

The Obsolete Package `scrpage2` *

Markus Kohm

2020/02/27

This is the former KOMA-Script package `scrpage2`. The package is deprecated and should not be used any longer. You can very easily replace it by the KOMA-Script package `scrlayer-scrpage` that is documented in the English and the German KOMA-Script guide.

Contents

English	1
1 Adapting Page Headers and Footers with <code>scrpage2</code>	2
1.1 Basic Functionality	2
1.1.1 Predefined Page Styles	3
1.1.2 Manual and Running Headings	6
1.1.3 Formatting of Header and Footer	7
1.1.4 Package Options	12
1.2 Defining Own Page Styles	15
1.2.1 The Interface for Beginners	15
1.2.2 The Interface for Experts	17
1.2.3 Managing Page Styles	21
 Deutsch	 22
2 Kopf- und Fußzeilen mit <code>scrpage2</code>	23
2.1 Grundlegende Funktionen	24
2.1.1 Vordefinierte Seitenstile	24
2.1.2 Manuelle und automatische Kolumnentitel	27
2.1.3 Formatierung der Kopf- und Fußzeilen	29
2.1.4 Optionen beim Laden des Paketes	34

*This file has revision number v3.30, last revised 2020/02/27.

2.2	Seitenstile selbst gestalten	37
2.2.1	Die Anwenderschnittstelle	37
2.2.2	Die Expertenschnittstelle	39
2.2.3	Seitenstile verwalten	44
3	Implementation	44
3.1	Options	44
3.2	Some Initialisation	48
3.3	Predefinitions	48
3.4	Expert-Pagestyle-Definition-Interface	48
3.5	User-Pagestyle-Definition-Interface	53
3.6	Automark or not	55
3.7	Width of head and foot	58
3.8	Fonts	61
3.9	Length and thickness of lines at head and foot	61
3.10	Extended multilanguage formats	62
3.11	Definition of default pagestyles	62
3.12	Configurationfile	71
3.13	Some checks	71
3.14	End	71

1 Adapting Page Headers and Footers with `scrpage2`

From KOMA-Script 3.12 the completely newly implemented package `scrlayer-scrpage` replaces the old `scrpage2`. The new package is a consequently developed extension of the design of `scrpage2`. In difference to `scrpage2` it provides the extended option interface of the KOMA-Script classes. Because `scrlayer-scrpage` predominates `scrpage2` it is recommended to not longer use `scrpage2` but `scrlayer-scrpage`. Because of this the current version of `scrpage2` is the final one. All development resources will go into `scrlayer-scrpage`. For more information about `scrlayer-scrpage` see the KOMA-Script guide.

In place of `scrpage2` or `scrlayer-scrpage` you can of course make use of `fancyhdr`. However, `scrpage2` and especially `scrlayer-scrpage` integrated markedly better with the KOMA-Script bundle. For this reason, and because at the time the forerunner to `fancyhdr` was missing many features, `scrpage2` was developed. Naturally, `scrpage2` and `scrlayer-scrpage` are not limited to use only with the KOMA-Script classes, but can just as easily be used with other document classes.

1.1 Basic Functionality

To understand the following description, an overview of L^AT_EX's fairly involved header and footer mechanism is needed. The L^AT_EX kernel defines

the page styles `empty`, which produces a completely empty header and footer, and `plain`, which produces usually only a page number in the footer and an empty header. Apart from these, many document classes provide the style `headings`, which allows more complex style settings and running headings. The `headings` style often has a related variant, `myheadings`, which is similar except for switching off the running headings and reverting them to manual control by the user. A more detailed description is given in the page style section of the KOMA-Script manual where it is also noted that some L^AT_EX commands automatically switch to another page style—usually page style `plain`—for the current page.

Package `scrpage2` does not distinguish between page styles with automatic, running headings and page styles with manual headings. The way to deal with automatic and manual headings is independent from the page style and so the page style is independent from the choice of automatic or manual headings. More information about this in section 1.1.2.

1.1.1 Predefined Page Styles

One of the basic features of `scrpage2` is a set of predefined, configurable page styles.

```
\pagestyle{scrheadings}
\pagestyle{scrplain}
```

Package `scrpage2` delivers its own page style, named `scrheadings`, which can be activated with the `\pagestyle{scrheadings}`. When this page style is in use, an appropriate `scrplain` page style is used for the plain page style. In this case *appropriate* means that this new plain page style is also configureable by the commands introduced in section 1.1.3, which, for example, configure the header and footer width and complies within the basic layout. Neither the activation of `scrheadings` nor the attendant change to the appropriate plain page style, `scrplain`, influences the mode of manual or automatic headings (see section 1.1.2). The `scrplain` page style can also be activated directly with `\pagestyle`.

<pre>\lehead \cehead \rehead \lefoot \cefoot \refoot \lohead \cohead \rohead \lofoot \cofoot \rofoot \ihead \chead \ohead \ifoot \cfoot \ofoot</pre>	<p>The page style of <code>scrpage2</code> are defined to have flexible configurable header and footer. To achieve this, the page styles include three boxes in both the header and the footer. The contents of these boxes may be modified easily. The commands modifying the content of these boxes can be seen in figure 1. Commands in the middle column modify the box contents on both odd and even pages. All of the commands have an optional and a mandatory argument. The option argument influences the content of corresponding box of the plain page style, <code>scrplain</code>. The mandatory argument influences the content of the corresponding box of the page style <code>scrheadings</code>.</p>
--	--

Example: If one wants the page number within `scrheadings` be placed in the middle of the footer, then following can be used:

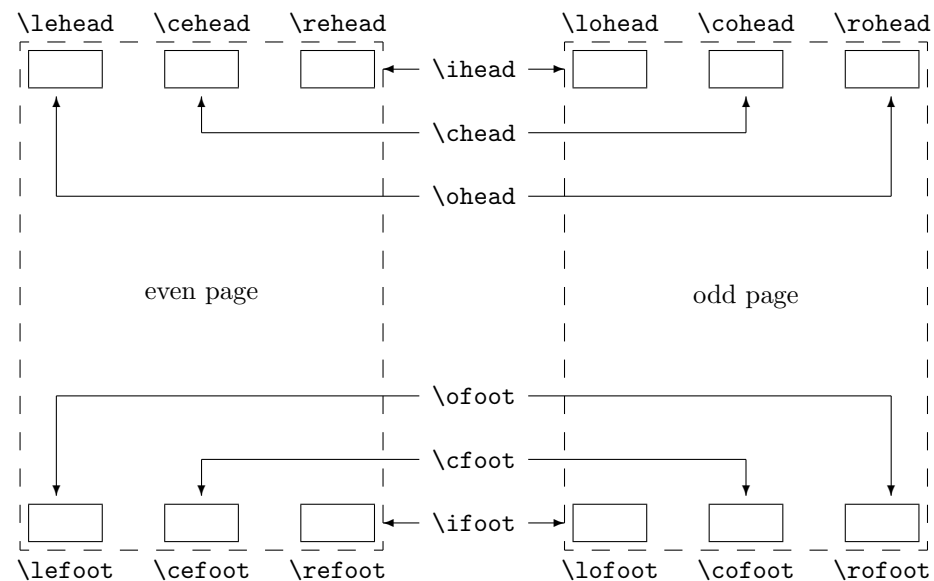


Figure 1: Commands for modification of page styles `scrheadings` and `scrplain` and their relationship to header and footer elements

```
\cfoot{\pagemark}
```

The next example shows how to place both running heading and page number in the header; the running heading inside and the page number outside:

```
\ohead{\pagemark}
\ihead{\headmark}
\cfoot{}
```

The command `\cfoot{}` is only required in order to empty the item in the middle of the footer, which normally contains the page number.

The commands which are associated with only one item can be used for more advanced settings.

Example: Assuming one has the order to write an annual report of a company, one could use commands like this:

```
\ohead{\pagemark}
\rehead{Annual Report 2001}
\lohead{\headmark}
\cefoot{TheCompanyName Inc.}
\cofoot{Department: Development}
```

In order to keep the data in the footer synchronized with the content of the document, the footer has to be updated using `\cofoot` when a new department is discussed in the report.

As mentioned above, there is a new plain page style which corresponds to `scrheadings`. Since it should also be possible to customize this style, the commands support an optional argument with which the contents of the appropriate fields of this plain page style can be modified.

Example: The position of the page number for the page style `scrheadings` can be declared as follows:

```
\cfoot[\pagemark]{}
\ohead[]{\pagemark}
```

When the command `\chapter`, after it has started a new page, now switches to the page style `plain`, then the page number is centered in the footer.

<code>\clearscrheadings</code> <code>\clearscrplain</code> <code>\clearscrheadfoot</code>	<p>If one wants to redefine both the page style <code>scrheadings</code> and the corresponding plain page style, frequently one must empty some already occupied page elements. Since one rarely fills all items with new content, in most cases several instructions with empty parameters are necessary. With the help of these three instructions the quick and thorough deletion is possible. While <code>\clearscrheadings</code> only deletes all fields of the page style <code>scrheadings</code>, and <code>\clearscrplain</code> deletes all fields of the corresponding plain page style, <code>\clearscrheadfoot</code> sets all fields of both page styles to empty.</p>
---	---

Example: If one wants to reset the page style to the default KOMA-Script settings, independent of the actual configuration, only these three commands are sufficient:

```
\clearscrheadfoot
\ohead{\headmark}
\ofoot[\pagemark]{\pagemark}
```

Without the commands `\clearscrheadfoot`, `\clearscrheadings` and `\clearscrplain`, six commands with additional nine empty arguments would be required:

```
\ihead[]{}
\chead[]{}
\ohead[]{\headmark}
\ifoot[]{}
\cfoot[]{}
\ofoot[\pagemark]{\pagemark}
```

Of course, for a specific configuration, some of them could be dropped.

In the previous examples two commands were used which have not been introduced yet. The description of these commands follows.

`\leftmark` `\rightmark` These two instructions make it possible to access the running headings, which are normally meant for the left or for the right page. These two instructions are not made available by `scrpage2`, but directly by the L^AT_EX kernel. When in this section running headings of the left page or the right page are mentioned, this refers to the contents of `\leftmark` or `\rightmark`, respectively.

`\headmark` This command gives access to the content of running headings. In contrast to `\leftmark` and `\rightmark`, one need not regard the proper assignment to left or right page.

`\pagemark` This command returns the formatted page number. The formatting can be controlled by `\pnumfont`, introduced in section 1.1.3, which `\pagemark` heeds automatically.

`\pagestyle{useheadings}` The package `scrpage2` is meant primarily for use of the supplied styles or for defining one's own styles. However, it may be necessary to shift back also to a style provided by the document class. It might appear that this should be done with `\pagestyle{headings}`, but this has the disadvantage that commands `\automark` and `\manualmark`, to be discussed shortly, do not function as expected. For this reason one should shift back to the original styles using `\pagestyle{useheadings}`, which chooses the correct page styles automatically for both manual and automatic running headings.

1.1.2 Manual and Running Headings

Usually there is a *my*-version of the `headings` page style. If such a page style is active, then the running headings are no longer updated no longer automatically and become manual headings. With `scrpage2` a different path is taken. Whether the headings are running or manual is determined by the instructions `\automark` and `\manualmark`, respectively. The default can be set already while loading of the package, with the options `automark` and `manualmark` (see section 1.1.4).

`\manualmark` As the name suggests, `\manualmark` switches off the updating of the running headings and makes them manual. It is left to the user to update and provide contents for the headings. For that purpose the instructions `\markboth` and `\markright` are available.

`\automark` The macro `\automark[<right page>]{<left page>}` activates the automatic

updating, that is, running headings. For the two parameters the designations of the document sectioning level whose title is to appear in appropriate place are to be used. Valid values for the parameters are: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, and `subparagraph`. For most of the classes use of `part` will not produce the expected result. So far only KOMA-Script classes from version 2.9s up are known to support this value. The optional argument *⟨right page⟩* is understandably meant only for double-sided documents. In the single-sided case one should normally not use it. With the help of the option `autooneside` one can also set that the optional argument in single-sided mode is ignored automatically (see section 1.1.4).

Example: Assuming that the document uses a *book* class, whose topmost section level is *chapter*, then after a preceding `\manualmark`

```
\automark[section]{chapter}
```

restores the original behaviour. If one prefers lower section levels in running headings, the following can be used:

```
\automark[subsection]{section}
```

For the upper section level, the data of the headings is set by the command `\markboth`, while that for the lower section level by `\markright` or `\markleft`. These commands are called indirectly by the sectioning commands. The macro `\markleft` is provided by the package `scrpage2` and is defined similarly to `\markright` in the L^AT_EX kernel. Although `\markleft` is not defined as an internal command, the direct use is not recommended.

1.1.3 Formatting of Header and Footer

The previous section concerned itself mainly with the contents of the header and footer. This is of course not sufficient to satisfy formative ambitions. Therefore we devote this section exclusively to this topic.

<code>\headfont</code> <code>\footfont</code> <code>\pnumfont</code>	<p>The command <code>\headfont</code> contains the commands which determine the font of header and footer lines. Command <code>\footfont</code> contains the difference of the footer to that. The difference for the style of the page number is defined by the command <code>\pnumfont</code>.</p>
--	--

Example: If, for example, one wants the header to be typeset in bold sans serif, the footer in non-bold sans serif, and the page number in a slanted serif style, then one can use the following definitions:

```
\renewcommand{\headfont}{\normalfont\sffamily\bfseries}
\renewcommand*{\footfont}{\normalfont\sffamily}
\renewcommand{\pnumfont}{\normalfont\rmfamily\slshape}
```

`\setkomafont{pagehead}`
`\setkomafont{pagefoot}`
`\setkomafont{pagenumber}`

From version 2.8p of the KOMA-Script classes a new unified user interface scheme is implemented for font attributes. If `scrpage2` is used together with one of these classes, then it is recommended to set up font attributes in the manner described in the KOMA-Script manual.

Instead of `\renewcommand` the command `\setkomafont` should be used to configure the font attributes. The previous definitions can then be written as:

```
\setkomafont{pagehead}\normalfont\sffamily\bfseries}
\setkomafont{pagefoot}{\normalfont\sffamily}
\setkomafont{pagenumber}{\normalfont\rmfamily\slshape}
```

`\setheadwidth`
`\setfootwidth`

Normally the widths of header and footer lines correspond to the width of the text body. The commands `\setheadwidth[⟨shift⟩]{⟨width⟩}` and `\setfootwidth[⟨shift⟩]{⟨width⟩}` enable the user to adapt in a simple manner the widths to his needs. The mandatory argument `⟨width⟩` takes the value of the desired width of the page header or footer, while `⟨shift⟩` is a length parameter by which amount the appropriate item is shifted toward the outside page edge.

For the most common situations the mandatory argument `⟨width⟩` accepts the following symbolic values:

<code>paper</code>	– the width of the paper
<code>page</code>	– the width of the page
<code>text</code>	– the width of the text body
<code>textwithmarginpar</code>	– the width of the text body including margin
<code>head</code>	– the current header width
<code>foot</code>	– the current footer width

The difference between `paper` and `page` is that `page` means the width of the paper less the binding correction if the package `typearea` is used (see the chapter about `typearea` in the KOMA-Script manual). Without `typearea` both values are identical.

Example: Assume that one wants a layout like that of *The L^AT_EX Companion*, where the header projects into the margin. This can be obtained with:

```
\setheadwidth[0pt]{textwithmarginpar}
```

which appears like this on an odd page:

KOMA-Script	3
This fill text is currently seized by 130 million receptors in your retina. Thereby the nerve cells Retina are put in a state of stimulation, which spreads into the rear part of your brain originating from	

If the footer line should have the same width and alignment, then two ways to set this up are possible. The first way simply repeats the settings for the case of the footer line:

```
\setfootwidth[0pt]{textwithmarginpar}
```

In the second way the symbolic value `head` is used, since the header already has the desired settings.

```
\setfootwidth[0pt]{head}
```

If no *⟨shift⟩* is indicated, i. e., without the optional argument, then the header or footer appears arranged symmetrically on the page. In other words, a value for the *⟨shift⟩* is determined automatically to correspond to the current page shape.

Example: Continuing with the previous example, we remove the optional argument:

```
\setheadwidth{textwithmarginpar}
```

which appears like this on an odd page:

KOMA-Script	3
This fill text is currently seized by 130 million receptors in your retina. Thereby the nerve cells Retina are put in a state of stimulation, which spreads into the rear part of your brain originating from	

As can be seen, the header is now shifted inward, while the header width has not changed. The shift is calculated in a way that the configuration of the typearea become visible also here.

<code>\setheadtopline</code>	Corresponding to the size configuration parameters of header and footer
<code>\setheadsepline</code>	there are commands to modify the rules above and below the header
<code>\setfootsepline</code>	and footer. But first of all the rules should be activated. See op-
<code>\setfootbotline</code>	tions <code>headtopline</code> , <code>headsepline</code> , <code>footsepline</code> , and <code>footbotline</code> in sec-
	tion 1.1.4 for this.

`\setheadtopline[⟨length⟩][⟨commands⟩]` – configures the line above the header

`\setheadsepline[⟨length⟩][⟨commands⟩]` – configures the line below the header

`\setfootsepline[⟨length⟩][⟨commands⟩]` – configures the line above the footer

`\setfootbotline[⟨length⟩][⟨commands⟩]` – configures the line below the footer

The mandatory argument *⟨thickness⟩* determines how strongly the line is drawn. The optional argument *⟨length⟩* accepts the same symbolic values as *⟨width⟩* for `\setheadwidth`, as well as also a normal length expression. As long as the optional argument *⟨length⟩* is not assigned a value, the appropriate line length adapts automatically the width of the header or the footer.

Use `auto` in the length argument to restore this automation for the length of a line.

`\setkomafont{headtopline}` The optional argument *⟨commands⟩* may be used to specify additional
`\setkomafont{headsepline}` commands to be executed before the respective line is drawn. For exam-
`\setkomafont{footsepline}` ple, such commands could be used for changing the color of the line.
`\setkomafont{footbotline}` When using a KOMA-Script class you could also use `\setkomafont` to
`\setkomafont{footbottomline}` specify commands for one of the elements `headtopline`, `headsepline`,
`footsepline`, `footbottomline`, or `footbotline`. These can then be extended via `\addtokomafont`. See the KOMA-Script manual for details on the `\setkomafont` and `\addtokomafont` commands.

You may also use symbolic values for *⟨length⟩* and *⟨thickness⟩*, e.g., `\setheadtopline[auto]{current}`, `\setheadtopline[auto]{}`, or `\setheadtopline[auto]{-}`. The arguments shown here for the command `\setheadtopline` are of course valid for the other three configuration commands too.

If the mandatory parameter *⟨thickness⟩* has the value `current` or has been left empty, then the line thickness is not changed. This may be used to modify the length of the line without changing its thickness.

If the optional argument *⟨commands⟩* is omitted, then all command settings that might have been specified before will remain active, while an empty *⟨commands⟩* argument will revoke any previously valid commands.

Example: If the header, for example, is to be contrasted by a strong line of 2pt above and a normal line of 0.4pt between header and body, one can achieve this with:

```
\setheadtopline{2pt}
\setheadsepline{.4pt}
```

Additionally the options `headtopline` and `headsepline` have to be used preferably globally in the optional argument of `\documentclass`. In this case the result may be the following.

<hr/>	
KOMA-Script	3
This fill text is currently seized by 130 million receptors in your retina. Thereby the nerve cells Retina are put in a state of stimulation, which spreads into the rear part of your brain originating from	

To specify that this line is to be drawn also, e.g., in red color, you would change the commands like this:

```
\setheadtopline{2pt}[\color{red}]
\setheadsepline{.4pt}[\color{red}]
```

In this example, as well as in the following one, line color is activated by applying the syntax of the `color` package, so this package must of course be loaded. Since `scrpage2` comes without built-in color handling, any package providing color support may be used.

KOMA-Script classes also support the following way of color specification:

```
\setheadtopline{2pt}
\setheadsepline{.4pt}
\setkomafont{headtopline}[\color{red}]
\setkomafont{headsepline}[\color{red}]
```

The automatic adjustment to the header and footer width is illustrated in the following example:

```
\setfootbotline{2pt}
\setfootsepline[text]{.4pt}
\setfootwidth[0pt]{textwithmarginpar}
```

This fill text is currently seized by 130 million receptors in your retina. Thereby the nerve cells Retina are put in a state of stimulation, which spreads	
KOMA-Script	3

Now not everyone will like the alignment of the line above the footer; instead, one would expect the line to be left-aligned. This can only be achieved with a global package option, which will be described together with other package options in the next section 1.1.4.

1.1.4 Package Options

In opposite to the KOMA-Script classes, where the most options may be changed using `\KOMAOptions` or `\KOMAoption` also after loading the class, package `scrpage2` does not provide this feature. All options to `scrpage2` have to be global options, i.e. be part of the optional argument of `\documentclass`, or package option, i.e. be part of the optional argument of `\usepackage`.

<code>headinclude</code>	Since version 2.3 of the package this options should not be passed to <code>scrpage2</code> any longer using <code>\PassOptionsToPackage</code> or the optional argument of <code>\usepackage</code> . Only for compatibility reason <code>scrpage2</code> still declares them and pass them as <code>headinclude</code> , <code>headinclude=false</code> , <code>footinclude</code> , and <code>footinclude=false</code> to package <code>typearea</code> .
<code>headexclude</code>	
<code>footinclude</code>	
<code>footexclude</code>	
<code>headtopline</code>	Basic adjustment of the lines under and over header and footer can be made with these options. These adjustments are then considered the default for all page styles defined with <code>scrpage2</code> . If one of these options is used, then a line thickness 0.4pt is set. Since there is a corresponding plain page style to the page style <code>scrheadings</code> , the corresponding line in the plain style can also be configured with the <code>plain...</code> options. These <code>plain</code> options do however work only if the corresponding options without <code>plain</code> are activated. Thus, <code>plainheadtopline</code> shows no effect without the <code>headtopline</code> option set.
<code>plainheadtopline</code>	
<code>headsepline</code>	
<code>plainheadsepline</code>	
<code>footsepline</code>	
<code>plainfootsepline</code>	
<code>footbotline</code>	
<code>plainfootbotline</code>	

With these options, it is to be noted that the appropriate page part, header or footer, is considered as a part of the text area for the calculation of the type area in case a line has been activated. This means that, if the separation line between header and text is activated with `headsepline`, then the package `typearea` calculates the type area in such a way that the page header is part of the text block automatically.

The conditions for the options of the preceding paragraph apply also to this automation. That means that the package `typearea` must be loaded after `scrpage2`, or that on use of a KOMA-Script class, the options `headinclude` and `footinclude` must be set explicitly with `\documentclass` in order to transfer header or footer line in the text area.

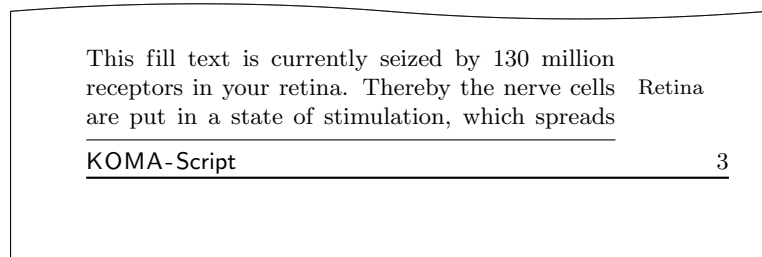
<code>ilines</code>	With the definition of the line lengths the case can arise where the lengths are set correctly, but the justification is not as desired because the line will be centered in the header or footer area. With the package options presented here, this specification can be modified for all page styles defined
<code>clines</code>	
<code>olines</code>	

with `scrpage2`. The option `ilines` sets the justification in such a way that the lines align to the inside edge. The option `clines` behaves like the default justification, and `olines` aligns at the outside edge.

