

The latex-lab-firstaid package

Temporary patches to external packages needed for the tagging project

L^AT_EX Project*

v0.86c 2026-09-15

Abstract

1 Introduction

The following contains small temporary changes to external packages to avoid errors with the new tagging code.

Similar to the main firstaid package the goal is to remove the patches once the packages have been updated.

2 Implementation

```
1 <*package>
2 <@@=tag>

3 \ProvidesPackage {latex-lab-testphase-firstaid} [%
4   \ltlabfirstaiddatetime\space v\ltlabfirstaidversion\space
5   Temporary patches to external packages needed for the tagging project]
```

`\FirstAidNeededT` This is a very simple help to ensure that we only apply first aid to an unmodified package or class. It only works in the case the file has already been loaded and the `csname \ver@#1.#2` got defined (holding the current date, version, and short description info). We then compare its content to a frozen string and make the modification `#3` only if both agree. If they differ we assume that the package/class in question got updated by its maintainer.

```
6 \ExplSyntaxOn
7 \providecommand\FirstAidNeededT[3]{
8   \exp_args:Nce\str_if_eq:onF{ver@#1.#2}{#3}
9   { \typeout{=>~ First~ Aid~ for~ #1.#2~ no~ longer~ applied!^^J
10     \@spaces Expected:^^J
11     \@spaces\@spaces #3^^J
12     \@spaces but~ found:^^J
13     \@spaces\@spaces \use:c{ver@#1.#2}^^J
14     \@spaces so~ I'm~ assuming~ it~ got~ fixed.
15   } }
```

*Initial implementation done by Ulrike Fischer

```

16 \exp_args:Nce\str_if_eq:ont{ver@#1.#2}{#3}
17 }

```

(End of definition for \FirstAidNeededT.)

2.1 tikz/pgf

tikz inputs libraries with the primitive `\input` command. This means that these libraries are not listed in the file list written by `\listfiles` and the new tagging status report created with the `check-tagging-status` key.

We therefore redefine one `pgf` command to use the L^AT_EX `\input` command. Check <https://github.com/pgf-tikz/pgf/issues/1424> for changes.

```

18 \AddToHook{package/pgfrcs/after}
19 {\def\pgfutil@InputIfFileExists#1#2#3{\pgfutil@IfFileExists{#1}{\input{#1}\relax#2}{#3}}}

```

2.2 AMS classes

Instead of putting the first aid code into the format we now write it to files that are loaded after the class has been loaded.

```

20 \AddToHook{class/amsbook/after}[latex-lab-firstaid]{\input{latex-lab-firstaid-amsbook.ltx}}
21 \AddToHook{class/amsart/after}[latex-lab-firstaid]{\input{latex-lab-firstaid-amsart.ltx}}
22 \AddToHook{class/amsproc/after}[latex-lab-firstaid]{\input{latex-lab-firstaid-amsproc.ltx}}

23 </package>
24 <*amscls>
25 \ExplSyntaxOn

```

The `amsart`, `amsbook` and `amsproc` classes do not use `\@author` to store the author list but a command `\authors`. To be able to nevertheless use the authors in the xmp-metadata we map `\@author` to this new command.

The authors are set with a `trivlist`, this leads to faulty spacing and wrong tagging. This is not yet handled here, but an example how to correct this is in the `test-amsart-title` test.

```

26 \def\@author{\authors}

```

In these classes the abstract is stored in a `\vbox`, as it ends with a list an explicit `\par` is needed.

```

27 \renewenvironment{abstract}{%
28   \ifx\maketitle\relax
29     \ClassWarning{\@classname}{Abstract~ should~ precede~
30       \protect\maketitle\space in~ AMS~ document~ classes;~ reported}%
31   \fi
32   \global\setbox\abstractbox=\vtop \bgroup
33   \normalfont\Small
34   \list{}{\labelwidth\z@
35     \leftmargin3pc \rightmargin\leftmargin
36     \listparindent\normalparindent \itemindent\z@
37     \parsep\z@ \@plus\p@
38     \let\fullwidthdisplay\relax
39   }%
40   \item[\hskip\labelsep\scshape\abstractname.]%
41 }{%
42   \endlist

```

```

43 \par % <--- added
44 \egroup
45 \ifx\@setabstract\relax \@setabstracta \fi
46 }

```

In the ams classes's `\newtheoremstyle` has different defaults compared to the standalone `amsthm` package. Therefore the classes have their own implementation of the command that we need to overwrite as well.

