

The `xhead` package*

FMi

2010/05/21

Contents

1 Introduction	2
2 Interfaces	2
2.1 The object type ‘heading’	2
2.2 The template ‘2e-vertical’ (object type heading)	3
3 Implementation	4
3.1 The heading object type	4
3.2 Interfaces to other components	4
3.2.1 Generating TOC and mark entries	4
3.2.2 Prepare heading number	7
3.2.3 Prepare indentation after heading	7
3.2.4 Evaluating the title arguments	7
3.3 Support variables for template code	8
3.4 The template <code>2e-vertical</code>	9
3.5 The template <code>3a-vertical</code>	10

Abstract

This module contains basic templates generating typical headings for various levels. This is very much work in progress and none of its content should be considered stable. Concepts and implementation are still under development.

*This file has version number 1914, last revised 2010/05/21.

1 Introduction

2 Interfaces

2.1 The object type ‘heading’

Arg: 1 (boolean) If `true` generate an unnumbered heading, otherwise a number is generated up to a certain heading level. If `true` TOC entries and running headers are suppressed unless explicitly forced by other arguments below.

Arg: 2 (<string>/\NoValue/<empty>) Specifies an alternate heading text for the TOC if one is produced. If `\NoValue` or `<empty>` the TOC receives the the main title text.

Arg: 3 (boolean) If `true` a TOC entry is forced (i.e., generated even if normally suppressed).

Arg: 4 (<string>/\NoValue/<empty>) Specifies an alternate heading text for the running header if one is produced. If `\NoValue` or `<empty>` the running header receives the the main title text.

Arg: 5 (boolean) If `true` a running header is forced (i.e., generated even if normally suppressed).

Arg: 6 (<string>) The title of the heading in the document. Also used for the TOC entry and running header (if produced) unless overwritten by other arguments.

Arg: 7 (<key/value list>) Extension argument to receive less use key value pairs to manipulate the heading. Exact syntax and semantics still need to be defined.

Semantics:

This template type defines the basic interface for headings both display headings as well as run-in headings.

A TOC entry is generated if `#1` is `false` or `#3` being `true`. The heading title for the TOC entry is `#6` unless overwritten by a non-empty `HASH2`. Whether or not this TOC entry actually appears in the TOC depends on settings for the TOC generation. (However, if not generated it will obviously not appear.)

A running header is generated if `#1` is `false` or `#5` or `true`. The heading title for the determined as follows: if `#4` is non-empty it will be used, otherwise if `#2` is non-empty it will be used, otherwise `HASH6` will be used.

`#7` will hold additional key/value pairs that define more specific semantics. This isn't implemented yet.

The above describes my current thinking about how those arguments should interact. In simpler templates (based on 2e semantics) the arguments #3 to #5 and #7 will get ignored.

2.2 The template ‘2e-vertical’ (object type heading)

Attributes:

- name (tokenlist)** Name of the heading. Used to refer to parts of its data structure, e.g., the counter associated with the heading will have this name. Must be set to a suitable string like “section” Default: none
- level (integer)** The level of the heading in the hierarchy. Required. Default: none
- break-penalty (penalty)** Penalty to break in front of the heading unless it is immediately preceded a higher-level heading in which case breaking is prohibited. Default: -300
- indent (length)** Indentation of the heading from the left margin. Default: 0pt
- above-skip (skip)** Vertical space above the heading. Default: 0pt
- below-skip (skip)** Vertical space below the heading. Default: 0pt
- title-decl (function 0)** Declarations to format the heading, e.g., font changes. Default: *<empty>*
- number-format (function 1)** Formatting instruction for the heading number (if present), receiving the heading counter representation as an argument. Default: `\xhead_default_number_format:n`
- indent-after (boolean)** Should the first paragraph after the heading be indented? Default: `false`

Semantics & Comments:

This is, or should be, a reimplementaion of the 2e `\@startsection` command for generating vertical headings. `\@startsection` is also able to produce run-in headings, but this is not covered by the current template.