Example: The next example illustrates the influence of the option `ilines`. Please compare to the example for `\setfootsepline` on page 11.

```
\usepackage[ilines]{scrpage2}
\setfootbotline{2pt}
\setfootsepline[text]{.4pt}
\setfootwidth[0pt]{textwithmarginpar}
```

The mere use of the option `ilines` leads to the different result shown below:



In contrast to the default configuration, the separation line between text and footer is now left-aligned, not centered.

automark These options set at the beginning of the document whether to use running
manualmark headings or manual ones. The option `automark` switches the automatic updating on, `manualmark` deactivates it. Without the use of one of the two options, the setting which was valid when the package was loaded is preserved.

Example: You load the package `scrpage2` directly after the document class `scrreprt` without any package options:

```
\documentclass{scrreprt}
\usepackage{scrpage2}
```

Since the default page style of `scrreprt` is `plain`, this page style is also now still active. Furthermore, `plain` means manual headings. If one now activates the page style `scrheadings` with

```
\pagestyle{scrheadings}
```

then the manual headings are nevertheless still active.

If you instead use the document class `scrbook`, then after

```
\documentclass{scrbook}
\usepackage{scrpage2}
```

the page style `headings` is active and the running headings are updated automatically. Switching to the page style `scrheadings` keeps this setting active. The marking commands of `scrbook` continue to be used.

However, the use of

```
\usepackage[automark]{scrpage2}
```

activates running headings independently of the used document class. The option does not of course affect the used page style `plain` of the class `scrreprt`. The headings are not visible until the page style is changed to `scrheadings`, `useheadings` or another user-defined page style with headings.

<code>autooneside</code>	This option ensures that the optional parameter of <code>\automark</code> will be ignored automatically in one-sided mode. See also the explanation of the command <code>\automark</code> in section 1.1.2.
<code>komastyle</code> <code>standardstyle</code>	<p>These options determine the look of the predefined page styles <code>scrheadings</code> and <code>scrplain</code>. The option <code>komastyle</code> configures a look like that of the KOMA-Script classes. This is the default for KOMA-Script classes and can in this way also be set for other classes.</p> <p>The option <code>standardstyle</code> configures a page style as it is expected by the standard classes. Furthermore, the option <code>markuppercase</code> will be activated automatically, but only if option <code>markusedcase</code> is not given.</p>
<code>markuppercase</code> <code>markusedcase</code>	<p>In order to achieve the functionality of <code>\automark</code>, the package <code>scrpage2</code> modifies internal commands which are used by the document structuring commands to set the running headings. Since some classes, in contrast to the KOMA-Script classes, write the headings in uppercase letters, <code>scrpage2</code> has to know how the used document class sets the headings.</p> <p>Option <code>markuppercase</code> shows <code>scrpage2</code> that the document class uses uppercase letters. If the document class does not set the headings in uppercase letters, then the option <code>markusedcase</code> should be given. These options are not suitable to force a representation; thus, unexpected effects may occur if the given option does not match the actual behaviour of the document class.</p>

nouppercase In the previous paragraph dealing with **markuppercase** and **markusedcase**, it has been already stated that some document classes set the running headings in uppercase letters using the commands `\MakeUppercase` or `\uppercase`. Setting the option **nouppercase** allows disabling both these commands in the headers and footers. However, this is valid only for page styles defined by **scrpage2**, including **scrheadings** and its corresponding plain page style.

The applied method is very brutal and can cause that desired changes of normal letters to uppercase letters do not occur. Since these cases do not occur frequently, the option **nouppercase** usually affords a useful solution.

Example: Your document uses the standard class **book**, but you do not want the uppercase headings but mixed case headings. Then the preamble of your document could start with:

```
\documentclass{book}
\usepackage[nouppercase]{scrpage2}
\pagestyle{scrheadings}
```

The selection of the page style **scrheadings** is necessary, since otherwise the page style **headings** is active, which does not respect the settings made by option **nouppercase**.

In some cases not only classes but also packages set the running headings in uppercase letters. Also in these cases the option **nouppercase** should be able to switch back to the normal mixed case headings.

1.2 Defining Own Page Styles

1.2.1 The Interface for Beginners

\deftripstyle Now one would not like to remain bound to only the provided page styles, but may wish to define one's own page styles. Sometimes there will be a special need, since a specific *Corporate Identity* may require the declaration of its own page styles. The easiest way to deal with this is:

```
\deftripstyle{<name>}
    [<LO>] [<LI>]
    {<HI>}{<HC>}{<HO>}
    {<FI>}{<FC>}{<FO>}
```

The individual parameters have the following meanings:

<name> – the name of the page style, in order to activate it using the command `\pagestyle{<name>}`

<LO> – the thickness of the outside lines, i. e., the line above the header and the line below the footer (optional)

- $\langle LI \rangle$ – the thickness of the separation lines, i.e., the line below the header and the line above the foot (optional)
- $\langle HI \rangle$ – contents of the inside box in the page header for two-sided layout or left for one-sided layout
- $\langle HC \rangle$ – contents of the centered box in the page header
- $\langle HO \rangle$ – contents of the outside box in the page header for two-sided layout or right for one-sided layout
- $\langle FI \rangle$ – contents of the inside box in the page footer for two-sided layout or left for one-sided layout
- $\langle FC \rangle$ – contents of the centered box in the page footer
- $\langle FO \rangle$ – contents of the outside box in the page footer for two-sided layout or right for one-sided layout

The command `\deftripstyle` definitely represents the simplest possibility of defining page styles. Unfortunately, there are also restrictions connected with this, since in a page range using a page style defined via `deftripstyle`, no modification of the lines above and below header and footer can take place.

Example: Assume a two-sided layout, where the running headings are placed on the inside. Furthermore, the document title, here “Report”, shall be placed outside in the header, the page number shall be centered in the footer.

```
\deftripstyle{TheReport}%
               {\headmark}{\Report}%
               {}{\pagemark}{}
```

If moreover the lines above the header and below the footer shall be drawn with a thickness of 2 pt, and the text body be separated from header and footer with 0.4 pt lines, then the definition has to be extended:

```
\deftripstyle{TheReport}[2pt][.4pt]%
               {\headmark}{\Report}%
               {}{\pagemark}{}
```

See figure 2 for the result.

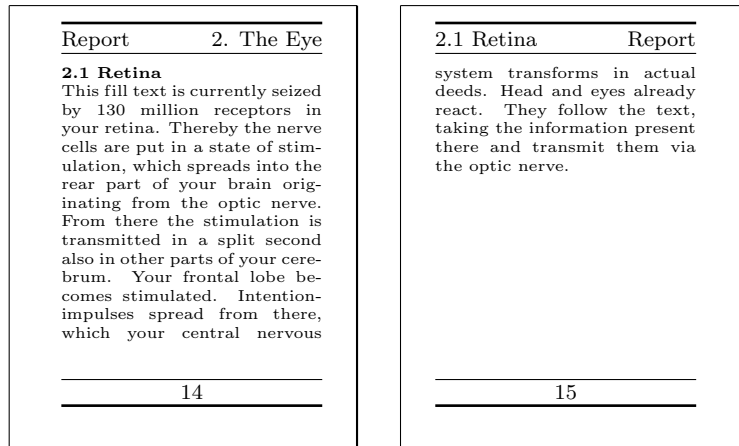


Figure 2: Example of a user defined, line dominated page style with a static and a running heading at the page header and a page number centered at the page footer

1.2.2 The Interface for Experts

`\defpagestyle` Simple page styles, as they can be defined with `\deftripstyle`, are fairly rare according to experience. Either a professor requires that the thesis looks like his or her own — and who seriously wants to argue against such a wish? — or a company would like that half the financial accounting emerges in the page footer. No problem, the solution is:

```
\defpagestyle{<name>}{<header definition>}{<footer definition>}
\newpagestyle{<name>}{<header definition>}{<footer definition>}
\renewpagestyle{<name>}{<header definition>}{<footer definition>}
\providepagestyle{<name>}{<header definition>}{<footer definition>}
```

These four commands give full access to the capabilities of `scrpage2` to define page styles. Their structure is identical, they differ only in the manner of working.

- `\defpagestyle` – defines a new page style. If a page style with this name already exists it will be overwritten.
- `\newpagestyle` – defines a new page style. If a page style with this name already exists a error message will be given.
- `\renewpagestyle` – redefines a page style. If a page style with this name does not exist a error message will be given.
- `\providepagestyle` – defines a new page style only if there is no page style with that name already present.

Using `\defpagestyle` as an example, the syntax of the four commands is explained below.

- $\langle name \rangle$ – the name of the page style for `\pagestyle{\langle name \rangle}`
- $\langle header\ definition \rangle$ – the declaration of the header, consisting of five element; elements in round parenthesis are optional:
 $(\langle ALL, ALT \rangle)\{\langle EP \rangle\}\{\langle OP \rangle\}\{\langle OS \rangle\}(\langle BLL, BLT \rangle)$
- $\langle footer\ definition \rangle$ – the declaration of the footer, consisting of five element; elements in round parenthesis are optional:
 $(\langle ALL, ALT \rangle)\{\langle EP \rangle\}\{\langle OP \rangle\}\{\langle OS \rangle\}(\langle BLL, BLT \rangle)$

As can be seen, header and footer declaration have identical structure. The individual parameters have the following meanings:

- $\langle ALL \rangle$ – above line length: (header = outside, footer = separation line)
- $\langle ALT \rangle$ – above line thickness
- $\langle EP \rangle$ – definition for *even* pages
- $\langle OP \rangle$ – definition for *odd* pages
- $\langle OS \rangle$ – definition for *one-sided* layout
- $\langle BLL \rangle$ – below line length: (header = separation line, footer = outside)
- $\langle BLT \rangle$ – below line thickness

If the optional line-parameters are omitted, then the line behaviour remains configurable by the commands introduced in section 1.1.3.

The three elements $\langle EP \rangle$, $\langle OP \rangle$ and $\langle OS \rangle$ are boxes with the width of page header or footer, as appropriate. The corresponding definitions are set left-justified in the boxes. To set something left- *and* right-justified into the boxes, the space between two text elements can be stretched using `\hfill`:

```
{\headmark\hfill\pagemark}
```

If one would like a third text-element centered in the box, then an extended definition must be used. The commands `\rlap` and `\llap` simply write the given arguments, but for \LaTeX they take up no horizontal space. Only in this way is the middle text really centered:

```
{\rlap{\headmark}\hfill centered text\hfill\llap{\pagemark}}
```

Example: This examples uses the document class `scrbook`, which means that the default page layout is two-sided. The package `scrpage2` is loaded with options `automark` and `headsepline`. The first switches on the automatic update of running headings, the second determines that a separation line between header and text body is drawn in the `scrheadings` page style.

```
\documentclass{scrbook}
\usepackage[automark,headsepline]{scrpage2}
```

The expert interface is used to define two page styles. The page style `withoutLines` does not define any line parameters. The second page style `withLines` defines a line thicknes of 1pt for the line above the header and 0pt for the separation line between header and text.

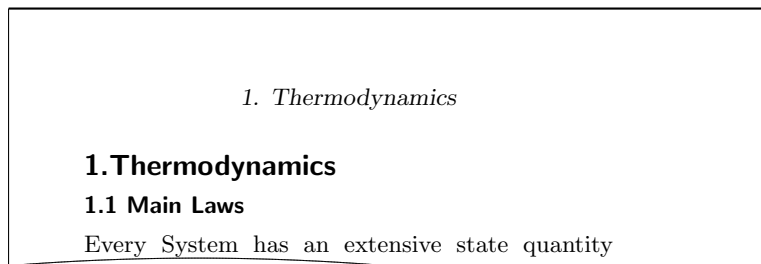
```
\defpagestyle{withoutLines}{%
  {Example\hfill\headmark}{\headmark\hfill without lines}
  {\rlap{Example}\hfill\headmark\hfill%
   \llap{without lines}}
}%
  {\pagemark\hfill}
  {\hfill\pagemark}
  {\hfill\pagemark\hfill}
}

\defpagestyle{withLines}{%
  (\textwidth,1pt)
  {with lines\hfill\headmark}{\headmark\hfill with lines}
  {\rlap{KOMAScript}\hfill \headmark\hfill%
   \llap{with lines}}
  (0pt,0pt)
}%
  (\textwidth,.4pt)
  {\pagemark\hfill}
  {\hfill\pagemark}
  {\hfill\pagemark\hfill}
  (\textwidth,1pt)
}
```

Right at the beginning of the document the page style `scrheadings` is activated. The command `\chapter` starts a new chapter and automatically sets the page rstyle for this page to `plain`. Even though not a prime example, the command `\chead` shows how running headings can be created even on a plain page. However, in principle running headings on chapter start-pages are to be avoided, since otherwise the special character of the `plain` page

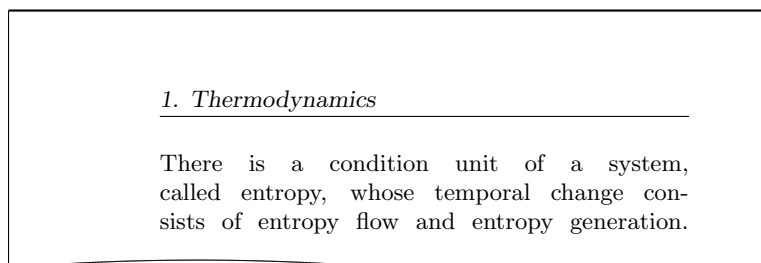
style is lost. It is more important to indicate that a new chapter starts here than that a section of this page has a special title.

```
\begin{document}
\pagestyle{scrheadings}
\chapter{Thermodynamics}
\chead[\leftmark]{}
\section{Main Laws}
Every system has an extensive state quantity called
Energy. In a closed system the energy is constant.
```



After starting a new page the page style `scrheadings` is active and thus the separation line below the header is visible.

There is a state quatity of a system, called entropy, whose temporal change consists of entropy flow and entropy generation.



After switching to the next page, the automatic update of the running headings is disabled using `\manualmark`, and the page style `withoutLines` becomes active. Since no line parameters are given in the definition of this page style, the default configuration is used, which draws a separation line between header and text body because `scrpage2` was called with `headsepline`.

```
\manualmark
\pagestyle{withoutLines}
\section{Exergy and Anergy}
```

```
\markright{Energy Conversion}
During the transition of a system to an equilibrium
state with its environment, the maximum work gainable
is called exergy.
```

<div style="display: flex; justify-content: space-between;"> <i>Energy Conversion</i> <i>without lines</i> </div> <hr style="width: 100%;"/> <p>1.2 Exergy and Anergy During the transition of a system to an equilibrium state with its environment, the maximum work gainable is called exergy.</p>

At the next page of the document, the page style `withLines` is activated. The line settings of its definition are taken in account and the lines are drawn accordingly.

```
\pagestyle{mitLinien}
\renewcommand{\headfont}{\itshape\bfseries}
The portion of an energy not convertible in exergy
is named anergy \Var{B}.
\[ B = U + T (S_1 - S_u) - p (V_1 - V_u)\]
\end{document}
```

<div style="display: flex; justify-content: space-between;"> <i>with lines</i> <i>1. Thermodynamics</i> </div> <hr style="width: 100%;"/> <p>The portion of an energy not convertible in exergy is named anergy <i>B</i>.</p> $B = U + T(S_1 - S_u) - p(V_1 - V_u)$

1.2.3 Managing Page Styles

Before long the work with different page styles will establish a common set of employed page styles, depending on taste and tasks. In order to make the management of page styles easier and avoid time-consuming copy operations each time a new project is started, `scrpage2` reads the file `scrpage.cfg` after initialisation. This file can contain a set of user-defined page styles which many projects can share.

Das obsoleete Paket `scrpage2`

Markus Kohm Jens-Uwe-Morawski

2014-06-25

Dies ist das frühere KOMA-Script-Paket `scrpage2`. Das Paket ist veraltet und sollte nicht länger verwendet werden. Sie können es sehr einfach durch das KOMA-Script-Paket `scrlayer-scrpage` ersetzen, das in der englischen und deutschen KOMA-Script-Anleitung dokumentiert ist.

Inhaltsverzeichnis

English	1
1 Adapting Page Headers and Footers with <code>scrpage2</code>	2
1.1 Basic Functionality	2
1.1.1 Predefined Page Styles	3
1.1.2 Manual and Running Headings	6
1.1.3 Formatting of Header and Footer	7
1.1.4 Package Options	12
1.2 Defining Own Page Styles	15
1.2.1 The Interface for Beginners	15
1.2.2 The Interface for Experts	17
1.2.3 Managing Page Styles	21
 Deutsch	 22
2 Kopf- und Fußzeilen mit <code>scrpage2</code>	23
2.1 Grundlegende Funktionen	24
2.1.1 Vordefinierte Seitenstile	24
2.1.2 Manuelle und automatische Kolumnentitel	27
2.1.3 Formatierung der Kopf- und Fußzeilen	29
2.1.4 Optionen beim Laden des Paketes	34
2.2 Seitenstile selbst gestalten	37
2.2.1 Die Anwenderschnittstelle	37
2.2.2 Die Expertenschnittstelle	39

2.2.3	Seitenstile verwalten	44
3	Implementation	44
3.1	Options	44
3.2	Some Initialisation	48
3.3	Predefinitions	48
3.4	Expert-Pagestyle-Definition-Interface	48
3.5	User-Pagestyle-Definition-Interface	53
3.6	Automark or not	55
3.7	Width of head and foot	58
3.8	Fonts	61
3.9	Length and thickness of lines at head and foot	61
3.10	Extended multilanguage formats	62
3.11	Definition of default pagestyles	62
3.12	Configurationfile	71
3.13	Some checks	71
3.14	End	71

2 Kopf- und Fußzeilen mit `scrpage2`

Mit KOMA-Script 3.12 wurde das komplett neu implementierte Paket `scrlayer-scrpage` vorgestellt, das eine konsequente Weiterführung des Designs von `scrpage2` mit anderen Mitteln darstellt. Siehe dazu das entsprechende Kapitel der KOMA-Script-Anleitung. Während `scrpage2` die erweiterte Optionen-Schnittstelle der KOMA-Script-Klassen nicht unterstützt, kommt diese in `scrlayer-scrpage` selbstverständlich zum Einsatz. Da der Ansatz von `scrlayer-scrpage` gegenüber `scrpage2` als überlegen betrachtet wird, wird somit empfohlen, `scrlayer-scrpage` an Stelle von `scrpage2` zu verwenden. Die aktuelle Version von `scrpage2` ist daher auch als final zu betrachten. Sämtliche Entwicklungsressourcen im Bereich Kopf- und Fußzeilen werden zukünftig in `scrlayer-scrpage` einfließen.

An Stelle von `scrpage2` oder `scrlayer-scrpage` kann natürlich auch `fancyhdr` verwendet werden. `scrpage2` und insbesondere `scrlayer-scrpage` harmonieren jedoch mit den KOMA-Script-Klassen deutlich besser. Genau deshalb und weil der Vorläufer von `fancyhdr` damals viele Möglichkeiten vermissen lies, ist `scrpage2` entstanden. Natürlich ist das Paket `scrpage2` ebenso wie das Paket `scrlayer-scrpage` nicht an eine KOMA-Script-Klasse gebunden, sondern kann auch sehr gut mit anderen Klassen verwendet werden.

2.1 Grundlegende Funktionen

Um die nachfolgende Beschreibung zu verstehen, muss noch einiges zu L^AT_EX gesagt werden. Im L^AT_EX-Kern sind die Standardseitenstile **empty**, welcher eine völlig undekorierte Seite erzeugt, und **plain**, welcher meist nur die Seitenzahl enthält, definiert. In vielen Klassen ist der Stil **headings** zu finden, welcher eine komplexe Seitendekoration mit automatischen Kolumnentitel erzeugt. Die Variante **myheadings** gleicht **headings**. Die Kolumnentitel müssen dabei aber manuell gesetzt werden. Ausführlicher wird das im Abschnitt über Seitenstile der KOMA-Script-Anleitung beschrieben. Dort wird auch erläutert, dass auf einigen Seiten der Seitenstil automatisch – in der Regel zu **plain** – wechselt.

Das Paket **scrpage2** unterscheidet nicht mehr zwischen Seitenstilen mit automatischem und mit manuellem Kolumnentitel. Die Wahl des Seitenstils erfolgt unabhängig davon, ob mit automatischem oder manuellem Kolumnentitel gearbeitet wird. Näheres dazu finden Sie in Abschnitt 2.1.2.

2.1.1 Vordefinierte Seitenstile

Zu den grundlegenden Funktionen von **scrpage2** gehören unter anderem vordefinierte, konfigurierbare Seitenstile.

```
\pagestyle{scrheadings}  
\pagestyle{scrplain}
```

Das Paket **scrpage2** liefert für Seiten mit Kolumnentitel einen eigenen Seitenstil namens **scrheadings**. Dieser Seitenstil kann mittels `\pagestyle{scrheadings}` aktiviert werden. Wird dieser Seitenstil benutzt, dann wird gleichzeitig der **plain**-Stil durch den dazu passenden Stil **scrplain** ersetzt. Passend bedeutet, dass auch der **plain**-Stil auf in Abschnitt 2.1.3 vorgestellte Befehle, die beispielsweise die Kopfbreite ändern, reagiert und im Grundlayout übereinstimmt. Die Aktivierung des Seitenstils **scrheadings** oder des zugehörigen **plain**-Stils, **scrplain**, hat keine Auswirkung darauf, ob mit manuellen oder automatischen Kolumnentiteln gearbeitet wird (siehe Abschnitt 2.1.2). Der Seitenstil **scrplain** kann auch direkt per `\pagestyle` aktiviert werden.