Provide a saved version of the `\newtheoremstyle` code in the format to be able to reinstall it after the class got loaded.

```

47 </amscls>
48 <+package> \cs_set_eq:NN \newtheoremstyle@saved \newtheoremstyle
49 <*amscls>

```

Then reinstall it and also change the default for the vertical space.

```

50 \cs_set_eq:NN \newtheoremstyle \newtheoremstyle@saved
51 \tl_set:Nn \newtheoremstyle@vspace@default {
52   .5\baselineskip \@plus .2\baselineskip \@minus .2\baselineskip
53 }

```

2.2.1 Headings

The classes redefine `\@startsection` and define their own heading commands which are not tagging compatible. This code is therefore disabled and replaced by standard definitions that process the heading arguments and call a suitable template instance. The top-level definitions are identical for all classes (only the heading instances then differ).

```

54 \DeclareDocumentCommand \part {s ={\shorttitle}o m}
55 { \ParseLaTeXeHeading {part} {#1} {#2} {#3} }

```

Chapters only exist in `amsbook` and their top-level definition is also slightly special as we need a different heading instance for numbered and unnumbered chapters.

```

56 <*amsbook>
57 \DeclareDocumentCommand \chapter {s ={\shorttitle}O{} m}
58 {
59   \IfBooleanTF{#1}
60   { \ParseLaTeXeHeading {chapter*} {#1} {#2} {#3} }
61   { \ParseLaTeXeHeading {chapter} {#1} {#2} {#3} }
62 }
63 </amsbook>

```

```

64 \DeclareDocumentCommand \section {s ={\shorttitle}o m}
65 { \ParseLaTeXeHeading {section} {#1} {#2} {#3} }

```

The `\specialsection` command is a variation of the section layout so has its own heading instance.

```

66 \DeclareDocumentCommand \specialsection {s ={\shorttitle}o m}
67 { \ParseLaTeXeHeading {specialsection} {#1} {#2} {#3} }

68 \DeclareDocumentCommand \subsection {s ={\shorttitle}o m}
69 { \ParseLaTeXeHeading {subsection} {#1} {#2} {#3} }

70 \DeclareDocumentCommand \subsubsection {s ={\shorttitle}o m}
71 { \ParseLaTeXeHeading {subsubsection} {#1} {#2} {#3} }

```

```

72 \DeclareDocumentCommand \paragraph {s ={\shorttitle}o m}
73   { \ParseLaTeXeHeading {paragraph} {#1} {#2} {#3} }

74 \DeclareDocumentCommand \subparagraph {s ={\shorttitle}o m}
75   { \ParseLaTeXeHeading {subparagraph} {#1} {#2} {#3} }

```

The heading template instances have some subtle differences in the different AMS classes. We therefore set all of them up by document class, even though some of the code is the same.

```

76 <amsbook>

77 \DeclareInstance{heading}{part}{display}
78   {
79     , name                = part
80     , level               = -1
81     , placement           = page
82     , start-code          = \cleardoublepage \thispagestyle{empty}
83                           \null \vfil \markboth{}{}
84     , final-code          = \vfil \vfil \newpage\newpage \thispagestyle{empty}
85     , mark-cmd            = \partmark {#1}
86     , heading-decls       = \centering\huge\bfseries
87     , headformat-instance = part
88     , prefix              = \partname
89     , number-format       = \theheading .
90     , title-decls         = \Huge
91   }
92 \DeclareInstance{headformat}{part}{display}
93   {
94     prefix-number-sep = \WordSpaceAmount{1},
95   }