3 Implementation

```
1 (*package)
2 \RequirePackage{expl3,xtemplate}
3 \GetIdInfo$Id: xhead.dtx 1914 2010-05-21 16:18:46Z schoepf $
4 {heading module}
5 \ProvidesExplPackage
6 {\filename}{\filedate}{\fileversion}{\filedescription}
```

3.1 The heading object type

Arguments:

```
#1 (boolean)           : produce no number
#2 (string/\NoValue/empty) : toc-title if different from title
#3 (boolean)           : force toc entry
#4 (string/\NoValue/empty) : mark-title if different from toc-title
#5 (Boolean)           : force mark entry
#6 (string)            : title
#7 (extra key vals)

7 \DeclareObjectType {heading} {7}
```

3.2 Interfaces to other components

This section holds code that is (or should be) common between different heading templates, for example, code for passing a title to the TOC system will always use the same TOC interface so it is encapsulated here. Effectively this means that this section should be the only part that needs modification if we update these other components and introduce new interfaces for them.

At the moment everything is set up to use the old 2e interfaces, but this will change over time.

3.2.1 Generating TOC and mark entries

```
\xhead_generate_toc:NNNNN This command provides the interface to the TOC system (currently the 2e one). It uses
\xhead_generate_toc:NNNNn its first two arguments to decide if a TOC entry should be produced and if so generates
the entry from the remaining arguments.
```

Arguments:

```
#1 (boolean)           : produce no number
#2 (boolean)           : force toc entry
#3 (string)            : heading name (\l_xhead_name_tl)
#4 (string)            : heading number if any (\l_xhead_number_tl)
#5 (string)            : toc-title (\l_xhead_toc_title_tl)
```

```

8 \cs_new:Npn \xhead_generate_toc:NNNNN #1#2#3#4#5 {
9   <*trace>
10   \typeout{xhead_generate_toc:NNNNN=#1:#2:#3:#4:#5}
11 </trace>

```

A TOC entry is produced if the heading number was not suppressed (!#1) or it is forced (#2).

```

12   \bool_if:nT
13   { #2 || ! #1 }
14   {
15     \group_begin:

```

Guard against use of mark commands in the heading title (from old sources).

```

16     \cs_set_eq:NN \@mkboth \use_none:nn
17     \cs_set_eq:NN \markboth \use_none:nn
18     \cs_set_eq:NN \markright \use_none:n

```

Use the old 2e \addcontentsline interface:

```

19     \addcontentsline{toc}{#3}
20     {
21       \tl_if_empty:NF #4
22       { \protect\numberline{ #4 } }
23       #5
24     }
25   \group_end:
26 }
27 }

```

In the templates we really want to make sure that the title in the last argument is passed and not a tokenlist in which it is saved. After all, this text will get passed around over long distances. We therefore provide the following variant for use in templates:

```

28 \cs_generate_variant:Nn \xhead_generate_toc:NNNNN {NNNNNo}

```

```

\xhead_generate_mark:NNNNN
\xhead_generate_mark:NNNNNo

```

This command provides the interface to the mark system for running headers (currently the 2e one). It uses its first two arguments to decide if a mark entry should be produced and if so generates the entry from the remaining arguments.

Arguments:

#1 (boolean)	: produce no number
#2 (boolean)	: force running header entry
#3 (string)	: heading name (\l_xhead_name_tl)
#4 (string)	: heading number if any (\l_xhead_number_tl)
#5 (string)	: mark-title (\l_xhead_mark_title_tl)

```

29 \cs_new:Npn \xhead_generate_mark:NNNNN #1#2#3#4#5 {
30   \*trace
31   \typeout{xhead_generate_mark:NNNNN=#1:#2:#3:#4:#5}
32 \*trace

```

A running header entry is produced if the heading number was not suppressed (!#1) or it is forced (#2).

```

33   \bool_if:nT
34   { #2 || ! #1 }
35   {
36     \group_begin:

```

Guard against use of mark command in heading titles (from old sources)¹

```

37     \cs_set_eq:NN \@mkboth \use_none:nn

```

Again we use the original 2e interface. `\markright`, etc. currently adds the section number so we do not add it here—for now.

```

38     \use:c {#3 mark} {#5}
39     \group_end:
40   }
41 }

```

In the templates we really want to make sure that the title in the last argument is passed and not a tokenlist in which it is saved. After all, this text will get passed around over long distances. We therefore provide the following variant for use in templates:

```

42 \cs_generate_variant:Nn \xhead_generate_mark:NNNNN {NNNNNo}

```

`\l_generate_toc_and_mark:NNNNNNN` This is just putting it all together, may not be worth having.