<code>\lehead</code>	Die Seitenstile von scrpage2 sind so definiert, dass ihr Kopf und Fuß flexibel angepasst werden kann. Hierzu sind sowohl im Kopf als auch im Fuß drei Felder vorhanden, deren Inhalt modifiziert werden kann. Die Befehle zur Modifikation sind in Abbildung 3 verdeutlicht. Die in der Mitte dargestellten Befehle modifizieren sowohl die Felder der linken als auch der rechten Seite. Alle Befehle haben sowohl ein optionales als auch ein obligatorisches Argument. Das optionale Argument bestimmt jeweils das durch den Befehl festgelegte Feld im plain -Seitenstil, scrplain . Das obligatorische Argument definiert das entsprechende Feld im Seitenstil scrheadings .
<code>\cehead</code>	
<code>\rehead</code>	
<code>\lefoot</code>	
<code>\cefoot</code>	
<code>\refoot</code>	
<code>\lohead</code>	
<code>\cohead</code>	
<code>\rohead</code>	
<code>\lofoot</code>	
<code>\cofoot</code>	
<code>\rofoot</code>	
<code>\ihead</code>	
<code>\chead</code>	
<code>\ohead</code>	
<code>\ifoot</code>	
<code>\cfoot</code>	
<code>\ofoot</code>	

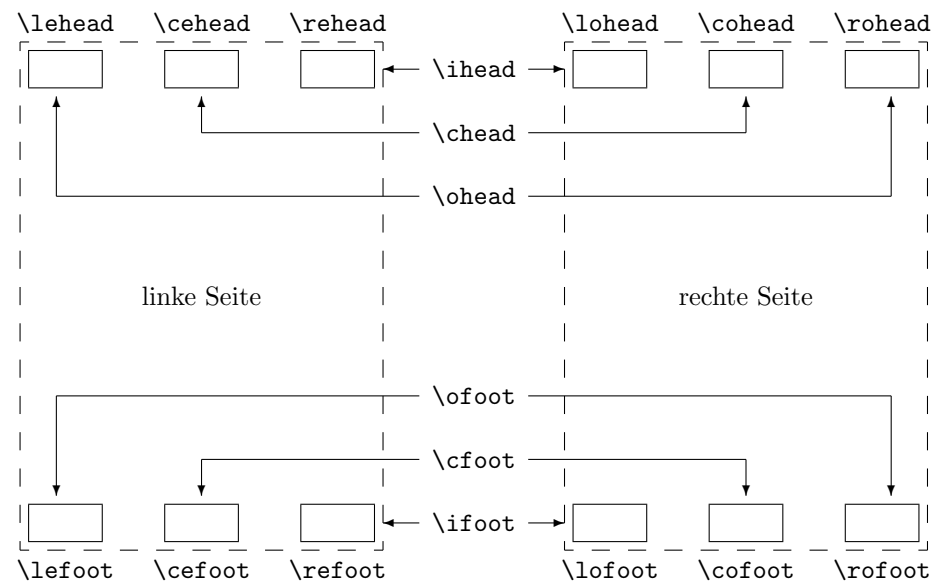


Abbildung 3: Zuordnung der Befehle zur Manipulation der Seitenstile **scrheadings** und **scrplain** zu den manipulierten Seitenelementen

Example: Angenommen, man möchte bei **scrheadings** zentriert im Seitenfuß die Seitenzahl dargestellt haben, dann benutzt man einfach:

```
\cfoot{\pagemark}
```

Sollen die Seitenzahlen im Kopf außen und die Kolumnentitel innen stehen, dann erfolgt dies mit:

```
\ohead{\pagemark}
\ihead{\headmark}
\cfoot{}
```

Das `\cfoot{}` ist nur notwendig, um eine möglicherweise in der Mitte des Fußes vorhandene Seitenzahl zu entfernen.

Die Befehle, die direkt nur einem Feld zugeordnet sind, können für anspruchsvollere Vorhaben genutzt werden.

Example: Angenommen, man hat den Auftrag, einen Jahresbericht einer Firma zu erstellen, dann könnte das so angegangen werden:

```
\ohead{\pagemark}
\rehead{Jahresbericht 2001}
\lohead{\headmark}
```

```
\cefoot{Firma WasWeißIch}
\cofoot{Abteilung Entwicklung}
```

Natürlich muss man hier dafür sorgen, dass mittels `\cofoot` der Fuß der rechten Seite aktualisiert wird, wenn eine neue Abteilung im Bericht besprochen wird.

Wie oben dargestellt, gibt es einen zu `scrheadings` korrespondierenden plain-Seitenstil. Da es auch möglich sein soll, diesen Stil anzupassen, unterstützen die Befehle ein optionales Argument. Damit kann der Inhalt des entsprechenden Feldes im plain-Seitenstil modifiziert werden.

Example: Um für die Nutzung von `scrheadings` die Position der Seitenzahlen festzulegen, kann man folgendes benutzen:

```
\cfoot[\pagemark]{}
\ohead[]{\pagemark}
```

Wird anschließend der Stil `plain` genutzt, beispielsweise weil `\chapter` eine neue Seite beginnt und darauf umschaltet, dann steht die Seitenzahl zentriert im Seitenfuß.

<pre>\clearscrheadings \clearscrplain \clearscrheadfoot</pre>	<p>Will man sowohl den Seitenstil <code>scrheadings</code> als auch den dazu gehörenden plain-Seitenstil von Grund auf neu definieren, muss man häufig zusätzlich einige der bereits belegten Seitenelemente löschen. Da man selten alle Elemente mit neuem Inhalt füllt, sind dazu in den meisten Fällen mehrere Befehle mit leeren Parametern notwendig. Mit Hilfe dieser drei Befehle ist das Löschen schnell und gründlich möglich. Während <code>\clearscrheadings</code> lediglich alle Felder des Seitenstils <code>scrheadings</code> und <code>\clearscrplain</code> alle Felder des zugehörigen plain-Seitenstils löscht, werden von <code>\clearscrheadfoot</code> alle Felder beider Seitenstile auf leere Inhalte gesetzt.</p>
---	---

Example: Sie wollen unabhängig davon, wie die Seitenstile derzeit aussehen, die Standardform der KOMA-Script-Klassen bei zweiseitigem Satz erreichen. Dies ist mit nur drei Befehlen möglich:

```
\clearscrheadfoot
\ohead{\headmark}
\ofoot[\pagemark]{\pagemark}
```

Ohne die Befehle `\clearscrheadings`, `\clearscrplain` und `\clearscrheadfoot` wären doppelt so viele Anweisungen und neun weitere leere Argumente notwendig:

```
\ihead[]{}

```

```

\thead[]{}
\ohead[]{\headmark}
\ifoot[]{}
\cfoot[]{}
\ofoot[\pagemark]{\pagemark}

```

Einige davon könnten natürlich entfallen, wenn man von einer konkreten Vorbelegung ausginge.

In den vorausgehenden Beispielen wurden schon zwei Befehle benutzt, die noch gar nicht besprochen wurden. Das soll jetzt nachgeholt werden.

`\leftmark` `\rightmark` Diese beiden Befehle erlauben es, auf die Kolumnentitel zuzugreifen, die normalerweise für die linke bzw. die rechte Seite gedacht sind. Diese beiden Befehle werden nicht von `scrpage2`, sondern direkt vom \LaTeX -Kern zur Verfügung gestellt. Wenn in diesem Kapitel vom Kolumnentitel der linken Seite oder vom Kolumnentitel der rechten Seite die Rede ist, dann ist damit eigentlich der Inhalt von `\leftmark` und `\rightmark` gemeint.

`\headmark` Dieser Befehl ermöglicht es, auf die Inhalte der Kolumnentitel zuzugreifen. Im Gegensatz zu den originalen \LaTeX -Befehlen `\leftmark` und `\rightmark` braucht man nicht auf die richtige Zuordnung zur linken oder rechten Seite zu achten.

`\pagemark` Dieser Befehl ermöglicht den Zugriff auf die Seitenzahl. Im Abschnitt 2.1.3 wird der Befehl `\pnumfont` zur Formatierung der Seitenzahl vorgestellt, den `\pagemark` automatisch berücksichtigt.

`\pagestyle{useheadings}` Das Paket `scrpage2` ist in erster Linie dafür gedacht, dass die bereitgestellten Stile benutzt oder eigene Stile definiert werden. Jedoch kann es notwendig sein, auch auf einen von der Dokumentklasse zur Verfügung gestellten Stil zurückzuschalten. Es wäre nahe liegend, dieses mit `\pagestyle{headings}` vorzunehmen. Das hätte aber den Nachteil, dass die nachfolgend besprochenen Befehle `\automark` und `\manualmark` nicht wie erwartet funktionierten. Daher sollte mit `\pagestyle{useheadings}` auf die originalen Stile umgeschaltet werden. Eine solche Umschaltung hat dann keine Auswirkung darauf, ob mit manuellen oder automatischen Kolumnentiteln gearbeitet wird.

2.1.2 Manuelle und automatische Kolumnentitel

Gewöhnlich gibt es zu einem `headings`-Stil eine *my*-Variante. Ist ein solcher Stil aktiv, dann werden die Kolumnentitel nicht mehr automatisch aktualisiert. Bei `scrpage2` wird ein anderer Weg beschritten. Ob die Kolumnentitel lebend sind oder nicht, bestimmen die Befehle `\automark` und

`\manualmark`. Die Voreinstellung kann auch bereits beim Laden des Paketes über die Optionen `automark` und `manualmark` beeinflusst werden (siehe Abschnitt 2.1.4).

`\manualmark` Wie der Name bereits verdeutlicht, schaltet `\manualmark` die Aktualisierung der Kolumnentitel aus. Es bleibt somit dem Nutzer überlassen, für eine Aktualisierung bzw. für den Inhalt der Kolumnentitel zu sorgen. Dazu stehen die Befehle `\markboth` und `\markright` aus dem L^AT_EX-Kern bereit. Diese Anweisungen sind im Abschnitt über Seitenstile in der KOMA-Script-Anleitung erklärt.

`\automark` Die Anweisung `\automark[<rechte Seite>]{<linke Seite>}` aktiviert die automatische Aktualisierung des Kolumnentitels. Für die beiden Parameter sind die Bezeichnungen der Gliederungsebenen einzusetzen, deren Titel an entsprechender Stelle erscheinen soll. Gültige Werte für die Parameter sind: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph` und `subparagraph`. Der Wert `part` führt bei Verwendung der meisten Klassen nicht zu dem gewünschten Ergebnis. Bisher ist nur von den KOMA-Script-Klassen ab Version 2.9s bekannt, dass dieser Wert unterstützt wird. Das optionale Argument *<rechte Seite>* ist verständlicherweise nur für zweiseitigen Satz gedacht. Im einseitigen Satz sollten Sie normalerweise darauf verzichten. Mit Hilfe der Option `autooneside` können Sie auch einstellen, dass das optionale Argument im einseitigen Satz automatisch ignoriert wird (siehe Abschnitt 2.1.4).

Example: Wird beispielsweise mit einer *book*-Klasse gearbeitet, deren höchste Gliederungsebene *chapter* ist, dann stellt nach einem vorhergehenden `\manualmark` der Befehl

```
\automark[section]{chapter}
```

den Originalzustand wieder her. Bevorzugt man stattdessen, die tieferen Gliederungsebenen angezeigt zu bekommen, dann erfolgt dies mit:

```
\automark[subsection]{section}
```

Die Markierung der jeweils höheren Gliederungsebene wird mit Hilfe von `\markboth` gesetzt. Die Markierung der tieferen Gliederungsebene wird mit `\markright` bzw. `\markleft` gesetzt. Der entsprechende Aufruf erfolgt indirekt über die Gliederungsbefehle. Die Anweisung `\markleft` wird von `scrpage2` bereitgestellt und ist vergleichbar zu `\markright` aus dem L^AT_EX-Kern definiert. Obwohl sie nicht als internes Makro definiert ist, wird von einem direkten Gebrauch abgeraten.

2.1.3 Formatierung der Kopf- und Fußzeilen

Im vorherigen Abschnitt ging es hauptsächlich um inhaltliche Dinge. Das genügt natürlich nicht, um die gestalterischen Ambitionen zu befriedigen. Deshalb soll es sich in diesem Abschnitt ausschließlich darum drehen.

`\headfont` Die Schriftformatierung für den Seitenkopf und -fuß übernimmt der Befehl
`\footfont` `\headfont`, `\footfont` die Abweichung davon für den Fuß und `\pnumfont`
`\pnumfont` wiederum die Abweichung davon für die Seitenzahl.

Example: Um beispielsweise den Kopf in fetter, serifenloser Schrift und den Fuß in nicht fetter, serifenloser Schrift zu setzen und die Seitenzahl geneigt mit Serifen erscheinen zu lassen, nutzt man folgende Definitionen:

```
\renewcommand*{\headfont}{%
  \normalfont\sffamily\bfseries}
\renewcommand*{\footfont}{%
  \normalfont\sffamily}
\renewcommand*{\pnumfont}{%
  \normalfont\rmfamily\slshape}
```

`\setkomafont{pagehead}` Ab Version 2.8p der KOMA-Script-Klassen wurde die Schnittstelle für
`\setkomafont{pagefoot}` Schriftattribute vereinheitlicht. Wird `scrpage2` in Verbindung mit einer
`\setkomafont{pagenumber}` dieser Klassen verwendet, dann sollte die Zuweisung in der Art erfolgen, wie sie in der KOMA-Script-Anleitung beschrieben wird.

Example: Statt `\renewcommand` wird bei Verwendung einer KOMA-Script-Klasse vorzugsweise der Befehl `\setkomafont` verwendet. Die vorhergehenden Definitionen lauten damit:

```
\setkomafont{pagehead}{%
  \normalfont\sffamily\bfseries}
\setkomafont{pagefoot}{%
  \normalfont\sffamily}
\setkomafont{pagenumber}{%
  \normalfont\rmfamily\slshape}
```

`\setheadwidth` Normalerweise entsprechen die Breiten von Kopf- und Fußzeile der Breite
`\setfootwidth` des Textbereichs. Die beiden Befehle `\setheadwidth[⟨Verschiebung⟩]{⟨Breite⟩}` und `\setfootwidth[⟨Verschiebung⟩]{⟨Breite⟩}` ermöglichen dem Anwender, auf einfache Weise die Breiten seinen Bedürfnissen anzupassen. Das obligatorische Argument `⟨Breite⟩` nimmt den Wert der Breite des Kopfes bzw. des Fußes auf, `⟨Verschiebung⟩` ist ein Längenmaß für die Verschiebung des entsprechenden Elements in Richtung des äußeren Seitenrandes.

Für die möglichen Standardfälle akzeptiert das obligatorische Argument `⟨Breite⟩` auch folgende symbolische Werte:

<code>paper</code>	– die Breite des Papiers
<code>page</code>	– die Breite der Seite
<code>text</code>	– die Breite des Textbereichs
<code>textwithmarginpar</code>	– die Breite des Textbereichs inklusive Seitenrand
<code>head</code>	– die aktuelle Breite des Seitenkopfes
<code>foot</code>	– die aktuelle Breite des Seitenfußes

Der Unterschied zwischen `paper` und `page` besteht darin, dass `page` die Breite des Papiers abzüglich der Bindekorrektur ist, falls das `typearea`-Paket verwendet wird (siehe das Kapitel über `typearea` in der KOMA-Script-Anleitung). Ohne Verwendung von `typearea` sind `paper` und `page` identisch.

Example: Angenommen, man möchte ein Seitenlayout wie im *L^AT_EX-Begleiter*, bei dem die Kopfzeile in den Rand ragt, dann geschieht das ganz einfach mit:

```
\setheadwidth[Opt]{textwithmarginpar}
```

und sieht dann auf einer rechten Seite folgendermaßen aus:

KOMA-Script	3
Dieser Blindtext wird gerade von 130 Millionen Rezeptoren Ihrer Netzhaut erfasst. Die Zellen werden dadurch in einen Erregungszustand versetzt, der sich vom Sehnerv in den	Netzhaut (<i>Retina</i>)

Soll der Seitenfuß die gleiche Breite und Ausrichtung haben, dann hat man jetzt zwei Wege. Der erste ist, man wiederholt das Gleiche für den Seitenfuß mit:

```
\setfootwidth[Opt]{textwithmarginpar}
```

oder man greift auf den anderen symbolischen Wert `head` zurück, da der Kopf bereits die gewünschte Breite hat.

```
\setfootwidth[Opt]{head}
```

Wird keine Verschiebung angegeben, das heißt auf das optionale Argument verzichtet, dann erscheint der Kopf bzw. der Fuß symmetrisch auf der Seite angeordnet. Es wird somit ein Wert für die Verschiebung automatisch ermittelt, der der aktuellen Seitengestalt entspricht.

Example: Entsprechend dem vorherigen Beispiel wird hier auf das optionale Argument verzichtet:

```
\setheadwidth{textwithmarginpar}
```

und sieht dann auf einer rechten Seite folgendermaßen aus:

KOMA-Script	3
Dieser Blindtext wird gerade von 130 Millionen Rezeptoren Ihrer Netzhaut erfasst. Die Zellen werden dadurch in einen Erregungszustand versetzt, der sich vom Sehnerv in den	
	Netzhaut (Retina)

Wie zu sehen, ist der Kopf jetzt nach innen verschoben, wobei die Kopfbreite sich nicht geändert hat. Die Verschiebung ist so berechnet, dass die Seitenproportionen auch hier sichtbar werden.

`\setheadtopline` Entsprechend den Größenparametern für die Kopf- und Fußzeile gibt es
`\setheadsepline` auch Befehle, die die Dimensionen der Linien im Kopf und Fuß modifi-
`\setfootsepline` zieren können. Dazu sollten diese Linien aber zunächst erst einmal ein-
`\setfootbotline` geschaltet werden. Siehe hierzu die Optionen `headtopline`, `headsepline`,
`footsepline`, `footbotline` in Abschnitt 2.1.4.

`\setheadtopline[⟨Länge⟩]{⟨Dicke⟩}[⟨Anweisungen⟩]` – modifiziert die Pa-
 rameter für die
 Linie über dem
 Seitenkopf

`\setheadsepline[⟨Länge⟩]{⟨Dicke⟩}[⟨Anweisungen⟩]` – modifiziert die Pa-
 rameter für die
 Linie zwischen Kopf
 und Textkörper

`\setfootsepline[⟨Länge⟩]{⟨Dicke⟩}[⟨Anweisungen⟩]` – modifiziert die Pa-
 rameter für die
 Linie zwischen Text
 und Fuß

`\setfootbotline[⟨Länge⟩]{⟨Dicke⟩}[⟨Anweisungen⟩]` – modifiziert die Parameter für die Linie unter dem Seitenfuß

Das obligatorische Argument *⟨Dicke⟩* bestimmt, wie stark die Linie gezeichnet wird. Das optionale Argument *⟨Länge⟩* akzeptiert die gleichen symbolischen Werte wie *⟨Breite⟩* bei `\setheadwidth`, also auch einen normalen Längenausdruck. Solange im Dokument dem optionalen Argument *⟨Länge⟩* kein Wert zugewiesen wurde, passt sich die entsprechende Linielänge automatisch der Breite des Kopfes bzw. des Fußes an.

Möchte man diesen Automatismus für die Länge einer Linie wieder restaurieren, dann nutzt man im Längenargument den Wert `auto`.

`\setkomafont{headtopline}` Mit dem optionalen Argument *⟨Anweisungen⟩* können zusätzliche Anweisungen definiert werden, die vor dem Zeichnen der jeweiligen Linie auszuführen sind. Das können beispielsweise Anweisungen sein, um die `\setkomafont{headsepline}` Farbe der Linie zu ändern. Bei Verwendung einer KOMA-Script-Klasse `\setkomafont{footsepline}` können diese Anweisungen auch über `\setkomafont` für eines der Elemente `\setkomafont{footbotline}` `headtopline`, `headsepline`, `footsepline`, `footbottomline` oder auch `footbotline` gesetzt und mit `\addtokomafont` erweitert werden. Die beiden Anweisungen `\setkomafont` und `\addtokomafont` sind in der KOMA-Script-Anleitung näher beschrieben.

Es sind auch symbolische Werte für *⟨Länge⟩* und *⟨Dicke⟩* möglich, beispielsweise `\setheadtopline[auto]{current}`, `\setheadtopline[auto]{}` oder `\setheadtopline[auto]{}{}`. Die hier am Befehl `\setheadtopline` illustrierten Argumente sind natürlich auch für die anderen drei Längenbefehle gültig.

Enthält das obligatorische Argument den Wert `current` oder wird leer gelassen, dann wird die Dicke der Linie nicht verändert. Das kann genutzt werden, wenn die Länge der Linie, aber nicht die Dicke modifiziert werden soll.

Wird das optionale Argument *⟨Anweisungen⟩* weggelassen, so bleiben eventuell zuvor gesetzte Anweisungen erhalten. Wird hingegen ein leeres Argument *⟨Anweisungen⟩* gesetzt, so werden eventuell zuvor gesetzte Anweisungen wieder gelöscht.

Example: Soll beispielsweise der Kopf mit einer kräftigen Linie von 2 pt darüber und einer normalen von 0,4 pt zwischen Kopf und Text abgesetzt werden, dann erfolgt das mit:

```
\setheadtopline{2pt}
\setheadsepline{.4pt}
```

Zusätzlich sind unbedingt die Optionen `headtopline` und `headsepline` vorzugsweise global im optionalen Argument von `\documentclass` zu setzen. Das Ergebnis könnte dann wie folgt aussehen.

KOMA-Script	3
Dieser Blindtext wird gerade von 130 Millionen Rezeptoren Ihrer Netzhaut erfasst. Die Zellen werden dadurch in einen Erregungszustand versetzt, der sich vom Sehnerv in den	
	Netzhaut (<i>Retina</i>)

Sollen diese Linien zusätzlich in roter Farbe gesetzt werden, dann sind die Anweisungen beispielsweise wie folgt zu ändern:

```
\setheadtopline{2pt}[\color{red}]
\setheadsepline{.4pt}[\color{red}]
```

In diesem und auch dem folgenden Beispiel wurde für die Aktivierung der Farbe die Syntax des `color`-Pakets verwendet, das dann natürlich auch geladen werden muss. `scrpage2` selbst bietet keine direkte Farbunterstützung. Damit ist jedes beliebige Farbunterstützungspaket verwendbar.

Mit einer KOMA-Script-Klasse kann alternativ

```
\setheadtopline{2pt}
\setheadsepline{.4pt}
\setkomafont{headtopline}{\color{red}}
\setkomafont{headsepline}{\color{red}}
```

verwendet werden.

Die automatische Anpassung an die Kopf- und Fußbreiten illustriert folgendes Beispiel, für das die Optionen `footbotline` und `footsepline` gesetzt sein sollten:

```
\setfootbotline{2pt}
\setfootsepline[text]{.4pt}
\setfootwidth[0pt]{textwithmarginpar}
```

Dieser Blindtext wird gerade von 130 Millionen Rezeptoren Ihrer Netzhaut erfasst. Die Zellen werden dadurch in einen Erregungszustand versetzt, der sich vom Sehnerv in den	
	Netzhaut (<i>Retina</i>)
KOMA-Script	3

Nun mag nicht jedem die Ausrichtung der Linie über der Fußzeile gefallen, sondern es wird in einem solchen Fall erwartet, dass sie wie der Kolummentitel linksbündig zum Text ist. Diese Einstellung kann nur global in Form einer Paketoption erfolgen und wird im folgenden Abschnitt 2.1.4 mit anderen Optionen beschrieben.

2.1.4 Optionen beim Laden des Paketes

Während bei den KOMA-Script-Klassen die Mehrzahl der Optionen auch noch nach dem Laden der Klasse mit `\KOMAOPTIONS` und `\KOMAOPTION` geändert werden kann, trifft dies für das Paket `scrpage2` nicht zu. Alle Optionen für dieses Paket müssen als globale Optionen, also im optionalen Argument von `\documentclass`, oder als Paketoptionen, also im optionalen Argument von `\usepackage`, angegeben werden.