```

Current heading templates aren't flexible enough and do not properly support altering the layout for unnumbered headings compared to their numbered counterparts. For this reason we have two heading instances, one for `\chapter` and one for `\chapter*`.

```

96 \DeclareInstance{heading}{chapter}{display}
97   {
98     , name                = chapter
99     , level               = 0
100    , mark-cmd            = \chaptermark {#1}
101    , para-indent         = true
102    , placement           = top
103    , after-penalty-vspace = 32pt
104    , after-vspace        = 22pt
105    , heading-decls       = \normalsize\mdseries\centering
106    , headformat-instance = chapter
107    , prefix              = \MakeUppercase{\chaptername}
108    , number-format       = \thechapter
109    , title-decls         = \fontsize{\@xivpt}{18} \bfseries
110    , contents-extra      = \addtocontents{lof}{\protect\addvspace{10\p@}}
111                           \addtocontents{lot}{\protect\addvspace{10\p@}}
112  }

113 \DeclareInstanceCopy{heading}{chapter*}{chapter}
114 \EditInstance{heading}{chapter*}{ after-penalty-vspace = 62pt }

```

```

115 \DeclareInstance{headformat}{chapter}{display}
116 {
117     , heading-indent = 0pt
118     , separator-b    = \par \nobreak \vspace{18pt}
119 }

120 \DeclareInstance{heading}{section}{display}
121 {
122     , name           = section
123     , level          = 1
124     , mark-cmd       = \sectionmark{#1}
125     , para-indent    = true
126     , before-vspace  = 0.7\linespacing plus \linespacing
127     , after-vspace   = 0.5\linespacing
128     , heading-decls  = \normalfont \bfseries \centering
129     , headformat-instance = section
130     , number-format  = \theheading .
131     , number-decls   = \upshape
132 }
133 \DeclareInstance{headformat}{section}{hang}
134 {
135     separator-b = \hspace{0.5em}
136 }

    A \specialsection is like a \section with two changes in the parameter settings.

137 \DeclareInstanceCopy{heading}{specialsection}{section}
138 \EditInstance{heading}{specialsection}
139 {
140     , before-vspace = 2\linespacing plus \linespacing
141     , heading-decls = \large \scshape \centering
142 }

143 \DeclareInstance{heading}{subsection}{runin}
144 {
145     , name           = subsection
146     , level          = 2
147     , mark-cmd       = \subsectionmark{#1}
148     , before-vspace  = 0.5\linespacing plus 0.7\linespacing
149     , heading-decls  = \normalfont \bfseries
150     , headformat-instance = subsection
151     , number-format  = \theheading .
152     , number-decls   = \upshape
153     , title-format   = #1 \AddPunct{.}
154 }
155 \DeclareInstance{headformat}{subsection}{runin}
156 {
157     , separator-b    = \hspace{0.5em}
158     , heading-indent = \normalparindent
159 }

160 \DeclareInstance{heading}{subsubsection}{runin}
161 {
162     , name           = subsubsection
163     , level          = 3
164     , mark-cmd       = \subsubsectionmark{#1}

