancy</code> (<i>internal</i>)	<i>2,</i>	<code>headings</code> (<i>opt.</i>)	<i>7</i>	<i>4, 5, 5, 6, 7, 8, 9, <u>44</u></i>
	<i>4, 5, 5, 6, 7, 8, 9, <u>44</u></i>	<code>headings</code> (<i>pgstyle</i>)	<i>3, 4, 7, 8</i>	<code>f@nch@fancycore</code>
C		M		<i>4, 5, 5, 6, 7, 8, 9, <u>44</u></i>
Commands:		<code>myheading</code> (<i>opt.</i>)	<i>3, 7</i>	<code>plain@fancy</code>
<code>\fancypagestyle</code>	<i>4, 5, 7, 8</i>	<code>myheadings</code> (<i>pgstyle</i>)	<i>3, 4, 7, 8</i>	page styles:
				<code>empty</code>
E		O		<i>3, 4, 7, 8</i>
<code>empty</code> (<i>pgstyle</i>)	<i>3, 4, 7, 8</i>	Options:		<code>fancy</code>
F		<code>heading</code>	<i>3</i>	<i>4, 5, 7, 8</i>
<code>f@nch@fancycore</code> (<i>inter-</i>		<code>headings</code>	<i>7</i>	<code>fancyplain</code>
<i>nal</i>)	<i>2,</i>	<code>myheading</code>	<i>3, 7</i>	<i>2, 4, 5, 6, 7, 8</i>
	<i>4, 5, 5, 6, 7, 8, 9, <u>44</u></i>	P		<code>headings</code>
<code>fancy</code> (<i>pgstyle</i>)	<i>4, 5, 7, 8</i>	Page styles (internal):		<i>3, 4, 7, 8</i>
<code>\fancypagestyle</code>	<i>4, 5, 7, 8</i>	<code>@empty</code>	<i>3, 5, 7, 8</i>	<code>plain</code>
<code>fancyplain</code> (<i>pgstyle</i>)	<i>2, 4, 5, 6, 7, 8</i>	T		<i>3, 4, 7, 8</i>
		TeX macros (internal):		<code>plain</code> (<i>pgstyle</i>)
		<code>\@mkboth</code>		<i>2, 4, 6, 7</i>
		<code>\@mkboth</code>		<i>2, 4, 6, 7</i>