<code>headinclude</code>	Diese Optionen sollten ab Version 2.3 des Pakets nicht mehr beispielsweise per optionalem Argument von <code>\usepackage</code> oder per <code>\PassOptionsToPackage</code> direkt an <code>scrpage2</code> übergeben werden. Lediglich aus Gründen der Kompatibilität sind sie noch in <code>scrpage2</code> deklariert und werden von diesem als <code>headinclude</code> , <code>headinclude=false</code> , <code>footinclude</code> und <code>footinclude=false</code> an das Paket <code>typearea</code> weitergereicht
<code>headexclude</code>	
<code>footinclude</code>	
<code>footexclude</code>	
<code>headtopline</code>	Eine Grundeinstellung für die Linien unter und über den Kopf- und Fußzeilen kann mit diesen Optionen vorgenommen werden. Diese Einstellungen gelten dann als Standard für alle mit <code>scrpage2</code> definierten Seitenstile. Wird eine von diesen Optionen verwendet, dann wird eine Linienstärke von 0,4 pt eingesetzt. Da es zum Seitenstil <code>scrheadings</code> einen entsprechenden plain-Stil gibt, kann mit den <code>plain...</code> -Optionen auch die entsprechende Linie des plain-Stils konfiguriert werden. Diese plain-Optionen wirken aber nur, wenn auch die korrespondierende Option ohne <code>plain</code> aktiviert wurde. Somit zeigt die Option <code>plainheadtopline</code> ohne <code>headtopline</code> keine Wirkung. Bei diesen Optionen ist zu beachten, dass der entsprechende Seitenteil in den Textbereich des Satzspiegels mit übernommen wird, wenn eine Linie aktiviert wurde. Wird also mittels <code>headsepline</code> die Trennlinie zwischen Kopf und Text aktiviert, dann wird automatisch mittels <code>typearea</code> der Satzspiegel so berechnet, dass der Seitenkopf Teil des Textblocks ist. Die Bedingungen für die Optionen des vorhergehenden Abschnitts gelten auch für diesen Automatismus. Das bedeutet, dass das Paket <code>typearea</code> nach <code>scrpage2</code> geladen werden muss, beziehungsweise, dass bei Verwendung einer KOMA-Script-Klasse die Optionen <code>headinclude</code> und <code>footinclude</code> explizit bei <code>\documentclass</code> gesetzt werden müssen, um Kopf- bzw. Fußzeile in den Textblock zu übernehmen.
<code>plainheadtopline</code>	
<code>headsepline</code>	
<code>plainheadsepline</code>	
<code>footsepline</code>	
<code>plainfootsepline</code>	
<code>footbotline</code>	
<code>plainfootbotline</code>	
<code>ilines</code>	Bei der Festlegung der Linienlängen kann es vorkommen, dass die Linie zwar die gewünschte Länge, aber nicht die erwünschte Ausrichtung hat,
<code>clines</code>	
<code>olines</code>	

da sie im Kopf- bzw. Fußbereich zentriert wird. Mit den hier vorgestellten Paketoptionen kann global für alle mit `scrpage2` definierten Seitenstile diese Vorgabe modifiziert werden. Dabei setzt `ilines` die Ausrichtung so, dass die Linien an den inneren Rand verschoben werden. Die Option `clines` verhält sich wie die Standardeinstellung und `olines` richtet am äußeren Rand aus.

Example: Hier gilt es, das Beispiel zu `\setfootsepline` auf Seite 33 mit dem folgenden zu vergleichen, um die Wirkung der Option `ilines` zu sehen.

```
\usepackage[ilines,footsepline,footbotline]
{scrpage2}
\setfootbotline{2pt}
\setfootsepline[text]{.4pt}
\setfootwidth[0pt]{textwithmarginpar}
```

Allein die Verwendung der Option `ilines` führt dabei zu der geänderten Ausgabe, die nachfolgend veranschaulicht wird:

Dieser Blindtext wird gerade von 130 Millionen Rezeptoren Ihrer Netzhaut erfasst. Die Zellen werden dadurch in einen Erregungs-	Netzhaut (<i>Retina</i>)
KOMA-Script	3

Die Trennlinie zwischen Text und Fuß wird bündig innen im Fußteil gesetzt und nicht wie bei der Standardeinstellung zentriert.

automark Diese Optionen bestimmen gleich zu Beginn des Dokuments, ob eine automatische Aktualisierung der Kolumnentitel erfolgt. Die Option **automark** schaltet die automatische Aktualisierung ein, **manualmark** deaktiviert sie. Ohne Verwendung einer der beiden Optionen bleibt die Einstellung erhalten, die beim Laden des Paketes gültig war.

Example: Sie laden das Paket `scrpage2` unmittelbar nach der Klasse `scrreprt` und ohne weitere Optionen. Dazu schreiben Sie:

```
\documentclass{scrreprt}
\usepackage{scrpage2}
```

Da bei `scrreprt` der Seitenstil `plain` voreingestellt ist, ist dies auch jetzt noch der Fall. Außerdem entspricht die Voreinstellung `plain` manuellen Kolumnentiteln. Wenn Sie also anschließend mit

```
\pagestyle{scrheadings}
```

auf den Seitenstil `scrheadings` umschalten, sind noch immer manuelle Kolumnentitel eingestellt.

Verwenden Sie stattdessen die Dokumentklasse `scrbook`, so ist nach

```
\documentclass{scrbook}
\usepackage{scrpage2}
```

der Seitenstil `headings` mit automatischen Kolumnentiteln aktiviert. Bei anschließender Umschaltung auf den Seitenstil `scrheadings` bleiben automatische Kolumnentitel eingeschaltet. Dabei werden dann weiterhin die Markierungsmakros von `scrbook` verwendet.

Verwenden Sie hingegen

```
\usepackage[automark]{scrpage2}
```

so wird unabhängig von der verwendeten Klasse auf automatische Kolumnentitel umgeschaltet, wobei die Markierungsmakros von `scrpage2` genutzt werden. Natürlich wirkt sich dies auf den Seitenstil `plain` von `scrreprt` nicht aus. Die Kolumnentitel werden erst sichtbar, wenn auf den Seitenstil `scrheadings` oder `useheadings` oder einen selbst definierten Seitenstil mit Kolumnentiteln umgeschaltet wird.

- | | |
|--|--|
| <code>autooneside</code> | Mit dieser Option wird das optionale Argument von <code>\automark</code> im einseitigen Satz automatisch ignoriert. Siehe hierzu auch die Erläuterung zum Befehl <code>\automark</code> in Abschnitt 2.1.2. |
| <code>komastyle</code>
<code>standardstyle</code> | <p>Diese Optionen bestimmen, wie die beiden vordefinierten Seitenstile <code>scrheadings</code> und <code>scrplain</code> gestaltet sind. Bei <code>komastyle</code> wird eine Definition vorgenommen, wie sie den KOMA-Script-Klassen entspricht. Bei den KOMA-Script-Klassen ist dies die Voreinstellung und kann somit auch für andere Klassen gesetzt werden.</p> <p>Die Option <code>standardstyle</code> definiert die beiden Seitenstile wie es von den Standardklassen erwartet wird. Außerdem wird hier automatisch <code>markuppercase</code> aktiviert, es sei denn, <code>markusedcase</code> wird ebenfalls als Option übergeben.</p> |

markuppercase Für die Funktionalität von `\automark` modifiziert `scrpage2` interne Befehle, die die Gliederungsbefehle benutzen, um die lebenden Kolumnentitel zu setzen. Da einige Klassen, im Gegensatz zu den KOMA-Script-Klassen, die Kolumnentitel in Großbuchstaben schreiben, muss `scrpage2` wissen, wie die genutzte Dokumentklasse die lebenden Kolumnentitel darstellt.

Die Option **markuppercase** zeigt `scrpage2`, dass die benutzte Klasse die Großschreibweise benutzt. Die Option **markusedcase** sollte angegeben werden, wenn die benutzte Dokumentklasse keine Großschreibweise verwendet. Die Optionen sind nicht geeignet, eine entsprechende Darstellung zu erzwingen. Es kann somit zu unerwünschten Effekten kommen, wenn die Angabe nicht dem Verhalten der Dokumentklasse entspricht.

nouppercase Wie in obiger Erklärung zu **markuppercase** und **markusedcase** bereits ausgeführt wurde, gibt es Klassen und auch Pakete, die beim Setzen der lebenden Kolumnentitel mit Hilfe einer der Anweisungen `\uppercase` oder `\MakeUppercase` den gesamten Eintrag in Großbuchstaben wandeln. Mit der Option **nouppercase** können diese beiden Anweisungen im Kopf und im Fuß außer Kraft gesetzt werden. Das gilt aber nur für Seitenstile, die mit Hilfe von `scrpage2` definiert werden. Dazu zählen auch **scrheadings** und der zugehörige plain-Seitenstil.

Die verwendete Methode ist äußerst brutal und kann dazu führen, dass auch erwünschte Änderungen von Klein- in Großbuchstaben unterbleiben. Da diese Fälle nicht sehr häufig sind, stellt **nouppercase** aber meist eine brauchbare Lösung dar.

Example: Sie verwenden die Standardklasse **book**, wollen aber, dass die lebenden Kolumnentitel nicht in Großbuchstaben, sondern in normaler gemischter Schreibweise gesetzt werden. Die Präambel Ihres Dokuments könnte dann wie folgt beginnen:

```
\documentclass{book}
\usepackage[nouppercase]{scrpage2}
\pagestyle{scrheadings}
```

Die Umschaltung auf den Seitenstil **scrheadings** ist notwendig, weil sonst der Seitenstil **headings** verwendet wird, der von der Option **nouppercase** nicht behandelt wird.

In einigen Fällen setzen nicht nur Klassen, sondern auch Pakete lebende Kolumnentitel in Großbuchstaben. Auch in diesen Fällen hilft **nouppercase** meist, um zu gemischter Schreibweise zurückzuschalten.

2.2 Seitenstile selbst gestalten

2.2.1 Die Anwenderschnittstelle

\deftripstyle Nun möchte man ja nicht immer an die vorgegebenen Seitenstile gebunden

sein, sondern auch seiner Kreativität freien Lauf lassen. Manchmal ist man auch dazu gezwungen, weil eine bestimmte *Corporate Identity* einer Firma es verlangt. Der einfachste Weg damit umzugehen ist

```
\deftripstyle{<Name>}
      [<LA>] [<LI>]
      {<KI>}{<KM>}{<KA>}
      {<FI>}{<FM>}{<FA>}
```

Die einzelnen Felder haben folgende Bedeutung:

- $\langle Name \rangle$ – die Bezeichnung des Seitenstils, die dann bei der Aktivierung mit `\pagestyle{<Name>}` oder `\thispagestyle{<Name>}` verwendet wird
- $\langle LA \rangle$ – die Dicke der äußeren Linien, d. h. der Linien über der Kopfzeile und unter der Fußzeile (optional)
- $\langle LI \rangle$ – die Dicke der inneren Linie, d. h. der Linien die Kopf und Fuß vom Textkörper trennen (optional)
- $\langle KI \rangle$ – Inhalt des Feldes im Kopf innenseitig oder bei einseitigem Layout links
- $\langle KM \rangle$ – Inhalt des Feldes im Kopf zentriert
- $\langle KA \rangle$ – Inhalt des Feldes im Kopf außenseitig oder bei einseitigem Layout rechts
- $\langle FI \rangle$ – Inhalt des Feldes im Fuß innenseitig oder bei einseitigem Layout links
- $\langle FM \rangle$ – Inhalt des Feldes im Fuß zentriert
- $\langle FA \rangle$ – Inhalt des Feldes im Fuß außenseitig oder bei einseitigem Layout rechts

Der Befehl `\deftripstyle` stellt sicherlich die einfachste Möglichkeit dar, Seitenstile zu definieren. Leider sind damit auch Einschränkungen verbunden, da in einem Seitenbereich mit einem durch `\deftripstyle` deklarierten Seitenstil keine Änderung der Kopf- und Fußlinien erfolgen kann.

Example: Vorgegeben sei ein doppelseitiges Layout, bei dem die Kolumnentitel innen erscheinen sollen. Weiterhin soll der Dokumenttitel, in diesem Fall kurz „Bericht“, an den Außenrand in den Kopf, die Seitenzahl soll zentriert in den Fuß.

```
\deftripstyle{DerBericht}%
      {\headmark}{\Bericht}%
      {}{\pagemark}{}
```

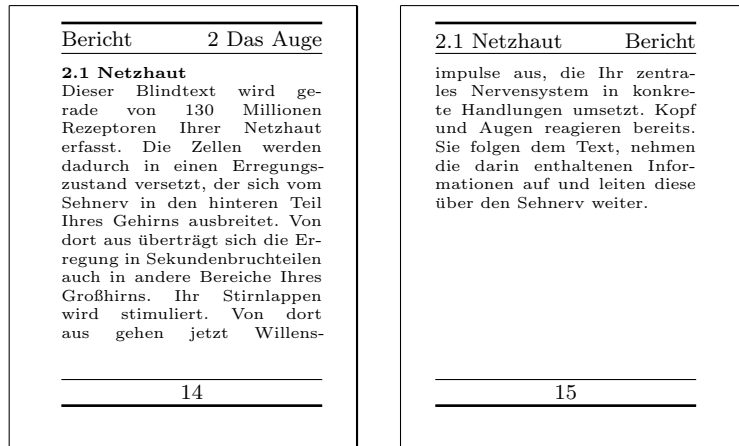


Abbildung 4: Beispiel für einen selbst definierten, von Linien dominierten Seitenstil mit einem statischen und einem lebenden Kolummentitel im Kopf und der Seitenzahl in der Mitte des Fußes.

Sollen weiterhin die Linien über dem Kopf und unter dem Fuß mit 2 pt erscheinen und der ganze Textkörper mit dünnen Linien von 0,4 pt von Kopf und Fuß abgesetzt werden, dann erweitert man vorherige Definition.

```
\deftripstyle{DerBericht}[2pt][.4pt]%
      {\headmark}{\Bericht}%
      {}{\pagemark}{}
```

Das Ergebnis ist in Abbildung 4 zu sehen.

2.2.2 Die Expertenschnittstelle

Einfache Seitenstile, wie sie mit `\deftripstyle` deklariert werden können, sind erfahrungsgemäß selten. Entweder verlangt ein Professor, dass die Diplomarbeit so aussieht wie seine eigene – und wer will ihm da *ernsthaft* widersprechen – oder eine Firma möchte, dass die halbe Finanzbuchhaltung im Seitenfuß auftaucht. Alles kein Problem, denn es gibt noch:

```
\defpagestyle{<Name>}{<Kopfdefinition>}{<Fußdefinition>}
\newpagestyle{<Name>}{<Kopfdefinition>}{<Fußdefinition>}
\renewpagestyle{<Name>}{<Kopfdefinition>}{<Fußdefinition>}
\providepagestyle{<Name>}{<Kopfdefinition>}{<Fußdefinition>}
```

Dies sind die Befehle, die die volle Kontrolle über die Gestaltung eines Seitenstils ermöglichen. Der Aufbau ist bei allen vier Definitionen gleich, sie unterscheiden sich nur hinsichtlich der Wirkungsweise.

- `\defpagestyle` – definiert einen neuen Seitenstil. Existiert bereits einer mit diesem Namen, wird dieser überschrieben.
- `\newpagestyle` – definiert einen neuen Seitenstil. Wenn schon einer mit diesem Namen existiert, wird ein Fehler ausgegeben.
- `\renewpagestyle` – definiert einen bestehenden Seitenstil um. Wenn noch keiner mit diesem Namen existiert, wird ein Fehler ausgegeben.
- `\providepagestyle` – definiert einen neuen Seitenstil nur dann, wenn dieser vorher noch nicht existiert.

Am Beispiel von `\defpagestyle` soll die Syntax der Definitionen im Folgenden erläutert werden.

- $\langle Name \rangle$ – die Bezeichnung des Seitenstils
- $\langle Kopfdefinition \rangle$ – die Deklaration des Seitenkopfes bestehend aus fünf Teilen, wobei die in runden Klammern stehenden Angaben optional sind:
 $(\langle OLL, OLD \rangle) \{ \langle GS \rangle \} \{ \langle US \rangle \} \{ \langle ES \rangle \} (\langle ULL, ULD \rangle)$
- $\langle Fußdefinition \rangle$ – die Deklaration des Seitenfußes bestehend aus fünf Teilen, wobei die in runden Klammern stehenden Angaben optional sind:
 $(\langle OLL, OLD \rangle) \{ \langle GS \rangle \} \{ \langle US \rangle \} \{ \langle ES \rangle \} (\langle ULL, ULD \rangle)$

Wie zu sehen ist, haben Kopf- und Fußdefinition identischen Aufbau. Die einzelnen Parameter haben folgende Bedeutung:

- $\langle OLL \rangle$ – obere Linienlänge: Kopf = außen, Fuß = Trennlinie
- $\langle OLD \rangle$ – obere Liniendicke
- $\langle GS \rangle$ – Definition für die *gerade* Seite
- $\langle US \rangle$ – Definition für die *ungerade* Seite
- $\langle ES \rangle$ – Definition für *einseitiges* Layout
- $\langle ULL \rangle$ – untere Linienlänge Kopf = Trennlinie, Fuß = außen
- $\langle ULD \rangle$ – untere Liniendicke

Werden die optionalen Linienargumente nicht gesetzt, dann bleibt das Verhalten weiterhin durch die in Abschnitt 2.1.3 vorgestellten Linienbefehle konfigurierbar.

Die drei Felder $\langle GS \rangle$, $\langle US \rangle$ und $\langle ES \rangle$ entsprechen Boxen, die die Breite des Kopf- bzw. Fußteils haben. Die entsprechenden Definitionen erscheinen in diesen Boxen linksbündig. Um somit etwas links- *und* rechtsseitig in den Boxen zu platzieren, kann der Zwischenraum mit `\hfill` gestreckt werden:

```
{\headmark\hfill\pagemark}
```

Um zusätzlich etwas zentriert erscheinen zu lassen, ist eine erweiterte Definition notwendig. Die Befehle `\rlap` und `\llap` setzen die übergebenen Argumente. Für L^AT_EX erscheint es aber so, dass diese Texte eine Breite von Null haben. Nur so erscheint der mittlere Text auch wirklich zentriert.

```
{\rlap{\headmark}\hfill zentriert\hfill\llap{\pagemark}}
```

Example: Angenommen es wird die Dokumentklasse `scrbook` genutzt. Damit liegt ein zweiseitiges Layout vor. Für das Paket `scrpage2` wird festgelegt, dass mit automatisch aktualisierten Kolumnentiteln gearbeitet wird und dass im Seitenstil `scrheadings` eine Trennlinie zwischen Kopf und Text gezogen wird.

```
\documentclass{scrbook}
\usepackage[automark,headsepline]{scrpage2}
```

Mit Hilfe der Expertenschnittstelle werden zwei Seitenstile definiert. Der erste legt keine Linienargumente fest, im zweiten wird die Linie über dem Kopf mit einer Dicke von 1 pt und die Linie unter dem Kopf mit 0 pt festgelegt.

```
\defpagestyle{ohneLinien}{%
  {\Beispiel\hfill\headmark}
  {\headmark\hfill ohne Linien}
  {\rlap{\Beispiel}\hfill\headmark\hfill%
   \llap{ohne Linien}}
}%
{\pagemark\hfill}
{\hfill\pagemark}
{\hfill\pagemark\hfill}
}
\defpagestyle{mitLinien}{%
  (\textwidth,1pt)
  {\mit Linien\hfill\headmark}
  {\headmark\hfill mit Linien}
  {\rlap{\KOMAScript}\hfill \headmark\hfill%
   \llap{mit Linien}}
  (0pt,0pt)
}%
(\textwidth,.4pt)
```

```

{\pagemark\hfill}
{\hfill\pagemark}
{\hfill\pagemark\hfill}
(\textwidth,1pt)
}

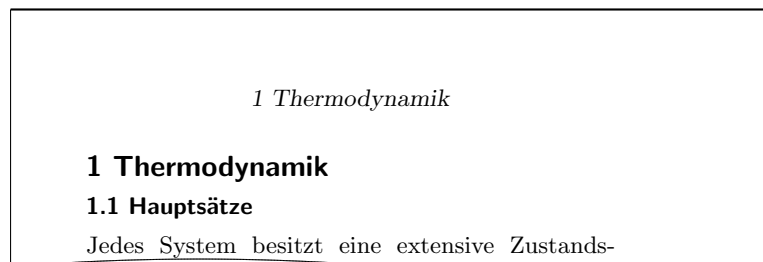
```

Gleich zu Beginn wird der Seitenstil `scrheadings` aktiviert. Mit `\chapter` wird ein neues Kapitel begonnen. Weiterhin wird automatisch durch `\chapter` der Seitenstil für diese Seite auf `plain` gesetzt. Das folgende `\chead` zeigt, wie durch Modifikation des `plain`-Stils ein Kolumnentitel erzeugt werden kann. Grundsätzlich sollte jedoch davon Abstand genommen werden, da sonst der Markierungscharakter der `plain`-Seite verloren geht. Es ist wichtiger anzuzeigen, dass hier ein neues Kapitel beginnt, als dass ein Abschnitt dieser Seite einen bestimmten Titel trägt.

```

\begin{document}
\pagestyle{scrheadings}
\chapter{Thermodynamik}
\chead[\leftmark]{}
\section{Hauptsätze}
Jedes System besitzt eine extensive Zustandsgröße
Energie. Sie ist in einem abgeschlossenen System
konstant.

```



Nach dem Seitenwechsel ist der Seitenstil `scrheadings` aktiv, und somit auch die Trennlinie aus den Paketoptionen sichtbar.

Es existiert eine Zustandsgröße, genannt die Entropie eines Systems, deren zeitliche Änderung sich aus Entropieströmung und Entropieerzeugung zusammensetzt.

1 Thermodynamik

Es existiert eine Zustandsgröße, genannt die Entropie eines Systems, deren zeitliche Änderung sich aus Entropieströmung und Entropie-

Wiederum nach einem Seitenwechsel wird auf manuelle Kolumnentitel gewechselt und der Seitenstil `ohneLinien` aktiviert. Da keine Linienargumente bei der Definition dieses Stils genutzt wurden, wird die Standard-Linienkonfiguration verwendet. Diese zeichnet hier eine Linie zwischen Kopf und Text, da `headsepline` als Argument für `scrpage2` angegeben wurde.

```
\manualmark
\pagestyle{ohneLinien}
\section{Exergie und Anergie}
\markright{Energieumwandlung}
Man bezeichnet die bei der Einstellung des
Gleichgewichts mit der Umgebung maximal gewinnbare
Arbeit als Exergie.
```

Energieumwandlung *ohne Linien*

1.2 Exergie und Anergie

Man bezeichnet die bei der Einstellung des Gleichgewichts mit der Umgebung maximal

Nach dem Wechsel auf die folgende linke Seite wird der Seitenstil `mitLinien` aktiviert. Die Linieneinstellungen werden hier nun angewendet und entsprechend der Definition dargestellt.

```
\pagestyle{mitLinien}
\renewcommand{\headfont}{\itshape\bfseries}
Den nicht in Exergie umwandelbaren Anteil einer
Energie nennt man Anergie  $\text{Var}\{B\}$ .

$$B = U + T (S_1 - S_u) - p (V_1 - V_u)$$

\end{document}
```

Den nicht in Exergie umwandelbaren Anteil einer Energie nennt man Anergie B .

$$B = U + T(S_1 - S_u) - p(V_1 - V_u)$$

2.2.3 Seitenstile verwalten

Bei längerer Arbeit mit verschiedenen Seitenstilen wird sich, je nach Geschmack und Aufgabenstellung, ein fester Satz an benutzten Stilen etablieren. Um nicht bei jedem neuen Projekt eine große Kopieraktion von den Daten eines Projekts zum neuen Projekt starten zu müssen, liest `scrpage2` am Ende seiner Initialisierungsphase die Datei `scrpage.cfg` ein. In dieser Datei können dann Seitenstile definiert sein, die viele Projekte gemeinsam nutzen können.