Arguments:

#1 (boolean)	: produce no number
#2 (boolean)	: force toc entry
#3 (boolean)	: force running header entry
#4 (string)	: heading name (<code>\l_xhead_name_tl</code>)
#5 (string)	: heading number or empty (<code>\l_xhead_number_tl</code>)
#6 (string)	: toc-title (<code>\l_xhead_toc_title_tl</code>)
#7 (string)	: mark-title (<code>\l_xhead_mark_title_tl</code>)

```

43 \cs_new:Npn \xhead_generate_toc_and_mark:NNNNNNN #1#2#3#4#5#6#7 {
44   \xhead_generate_toc:NNNNNo
45   #1 #2 #4 #5 #6
46   \xhead_generate_mark:NNNNNo
47   #1 #3 #4 #5 #7
48 }

```

¹This is wrong here, needs thought.

3.2.2 Prepare heading number

`\xhead_update_prepare_number:N` This command checks if a number should be produced and if so increments the corresponding counter and stores the number in `\l_xhead_number_tl`. Otherwise `\l_xhead_number_tl` will be set to empty to indicate that no number should be displayed.

Arguments:

`#1` (boolean) : produce no number

Assumptions:

`\l_xhead_name_tl` : holds heading name
`\l_xhead_level_int` : holds heading level

Outputs:

`\l_xhead_number_tl` : holds heading number or returns empty

```
49 \cs_new:Npn \xhead_update_prepare_number:N #1 {
50   \bool_if:nTF
51     { #1 || \intexpr_compare_p:n { \l_xhead_level_int > \c@secnumdepth } }
52     { \tl_clear:N \l_xhead_number_tl }
53     {
54       \refstepcounter \l_xhead_name_tl
55       \tl_set:Nx \l_xhead_number_tl { \use:c {the \l_xhead_name_tl} }
56     }
57 }
```

3.2.3 Prepare indentation after heading

To set the paragraph indentation after a heading we need to interact with the galley mechanism, which at one point will be `xgalley`. Right now we use the `2e` interface.

`\xhead_set_after_indention:N` Here we set the `2e` flag depending

```
58 \cs_new:Npn \xhead_set_after_indention:N #1 {
59   \bool_if:NTF #1 \@afterindenttrue \@afterindentfalse
60 }
```

3.2.4 Evaluating the title arguments

According to the heading object type spec the title text for the toc and running header may differ from the main title in dependence of the arguments passed to the template. The code below implements the documented logic.

`\xhead_set_titles:nmn` Arguments:

`#1` (string/\NoValue/empty) : toc-title if different from title
`#2` (string/\NoValue/empty) : mark-title if different from toc-title
`#3` (string) : title

```
61 \cs_new:Npn \xhead_set_titles:nmn #1#2#3 {
```

Determine title for toc: If `#1` is `\NoValue` or `<empty>` use `#3` for the toc-title. Otherwise use `#1`.

```
62   \IfNoValueTF{#1}
63     { \tl_set:Nn \l_xhead_toc_title_tl {#3} }
64     { \tl_if_empty:nTF {#1}
65       { \tl_set:Nn \l_xhead_toc_title_tl {#3} }
66       { \tl_set:Nn \l_xhead_toc_title_tl {#1} }
67   }
```