```

```

165     , before-vspace      = 0pt
166     , heading-decls      = \normalfont \itshape
167     , headformat-instance = subsubsection
168     , number-format      = \theheading .
169     , number-decls       = \upshape
170     , title-format       = #1 \AddPunct{.}
171   }
172 \DeclareInstance{headformat}{subsubsection}{runin}
173 {
174     , separator-b        = \hspace{0.5em}
175     , heading-indent     = \normalparindent
176 }

177 \DeclareInstance{heading}{paragraph}{runin}
178 {
179     , name                = paragraph
180     , level               = 4
181     , mark-cmd            = \paragraphmark{#1}
182     , before-vspace      = 0pt
183     , after-vspace       = \fontdimen2 \font
184     , heading-decls      = \normalfont
185     , headformat-instance = paragraph
186     , number-format      = \theheading .
187     , number-decls       = \upshape
188     , title-format       = #1 \AddPunct{.}
189 }
190 \DeclareInstance{headformat}{paragraph}{runin}
191 {
192     , separator-b        = \hspace{0.5em}
193     , heading-indent     = \normalparindent
194 }

195 \DeclareInstance{heading}{subparagraph}{runin}
196 {
197     , name                = subparagraph
198     , level               = 5
199     , mark-cmd            = \subparagraphmark{#1}
200     , before-vspace      = 0pt
201     , after-vspace       = \fontdimen2 \font
202     , heading-decls      = \normalfont
203     , headformat-instance = subparagraph
204     , number-format      = \theheading .
205     , number-decls       = \upshape
206     , title-format       = #1 \AddPunct{.}
207 }
208 \DeclareInstance{headformat}{subparagraph}{runin}
209 {
210     , separator-b        = \hspace{0.5em}
211 }

212 </amsbook>

213 <*amsart>

```

The use of `\mdseries` in several of the instances is really odd (and not done in

amsbook or amsproc) but it is what the current class generates as a layout, so we keep it that way.

```

214 \DeclareInstance{heading}{part}{display}
215 {
216   , name                = part
217   , level                = -1                % -1 or 0 ?
218   , mark-cmd            = \partmark{#1}
219   , para-indent         = true
220   , before-vspace       = \linespacing plus \linespacing
221   , after-vspace        = 0.5 \linespacing
222   , heading-decls       = \normalfont \bfseries \raggedright
223   , headformat-instance = part
224   , prefix              = \partname
225   , number-format       = \theheading .
226   , number-decls       = \mdseries \upshape % proc: has just \upshape
227 }
228 \DeclareInstance{headformat}{part}{hang}
229 {
230   , separator-b = \hspace{0.5em}
231 }
232 \DeclareInstance{heading}{section}{display}
233 {
234   , name                = section
235   , level                = 1
236   , mark-cmd            = \sectionmark{#1}
237   , para-indent         = true
238   , before-vspace       = 0.7\linespacing plus \linespacing
239   , after-vspace        = 0.5\linespacing
240   , heading-decls       = \normalfont \scshape \centering
241   , headformat-instance = section
242   , number-format       = \theheading .
243   , number-decls       = \mdseries \upshape
244 }
245 \DeclareInstance{headformat}{section}{hang}
246 {
247   , separator-b = \hspace{0.5em}
248 }

```

The `\specialsection` command is a variation of the `section` layout.

```

249 \DeclareInstanceCopy{heading}{specialsection}{section}
250 \EditInstance{heading}{specialsection}
251 {
252   , before-vspace = \linespacing plus \linespacing
253   , heading-decls = \normalfont \centering
254 }
255 \DeclareInstance{heading}{subsection}{runin}
256 {
257   , name                = subsection
258   , level                = 2
259   , mark-cmd            = \subsectionmark{#1}
260   , before-vspace       = 0.5\linespacing plus 0.7\linespacing
261   , heading-decls       = \normalfont \bfseries