3 Implementation

Currently this package doesn't share any code but the KOMA-Script logo with other packages. So all the code of this package but the implementation of the KOMA-Script logo is at this file and there is no more code at this file.

```
1 \scrpage2
2 \PackageWarning{scrpage2}{usage of obsolete package!\MessageBreak
3   Package 'scrpage2' is obsolete.\MessageBreak
4   You should not longer use package 'scrpage2'!\MessageBreak
5   You should replace usage of package 'scrpage2'\MessageBreak
6   by 'scrlayer-scrpage'}
```

3.1 Options

```
headinclude scrpage2 has known some options. They are similar to the old KOMA-
headexclude Script classes. To work with and without the KOMA-Script classes their
footinclude definitions are not quite simple.
footexclude 7 \newcommand*{\@Pass@Obsolete@Option@To@Typearea}[1]{%
8   \@expandtwoargs\in{,\CurrentOption,}{,\@curroptions,}%
9   \ifin@
10    \PackageWarningNoLine{scrpage2}{%
11      deprecated passing option '\CurrentOption' to scrpage2.\MessageBreak
12      If you are using package typearea or\MessageBreak
13      a KOMA-Script class since version 3.00\MessageBreak
14      you may use '\string\KOMAoptions{#1}' and\MessageBreak
```

```

15     remove passing ‘\CurrentOption’ to scrpage2%
16 }%
17 \begingroup\expandafter\expandafter\expandafter\endgroup
18 \expandafter\ifx\csname KOMAoptions\endcsname\relax
19   \PassOptionsToPackage{#1}{typearea}%
20 \else
21   \KOMAOptions{#1}\csname relcalctypearea\endcsname
22 \fi
23 \fi
24 }
25 \DeclareOption{headinclude}{%
26   \@Pass@Obsolete@Option@To@Typearea{headinclude}%
27 }
28 \DeclareOption{headexclude}{%
29   \@Pass@Obsolete@Option@To@Typearea{headinclude=false}%
30 }
31 \DeclareOption{footinclude}{%
32   \@Pass@Obsolete@Option@To@Typearea{footinclude}%
33 }
34 \DeclareOption{footexclude}{%
35   \@Pass@Obsolete@Option@To@Typearea{footinclude=false}%
36 }

```

We do so, because we want so set `headinclude`, `footinclude` by default at old version (see below)!

We have additional options designed for the additional interface.

`mpinclude` There are documents where the head and foot should span not only the text
`mpexclude` but also the marginpar. We have an option to do this at initialization. This
option corresponds with the option of package `typearea` and the classes.

```

\if@mincl
\@mincltrue 37 \ifundefined{@mincltrue}{\newif\if@mincl}{\fi}
\@minclfalse 38 \DeclareOption{mpinclude}{\@mincltrue}
39 \DeclareOption{mpexclude}{\@minclfalse}

```

Other new options redefine macros, so we have to define them first.

`\scr@headabove@linethickness` These macros define the thickness of the line above and below the head
`\scr@headbelow@linethickness` and the foot.
`\scr@footabove@linethickness` 40 \newcommand*{\scr@headabove@linethickness}{0pt}
`\scr@footbelow@linethickness` 41 \newcommand*{\scr@headbelow@linethickness}{0pt}
42 \newcommand*{\scr@footabove@linethickness}{0pt}
43 \newcommand*{\scr@footbelow@linethickness}{0pt}

`\scr@headabove@linelength` Because we want to have the “line definitions” at one place, we also define
`\scr@headbelow@linelength` the length of the lines here. We have own length at to be used at pagestyle
`\scr@footabove@linelength` `scrplain`.
`\scr@footbelow@linelength`
`scrplain@headabove@linelength`
`scrplain@headbelow@linelength`
`scrplain@footabove@linelength`
`scrplain@footbelow@linelength`

```

44 \newcommand*{\scr@headabove@linelength}{\@headwidth}
45 \newcommand*{\scr@headbelow@linelength}{\@headwidth}
46 \newcommand*{\scr@footabove@linelength}{\@footwidth}
47 \newcommand*{\scr@footbelow@linelength}{\@footwidth}
48 \newcommand*{\scrplain@headabove@linelength}{0pt}
49 \newcommand*{\scrplain@headbelow@linelength}{0pt}
50 \newcommand*{\scrplain@footabove@linelength}{0pt}
51 \newcommand*{\scrplain@footbelow@linelength}{0pt}

headtopline The thickness can be set by four simple options:
headsepline 52 \DeclareOption{headtopline}{%
footsepline 53 \renewcommand*{\scr@headabove@linethickness}{.4pt}
footbotline 54 \PassOptionsToPackage{headinclude}{typearea}}
55 \DeclareOption{headsepline}{%
plainheadtopline 56 \renewcommand*{\scr@headbelow@linethickness}{.4pt}
plainheadsepline 57 \PassOptionsToPackage{headinclude}{typearea}}
plainfootsepline 58 \DeclareOption{footsepline}{%
plainfootbotline 59 \renewcommand*{\scr@footabove@linethickness}{.4pt}
60 \PassOptionsToPackage{footinclude}{typearea}}
61 \DeclareOption{footbotline}{
62 \renewcommand*{\scr@footbelow@linethickness}{.4pt}
63 \PassOptionsToPackage{footinclude}{typearea}}
64 \DeclareOption{plainheadtopline}{%
65 \renewcommand*{\scrplain@headabove@linelength}{\scr@headabove@linelength}
66 \PassOptionsToPackage{headinclude}{typearea}}
67 \DeclareOption{plainheadsepline}{%
68 \renewcommand*{\scrplain@headbelow@linelength}{\scr@headbelow@linelength}
69 \PassOptionsToPackage{headinclude}{typearea}}
70 \DeclareOption{plainfootsepline}{%
71 \renewcommand*{\scrplain@footabove@linelength}{\scr@footabove@linelength}
72 \PassOptionsToPackage{footinclude}{typearea}}
73 \DeclareOption{plainfootbotline}{
74 \renewcommand*{\scrplain@footbelow@linelength}{\scr@footbelow@linelength}
75 \PassOptionsToPackage{footinclude}{typearea}}

\hfline@adjust At Version 2 lines may be inner adjusted (0), centered (1) or outer adjusted
ilines (2):
clines 76 \newcommand*{\hfline@adjust}{1}
olines 77 \DeclareOption{ilines}{\renewcommand*{\hfline@adjust}{0}}
78 \DeclareOption{clines}{\renewcommand*{\hfline@adjust}{1}}
79 \DeclareOption{olines}{\renewcommand*{\hfline@adjust}{2}}

automark At version 2 we want to distinguish between running headline and static
manualmark headline not only using different pagestyles. We do this with options and
commands. But here only the options are defined.
80 \DeclareOption{automark}{%
81 \AtEndOfPackage{\if@twoside%
82 \if@chapter\automark[section]{chapter}%
83 \else\automark[subsection]{section}\fi%

```

```

84     \else%
85         \if@chapter\automark{chapter}\else\automark{section}\fi%
86     \fi}}
87 \DeclareOption{manualmark}{\AtEndOfPackage{\manualmark}}

autoonesize  If the optional argument of \automark should be ignored at inside mode,
\if@autooneside use this option:
88 \newif\if@autooneside\@autoonesidefalse
89 \DeclareOption{autooneside}{\@autoonesidetrue}

markuppercase  For use of running headlines there are also options to say “class uses upper
markusedcase   case at the automatic marks” or “class use same case at the automatic
                marks like used writing the heading”.
90 \DeclareOption{markuppercase}{\let\MakeMarkcase\MakeUppercase}
91 \DeclareOption{markusedcase}{\let\MakeMarkcase\@empty}

\scr@nouppercase  Sometimes scrpage2 may be used with a class, which uses \uppercase or
nouppercase       \MakeUppercase at \markboth or \markright. But the author does not
                want upper case at the headlines. So we define an option, that simply deak-
                tivates \uppercase and \MakeUppercase at all heads and foots, defined by
                scrpage2.
92 \newcommand*{\scr@nouppercase}{}
93 \DeclareOption{nouppercase}{\renewcommand*{\scr@nouppercase}{%
94     \let\uppercase\relax\let\MakeUppercase\relax
95     \expandafter\let\csname MakeUppercase \endcsname\relax}
96 \ifx\MakeMarkcase\MakeUppercase
97     \PackageWarning{scrpage2}{%
98         Option 'markuppercase' will be ignored.\MessageBreak%
99         You shouldn't use option 'markuppercase' together\MessageBreak%
100        with option 'nouppercase'}
101     \let\MakeMarkcase\@empty
102 \fi}

komastyle  At version 2 there is an predefined pagestyle “scrheadings”. With the two
standardstyle options you may decide, whether it has to behave like “headings” at the
                standard classes or the KOMA-Script classes. The macro saves the name
                of the style.
103 \DeclareOption{komastyle}{\def\scrheadings@style{koma}
104 \@ifundefined{MakeMarkcase}{\let\MakeMarkcase\relax}{}}
105 \DeclareOption{standardstyle}{\def\scrheadings@style{standard}
106 \@ifundefined{MakeMarkcase}{\let\MakeMarkcase\MakeUppercase}{}}

                We don't need default options.
107 \ExecuteOptions{}

                But you may set other Options:
108 \ProcessOptions\relax

```

3.2 Some Initialisation

For easier handling of the differences between article-, report- and book-classes we define some more switches.

```
\if@chapter First distinguish between article and others.
109 \newif\if@chapter
110 \begingroup\expandafter\expandafter\expandafter\endgroup
111 \expandafter\ifx\csname chapter\endcsname\relax
112   \@chapterfalse
113 \else
114   \@chaptertrue
115 \fi

\if@mainmatter Next distinguish between matter- and nomatter-classes.
116 \def\@tempa{\newif\if@mainmatter\@mainmattertrue}
117 \begingroup\expandafter\expandafter\expandafter\endgroup
118 \expandafter\ifx\csname mainmatter\endcsname\relax
119 \else
120   \@ifundefined{@mainmattertrue}{%
121     \PackageWarningNoLine{scrpage2}{%
122       \string\mainmatter\space defined without
123       \expandafter\string\csname if@mainmatter\endcsname!\MessageBreak
124       This switch is obligatory for scrpage2 to omit\MessageBreak
125       running head chapter numbers in front and\MessageBreak
126       back matter and will be defined now%
127     }%
128   }{%
129     \let\@tempa\relax
130   }%
131 \fi
132 \@tempa
```

3.3 Predefinitions

There are some commands, you can use at pagestyle-definition.

```
\headmark This macro is \rightmark or \leftmark. But outside pagestyle-definition
it's nothing.
133 \let\headmark\relax
```

```
\pagemark This macro is the number of the page at the pagenumberfont:
134 \DeclareRobustCommand\pagemark{{\pnumfont\thepage}}
```

3.4 Expert-Pagestyle-Definition-Interface

The pagestyle-definition-interface for experts is not as easy as the simple interface we'll define later. But it's more flexible. We'll later use it to define the pagestyle-definition-interface for users.

```

\if@automark We need this switch to decide to switch on or off \@mkboth at the pagestyles.
\@automarktrue 135 \newif\if@automark\@automarkfalse
\@automarkfalse
\defpagestyle First we define the simple definition-interface. There's no test, if the
pagestyle's defined twice.
First there has to be the definition of head and foot.
136 \def\defpagestyle{%
We have to decide, whether it is a my-version or not:
137 \@ifstar
138 {\@defpagestyle[-]}%
139 {\@defpagestyle[+]}%

\@defpagestyle Now the we can define the head and the foot. Before version 2.5 we have
distinguished one- and two-side definitions while loading the package and
have had all the definitions of \@mkboth, \chaptermark, \sectionmark,
and \subsectionmark at the definition of the page style. Since version 2.5
the difference of one- and two-side is made at the definition of the heads and
foots themself and \chaptermark, \sectionmark, and \subsectionmark
are defined by \automark and \manualmark. So the code is much easier.
140 \def\@defpagestyle[#1]#2#3#4{%
141 \expandafter\def\csname ps@#2\endcsname{%
142 \if@automark\let\@mkboth\markboth\else\let\@mkboth\@gobbletwo\fi
143 \def@head#3
144 \def@foot#4
145 }%
146 }

\def@head This was real simple, but we already have to do the real definition of the
\def@@head heads and foots. First the head:
147 \def\def@head{%
148 \kernel@ifnextchar(%)
149 {\def@@head}%
150 {\def@@head(\scr@headabove@linelength,%
151 \scr@headabove@linethickness)}}
152 \def\def@@head(#1,#2)#3#4#5{%
153 \kernel@ifnextchar(%)
154 {\def@@@head({#1},{#2}){#3}{#4}{#5}}%
155 {\def@@@head({#1},{#2}){#3}{#4}{#5}%
156 (\scr@headbelow@linelength,%
157 \scr@headbelow@linethickness)}}
158 \def\def@@@head(#1,#2)#3#4#5(#6,#7){%
159 \def\@evenhead{%
160 \if@twoside
161 \let\headmark\leftmark%
162 \hss\hskip\@evenheadshift\vbox{\hsize=\@headwidth\relax%
163 \hf@rule{#1}{#2}{\@headwidth}{headabove}}%
164 \vskip#2

```

```

165         \vskip\baselineskip
166         \hb@xt@\@headwidth{%
167             \scr@nouppercase
168             \headfont\strut #3}}%
169         \hf@rule{#6}{#7}{\@headwidth}{headbelow}%
170     }\hskip\@oddheadshift\hss
171 \fi
172 }%
173 \def\@oddhead{\let\headmark\rightmark
174     \hss\hskip\@oddheadshift\vbox{\hsize=\@headwidth\relax
175         \hf@rule{#1}{#2}{\@headwidth}{headabove}%
176         \vskip#2%
177         \vskip\baselineskip
178         \hb@xt@\@headwidth{%
179             \scr@nouppercase
180             \headfont\strut \if@twoside #4\else #5\fi}}%
181         \hf@rule{#6}{#7}{\@headwidth}{headbelow}%
182     }\hskip\@evenheadshift\hss}%
183 }

\def@foot And last twosided and onesided foot:
\def@@foot 184 \def\def@foot{%
\def@@@foot 185     \kernel@ifnextchar(%)
186     {\def@@foot}
187     {\def@@foot(\scr@footabove@linelength,%
188         \scr@footabove@linethickness)}}
189 \def\def@@foot(#1,#2)#3#4#5{%
190     \kernel@ifnextchar(%)
191     {\def@@@foot({#1},{#2}){#3}{#4}{#5}}%
192     {\def@@@foot({#1},{#2}){#3}{#4}{#5}%
193         (\scr@footbelow@linelength,%
194         \scr@footbelow@linethickness)}}
195 \def\def@@@foot(#1,#2)#3#4#5(#6,#7){%
196     \def\@evenfoot{%
197         \if@twoside
198             \let\headmark\leftmark
199             \hss\hskip\@evenfootshift\vbox{\hsize=\@footwidth\relax
200                 \topfoot@rule{#1}{#2}\hb@xt@\@footwidth{%
201                     \scr@nouppercase
202                     \headfont\footfont\strut #3}}%
203                 \botfoot@rule{#6}{#7}}\hskip\@oddfootshift\hss
204         \fi
205     }%
206     \def\@oddfoot{\let\headmark\rightmark
207         \hss\hskip\@oddfootshift\vbox{\hsize=\@footwidth\relax
208             \topfoot@rule{#1}{#2}\hb@xt@\@footwidth{%
209                 \scr@nouppercase
210                 \headfont\footfont\strut \if@twoside #4\else #5\fi}}%
211             \botfoot@rule{#6}{#7}}\hskip\@evenfootshift\hss}%

```

212 }

We now define the rules used at head and foot.

\topfoot@rule

```
213 \newcommand\topfoot@rule[2]{%
214   \@tempdima\baselineskip\advance\@tempdima by-.7\normalbaselineskip
215   \advance\@tempdima by -#2
216   \vskip\@tempdima\hf@rule{#1}{#2}{\@footwidth}{footabove}%
217   \vskip-\@tempdima}
```

\botfoot@rule

```
218 \newcommand\botfoot@rule[2]{%
219   \@tempdima-\baselineskip\advance\@tempdima by .3\normalbaselineskip
220   \advance\@tempdima by #2
221   \vskip\@tempdima\hf@rule{#1}{#2}{\@footwidth}{footbelow}%
222 }
```

\hf@rule

```
223 \newcommand*\hf@rule[4]{%
224   \setlength{\@tempdimb}{#1}
225   \begingroup
226     \csname scr@pre@#4@line\endcsname
227     \if@twoside
228       \ifodd\c@page
229         \edef\@tempa{\hfline@adjust}%
230       \else
231         \@tempcnta=2\advance\@tempcnta -\hfline@adjust
232         \edef\@tempa{\the\@tempcnta}%
233       \fi
234     \else
235       \edef\@tempa{\hfline@adjust}%
236     \fi
237     \ifnum\@tempa=1
238       \setlength{\@tempdimb}{.5\@tempdimb}%
239     \fi
240     \hb@xt@#3{%
241       \ifnum\@tempa>0
242         \hfill%
243         \llap{\vrule\@depth#2\@height\z@\@width\@tempdimb}%
244       \fi
245       \ifnum\@tempa<2
246         \rlap{\vrule\@depth#2\@height\z@\@width\@tempdimb}%
247         \hfill%
248       \fi
249     }%
250   \endgroup
251 }
```

```

\scr@pre@headabove@line
\scr@fnt@headtopline 252 \newcommand*{\scr@pre@headabove@line}{}
\scr@pre@headbelow@line 253 \newcommand*{\scr@fnt@headtopline}{\scr@pre@headabove@line}
\scr@fnt@headsepline 254 \newcommand*{\scr@pre@headbelow@line}{}
\scr@pre@footabove@line 255 \newcommand*{\scr@fnt@headsepline}{\scr@pre@headbelow@line}
\scr@fnt@footsepline 256 \newcommand*{\scr@pre@footabove@line}{}
\scr@pre@footbelow@line 257 \newcommand*{\scr@fnt@footsepline}{\scr@pre@footabove@line}
\scr@fnt@footbottomrule 258 \newcommand*{\scr@pre@footbelow@line}{}
\scr@fnt@instead@footbotline 259 \newcommand*{\scr@fnt@footbottomrule}{\scr@pre@footbelow@line}
\scr@fnt@instead@footbotline 260 \newcommand*{\scr@fnt@instead@footbotline}{\scr@pre@footbelow@line}

```

Using these definitions, we can define the testing macros.

`\newpagestyle` First the simple star/starless-selection:

```
261 \def\newpagestyle{%
```

We have to decide, whether it is a my-version or not:

```

262 \@ifstar
263 {\@newpagestyle[-]}%
264 {\@newpagestyle[+]}

```

`\@newpagestyle` There we have to distinguish, whether the pagestyle is already defined or not.

```

265 \def\@newpagestyle[#1]#2#3#4{%
266 \expandafter\ifx\csname ps@#2\endcsname\relax

```

Then we can use the already defined `\@defpagestyle`.

```

267 \@defpagestyle[#1]{#2}{#3}{#4}%
268 \else
269 \PackageError
270 {scrpage2}%
271 {Your command was ignored}%
272 {There is already a pagestyle named ‘‘#1’’.\MessageBreak%
273 Use \protect\defpagestyle\space, \protect\renewpagestyle or
274 \protect\providepagestyle.}%
275 \fi
276 }

```

`\renewpagestyle` First the simple star/starless-selection:

```
277 \def\renewpagestyle{%
```

We have to decide, whether it is a my-version or not:

```

278 \@ifstar
279 {\@renewpagestyle[-]}%
280 {\@renewpagestyle[+]}

```

`\@renewpagestyle` There we have to distinguish, whether the pagestyle is already defined or not.

```
281 \def\@renewpagestyle[#1]#2#3#4{%
```

```

282 \begingroup\expandafter\expandafter\expandafter\endgroup
283 \expandafter\ifx\csname ps@#2\endcsname\relax
284   \PackageError
285     {scrpage2}%
286     {Your command was ignored}%
287     {There is no pagestyle named ‘‘#1’’. \MessageBreak%
288       Use \protect\defpagestyle, \protect\newpagestyle or
289       \protect\providepagestyle.}%
290 \else
  Then we can use the already defined \@defpagestyle.
291   \@defpagestyle[#1]{#2}{#3}{#4}%
292 \fi
293 }

```

`\providepagestyle` First the simple star/starless-selection:

```

294 \def\providepagestyle{%
  We have to decide, whether it is a my-version or not:
295   \@ifstar
296   {\@providepagestyle[-]}%
297   {\@providepagestyle[+]}%

```

`\@providepagestyle` There we have to distinguish, whether the pagestyle is already defined or not.

```

298 \def\@providepagestyle[#1]#2#3#4{%
299   \expandafter\ifx\csname ps@#2\endcsname\relax
    Then we can use the already defined \@defpagestyle.
300     \@defpagestyle[#1]{#2}{#3}{#4}%
301   \else
302     \PackageInfo
303       {scrpage2}%
304       {\protect\providepagestyle\protect{#1\protect} ignored.}%
305   \fi
306 }

```

3.5 User-Pagestyle-Definition-Interface

The pagestyle-definition-interface for users is much easier than the expert-version. Using the expert-macros, the definition is easy, too.

`\deftripstyle`

At version 1 we define the selection of star- or starless-version first:

```

307 \def\deftripstyle{%
308   \@ifstar
309   {\@deftripstyle[-]}%
310   {\@deftripstyle[+]}%

```

Next we check, if there is an optional argument:

```
\@deftripstyle
311 \def\@deftripstyle[#1]#2{%
312   \kernel@ifnextchar[%]
313   {\@@deftripstyle[#1]{#2}}%
314   {\@@deftripstyle[#1]{#2}[\z@]}}
```

or perhaps two optional arguments:

```
\@@deftripstyle
315 \def\@@deftripstyle[#1]#2[#3]{%
```

Ok, now we have the problem, that there can be only 9 arguments. So we have to distinguish the star-versions by a macro:

```
316 \def\@tempa{#1}%
317 \kernel@ifnextchar[%]
318 {\@@@deftripstyle#2[#3]}%
319 {\@@@deftripstyle#2[\z@] [#3]}
```

Now the main-definition:

```
\@@@deftripstyle
320 \def\@@@deftripstyle#1[#2] [#3]#4#5#6#7#8#9{%
321   \def\@tempb{+}%
322   \ifx\@tempa\@tempb
323     \defpagestyle{#1}%
324     {(\@headwidth,#2)%
325      {\rlap{#6}\hfill{#5}\hfill\llap{#4}}%
326      {\rlap{#4}\hfill{#5}\hfill\llap{#6}}%
327      {\rlap{#4}\hfill{#5}\hfill\llap{#6}}%
328      (\@headwidth,#3)}%
329     {(\@footwidth,#3)%
330      {\rlap{#9}\hfill{#8}\hfill\llap{#7}}%
331      {\rlap{#7}\hfill{#8}\hfill\llap{#9}}%
332      {\rlap{#7}\hfill{#8}\hfill\llap{#9}}%
333      (\@footwidth,#2)}%
334   \else
335     \defpagestyle*{#1}%
336     {(\@headwidth,#2)%
337      {\rlap{#6}\hfill{#5}\hfill\llap{#4}}%
338      {\rlap{#4}\hfill{#5}\hfill\llap{#6}}%
339      {\rlap{#4}\hfill{#5}\hfill\llap{#6}}%
340      (\@headwidth,#3)}%
341     {(\@footwidth,#3)%
342      {\rlap{#9}\hfill{#8}\hfill\llap{#7}}%
343      {\rlap{#7}\hfill{#8}\hfill\llap{#9}}%
344      {\rlap{#7}\hfill{#8}\hfill\llap{#9}}%
345      (\@footwidth,#2)}%
346   \fi
347 }
```

3.6 Automark or not

This is new since version 2.

`\automark` This macro activates automatic mark. It has two arguments. The first, optional argument is the sectioning level name of the mark at left pages, the second, not optional argument is the same at right pages. At oneside mode all pages are right.