Determine title for mark: If `#2` is `\NoValue` then use whatever was determined for the toc-title as value (i.e., if a special toc-title was used this will be also used for the running header, if not the main title is used). Otherwise, if `#2` is `<empty>` use `#3`. Otherwise use `#2` (i.e., a special title was passed for the running header).

```
68   \IfNoValueTF{#2}
69     { \tl_set_eq:NN \l_xhead_mark_title_tl \l_xhead_toc_title_tl }
70     { \tl_if_empty:nTF {#2}
71       { \tl_set:Nn \l_xhead_mark_title_tl {#3} }
72       { \tl_set:Nn \l_xhead_mark_title_tl {#2} }
73   }
74 }
```

3.3 Support variables for template code

```
75 \dim_new:N \l_xhead_indent_dim
76 \skip_new:N \l_xhead_above_skip
77 \skip_new:N \l_xhead_below_skip
78 \int_new:N \l_xhead_level_int
79 \int_new:N \l_xhead_break_penalty_int
80 \tl_new:N \l_xhead_name_tl
81 \tl_new:N \l_xhead_number_tl
82 \tl_new:N \l_xhead_toc_title_tl
83 \tl_new:N \l_xhead_mark_title_tl
84 \bool_new:N \l_xhead_indent_after_bool
85
86 \cs_new:Npn \xhead_title_decl: {}
87
88 \cs_new:Npn \xhead_number_format:n #1 {}
```



```

89
90 \cs_new:Npn \xhead_default_number_format:n #1 { #1 \quad}

```

3.4 The template 2e-vertical

```

91 \DeclareTemplateInterface {heading} {2e-vertical} {7}
92 {
93   name      : tokenlist   = ???   ,
94   level     : integer     = 0     ,
95   break-penalty: integer  = -300  ,
96   indent    : length     = Opt   ,
97   above-skip : skip       = Opt   ,
98   below-skip : skip       = Opt   ,
99   title-decl : function {0} =     ,
100  number-format: function {1} = \xhead_default_number_format:n {#1} ,
101  indent-after : boolean   = false ,
102 }
103 \DeclareTemplateCode {heading} {2e-vertical} {7}
104 {
105   name      = \l_xhead_name_tl      ,
106   level     = \l_xhead_level_int    ,
107   break-penalty = \l_xhead_break_penalty_int ,
108   indent    = \l_xhead_indent_dim  ,
109   above-skip = \l_xhead_above_skip  ,
110   below-skip = \l_xhead_below_skip  ,
111   title-decl = \xhead_title_decl:   ,
112   number-format = \xhead_number_format:n ,
113   indent-after = \l_xhead_indent_after_bool ,
114 }
115 {
116   \AssignTemplateKeys

```

Determine title for toc and mark using the 2e logic: Do not generate them if the star-form of the heading is used and always use the same text for TOC and mark:

```

117   \bool_if:NF #1
118   {
119     \IfNoValueTF{#2}
120     { \tl_set:Nn \l_xhead_toc_title_tl {#6} }
121     { \tl_set:Nn \l_xhead_toc_title_tl {#2} }
122     \tl_set_eq:NN \l_xhead_mark_title_tl \l_xhead_toc_title_tl
123   }

```

If heading number increment counter and set `\l_xhead_number_tl`

```

124   \xhead_update_prepare_number:N #1

```

Indent after heading if requested (use old mechanism):

```

125   \xhead_set_after_indention:N \l_xhead_indent_after_bool

```

Produce the space above the heading (using old 2e methods):

```

126 \if@noskipsec \leavevmode \fi
127 \par
128 \if@nobreak
129   \everypar{}
130 \else:
131   \addpenalty \l_xhead_break_penalty_int
132   \addvspace \l_xhead_above_skip
133 \fi:

```

Produce the title of the heading (using old 2e methods)

```

134 \group_begin:
135   \xhead_title_decl:
136   {
137     \@changefrom{\hskip \l_xhead_indent_dim \scan_stop:
138               \tl_if_empty:NF \l_xhead_number_tl
139               { \xhead_number_format:n \l_xhead_number_tl }
140             }
141     \interlinepenalty \@M #6\@par
142   }
143 \group_end:

```

Generate toc entry and mark if required:

```

144 \xhead_generate_toc_and_mark:NNNNNNN
145   #1 \BooleanFalse \BooleanFalse % no forcing in 2e
146   \l_xhead_name_tl
147   \l_xhead_number_tl
148   \l_xhead_toc_title_tl
149   \l_xhead_mark_title_tl