```

```

262     , headformat-instance = subsection
263     , number-format      = \theheading .
264     , number-decls       = \mdseries \upshape
265     , title-format       = #1 \AddPunct{.}
266   }
267 \DeclareInstance{headformat}{subsection}{runin}
268 {
269   , separator-b = \hspace{0.5em}
270 }

271 \DeclareInstance{heading}{subsubsection}{runin}
272 {
273   , name              = subsubsection
274   , level             = 3
275   , mark-cmd          = \subsubsectionmark{#1}
276   , before-vspace     = 0.5\linespacing plus 0.7\linespacing
277   , heading-decls     = \normalfont \itshape
278   , headformat-instance = subsubsection
279   , number-format     = \theheading .
280   , number-decls      = \mdseries \upshape
281   , title-format      = #1 \AddPunct{.}
282 }
283 \DeclareInstance{headformat}{subsubsection}{runin}
284 {
285   , separator-b = \hspace{0.5em}
286 }

287 \DeclareInstance{heading}{paragraph}{runin}
288 {
289   , name              = paragraph
290   , level             = 4
291   , mark-cmd          = \paragraphmark{#1}
292   , before-vspace     = 0pt
293   , after-vspace      = \fontdimen2\font
294   , heading-decls     = \normalfont
295   , headformat-instance = paragraph
296   , number-format     = \theheading .
297   , number-decls      = \mdseries \upshape
298   , title-format      = #1 \AddPunct{.}
299 }
300 \DeclareInstance{headformat}{paragraph}{runin}
301 {
302   , separator-b = \hspace{0.5em}
303 }

304 \DeclareInstance{heading}{subparagraph}{runin}
305 {
306   , name              = subparagraph
307   , level             = 5
308   , mark-cmd          = \subparagraphmark{#1}
309   , before-vspace     = 0pt
310   , after-vspace      = \fontdimen2\font
311   , heading-decls     = \normalfont
312   , headformat-instance = subparagraph
313   , number-format     = \theheading .

```

```

314     , number-decls      = \mdseries \upshape
315     , title-format      = #1 \AddPunct{.}
316   }
317 \DeclareInstance{headformat}{subparagraph}{runin}
318 {
319     , separator-b = \hspace{0.5em}
320 }
321 \</amsart>
322 \<*amsproc>
323 \DeclareInstance{heading}{part}{display}
324 {
325     , name              = part
326     , level              = -1                % -1 or 0 ?
327     , mark-cmd          = \partmark{#1}
328     , para-indent       = true
329     , before-vspace     = \linespacing plus \linespacing
330     , after-vspace      = 0.5 \linespacing
331     , heading-decls     = \normalfont \bfseries \raggedright
332     , headformat-instance = part
333     , prefix            = \partname
334     , number-format     = \theheading .
335     , number-decls      = \upshape
336 }
337 \DeclareInstance{headformat}{part}{hang}
338 {
339     , separator-b = \hspace{0.5em}
340 }
341 \DeclareInstance{heading}{section}{display}
342 {
343     , name              = section
344     , level              = 1
345     , mark-cmd          = \sectionmark{#1}
346     , para-indent       = true
347     , before-vspace     = 0.7\linespacing plus \linespacing
348     , after-vspace      = 0.5\linespacing
349     , heading-decls     = \normalfont \bfseries \centering
350     , headformat-instance = section
351     , number-format     = \theheading .
352     , number-decls      = \upshape
353 }
354 \DeclareInstance{headformat}{section}{hang}
355 {
356     , separator-b = \hspace{0.5em}
357 }

```

The `\specialsection` command is a variation of the `section` layout.

```

358 \DeclareInstanceCopy{heading}{specialsection}{section}
359 \EditInstance{heading}{specialsection}
360 {
361     , before-vspace = \linespacing plus \linespacing
362     , heading-decls = \large \scshape \centering
363 }