`\@iflevelok` To define `\automark` we also need these three macros. The first one decides, if level of second parameter is less than level of first parameter and does
`\markleft` then third or fourth parameter. `\markleft` and `\@markleft` are similar to
`\@markleft` `\markright` and `\@markright` at L^AT_EX kernel.

```

348 \newcommand*{\@iflevelok}[4]{%
349   \begingroup
350     \edef\@tempa{#1}\ifx\@tempa\@empty\def\@tempa{\endgroup #3}\else
351       \def\@tempa{\csname #1level\endcsname}%
352       \def\@tempb{\csname #2level\endcsname}%
353       \ifnum \@tempa>\@tempb
354         \def\@tempa{\endgroup#3}%
355       \else
356         \def\@tempa{\endgroup#4}%
357     \fi\fi\@tempa}
358 \providecommand{\markleft}[1]{%
359   \begingroup
360     \let\label\relax \let\index\relax \let\glossary\relax
361     \expandafter\@markleft\@themark {#1}%
362     \@temptokena \expandafter{\@themark}%
363     \mark{\the\@temptokena}%
364   \endgroup
365   \if@nobreak\ifvmode\nobreak\fi\fi}
366 \providecommand{\@markleft}[3]{\@temptokena {#2}%
367   \unrestored@protected@xdef\@themark{#{3}{\the\@temptokena}}}
```

```

368 \newcommand*{\automark}[2][\@empty]{%
369   \manualmark
370   \@automarktrue
371   \begingroup
372     \edef\@tempa{#2}\ifx\@tempa\@empty
373       \endgroup
374       \PackageWarning{scrpage2}%
375         {\string\automark\space with empty obligatory argument}%
376     \else
377       \endgroup
378       \expandafter\def\csname #2mark\endcsname##1{%
379         \begingroup
380           \def\@tempa{\csname #2level\endcsname}%
381           \@tempswafalse
382           \ifnum \c@secnumdepth <\@tempa\else\if@mainmatter
```

```

383         \@tempswatrue
384     \fi\fi
385     \edef\@tempa{#1}%
386     \if@twoside\edef\@tempb{#2}\else\let\@tempb\@tempa\fi
387     \ifx\@tempa\@tempb
388         \if@tempswa
389             \@mkboth{\MakeMarkcase{\csname #2markformat\endcsname ##1}}%
390                 {\MakeMarkcase{\csname #2markformat\endcsname ##1}}%
391         \else%
392             \@mkboth{\MakeMarkcase{##1}}{\MakeMarkcase{##1}}%
393         \fi
394     \else
395         \if@tempswa
396             \@iflevelok{#1}{#2}{%
397                 \@mkboth{\MakeMarkcase{\csname #2markformat\endcsname
398                     ##1}}{\}%
399                 \ifx\@mkboth\markboth\markleft{\MakeMarkcase{\csname
400                     #2markformat\endcsname ##1}}%
401                 \fi}
402             \else
403                 \@iflevelok{#1}{#2}{%
404                     \@mkboth{\MakeMarkcase{##1}}{\}%
405                     \ifx\@mkboth\markboth\markleft{\MakeMarkcase{##1}}\fi}%
406             \fi
407         \fi
408     \endgroup
409 }
410 \fi
411 \begingroup
412 \edef\@tempa{#1}\edef\@tempb{#2}%
413 \if@autooneside\if@twoside\else\let\@tempa\@empty\fi\fi
414 \ifx\@tempa\@empty
415     \endgroup
416 \else
417     \ifx\@tempa\@tempb
418         \endgroup
419     \else
420         \endgroup
421         \expandafter\def\csname #1mark\endcsname##1{%
422             \begingroup
423             \def\@tempa{\csname #1level\endcsname}%
424             \@tempswafalse
425             \ifnum \c@secnumdepth <\@tempa\else\if@mainmatter
426                 \@tempswatrue
427             \fi\fi
428             \if@tempswa
429                 \@iflevelok{#1}{#2}{%
430                     \ifx\@mkboth\markboth
431                         \markright{\MakeMarkcase{\csname #1markformat\endcsname

```

```

432         ##1}}%
433     \fi}{%
434     \@mkboth{}\MakeMarkcase{\csname #1markformat\endcsname
435         ##1}}}%
436     \else
437         \@iflevelok{#1}{#2}{%
438             \ifx\@mkboth\markboth
439                 \markright{\MakeMarkcase{##1}}}%
440             \fi}{%
441             \@mkboth{}\MakeMarkcase{##1}}}%
442     \fi
443 \endgroup
444 }
445 \fi\fi%
446 \let\@mkboth\markboth
447 }

```

`\manualmark` This macro switches to the manual set of marks (using `\markboth` and `\markright`. This is very simple, we only have to deactivate the mark macros.

```

448 \newcommand*\manualmark{%
449     \if@chapter\let\chaptermark\@gobble\fi
450     \let\sectionmark\@gobble
451     \let\subsectionmark\@gobble
452     \let\subsubsectionmark\@gobble
453     \let\paragraphmark\@gobble
454     \let\subparagraphmark\@gobble
455     \let\@mkboth\@gobbletwo
456     \@automarkfalse
457 }

```

`\partlevel` At `\automark` we use some macros to get the numeric level of a sectioning name. We simply define these:

```

\chapterlevel 458 \newcommand*\partlevel{-1}
\sectionlevel 459 \newcommand*\chapterlevel{0}
\subsectionlevel 460 \newcommand*\sectionlevel{1}
\subsubsectionlevel 461 \newcommand*\subsectionlevel{2}
\paragraphlevel 462 \newcommand*\subsubsectionlevel{3}
\subparagraphlevel 463 \newcommand*\paragraphlevel{4}
464 \newcommand*\subparagraphlevel{5}

```

`\MakeMarkcase` At `\automark` we use a new macro to enable usability of `\MakeUppercase` at the head-marks. We set this macro to `\MakeUppercase` if one of the standard-classes was loaded:

```

465 \@ifundefined{MakeMarkcase}{%
466     \@ifclassloaded{scrbook}{\let\MakeMarkcase\relax}{%
467         \@ifclassloaded{scrreprt}{\let\MakeMarkcase\relax}{%
468             \@ifclassloaded{scrartcl}{\let\MakeMarkcase\relax}{%

```

```

469 \ifclassloaded{scrlettr2}{\let\MakeMarkcase\relax}{%
470 \ifclassloaded{book}{}{%
471 \ifclassloaded{report}{}{%
472 \ifclassloaded{article}{}{%
473 \PackageWarningNoLine{scrpage2}{%
474 use of scrpage2 with unknown class.\MessageBreak
475 Please mail to markus.kohm@gmx.de, if
476 headmarks\MessageBreak
477 are lower case without use of scrpage2}}}%
478 \let\MakeMarkcase\MakeUppercase}}}}{}

```

`\scrheadings@style` This macro defines, which style for scrheadings should be used. If it is not defined, we have to define it in a proper way.

```

479 \ifundefined{scrheadings@style}{%
480 \ifclassloaded{scrbook}{\def\scrheadings@style{koma}}{%
481 \ifclassloaded{scrreprt}{\def\scrheadings@style{koma}}{%
482 \ifclassloaded{scrartcl}{\def\scrheadings@style{koma}}{%
483 \ifclassloaded{scrlettr2}{\def\scrheadings@style{koma}}{%
484 \def\scrheadings@style{standard}}}}}}{}

```

3.7 Width of head and foot

`\setheadwidth` Until version 1.3, head and foot of a page had same width like the text
`\setfootwidth` block. Since version 1.3 this was changed. Now the head and foot have
`\set@hf@width` their own width and an optional shift relative to the text part of the page.
`\settowidthof` This shift can be positive or negative. We set the width and shift with
`\deftowidthof` one macro and save them at macros. The set macros have an optional
argument: the shift.

For definition of symbolic length values there is the macro `\settowidthof`.
And to define a macro with this value, there is `\deftowidthof`.

```

485 \newcommand*{\settowidthof}[2]{%
486 \edef\@tempa{#2}%
487 \edef\@tempb{paper}\ifx\@tempa\@tempb
488 \setlength{#1}{\paperwidth}%
489 \else\edef\@tempb{text}\ifx\@tempa\@tempb
490 \setlength{#1}{\textwidth}%
491 \else\edef\@tempb{textwithmarginpar}\ifx\@tempa\@tempb
492 \setlength{#1}{\textwidth}%
493 \addtolength{#1}{\marginparwidth}%
494 \addtolength{#1}{\marginparsep}%
495 \else\edef\@tempb{head}\ifx\@tempa\@tempb
496 \setlength{#1}{\@headwidth}%
497 \else\edef\@tempb{foot}\ifx\@tempa\@tempb
498 \setlength{#1}{\@footwidth}%
499 \else\edef\@tempb{headtopline}\ifx\@tempa\@tempb
500 \setlength{#1}{\scr@headabove@linelength}%
501 \else\edef\@tempb{headsepline}\ifx\@tempa\@tempb
502 \setlength{#1}{\scr@headbelow@linelength}%

```

```

503         \else\edef\@tempb{footsepline}\ifx\@tempa\@tempb
504             \setlength{#1}{\scr@footabove@linelength}%
505         \else\edef\@tempb{footbotline}\ifx\@tempa\@tempb
506             \setlength{#1}{\scr@footbelow@linelength}%
507         \else\edef\@tempb{page}\ifx\@tempa\@tempb
508             \setlength{#1}{\paperwidth}%
509             \begingroup\expandafter\expandafter\expandafter\endgroup
510             \expandafter\ifx\csname ta@bcor\endcsname\relax\else
511                 \addtolength{#1}{-\ta@bcor}%
512             \fi
513         \else
514             \setlength{#1}{#2}%
515         \fi
516     \fi
517 \fi
518 \fi
519 \fi
520 \fi
521 \fi
522 \fi
523 \fi
524 \fi
525 }
526 \newcommand*\@deftowidthof}[2]{%
527     \edef\@tempa{#2}\edef\@tempb{autohead}\ifx\@tempa\@tempb
528         \expandafter\def\csname #1\endcsname{\@headwidth}%
529     \else\edef\@tempb{autofoot}\ifx\@tempa\@tempb
530         \expandafter\def\csname #1\endcsname{\@footwidth}%
531     \else
532         \settowidthof{\@tempdima}{#2}%
533         \expandafter\edef\csname #1\endcsname{\the\@tempdima}%
534     \fi
535 \fi
536 }
537 \newcommand*\@setheadwidth#[2][\@empty]{%
538     \set@hf@width{head}{#1}{#2} %
539 }
540 \newcommand*\@setfootwidth#[2][\@empty]{%
541     \set@hf@width{foot}{#1}{#2}%
542 }
543 \newcommand*\@set@hf@width#[3]{%
544     \settowidthof{\@tempdima}{#3}%
545     \ifdim\@tempdima>\paperwidth
546         \PackageWarning
547             {scrpage2}%
548             {You've set width of #1 to a value\MessageBreak
549             greater than width of page!\MessageBreak%
550             I'll reduce width of #1 to width of page}%
551     \setlength{\@tempdima}{\paperwidth}%

```

```

552 \else
553   \ifdim\@tempdima<\z@
554     \PackageWarning
555       {scrpage2}%
556       {You've set width of #1 to negative value!\MessageBreak%
557         I'll set it to 0pt}%
558     \setlength{\@tempdima}{\z@}%
559   \fi
560 \fi
561 \expandafter\edef\csname @#1width\endcsname{\the\@tempdima}%
562 \edef\@tempa{#2}%
563 \setlength{\@tempdima}{\textwidth}%
564 \addtolength{\@tempdima}{-\csname @#1width\endcsname}%
565 \ifx\@tempa\@empty
566   \if@twoside
567     \setlength{\@tempdimb}{.333333333\@tempdima}%
568   \else
569     \setlength{\@tempdimb}{.5\@tempdima}%
570   \fi
571 \else
572   \setlength{\@tempdimb}{#2}%
573 \fi
574 \addtolength{\@tempdima}{-\@tempdimb}%
575 \ifdim\@tempdimb<\z@
576   \setlength{\@tempdimb}{-\@tempdimb}%
577   \expandafter\edef\csname @odd#1shift\endcsname{-\the\@tempdimb}%
578 \else
579   \expandafter\edef\csname @odd#1shift\endcsname{\the\@tempdimb}%
580 \fi
581 \ifdim\@tempdima<\z@
582   \setlength{\@tempdima}{-\@tempdima}%
583   \expandafter\edef\csname @even#1shift\endcsname{-\the\@tempdima}%
584 \else
585   \expandafter\edef\csname @even#1shift\endcsname{\the\@tempdima}%
586 \fi
587 }

```

\@headwidth The values of width and shift of head and foot are saved at internal macros.

```

\@footwidth 588 \newcommand*{\@headwidth}{\textwidth}%
\@oddheadshift 589 \newcommand*{\@oddheadshift}{\z@}%
\@evenheadshift 590 \newcommand*{\@evenheadshift}{\z@}%
\@oddfootshift 591 \newcommand*{\@footwidth}{\textwidth}%
\@evenfootshift 592 \newcommand*{\@oddfootshift}{\z@}%
593 \newcommand*{\@evenfootshift}{\z@}%

```

If option `mpinclude` was used, we use a special trick to set width of head and foot but only if the user doesn't:

```

594 \if@mincl
595   \renewcommand*{\@headwidth}{}%
596   \renewcommand*{\@footwidth}{}%

```

```

597 \AtBeginDocument{%
598   \ifx\@headwidth\@empty
599     \setheadwidth[0pt]{textwithmarginpar}%
600   \fi
601   \ifx\@footwidth\@empty
602     \setfootwidth[0pt]{textwithmarginpar}%
603   \fi
604 }
605 \fi

```

3.8 Fonts

`\pnumfont` We have to define a macro defining the font, used by `\pagemark`. Because of this macro may already be defined by a KOMA-script-class, we have to do this, using `\providecommand`

```
606 \providecommand*\pnumfont{\normalfont}
```

`\headfont` There is also a macro used by `\@evenhead`, `\@oddhead` and `\@evenfoot`, `\@oddfoot`:

```

607 \providecommand*\headfont{\normalfont\slshape}
608 \def\scr@fnt@wrn@pagehead#1{%
609   'pagehead' is only an alias of 'pageheadfoot'.\MessageBreak
610   Font of page foot may also be changed%
611 }

```

`\footfont` There is also a macro used by `\@evenfoot`, `\@oddfoot` after `\headfont`:

```

\scr@fnt@pagefoot 612 \providecommand*\footfont{\normalfont}
\scr@fnt@wrn@pagefoot 613 \providecommand*\scr@fnt@pagefoot{\footfont}
614 \let\scr@fnt@wrn@pagefoot\relax

```

3.9 Length and thickness of lines at head and foot

`\setheadtopline` Since version 2 there are new macros to set the length and thickness of the lines at head and foot. We use macros not dimensions, because the amount of dimensions is very small at T_EX.

```

\setheadsepline
\setfootsepline
\setfootbotline 615 \newcommand*\setheadtopline{\scr@setline{head}{above}}
616 \newcommand*\setheadsepline{\scr@setline{head}{below}}
617 \newcommand*\setfootsepline{\scr@setline{foot}{above}}
618 \newcommand*\setfootbotline{\scr@setline{foot}{below}}

```

`\scr@setline` The real change of the lines is done with the macros `\scr@setline`, `\@scr@setline` and `\@@scr@setline`:

```

\@@scr@setline 619 \def\scr@setline#1#2{%
\scr@set@pre@line 620   \kernel@ifnextchar[%
621     {\@scr@setline{#1}{#2}}{\@@scr@setline{#1#2}}%
622 }
623 \def\@scr@setline#1#2[#3]{%

```

```

624 \edef\@tempa{#3}\edef\@tempa{auto}\ifx\@tempa\@tempb
625 \deftowidthof{scr@#1#2@linelength}{#3#1}%
626 \else
627 \deftowidthof{scr@#1#2@linelength}{#3}%
628 \fi
629 \@scr@setline{#1#2}%
630 }
631 \def\@scr@setline#1#2{%
632 \edef\@tempa{#2}\edef\@tempb{current}\ifx\@tempa\@empty\else
633 \ifx\@tempa\@tempb\else
634 \setlength{\@tempdima}{#2}%
635 \expandafter\edef\csname scr@#1@linethickness\endcsname%
636 {\the\@tempdima}%
637 \fi\fi
638 \kernel@ifnextchar[%]
639 {\@src@set@pre@line{#1}}{}%
640 }
641 \def\@src@set@pre@line#1[#2]{%
642 \expandafter\def\csname scr@pre@#1@line\endcsname{#2}%
643 }

```

3.10 Extended multilanguage formats

<p>\partmarkformat</p> <p>\chaptermarkformat</p> <p>\sectionmarkformat</p> <p>\subsectionmarkformat</p> <p>\subsubsectionmarkformat</p> <p>\paragraphmarkformat</p> <p>\subparagraphmarkformat</p> <p>\@secCNTmarkformat</p> <p>\@secCNTformat</p>	<p>There are some macros to handle languages like Chinese, Japanese or Korean. These macros were defined first by Werner Lemberg at his CJK-bundle. See <code>scrclass.dtx</code> for further information.</p>	<pre> 644 \providecommand{\partmarkformat} {\partname\ \thepart. \ } 645 \ifchapter 646 \providecommand{\chaptermarkformat} {\@chapapp\ \thechapter. \ } 647 \providecommand{\sectionmarkformat} {\thesection. \ } 648 \providecommand{\subsectionmarkformat} {\@secCNTmarkformat{subsection}} 649 \else 650 \providecommand{\sectionmarkformat} {\thesection\quad} 651 \providecommand{\subsectionmarkformat} {\thesubsection\quad} 652 \fi 653 \providecommand{\subsubsectionmarkformat}{\@secCNTmarkformat{subsubsection}} 654 \providecommand{\paragraphmarkformat} {\@secCNTmarkformat{paragraph}} 655 \providecommand{\subparagraphmarkformat} {\@secCNTmarkformat{subparagraph}} 656 \providecommand{\@secCNTmarkformat}[1] {\@secCNTformat{#1}} </pre>
--	--	---

3.11 Definition of default pagestyles

\ps@save@plain We define two default pagestyles, which can be used instead of the standard pagestyles. The two pagestyles are something like a family. A pagestyle for *normal* pages with head and foot and a pagestyle for *plain* pages with not so much head and foot. So we have to redefine the standard pagestyle `plain`. But maybe someone want to use the old standard pagestyle. So we first define new names for old pagestyles.

```

657 \newcommand*{\ps@save@plain}{\}
658 \let\ps@save@plain\ps@plain
659 \newcommand*{\ps@use@plain}{\let\save@@mkboth\@mkboth
660   \ps@save@plain
661   \let\@mkboth\save@@mkboth}
662 \let\ps@plain\ps@use@plain

```

`\ps@save@headings` If standard `pagestyle headings` or `myheadings` is used, also standard
`\ps@save@myheadings` `pagestyle use@plain` should be used. So we also define new names for
`\ps@headings` the other old pagestyles and new pagestyles with the old names, doing the
`\ps@myheadings` switch back for the plain pagestyle. These are defined even if the pagestyles
are unknown.

```

663 \newcommand*{\ps@save@headings}{\}
664 \let\ps@save@headings\ps@headings
665 \newcommand*{\ps@save@myheadings}{\}
666 \let\ps@save@myheadings\ps@myheadings
667 \def\ps@headings{%
668   \PackageWarning{scrpage2}%
669     {you've used pagestyle 'headings'.\MessageBreak
670     You should not use this pagestyle with a\MessageBreak
671     scrpage2-pagestyle!\MessageBreak
672     You should use pagestyle 'useheadings' instead\MessageBreak}%
673   \let\ps@plain\ps@save@plain%
674   \ps@save@headings\@automarktrue}
675 \def\ps@myheadings{%
676   \PackageWarning{scrpage2}%
677     {you've used pagestyle 'myheadings'.\MessageBreak
678     You should not use this pagestyle with a\MessageBreak
679     scrpage2-pagestyle!\MessageBreak
680     You should use pagestyle 'useheadings' instead\MessageBreak}%
681   \let\ps@plain\ps@save@plain%
682   \ps@save@myheadings\@automarkfalse}

```

`\ps@useheadings` This is the pagestyle, you should use instead of `headings` or `myheadings`.
If you are using manual marks, this behaves like pagestyle `myheadings`.
If you are using automatic marks, this behaves like `headings`, but without
changing `\chaptermark`, `\sectionmark` or `\subsectionmark`.

```

683 \newcommand*{\ps@useheadings}{%
684   \if@automark\let\@mkboth\markboth\else\let\@mkboth\@gobbletwo\fi
685   \let\ps@plain\ps@use@plain
686   \let\save@@mkboth\@mkboth
687   \let\save@chaptermark\chaptermark
688   \let\save@sectionmark\sectionmark
689   \let\save@subsectionmark\subsectionmark
690   \ifx\@mkboth\markboth\ps@save@headings\else\ps@save@myheadings\fi
691   \let\@mkboth\save@@mkboth
692   \let\chaptermark\save@chaptermark
693   \let\sectionmark\save@sectionmark

```

```

694 \let\subsectionmark\save@subsectionmark
695 }

```

Note: Because of saving \@mkboth, \chaptermark, \sectionmark and \subsection before changing the pagestyle and restore them after changing the pagestyle it does not matter, if we set pagestyle headings or myheadings. So it would be possible to remove the \ifx...\fi-line above and simply use \ps@save@headings or \ps@save@myheadings instead of.

```

\ps@scrplain After this, we can define the new pagestyles, first a new plain one. It could
\scrplain@even@left@head be variable and so we use macros to define the parts.
\scrplain@even@middle@head 696 \newcommand*{\ps@scrplain}{\let\ps@plain\ps@scrplain\ps@plain}
\scrplain@even@right@head 697 \newpagestyle{\scrplain}
\scrplain@odd@left@head 698 {%
\scrplain@odd@middle@head 699 (\scrplain@headabove@linelength,\scr@headabove@linethickness)%
\scrplain@odd@right@head 700 {%
\scrplain@even@left@foot 701 \rlap{\parbox{\@headwidth}{\raggedright
\scrplain@even@middle@foot 702 \strut\scrplain@even@left@head\strut}}}%
\scrplain@even@right@foot 703 \hfill
\scrplain@odd@left@foot 704 \parbox{\@headwidth}{\centering\strut
\scrplain@odd@middle@foot 705 \strut\scrplain@even@middle@head\strut}}}%
\scrplain@odd@middle@foot 706 \hfill
\scrplain@odd@right@foot 707 \llap{\parbox{\@headwidth}{\raggedleft
708 \strut\scrplain@even@right@head\strut}}}%
709 {%
710 \rlap{\parbox{\@headwidth}{\raggedright
711 \strut\scrplain@odd@left@head\strut}}}%
712 \hfill
713 \parbox{\@headwidth}{\centering
714 \strut\scrplain@odd@middle@head\strut}}}%
715 \hfill
716 \llap{\parbox{\@headwidth}{\raggedleft
717 \strut\scrplain@odd@right@head\strut}}}%
718 {%
719 \rlap{\parbox{\@headwidth}{\raggedright
720 \strut\scrplain@odd@left@head\strut}}}%
721 \hfill
722 \parbox{\@headwidth}{\centering
723 \strut\scrplain@odd@middle@head\strut}}}%
724 \hfill
725 \llap{\parbox{\@headwidth}{\raggedleft
726 \strut\scrplain@odd@right@head\strut}}}%
727 (\scrplain@headbelow@linelength,\scr@headbelow@linethickness)
728 }{%
729 (\scrplain@footabove@linelength,\scr@footabove@linethickness)
730 {%
731 \rlap{\parbox{\@footwidth}{\raggedright
732 \strut\scrplain@even@left@foot\strut}}}%
733 \hfill