```

Produce the space below the heading and the handling of following paragraph (using old 2e methods):

```

150 \par \nobreak
151 \vspace \l_xhead_below_skip
152 \@afterheading
153 \ignorespaces
154 }

```

3.5 The template 3a-vertical

Right now this template is more or less equal to 2e-vertical except that it supports the extended heading arguments with the exception of #7.

```

155 \DeclareTemplateInterface {heading} {3a-vertical} {7}
156 {
157   name           : tokenlist           = ??? ,
158   level          : integer             = 0 ,
159   break-penalty  : integer             = -300 ,
160   indent         : length              = 0pt ,

```

```

161   above-skip      : skip                = Opt    ,
162   below-skip     : skip                = Opt    ,
163   format         : tokenlist           = indent-by-number ,
164   parshape       : instance {parshape} = plain  ,
165   justification  : instance {justification} = flush-left ,
166   font           : tokenlist           =          ,
167   number-format  : function {1}        = \xhead_default_number_format:n {#1} ,
168   indent-after   : boolean             = false  ,
169 }

```

`\l_xhead_title_format_tl`

```

170 \tl_new:N \l_xhead_title_format_tl
171 \tl_new:N \l_xhead_title_font_tl
172
173 \cs_new:Npn \xhead_title_parshape:    {}
174 \cs_new:Npn \xhead_title_justification: {}
175
176 \DeclareTemplateCode {heading} {3a-vertical} {7}
177 {
178   name           = \l_xhead_name_tl      ,
179   level         = \l_xhead_level_int    ,
180   break-penalty = \l_xhead_break_penalty_int ,
181   indent        = \l_xhead_indent_dim   ,
182   above-skip    = \l_xhead_above_skip   ,
183   below-skip    = \l_xhead_below_skip   ,
184   format        = \l_xhead_title_format_tl ,
185   parshape      = \xhead_title_parshape: ,
186   justification = \xhead_title_justification: ,
187   font          = \l_xhead_title_font_tl ,
188   number-format = \xhead_number_format:n ,
189   indent-after  = \l_xhead_indent_after_bool ,
190 }
191 {
192   \AssignTemplateKeys

```

Set title text for TOC and mark entry:

```

193   \xhead_set_titles:nnn {#2} {#4} {#6}

```

If heading gets a number, increment counter and set `\l_xhead_number_tl`.

```

194   \xhead_update_prepare_number:N #1

```

Indent after heading if requested (use old mechanism)

```

195   \xhead_set_after_indention:N \l_xhead_indent_after_bool

```

Produce the space above the heading (using old 2e methods)

```

196 \if@noskipsec \leavevmode \fi
197 \par
198 \if@nobreak
199 \everypar{}
200 \else:
201 \addpenalty \l_xhead_break_penalty_int
202 \addvspace \l_xhead_above_skip
203 \fi:

```

Produce the title of the heading

```

204 \xhead_use_title_format:nn \l_xhead_title_format_tl {#6}
205

```

Generate toc entry and mark:

```

206 \xhead_generate_toc_and_mark:NNNNNNN
207 #1 #3 #5
208 \l_xhead_name_tl
209 \l_xhead_number_tl
210 \l_xhead_toc_title_tl
211 \l_xhead_mark_title_tl

```

Produce the space below the heading and the handling of following paragraph (using old 2e methods)

```

212 \par \nobreak
213 \vspace \l_xhead_below_skip
214 \@afterheading
215 \ignorespaces
216 }

```

`\xhead_use_title_format:nn` Select a formatting command (based on first arg string) and use it to format the title (second arg).