```

```

364 \DeclareInstance{heading}{subsection}{runin}
365 {
366   , name           = subsection
367   , level          = 2
368   , mark-cmd       = \subsectionmark{#1}
369   , before-vspace  = 0.5\linespacing plus 0.7\linespacing
370   , heading-decls  = \normalfont \bfseries
371   , headformat-instance = subsection
372   , number-format  = \theheading .
373   , number-decls   = \upshape
374   , title-format   = #1 \AddPunct{.}
375 }
376 \DeclareInstance{headformat}{subsection}{runin}
377 {
378   , separator-b    = \hspace{0.5em}
379   , heading-indent = \normalparindent
380 }
381 \DeclareInstance{heading}{subsubsection}{runin}
382 {
383   , name           = subsubsection
384   , level          = 3
385   , mark-cmd       = \subsubsectionmark{#1}
386   , before-vspace  = 0.5\linespacing plus 0.7\linespacing
387   , heading-decls  = \normalfont \itshape
388   , headformat-instance = subsubsection
389   , number-format  = \theheading .
390   , number-decls   = \upshape
391   , title-format   = #1 \AddPunct{.}
392 }
393 \DeclareInstance{headformat}{subsubsection}{runin}
394 {
395   , separator-b    = \hspace{0.5em}
396   , heading-indent = \normalparindent
397 }
398 \DeclareInstance{heading}{paragraph}{runin}
399 {
400   , name           = paragraph
401   , level          = 4
402   , mark-cmd       = \paragraphmark{#1}
403   , before-vspace  = 0pt
404   , after-vspace   = \fontdimen2\font
405   , heading-decls  = \normalfont
406   , headformat-instance = paragraph
407   , number-format  = \theheading .
408   , number-decls   = \upshape
409   , title-format   = #1 \AddPunct{.}
410 }
411 \DeclareInstance{headformat}{paragraph}{runin}
412 {
413   , separator-b    = \hspace{0.5em}
414   , heading-indent = \normalparindent
415 }
416 \DeclareInstance{heading}{subparagraph}{runin}

```

```

417 {
418   , name                = subparagraph
419   , level                = 5
420   , mark-cmd            = \subparagraphmark{#1}
421   , before-vspace       = 0pt
422   , after-vspace        = \fontdimen2\font
423   , heading-decls       = \normalfont
424   , headformat-instance = subparagraph
425   , number-format       = \theheading .
426   , number-decls        = \upshape
427   , title-format = #1 \AddPunct{.}
428 }
429 \DeclareInstance{headformat}{subparagraph}{runin}
430 {
431   , separator-b = \hspace{0.5em}
432 }
433 \</amsproc>
434 \ExplSyntaxOff
435 \</amscs>
436 \<*package>
437 \ExplSyntaxOff

```

3 fncychap

```

438 \AddToHook{package/fncychap/after}[latex-lab-firstaid]{\input{latex-lab-firstaid-
fncychap.ltx}}
439 \</package>
440 \<*fncychap>
441 \ExplSyntaxOn

```

The `fncychap` package has its own way of offering customizations including arbitrary rewrites for the commands `\DOCH`, `\DOTI`, and `\DOTIS`. In order to support customizations in existing documents (defining chapter layout variations in this fashion) we define a `headformat` in which the three commands are simply called to typeset the heading, i.e., the template doesn't offer any customizations through keys—everything is done through the interfaces of `fncychap` as before.

For the same reason most of the keys of the chapter instance are nonfunctional, i.e., they exist but aren't used with a few exceptions. After all, the purpose here is to make `fncychap` tagging compatible and not to extend its functionality or offer additional customization possibilities.

The two values in the heading chapter instance that we need to alter are `after-vspace` and set it to zero (because this is handled inside `\DOTI` or `\DOTIS`) and `headformat-instance` as we want to call our own instance suitable to deal with the customization possibilities offered by `fncychap`. Thus we edit the default `heading chapter` instance in the following way.

```

442 \EditInstance{heading}{chapter}{
443   after-vspace      = 0pt
444   , headformat-instance = fncychap
445 }

```

This means we have to declare a new template `fncychap` of type `headformat` (with no keys as explained above):

```
446 \DeclareTemplateInterface{headformat}{fncychap}{6}{}

```

The code for this template is fairly simple.

```
447 \DeclareTemplateCode{headformat}{fncychap}{6}{}
448 {

```

Output some debugging info at the start.

```
449   \__head_show_arguments:nnnnnn {#1}{#2}{#3}{#4}{#5}{#6}

```

Everything else happens in a group to keep changes local.

```
450   \group_begin:

```

First, make a link target (with slight difference depending on whether or not the chapter is numbered).

```
451   \bool_if:NTF \l__head_unnumbered_bool
452     { \MakeLinkTarget [ \l__head_name_tl ] {} }
453     { \MakeLinkTarget { \l__head_name_tl } }