```

```

734 \parbox{\@footwidth}{\centering
735 \strut\scrplain@even@middle@foot\strut}%
736 \hfill
737 \llap{\parbox{\@footwidth}{\raggedleft
738 \strut\scrplain@even@right@foot\strut}}}%
739 {%
740 \rlap{\parbox{\@footwidth}{\raggedright
741 \strut\scrplain@odd@left@foot\strut}}}%
742 \hfill
743 \parbox{\@footwidth}{\centering
744 \strut\scrplain@odd@middle@foot\strut}%
745 \hfill
746 \llap{\parbox{\@footwidth}{\raggedleft
747 \strut\scrplain@odd@right@foot\strut}}}%
748 {%
749 \rlap{\parbox{\@footwidth}{\raggedright
750 \strut\scrplain@odd@left@foot\strut}}}%
751 \hfill
752 \parbox{\@footwidth}{\centering
753 \strut\scrplain@odd@middle@foot\strut}%
754 \hfill
755 \llap{\parbox{\@footwidth}{\raggedleft
756 \strut\scrplain@odd@right@foot\strut}}}%
757 (\scrplain@footbelow@linelength,\scr@footbelow@linethickness)
758 }
759 \newcommand*{\scrplain@even@left@head}{\}
760 \newcommand*{\scrplain@even@middle@head}{\}
761 \newcommand*{\scrplain@even@right@head}{\}
762 \newcommand*{\scrplain@odd@left@head}{\}
763 \newcommand*{\scrplain@odd@middle@head}{\}
764 \newcommand*{\scrplain@odd@right@head}{\}
765 \newcommand*{\scrplain@even@left@foot}{\}
766 \newcommand*{\scrplain@even@middle@foot}{\}
767 \newcommand*{\scrplain@even@right@foot}{\}
768 \newcommand*{\scrplain@odd@left@foot}{\}
769 \newcommand*{\scrplain@odd@middle@foot}{\}
770 \newcommand*{\scrplain@odd@right@foot}{\}

```

\ps@scrheadings

We also define a new headings one. It could be variable and so we use macros to define the parts. But we need an *also action*, so we first define a inner pagestyle and then use this at a outer one (definition is done in reverse order):

```

\scrheadings@even@left@head
\scrheadings@even@middle@head
\scrheadings@even@right@head
\scrheadings@odd@left@head
\scrheadings@odd@middle@head
\scrheadings@odd@right@head
\scrheadings@even@left@foot
\scrheadings@even@middle@foot
\scrheadings@even@right@foot
\scrheadings@odd@left@foot
\scrheadings@odd@middle@foot
\scrheadings@odd@right@foot

```

```

771 \newcommand*{\ps@scrheadings}{\let\ps@plain\ps@scrplain\ps@scrheadings}
772 \newpagestyle{@scrheadings}
773 {%
774 {%
775 \rlap{\parbox{\@headwidth}{\raggedright
776 \strut\scrheadings@even@left@head\strut}}}%
777 \hfill

```

```

778 \parbox{\@headwidth}{\centering
779 \strut\scrheadings@even@middle@head\strut}%
780 \hfill
781 \llap{\parbox{\@headwidth}{\raggedleft%
782 \strut\scrheadings@even@right@head\strut}}}%
783 {%
784 \rlap{\parbox{\@headwidth}{\raggedright
785 \strut\scrheadings@odd@left@head\strut}}}%
786 \hfill
787 \parbox{\@headwidth}{\centering
788 \strut\scrheadings@odd@middle@head\strut}%
789 \hfill
790 \llap{\parbox{\@headwidth}{\raggedleft%
791 \strut\scrheadings@odd@right@head\strut}}}%
792 {%
793 \rlap{\parbox{\@headwidth}{\raggedright
794 \strut\scrheadings@odd@left@head\strut}}}%
795 \hfill
796 \parbox{\@headwidth}{\centering
797 \strut\scrheadings@odd@middle@head\strut}%
798 \hfill
799 \llap{\parbox{\@headwidth}{\raggedleft
800 \strut\scrheadings@odd@right@head\strut}}}%
801 {%
802 {%
803 \rlap{\parbox{\@footwidth}{\raggedright
804 \strut\scrheadings@even@left@foot\strut}}}%
805 \hfill
806 \parbox{\@footwidth}{\centering
807 \strut\scrheadings@even@middle@foot\strut}%
808 \hfill
809 \llap{\parbox{\@footwidth}{\raggedleft
810 \strut\scrheadings@even@right@foot\strut}}}%
811 {%
812 \rlap{\parbox{\@footwidth}{\raggedright
813 \strut\scrheadings@odd@left@foot\strut}}}%
814 \hfill
815 \parbox{\@footwidth}{\centering
816 \strut\scrheadings@odd@middle@foot\strut}%
817 \hfill
818 \llap{\parbox{\@footwidth}{\raggedleft
819 \strut\scrheadings@odd@right@foot\strut}}}%
820 {%
821 \rlap{\parbox{\@footwidth}{\raggedright
822 \strut\scrheadings@odd@left@foot\strut}}}%
823 \hfill
824 \parbox{\@footwidth}{\centering
825 \strut\scrheadings@odd@middle@foot\strut}%
826 \hfill

```

```

827 \llap{\parbox{\@footwidth}{\raggedleft
828 \strut\scrheadings@odd@right@foot\strut}}%
829 }%
830 }
831 \newcommand*\scrheadings@even@left@head{}
832 \newcommand*\scrheadings@even@middle@head{}
833 \newcommand*\scrheadings@even@right@head{}
834 \newcommand*\scrheadings@odd@left@head{}
835 \newcommand*\scrheadings@odd@middle@head{}
836 \newcommand*\scrheadings@odd@right@head{}
837 \newcommand*\scrheadings@even@left@foot{}
838 \newcommand*\scrheadings@even@middle@foot{}
839 \newcommand*\scrheadings@even@right@foot{}
840 \newcommand*\scrheadings@odd@left@foot{}
841 \newcommand*\scrheadings@odd@middle@foot{}
842 \newcommand*\scrheadings@odd@right@foot{}

```

`\ihead` We also need macros to define the contents of the three part head and
`\ohead` foot. Each macro defines one of the part of head or foot of the scrheadings
`\chead` pagestyle and optional of the scrplain pagestyle.

```

\lehead 843 \newcommand{\ihead}{\kernel@ifnextchar[{\@lhead}{\@lhead}}
\lohead 844 \newcommand{\@lhead}[1]{\@rehead{#1}\@lohead{#1}}
\rehead 845 \newcommand{\@lhead}[2][\@rehead{#1}{#2}\@lohead{#1}{#2}}
\rohead 846 \newcommand{\ohead}{\kernel@ifnextchar[{\@rhead}{\@rhead}}
\cehead 847 \newcommand{\@rhead}[1]{\@lehead{#1}\@rohead{#1}}
\cohead 848 \newcommand{\@rhead}[2][\@lehead{#1}{#2}\@rohead{#1}{#2}}
\ifoot 849 \newcommand{\chead}{\kernel@ifnextchar[{\@chead}{\@chead}}
\ofoot 850 \newcommand{\@chead}[1]{\@cehead{#1}\@cohead{#1}}
\cfoot 851 \newcommand{\@chead}[2][\@cehead{#1}{#2}\@cohead{#1}{#2}}
\lefoot 852 \newcommand{\lehead}{\kernel@ifnextchar[{\@lehead}{\@lehead}}
\lofoot 853 \newcommand{\@lehead}[1]{\renewcommand*\scrheadings@even@left@head{#1}}
\lofoot 854 \newcommand{\@lehead}[2][\@lohead{#1}{#2}}
\refoot 855 \renewcommand*\scrplain@even@left@head{#1}%
\rofoot 856 \renewcommand*\scrheadings@even@left@head{#2}}
\cefoot 857 \newcommand{\lohead}{\kernel@ifnextchar[{\@lohead}{\@lohead}}
\cofoot 858 \newcommand{\@lohead}[1]{\renewcommand*\scrheadings@odd@left@head{#1}}
\cofoot 859 \newcommand{\@lohead}[2][\@lohead{#1}{#2}}
860 \renewcommand*\scrplain@odd@left@head{#1}%
861 \renewcommand*\scrheadings@odd@left@head{#2}}
862 \newcommand{\rehead}{\kernel@ifnextchar[{\@rehead}{\@rehead}}
863 \newcommand{\@rehead}[1]{\renewcommand*\scrheadings@even@right@head{#1}}
864 \newcommand{\@rehead}[2][\@rehead{#1}{#2}}
865 \renewcommand*\scrplain@even@right@head{#1}%
866 \renewcommand*\scrheadings@even@right@head{#2}}
867 \newcommand{\rohead}{\kernel@ifnextchar[{\@rohead}{\@rohead}}
868 \newcommand{\@rohead}[1]{\renewcommand*\scrheadings@odd@right@head{#1}}
869 \newcommand{\@rohead}[2][\@rohead{#1}{#2}}
870 \renewcommand*\scrplain@odd@right@head{#1}%
871 \renewcommand*\scrheadings@odd@right@head{#2}}

```

```

872 \newcommand{\cehead}{\kernel@ifnextchar[{\@@cehead}{\@cehead}}
873 \newcommand{\@cehead}[1]{\renewcommand*{\scrheadings@even@middle@head}{#1}}
874 \newcommand{\@@cehead}[2][]{%
875   \renewcommand*{\scrplain@even@middle@head}{#1}%
876   \renewcommand*{\scrheadings@even@middle@head}{#2}}
877 \newcommand{\cohead}{\kernel@ifnextchar[{\@@cohead}{\@cohead}}
878 \newcommand{\@cohead}[1]{\renewcommand*{\scrheadings@odd@middle@head}{#1}}
879 \newcommand{\@@cohead}[2][]{%
880   \renewcommand*{\scrplain@odd@middle@head}{#1}%
881   \renewcommand*{\scrheadings@odd@middle@head}{#2}}
882 \newcommand{\ifoot}{\kernel@ifnextchar[{\@@lfoot}{\@lfoot}}
883 \newcommand{\@lfoot}[1]{\@refoot{#1}\@lofoot{#1}}
884 \newcommand{\@@lfoot}[2][]{\@@refoot[{#1}]{#2}\@lofoot[{#1}]{#2}}
885 \newcommand{\ofoot}{\kernel@ifnextchar[{\@@rfoot}{\@rfoot}}
886 \newcommand{\@rfoot}[1]{\@lefoot{#1}\@rofoot{#1}}
887 \newcommand{\@@rfoot}[2][]{\@@lefoot[{#1}]{#2}\@rofoot[{#1}]{#2}}
888 \newcommand{\cfoot}{\kernel@ifnextchar[{\@@cfoot}{\@cfoot}}
889 \newcommand{\@cfoot}[1]{\@cefoot{#1}\@cofoot{#1}}
890 \newcommand{\@@cfoot}[2][]{\@@cefoot[{#1}]{#2}\@cofoot[{#1}]{#2}}
891 \newcommand{\lefoot}{\kernel@ifnextchar[{\@@lefoot}{\@lefoot}}
892 \newcommand{\@lefoot}[1]{\renewcommand*{\scrheadings@even@left@foot}{#1}}
893 \newcommand{\@@lefoot}[2][]{%
894   \renewcommand*{\scrplain@even@left@foot}{#1}%
895   \renewcommand*{\scrheadings@even@left@foot}{#2}}
896 \newcommand{\lofoot}{\kernel@ifnextchar[{\@@lofoot}{\@lofoot}}
897 \newcommand{\@lofoot}[1]{\renewcommand*{\scrheadings@odd@left@foot}{#1}}
898 \newcommand{\@@lofoot}[2][]{%
899   \renewcommand*{\scrplain@odd@left@foot}{#1}%
900   \renewcommand*{\scrheadings@odd@left@foot}{#2}}
901 \newcommand{\refoot}{\kernel@ifnextchar[{\@@refoot}{\@refoot}}
902 \newcommand{\@refoot}[1]{\renewcommand*{\scrheadings@even@right@foot}{#1}}
903 \newcommand{\@@refoot}[2][]{%
904   \renewcommand*{\scrplain@even@right@foot}{#1}%
905   \renewcommand*{\scrheadings@even@right@foot}{#2}}
906 \newcommand{\rofoot}{\kernel@ifnextchar[{\@@rofoot}{\@rofoot}}
907 \newcommand{\@rofoot}[1]{\renewcommand*{\scrheadings@odd@right@foot}{#1}}
908 \newcommand{\@@rofoot}[2][]{%
909   \renewcommand*{\scrplain@odd@right@foot}{#1}%
910   \renewcommand*{\scrheadings@odd@right@foot}{#2}}
911 \newcommand{\cefoot}{\kernel@ifnextchar[{\@@cefoot}{\@cefoot}}
912 \newcommand{\@cefoot}[1]{\renewcommand*{\scrheadings@even@middle@foot}{#1}}
913 \newcommand{\@@cefoot}[2][]{%
914   \renewcommand*{\scrplain@even@middle@foot}{#1}%
915   \renewcommand*{\scrheadings@even@middle@foot}{#2}}
916 \newcommand{\cofoot}{\kernel@ifnextchar[{\@@cofoot}{\@cofoot}}
917 \newcommand{\@cofoot}[1]{\renewcommand*{\scrheadings@odd@middle@foot}{#1}}
918 \newcommand{\@@cofoot}[2][]{%
919   \renewcommand*{\scrplain@odd@middle@foot}{#1}%
920   \renewcommand*{\scrheadings@odd@middle@foot}{#2}}

```

`\clearscrheadings` Maybe you want to clear the currently defined head and foot. First let's
`\clearscrheadfoot` clear only scrheadings pagestyle but not scrplain pagestyle. We do this
`\clearscrplain` simply using the already known command:

```
921 \newcommand*{\clearscrheadings}{%
922   \ihead{}\chead{}\ohead{}\ifoot{}\cfoot{}\ofoot{}}
```

We can do almost the same to clear scrheadings pagestyle and scrplain pagestyle together:

```
923 \newcommand*{\clearscrheadfoot}{%
924   \ihead[]{}\chead[]{}\ohead[]{}\ifoot[]{}\cfoot[]{}\ofoot[]{}}
```

But, for only clearing the scrplain pagestyle, we have to do the redefinition directly:

```
925 \newcommand*{\clearscrplain}{%
926   \renewcommand*{\scrplain@even@left@head}{}%
927   \renewcommand*{\scrplain@even@middle@head}{}%
928   \renewcommand*{\scrplain@even@right@head}{}%
929   \renewcommand*{\scrplain@odd@left@head}{}%
930   \renewcommand*{\scrplain@odd@middle@head}{}%
931   \renewcommand*{\scrplain@odd@right@head}{}%
932   \renewcommand*{\scrplain@even@left@foot}{}%
933   \renewcommand*{\scrplain@even@middle@foot}{}%
934   \renewcommand*{\scrplain@even@right@foot}{}%
935   \renewcommand*{\scrplain@odd@left@foot}{}%
936   \renewcommand*{\scrplain@odd@middle@foot}{}%
937   \renewcommand*{\scrplain@odd@right@foot}{}%
938 }
```

Let's do the default setting:

```
939 \def\@tempa{koma}\ifx\scrheadings@style\@tempa
940   \@ifclassloaded{scrلتtr2}{%
941     \ifcase\@pageat
942       \lehead[\pagemark]{\pagemark}\lohead[\pagemark]{\pagemark}%
943       \chead{\rightmark}%
944       \rehead{\leftmark}\rohead{\leftmark}%
945     \or
946       \lehead{\leftmark}\lohead{\leftmark}%
947       \chead[\pagemark]{\pagemark}%
948       \rehead{\rightmark}\rohead{\rightmark}%
949     \or
950       \lehead{\leftmark}\lohead{\leftmark}%
951       \chead{\rightmark}%
952       \rehead[\pagemark]{\pagemark}\rohead[\pagemark]{\pagemark}%
953     \or
954       \lehead{\leftmark}\lohead{\leftmark}%
955       \rehead{\rightmark}\rohead{\rightmark}%
956       \lefoot[\pagemark]{\pagemark}\lofoot[\pagemark]{\pagemark}%
957     \or
958       \lehead{\leftmark}\lohead{\leftmark}%
959     \fi
960   }
```

```

959     \rehead{\rightmark}\rohead{\rightmark}%
960     \cfoot[\pagemark]{\pagemark}%
961 \or
962     \lehead{\leftmark}\lohead{\leftmark}%
963     \rehead{\rightmark}\rohead{\rightmark}%
964     \refoot[\pagemark]{\pagemark}\rofoot[\pagemark]{\pagemark}%
965 \fi
966 \let\ps@save@myheadings=\ps@save@headings
967 }{%
968 \ohead{\if@twoside\headmark\fi}%
969 \chead{\if@twoside\else\headmark\fi}%
970 \cfoot[\if@twoside\else\pagemark\fi]{\if@twoside\else\pagemark\fi}%
971 \ofoot[\if@twoside\pagemark\fi]{\if@twoside\pagemark\fi}%
972 }%
973 \else\def\@tempa{standard}\ifx\scrheadings@style\@tempa
974     \ihead{\headmark}
975     \ohead{\pagemark}
976     \cfoot[\pagemark]{\fi}
977 \else\PackageError{scrpage2}{unknown scrheadings style}%
978     {Somehow \string\scrheadings@style was
979     defined\MessageBreak
980     as ‘‘\scrheadings@style’’.\MessageBreak
981     scrpage2 doesn’t know what to do.
982     If you’ll continue, pagestyle
983     ‘‘scrheadings’’\MessageBreak
984     will be predefined empty.}
985 \fi\fi
986 \begingroup\expandafter\expandafter\expandafter\endgroup\expandafter
987 \ifx\csname chaptermark\endcsname\relax
988 \begingroup\expandafter\expandafter\expandafter\endgroup\expandafter
989 \ifx\csname sectionmark\endcsname\relax
990     \ifx\@mkboth\markboth\@automarktrue\fi
991 \else
992     \ifx\sectionmark\@gobble\else
993         \begingroup
994             \def\@gobble#1{}\ifx\sectionmark\@gobble\else
995                 \aftergroup\@automarktrue
996             \fi
997         \endgroup
998     \fi
999 \fi
1000 \else
1001 \ifx\chaptermark\@gobble\else
1002     \begingroup
1003         \def\@gobble#1{}\ifx\chaptermark\@gobble\else
1004             \aftergroup\@automarktrue
1005         \fi
1006     \endgroup
1007 \fi

```

1008 \fi

3.12 Configurationfile

Not last and not least we include the local configuration-file `scrpage.cfg`, if it exists.

```
1009 \InputIfFileExists{scrpage.cfg}
1010     {\typeout{*****^~J%
1011             * Local config file scrpage.cfg used^~J%
1012             *****}}
1013     {}
```

3.13 Some checks

Last but not least we do some checks at version 2. Some means one at the moment. And the one is a check, if the `\headheight` should be increased.

```
1014 \AtBeginDocument{%
1015   \ifdim\headheight=12pt
1016     \PackageWarningNoLine{scrpage2}%
1017       {seems you are using a constant headheight.\MessageBreak
1018       If you get overfull \string\vbox\space warnings at almost every%
1019       \MessageBreak
1020       page, you should increase headheight e.g. writing\MessageBreak
1021       \space\string\setlength{\string\headheight}{%
1022       1.1\string\baselineskip}\MessageBreak
1023       into the preamble of your document.\MessageBreak
1024       A better solution would be to use typearea package}%
1025   \else\ifdim\headheight<1.07\baselineskip
1026     \PackageWarningNoLine{scrpage2}%
1027       {seems you are using a very small headheight.\MessageBreak
1028       If you get overfull \string\vbox\space warnings at almost every%
1029       \MessageBreak
1030       page, you should increase headheight e.g. writing\MessageBreak
1031       \space\string\setlength{\string\headheight}{%
1032       1.1\string\baselineskip}\MessageBreak
1033       into the preamble of your document.\MessageBreak
1034       A better solution would be to use typearea package}%
1035   \fi\fi}
```

3.14 End

```
1036 \scrpage2
```

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		
<code>\@@@deftripstyle</code>	<u>320</u>	<code>\clearscrheadings</code> 5, 26, <u>921</u>
<code>\@@deftripstyle</code>	<u>315</u>	<code>\clearscrplain</code> <u>921</u>
<code>\@scr@setline</code>	<u>619</u>	<code>\clearscrplain</code> 5, 26
<code>\@automarkfalse</code>	<u>135</u>	<code>clines</code> (Option) 0, 76
<code>\@automarktrue</code>	<u>135</u>	<code>\cofoot</code> 3, 24, <u>843</u>
<code>\@defpagestyle</code>	<u>140</u>	<code>\cohead</code> 3, 24, <u>843</u>
<code>\@deftripstyle</code>	<u>311</u>	color
<code>\@evenfootshift</code>	588	in footer 10
<code>\@evenheadshift</code>	588	in header 10
<code>\@footwidth</code>	588	
<code>\@headwidth</code>	588	
<code>\@iflevelok</code>	348	
<code>\@markleft</code>	348	
<code>\@minclfalse</code>	37	
<code>\@mincltrue</code>	37	
<code>\@newpagestyle</code>	<u>265</u>	
<code>\@oddfootshift</code>	588	
<code>\@oddheadshift</code>	588	
<code>\@providepagestyle</code>	298	
<code>\@renewpagestyle</code>	281	
<code>\@scr@set@pre@line</code>	<u>619</u>	
<code>\@scr@setline</code>	<u>619</u>	
<code>\@seccntformat</code>	<u>644</u>	
<code>\@seccntmarkformat</code>	<u>644</u>	
	A	
<code>\automark</code>	6, 28, <u>348</u>	
<code>automark</code> (Option)	0, <u>80</u>	
<code>autooneside</code> (Option)	0	
<code>autoonesize</code> (Option)	88	
	B	
<code>\botfoot@rule</code>	<u>218</u>	
	C	
<code>\cefoot</code>	3, 24, <u>843</u>	
<code>\cehead</code>	3, 24, <u>843</u>	
<code>\cfoot</code>	3, 24, <u>843</u>	
<code>\chapterlevel</code>	<u>458</u>	
<code>\chaptermarkformat</code>	<u>644</u>	
<code>\chead</code>	3, 24, <u>843</u>	
<code>\clearscrheadfoot</code>	5, 26, <u>921</u>	
		D
		<code>\def@@@foot</code> <u>184</u>
		<code>\def@@foot</code> <u>184</u>
		<code>\def@head</code> <u>147</u>
		<code>\def@foot</code> <u>184</u>
		<code>\def@head</code> <u>147</u>
		<code>\defpagestyle</code> 17, <u>136</u>
		<code>\deftowidthof</code> <u>485</u>
		<code>\deftripstyle</code> 15, 37, <u>307</u>
		E
		<code>einseitig</code> 28
		F
		Farbe
		im Kopf 32
		im Fuß 32
		foot
		color 10
		<code>footbotline</code> (Option) 0, <u>52</u>
		footer
		color 10
		<code>footexclude</code> (Option) 0, <u>7</u>
		<code>\footfont</code> 7, 29, <u>612</u>
		<code>footinclude</code> (Option) 0, <u>7</u>
		<code>footsepline</code> (Option) 0, <u>52</u>
		Fuß
		Breite 29
		Farbe 32
		G
		<code>Gliederungsebenen</code> 28
		<code>Großbuchstaben</code> 37