The formatting command makes a bunch of assumptions with respect to the variable names used by the templates (it might ignore some of them):

```

\l_xhead_title_font_tl      : font declarations for title text
\l_xhead_indent_dim        : indentation from the left of title
\xhead_title_parshape:     : parshape instance to use for title
\xhead_title_justification: : justification instance to use for title
\l_xhead_number_tl         : raw title number or empty
\xhead_number_format:n     : formatting command for title number

```

In addition it takes a single mandatory argument holding the title text.

```

217 \cs_new:Npn \xhead_use_title_format:nn #1 {
218 \cs_if_exist:cTF {xhead_format_ #1:n}
219 { \use:c{xhead_format_ #1:n} }
220 { \ERROR xhead format not defined }
221 }

```

xhead_format_indent-by-number:n Implementing vertical version of \@startsection ...

```
222 \cs_new:cpn {xhead_format_indent-by-number:n} #1 {
223   \group_begin:
224     \l_xhead_title_font_tl
225     \hbox_set:Nn \l_tmpa_box
226       {
227         \tl_if_empty:NF \l_xhead_number_tl
228           { \xhead_number_format:n \l_xhead_number_tl }
229       }
230
231     \UseInstance{measure}{fullwidth}
232       {\l_xhead_indent_dim + \box_wd:N \l_tmpa_box }
233
234     \xhead_title_parshape:
235     \xhead_title_justification:
236     \interlinepenalty \@M
237
238     \noindent \hbox_overlap_left:n { \box_use:N \l_tmpa_box }
239     #1\@@par
240   \group_end:
241 }
```

xhead_format_number-above-text:n

```
242 \cs_new:cpn {xhead_format_number-above-text:n} #1 {
243   \group_begin:
244     \l_xhead_title_font_tl
245     \xhead_title_parshape:
246     \xhead_title_justification:
247     \interlinepenalty \@M
248
249     \tl_if_empty:NF \l_xhead_number_tl
250     {
251       \xhead_number_format:n \l_xhead_number_tl
252     }
253
254     #1\@@par
255   \group_end:
256 }
```

```
257
258 \input{xtextblock.sty}
```

```
259
260
261 \DeclareTemplateInterface {heading} {3b-vertical} {7}
262 {
263   name           : tokenlist           = ???   ,
264   level          : integer             = 0     ,
265   break-penalty  : integer             = -300  ,
```

```

266 indent           : length           = Opt   ,
267 above-skip       : skip             = Opt   ,
268 below-skip        : skip             = Opt   ,
269 title-format      : instance {textblock-2} = heading ,
270 number-format     : function {1}       = \xhead_default_number_format:n {#1} ,
271 indent-after      : boolean           = false ,
272 }
273
274 \cs_new:Npn \xhead_title_format:nn #1#2 {}
275
276 \DeclareTemplateCode {heading} {3b-vertical} {7}
277 {
278   name           = \l_xhead_name_tl           ,
279   level          = \l_xhead_level_int        ,
280   break-penalty  = \l_xhead_break_penalty_int ,
281   indent         = \l_xhead_indent_dim       ,
282   above-skip     = \l_xhead_above_skip       ,
283   below-skip     = \l_xhead_below_skip       ,
284   title-format   = \xhead_title_format:nn    ,
285   number-format  = \xhead_number_format:n    ,
286   indent-after   = \l_xhead_indent_after_bool ,
287 }
288 {
289   \AssignTemplateKeys
290
291   \xhead_set_titles:nnn {#2} {#4} {#6}
292
293   \xhead_update_prepare_number:N #1
294
295   \xhead_set_after_indention:N \l_xhead_indent_after_bool
296
297   \if@noskipsec \leavevmode \fi
298   \par
299   \if@nobreak
300     \everypar{}
301   \else:
302     \addpenalty \l_xhead_break_penalty_int
303     \addvspace \l_xhead_above_skip
304   \fi:
305
306   \xhead_title_format:nn \l_xhead_number_tl {#6}
307
308   \xhead_generate_toc_and_mark:NNNNNNN
309     #1 #3 #5
310     \l_xhead_name_tl
311     \l_xhead_number_tl
312     \l_xhead_toc_title_tl
313     \l_xhead_mark_title_tl
314
315   \par \nobreak

```

```
316 \vspace \l_xhead_below_skip
317 \@afterheading
318 \ignorespaces
319 }
320
321
322 \endinput
323 \end{package}
```