```

Next follow some default paragraph settings that may or may not be overwritten by declarations for a specific chapter layout from the package.

```
454   \parindent Opt
455   \raggedright
456   \interlinepenalty 10000
457   \normalfont

```

Then we have a few tagging socket settings for `\parbox` or `minipage` environments, because some heading designs use `\parboxes` and they need the special settings in the context of a heading to make the tagging work.

```
458   \AssignTaggingSocketPlug{parbox/before}{heading}
459   \AssignTaggingSocketPlug{minipage/before}{heading}
460   \AssignTaggingSocketPlug{para/restore}{noop}

```

Finally, we have the bare heading design as used in the package. When the heading is numbered then `\DOCH` is responsible to place the heading prefix (e.g., Chapter) and the chapter number.

```
461 %   \leavevmode % this can wait ...
462   \bool_if:NF \l__head_unnumbered_bool
463     { \DOCH }

```

If we are in “main matter” then `\DOTI` typesets the chapter title, otherwise `\DOTIS` is used to typeset the chapter title.

```
464   \legacy_if:NTF { @mainmatter }
465     { \DOTI{#4} }
466     { \DOTIS{#4} }

```

We finish off with a `\par` to ensure that we are back in vertical mode and then close the group.

```
467   \par
468   \group_end:
469 }

```

We use this template to create the `headformat` instance used in the `heading` instance. Since the template has no keys this is a trivial declaration:

```
470 \DeclareInstance{headformat}{fncychap}{fncychap}{}
```

To put a `<heading-number>` structure around the number without really knowing how the heading layout is done, we hook into `\thechapter`. The `\thespecialchapter` command takes one argument which is normally `\arabic` but redefined in `\appendix` to become `\Alph`. This isn't very flexible but works for the situations for which `fncychap` was written.

```
471 \protected\def\thechapter{\thespecialchapter\arabic}

472 \def\appendix{\par
473   \setcounter{chapter}{0}
474   \setcounter{section}{0}
475   \gdef\@chapapp{\appendixname}
476   \protected\gdef\thechapter{\thespecialchapter\Alph}
477 }
```

The `\thespecialchapter` ensures that `hmode` is started and then normally generates a `<heading-number>` structure but only if we aren't already in an `<Lb1>`. It then typesets the current chapter counter either using `\arabic` or `\Alph`

```
478 \protected\def\thespecialchapter#1{
479   \leavevmode
480   \tag_if_in:nTF{Lb1}
481     { #1 {chapter} }
482     {
483       \UseTaggingSocket{inline/begin}{tag=heading-number}
484       #1 {chapter}
485       \UseTaggingSocket{inline/end}
486     }
487 }
```

`\mghrulefill`

This here is a special rule definition from `fncychap` that we overwrite to add the necessary tagging support.

```
488 \protected\def\mghrulefill#1{\leavevmode
489   \UseTaggingSocket{rule}{}{
490     \leaders\hrule\@height #1\hfill\kern\z@
491   }
492 }
```

(End of definition for `\mghrulefill`. This function is documented on page ??.)

```
493 \ExplSyntaxOff
```

```
494 </fncychap>
495 <*package>
```

3.1 verse

The `verse` package has its own definition of the `verse` environment, which would tag correctly, except that it is overwritten by the block code in the hook `begindocument/before`. So the simplest way to make tagging work is to reinstall the package version afterwards, which is what we are doing here.

```

496 \AddToHook{package/verse/after}[latex-lab-firstaid]{%
497   \FirstAidNeededT{verse}{sty}{2014/05/10 v2.4b verse typesetting}%
498   {%
499     \AtBeginDocument{%
500       \renewenvironment{verse}[1][\linewidth]{%
501         \stepcounter{verse@envctr}%
502         \setcounter{poemline}{0}\refstepcounter{poemline}%
503         \setcounter{vslineno}{1}%
504         \let\@vscentercr
505         \list{}{\itemsep \z@
506               \itemindent -\vindent
507               \listparindent\itemindent
508               \parsep \stanzaskip
509               \ifdim #1 < \linewidth
510                 \rightmargin \z@
511                 \setlength{\leftmargin}{\linewidth}%
512                 \addtolength{\leftmargin}{-#1}%
513                 \addtolength{\leftmargin}{-0.5\leftmargin}%
514               \else
515                 \rightmargin \leftmargin
516               \fi
517               \addtolength{\leftmargin}{\vindent}}}%
518         \item[]%
519       }%
520       {\endlist}%
521     }%
522   }%
523 }
```