H		<code>\manualmark</code> 6, 28, 448
head		<code>manualmark (Option)</code> 0, 80
color	10	<code>\markleft</code> 348
header		<code>markuppercase (Option)</code> 0, 90
color	10	<code>markusedcase (Option)</code> 0, 90
<code>headexclude (Option)</code>	0, 7	<code>mpexclude (Option)</code> 37
<code>\headfont</code>	7, 29, 607	<code>mpinclude (Option)</code> 37
<code>headinclude (Option)</code>	0, 7	
heading	15	N
headings	4	<code>\newpagestyle</code> 17, 261
automatic	3	<code>nouppercase (Option)</code> 0, 92
manual	3	
running	3	O
<code>\headmark</code>	6, 27, 133	<code>\ofoot</code> 3, 24, 843
<code>headsepline (Option)</code>	0, 52	<code>\ohead</code> 3, 24, 843
<code>headtopline (Option)</code>	0, 52	<code>olines (Option)</code> 0, 76
<code>\hf@rule</code>	223	Optionen:
<code>\hflines@adjust</code>	76	<code>automark</code> 0, 80
		<code>autooneside</code> 0
		<code>autoonesize</code> 88
I		<code>clines</code> 0, 76
<code>\if@automark</code>	135	<code>footbotline</code> 0, 52
<code>\if@autooneside</code>	88	<code>footexclude</code> 0, 7
<code>\if@chapter</code>	109	<code>footinclude</code> 0, 7
<code>\if@mainmatter</code>	116	<code>footsepline</code> 0, 52
<code>\if@mincl</code>	37	<code>headexclude</code> 0, 7
<code>\ifoot</code>	3, 24, 843	<code>headinclude</code> 0, 7
<code>\ihead</code>	3, 24, 843	<code>headsepline</code> 0, 52
<code>ilines (Option)</code>	0, 76	<code>headtopline</code> 0, 52
		<code>ilines</code> 0, 76
K		<code>komastyle</code> 0, 103
Kolumnentitel	25	<code>manualmark</code> 0, 80
automatisch	24, 36, 37	<code>markuppercase</code> 0, 90
manuell	24	<code>markusedcase</code> 0, 90
<code>komastyle (Option)</code>	0, 103	<code>mpexclude</code> 37
Kopf		<code>mpinclude</code> 37
Breite	29	<code>nouppercase</code> 0, 92
Farbe	32	<code>olines</code> 0, 76
		<code>plainfootbotline</code> 0, 52
L		<code>plainfootsepline</code> 0, 52
<code>\lefoot</code>	3, 24, 843	<code>plainheadsepline</code> 0, 52
<code>\leftmark</code>	6, 27	<code>plainheadtopline</code> 0, 52
<code>\lehead</code>	3, 24, 843	<code>standardstyle</code> 0, 103
line		
alignment	12	
Linienausrichtung	34	P
<code>\lofoot</code>	3, 24, 843	page
<code>\lohead</code>	3, 24, 843	number
		<code>\pagemark</code> 6, 27, 134
M		pagination
<code>\MakeMarkcase</code>	465	Paginierung

<code>\paragraphlevel</code>	458	<code>\scr@setline</code>	619
<code>\paragraphmarkformat</code>	644	<code>\scrheadings@even@left@foot</code>	771
<code>\partlevel</code>	458	<code>\scrheadings@even@left@head</code>	771
<code>\partmarkformat</code>	644	<code>\scrheadings@even@middle@foot</code>	771
<code>plainfootbotline</code> (Option) ..	0, 52	<code>\scrheadings@even@middle@head</code>	771
<code>plainfootsepline</code> (Option) ..	0, 52	<code>\scrheadings@even@right@foot</code>	771
<code>plainheadsepline</code> (Option) ..	0, 52	<code>\scrheadings@even@right@head</code>	771
<code>plainheadtopline</code> (Option) ..	0, 52	<code>\scrheadings@odd@left@foot</code> .	771
<code>\pnumfont</code>	7, 29, 606	<code>\scrheadings@odd@left@head</code> .	771
<code>\providepagestyle</code>	17, 294	<code>\scrheadings@odd@middle@foot</code>	771
<code>\ps@headings</code>	663	<code>\scrheadings@odd@middle@head</code>	771
<code>\ps@myheadings</code>	663	<code>\scrheadings@odd@right@foot</code>	771
<code>\ps@save@headings</code>	663	<code>\scrheadings@odd@right@head</code>	771
<code>\ps@save@myheadings</code>	663	<code>\scrheadings@style</code>	479
<code>\ps@save@plain</code>	657	<code>\scrplain@even@left@foot</code> ...	696
<code>\ps@scrheadings</code>	771	<code>\scrplain@even@left@head</code> ...	696
<code>\ps@scrplain</code>	696	<code>\scrplain@even@middle@foot</code> .	696
<code>\ps@useheadings</code>	683	<code>\scrplain@even@middle@head</code> .	696
R			
<code>\refoot</code>	3, 24, 843	<code>\scrplain@even@right@foot</code> ..	696
<code>\rehead</code>	3, 24, 843	<code>\scrplain@even@right@head</code> ..	696
<code>\renewpagestyle</code>	17, 277	<code>\scrplain@footabove@linelength</code>	44
<code>\rightmark</code>	6, 27	44
<code>\rofoot</code>	3, 24, 843	<code>\scrplain@footbelow@linelength</code>	44
<code>\rohead</code>	3, 24, 843	44
<code>rule</code>		<code>\scrplain@headabove@linelength</code>	44
<code>alignment</code>	12	44
S			
<code>\scr@fnt@footbottomrule</code>	252	<code>\scrplain@headbelow@linelength</code>	44
<code>\scr@fnt@footsepline</code>	252	44
<code>\scr@fnt@headsepline</code>	252	<code>\scrplain@odd@left@foot</code>	696
<code>\scr@fnt@headtopline</code>	252	<code>\scrplain@odd@left@head</code>	696
<code>\scr@fnt@instead@footbotline</code>	252	<code>\scrplain@odd@middle@foot</code> ..	696
<code>\scr@fnt@pagefoot</code>	612	<code>\scrplain@odd@middle@head</code> ..	696
<code>\scr@fnt@wrn@pagefoot</code>	612	<code>\scrplain@odd@right@foot</code> ...	696
<code>\scr@footabove@linelength</code> ...	44	<code>\scrplain@odd@right@head</code> ...	696
<code>\scr@footabove@linethickness</code>	40	<code>\sectionlevel</code>	458
<code>\scr@footbelow@linelength</code> ...	44	<code>\sectionmarkformat</code>	644
<code>\scr@footbelow@linethickness</code>	40	<code>\set@hf@width</code>	485
<code>\scr@headabove@linelength</code> ...	44	<code>\setfootbotline</code>	9, 31, 615
<code>\scr@headabove@linethickness</code>	40	<code>\setfootsepline</code>	9, 31, 615
<code>\scr@headbelow@linelength</code> ...	44	<code>\setfootwidth</code>	8, 29, 485
<code>\scr@headbelow@linethickness</code>	40	<code>\setheadsepline</code>	9, 31, 615
<code>\scr@nouppercase</code>	92	<code>\setheadtopline</code>	9, 31, 615
<code>\scr@pre@footabove@line</code>	252	<code>\setheadwidth</code>	8, 29, 485
<code>\scr@pre@footbelow@line</code>	252	<code>\settowidthof</code>	485
<code>\scr@pre@headabove@line</code>	252	<code>standardstyle</code> (Option)	0, 103
<code>\scr@pre@headbelow@line</code>	252	<code>\subparagraphlevel</code>	458
		<code>\subparagraphmarkformat</code>	644
		<code>\subsectionlevel</code>	458
		<code>\subsectionmarkformat</code>	644

<code>\subsubsectionlevel</code>	458		
<code>\subsubsectionmarkformat</code> . . .	644		
		U	
		uppercase letters	15
		T	
<code>\topfoot@rule</code>	213		

Change History

2020/02/27		<code>\def@foot</code> : Use of <code>\@footwidth</code> instead of <code>\textwidth</code> . . .	50
General: Final release	1	<code>\hf@rule</code> : New, third parameter “boxwidth”	51
Manual integrated into dtx-file	1	<code>\set@hf@width</code> : New	58
v1.0b		<code>\setfootwidth</code> : New	58
<code>\@defpagestyle</code> : Dot deleted after number at chapter- and sectionmark	49	<code>\setheadwidth</code> : New	58
Use CJK at <code>\chaptermark</code> , <code>\sectionmark</code> and <code>\subsectionmark</code>	49	v1.3a	
v1.1		<code>\set@hf@width</code> : symbolic values “paper”, “text”, “textwidthmarginpar”	58
General: <code>\hbox</code> to changed into <code>\hb@xt@</code>	44	v1.3e	
<code>\chaptermarkformat</code> : Space after chapternumber increased.	62	<code>\@defpagestyle</code> : usage of <code>\if@mainmatter</code> fixed	49
<code>\sectionmarkformat</code> : Space after sectionnumber increased.	62	v2.0	
<code>\subsectionmarkformat</code> : Space after subsectionnumber increased.	62	<code>\@defpagestyle</code> : Version 2 uses <code>\automark</code> and <code>\manualmark</code> , so star and non-star version are same . .	49
v1.2		<code>\@seccntformat</code> : New	62
<code>\if@mainmatter</code> : Definition changed	48	<code>\@seccntmarkformat</code> : New . .	62
v1.2a		General: <code>\headheight-check</code> . .	71
<code>\@renewpagestyle</code> : Avoid to define pagestyle as <code>\relax</code> if it wasn’t defined (Thanks to Bernd).	52	<code>\automark</code> : New	55
v1.3		<code>\cefoot</code> : New	67
<code>\@evenfootshift</code> : New	60	<code>\cehead</code> : New	67
<code>\@evenheadshift</code> : New	60	<code>\cfoot</code> : New	67
<code>\@footwidth</code> : New	60	<code>\chapterlevel</code> : New	57
<code>\@headwidth</code> : New	60	<code>\chead</code> : New	67
<code>\@oddfootshift</code> : New	60	<code>\cofoot</code> : New	67
<code>\@oddheadshift</code> : New	60	<code>\cohead</code> : New	67
		<code>\def@@head</code> : Thickness and length of the lines are optional	49
		<code>\def@foot</code> : Thickness and length of the lines are optional	50
		<code>footbotline</code> : New option . . .	46
		<code>footsepline</code> : New option . . .	46
		<code>headsepline</code> : New option . . .	46
		<code>headtopline</code> : New option . . .	46

<code>\ifoot: New</code>	67	<code>\scrheadings@even@middle@head:</code>	
<code>\ihead: New</code>	67	<code> New</code>	65
<code>\lefoot: New</code>	67	<code>\scrheadings@even@right@foot:</code>	
<code>\lehead: New</code>	67	<code> New</code>	65
<code>\lofoot: New</code>	67	<code>\scrheadings@even@right@head:</code>	
<code>\lohead: New</code>	67	<code> New</code>	65
<code>\manualmark: New</code>	57	<code>\scrheadings@odd@left@foot:</code>	
<code>\ofoot: New</code>	67	<code> New</code>	65
<code>\ohead: New</code>	67	<code>\scrheadings@odd@left@head:</code>	
<code>\paragraphlevel: New</code>	57	<code> New</code>	65
<code>\paragraphmarkformat: New</code>	62	<code>\scrheadings@odd@middle@foot:</code>	
<code>plainfootbotline: New option</code>	46	<code> New</code>	65
<code>plainfootsepline: New option</code>	46	<code>\scrheadings@odd@middle@head:</code>	
<code>plainheadsepline: New option</code>	46	<code> New</code>	65
<code>plainheadtopline: New option</code>	46	<code>\scrheadings@odd@right@foot:</code>	
<code>\ps@headings: New</code>	63	<code> New</code>	65
<code>\ps@myheadings: New</code>	63	<code>\scrheadings@odd@right@head:</code>	
<code>\ps@save@headings: New</code>	63	<code> New</code>	65
<code>\ps@save@myheadings: New</code>	63	<code>\scrplain@even@left@foot:</code>	
<code>\ps@save@plain: New</code>	62	<code> New</code>	64
<code>\ps@scrheadings: New</code>	65	<code>\scrplain@even@left@head:</code>	
<code>\ps@scrplain: New</code>	64	<code> New</code>	64
<code>\refoot: New</code>	67	<code>\scrplain@even@middle@foot:</code>	
<code>\rehead: New</code>	67	<code> New</code>	64
<code>\rofoot: New</code>	67	<code>\scrplain@even@middle@head:</code>	
<code>\rohead: New</code>	67	<code> New</code>	64
<code>\scr@footabove@linelength:</code>		<code>\scrplain@even@right@foot:</code>	
<code> New</code>	45	<code> New</code>	64
<code>\scr@footabove@linethickness:</code>		<code>\scrplain@even@right@head:</code>	
<code> New</code>	45	<code> New</code>	64
<code>\scr@footbelow@linelength:</code>		<code>\scrplain@footabove@linelength:</code>	
<code> New</code>	45	<code> New</code>	45
<code>\scr@footbelow@linethickness:</code>		<code>\scrplain@footbelow@linelength:</code>	
<code> New</code>	45	<code> New</code>	45
<code>\scr@headabove@linelength:</code>		<code>\scrplain@headabove@linelength:</code>	
<code> New</code>	45	<code> New</code>	45
<code>\scr@headabove@linethickness:</code>		<code>\scrplain@headbelow@linelength:</code>	
<code> New</code>	45	<code> New</code>	45
<code>\scr@headbelow@linelength:</code>		<code>\scrplain@odd@left@foot:</code>	
<code> New</code>	45	<code> New</code>	64
<code>\scr@headbelow@linethickness:</code>		<code>\scrplain@odd@left@head:</code>	
<code> New</code>	45	<code> New</code>	64
<code>\scrheadings@even@left@foot:</code>		<code>\scrplain@odd@middle@foot:</code>	
<code> New</code>	65	<code> New</code>	64
<code>\scrheadings@even@left@head:</code>		<code>\scrplain@odd@middle@head:</code>	
<code> New</code>	65	<code> New</code>	64
<code>\scrheadings@even@middle@foot:</code>		<code>\scrplain@odd@right@foot:</code>	
<code> New</code>	65	<code> New</code>	64

\scrplain@odd@right@head:	\@scr@setline: symbolic values
New 64	implemented 61
\sectionlevel: New 57	\setfootbotline: footbot
\setfootbotline: New 61	changed into footbelow .. 61
\setfootsepline: New 61	\setfootsepline: foottop
\setheadsepline: New 61	changed into footabove .. 61
\setheadtopline: New 61	\setheadsepline: headbot
\subparagraphlevel: New .. 57	changed into headbelow .. 61
\subparagraphmarkformat:	\setheadtopline: headtop
New 62	changed into headabove .. 61
\subsectionlevel: New 57	v2.0e
\subsubsectionlevel: New .. 57	\deftowidthof: New 58
\subsubsectionmarkformat:	\settowidthof: New 58
New 62	v2.0f
v2.0a	clines: New 46
\ps@scrheadings: \strut	\deftowidthof: new symbolic
inserted 65	values “autohead” and
\ps@scrplain: \strut inserted 64	“autofoot” 58
v2.0b	\hf@rule: Rule can be adjusted
automark: use	at inner margin, centered or
\AtEndOfPackage to run not	adjusted at outer margin . 51
yet defined macro 46	\hflin@adjust: New 46
manualmark: use	ilines: New 46
\AtEndOfPackage to run not	olines: New 46
yet defined macro 46	\ps@scrheadings: \strut
\ps@save@headings: use this	inserted 65
instead of	\ps@scrplain: \strut inserted 64
\ps@standardheadings .. 63	\setfootbotline: new
\ps@save@myheadings: use this	symbolic value “auto” 61
instead of	\setfootsepline: new
\ps@standardmyheadings . 63	symbolic value “auto” 61
\ps@save@plain: use this	\setheadsepline: new
instead of	symbolic value “auto” 61
\ps@standardplain 62	\setheadtopline: new
\ps@useheadings: New 63	symbolic value “auto” 61
\set@hf@width: symbolic values	v2.0g
“paper”, “text”,	\automark: Use of \manualmark 55
“textwidthmarginpar” 58	\manualmark: also deactivating
v2.0c	\...mark 57
\automark: Use of	\scrheadings@style: New .. 58
\MakeMarkcase 55	v2.0i
markuppercase: New 47	\@iflevelok: New 55
markusedcase: New 47	\@markleft: New 55
\set@hf@width: error fix at	\automark: \markboth behavior
warning 58	fixed 55
v2.0d	Use of \if@autooneside .. 55
\@@scr@setline: macro name	Warning, if arguments are
fixed 61	mixed up 55
	autoonesize: New 47

<code>\deftowidthof</code> : new symbolic value “page”	58	v2.1c	<code>\ps@scrplain</code> : thickness of footbelow line fixed	64
<code>\if@autooneside</code> : New	47			
<code>manualmark</code> : don’t ignore oneside any longer	46	v2.2	<code>\@scr@set@pre@line</code> : New internal	61
<code>\markleft</code> : New	55		<code>\deftowidthof</code> : value “foot” fixed	58
v2.0j			<code>\hf@rule</code> : Last parameter of the rule tells, if it is a headabove, headbelow, footabove or footbelow rule	51
<code>\clearscrheadfoot</code> : New	69		Macro is not longer long	51
<code>\clearscrheadings</code> : New	69		Precommands before the rules are allowed	51
<code>\clearscrpain</code> : New	69		<code>\partlevel</code> : New	57
<code>\def@@@foot</code> : Use of <code>\scr@nouppercase</code>	50		<code>\partmarkformat</code> : New	62
<code>\def@@@head</code> : Use of <code>\scr@nouppercase</code>	49		<code>\scr@fnt@footbottomrule</code> : New KOMA-Script font command	52
<code>nouppercase</code> : New	47		<code>\scr@fnt@footsepline</code> : New KOMA-Script font command	52
<code>\scr@nouppercase</code> : New	47		<code>\scr@fnt@headsepline</code> : New KOMA-Script font command	52
v2.0k			<code>\scr@fnt@headtopline</code> : New KOMA-Script font command	52
<code>\deftowidthof</code> : <code>\pagewidth</code> korrigiert	58		<code>\scr@fnt@instead@footbotline</code> : New KOMA-Script font command	52
v2.0m			<code>\scr@pre@footabove@line</code> : New internal	52
General: Active support for new letter class <code>scrlttr2</code>	69		<code>\scr@pre@footbelow@line</code> : New internal	52
<code>\MakeMarkcase</code> : <code>scrlttr2</code> is a known class	57		<code>\scr@pre@headabove@line</code> : New internal	52
<code>\ps@useheadings</code> : Fatal bug (<code>\mkboth</code> instead of <code>\@mkboth</code>) fixed	63		<code>\scr@pre@headbelow@line</code> : New internal	52
<code>\scrheadings@style</code> : Take care of class <code>scrlttr2</code>	58	v2.2a	<code>\automark</code> : all changes of helper macros and switches are local to fix e.g. <code>twocolumn</code> problem with <code>\part</code>	55
v2.1				
<code>\@footwidth</code> : use of option <code>mpinclude</code>	60	v2.2b	<code>\hf@rule</code> : No even alignment at oneside	51
<code>\@headwidth</code> : use of option <code>mpinclude</code>	60			
<code>\if@automark</code> : New switch	49			
v2.1a				
<code>\@defpagestyle</code> : Use <code>\if@automark</code>	49			
<code>\automark</code> : Switch on <code>\if@automark</code>	55			
<code>\manualmark</code> : Switch off <code>\if@automark</code>	57			
<code>\ps@useheadings</code> : Use of switch <code>\if@automark</code>	63			
v2.1b				
<code>\deftowidthof</code> : more save existance test	58			
<code>\if@chapter</code> : more save test	48			
<code>\if@mainmatter</code> : more save test	48			

v2.2c	General: a lot of % added and removed	44	v3.20	\@@deftripstyle:	
				\@ifnextchar replaced by	
v2.2d	\@defpagestyle: Support of article-classes with			\kernel@ifnextchar	54
	\if@mainmatter	49		\@@scr@setline: \@ifnextchar	
	\if@mainmatter: additional			replaced by	
	\if@mainmatter test	48		\kernel@ifnextchar	61
v2.2f				\@deftripstyle: \@ifnextchar	
	\def@@@foot: Use of \footfont	50		replaced by	
	\footfont: New	61		\kernel@ifnextchar	54
	\scr@fnt@pagefoot: New (internal)	61		\cefoot: \@ifnextchar	
	\scr@fnt@wrn@pagefoot: deactivated	61		replaced by	
v2.2g				\kernel@ifnextchar	67
	\@minclfalse: Test fixed	45		\cehead: \@ifnextchar	
	\if@mainmatter: usage of			replaced by	
	\if@mainmatter at message removed	48		\kernel@ifnextchar	67
v2.3				\cfoot: \@ifnextchar replaced	
	footexclude: obsolete	44		by \kernel@ifnextchar . .	67
	footinclude: obsolete	44		\chead: \@ifnextchar replaced	
	headexclude: obsolete	44		by \kernel@ifnextchar . .	67
	headinclude: obsolete	44		\cofoot: \@ifnextchar	
v2.4				replaced by	
	\partlevel: part level is always -1	57		\kernel@ifnextchar	67
v2.5				\cohead: \@ifnextchar	
	\@defpagestyle: moved			replaced by	
	\if@twoside inside of definition	49		\kernel@ifnextchar	49
	\def@@@foot: new (internal) .	50		\def@foot: \@ifnextchar	
	\def@@foot: new (internal) . .	50		replaced by	
	\def@@head: new (internal) . .	49		\kernel@ifnextchar	50
	\def@foot: new (internal) . . .	50		\def@head: \@ifnextchar	
	\def@head: new (internal) . . .	49		replaced by	
v2.5a				\kernel@ifnextchar	49
	General: \@gobble is long, but			\ifoot: \@ifnextchar replaced	
	\sectionmark and			by \kernel@ifnextchar . .	67
	\chaptermark may be short	70		\ihead: \@ifnextchar replaced	
v2.8q				by \kernel@ifnextchar . .	67
	mpexclude: New option	45		\lefoot: \@ifnextchar	
	mpinclude: New option	45		replaced by	
v3.12				\kernel@ifnextchar	67
	General: fix \rohead if page number left at head	69		\lehead: \@ifnextchar	
				replaced by	
				\kernel@ifnextchar	67
				\lofoot: \@ifnextchar	

replaced by		replaced by	
\kernel@ifnextchar	67	\kernel@ifnextchar	67
\lohead: \@ifnextchar		\rohead: \@ifnextchar	
replaced by		replaced by	
\kernel@ifnextchar	67	\kernel@ifnextchar	67
\ofoot: \@ifnextchar replaced		\scr@setline: \@ifnextchar	
by \kernel@ifnextchar ..	67	replaced by	
\ohead: \@ifnextchar replaced		\kernel@ifnextchar	61
by \kernel@ifnextchar ..	67		
\refoot: \@ifnextchar		v3.24	
replaced by		General: warning about obsolete	
\kernel@ifnextchar	67	package added	44
\rehead: \@ifnextchar		v3.28	
replaced by		General: \PackageErrorNoLine	
\kernel@ifnextchar	67	durch \PackageError	
\rofoot: \@ifnextchar		ersetzt	69