Of course, this means that the optional argument of the environment then only accepts a length value and not any more a key value list for altering the environment settings.

A more elaborate version could be something like this that allows key/val and legacy interface. Or one could extend the list template to support a `list-width` key.

```

\ExplSyntaxOn
\cs_new_protected:Npn \ExtractAndDropKey #1#2#3#4#5 {
  \tl_set_eq:NN #4 \c_novalue_tl % or empty?
  \keys_define:nn { #1 } { #2 .code:n = \tl_set:Nn #4{##1} }
  \keys_set_known:nnN { #1 } { #3 } #5
}
\ExplSyntaxOff
```

```

% Change the env definition for verse matching verse.sty
% This keeps the verse.sty interface as it is and only adjusts the
% main environment to use the basic list env with the verse.sty
% specific settings.
\makeatletter
```

```

\AddToHook{package/verse/after}{%
  \AtBeginDocument{%
    \RenewDocumentEnvironment{verse}{={verse-width}!0{\linewidth}}%
    {%
      \stepcounter{verse@envctr}%
      \setcounter{poemline}{0}\refstepcounter{poemline}%
      \setcounter{vslineno}{1}%
      \let\=\@vscentercr
    }%
    %
    \ExtractAndDropKey{verse}{verse-width}{#1}\@vswidth\@vsremainingkvlist
    % If other keys have been specified but not verse-width we have no
    % default for \@vswidth and need to set it again
    \ExpandArgs{o}\IfNoValueT \@vswidth
      {\def\@vswidth{\linewidth}}}%
    %
    % This is a bit ugly but we can't stick \cs{@vsremainingkvlist} into
    % the instance argument as keys are expected to be visible on
    % top-level not hidden inside a macro. The alternative is to push
    % in \verb=#1= but then the key/value \verb/verse-width=.../ is
    % passed into the instance which is not known there (not harmful as
    % it will get ignored but noticeably more and unnecessary
    % processing).
    %
    \def\next##1{%
      \UseInstance{blockenv}{list}%
      {%
        item-indent = -\vindent,%
        para-indent = -\vindent,%
        para-vspace = \stanzaskip,%
        item-skip = Opt,%
        left-margin = (\linewidth-\@vswidth)/2+\vindent,%
        right-margin = \ifdim\@vswidth<\linewidth Opt
                      \else (\linewidth-\@vswidth)/2\fi,%
        ##1%
      }%
      \ExpandArgs{o}\next\@vsremainingkvlist
      \item\relax
    }\endblockenv}%
  }%
}
\makeatother

```

3.2 cleveref

The cleveref package redefines `\@makefnstext` and this means that the patches in the new footnote code fails. We use a hook instead.

```

524 \AddToHook{package/cleveref/after}
525 {
526   \let\@makefnstext\cref@old@makefnstext

```

```

527 \AddToHook{cmd/@makefnmark/before}{%
528   \cref@constructprefix{footnote}{\cref@result}%
529   \protected@edef\cref@currentlabel{%
530     [footnote][\arabic{footnote}][\cref@result]%
531     \p@footnote\@thefnmark}}
532 }

```

3.3 booktabs

In some cases booktabs inserts a `\multispan` into the table (through the commands `\@cmidruleb` and `\@cmidrulea` and this then errors with the tagging code. This affects both tabular and longtable (but longtable more as booktabs handles lines in longtable differently). See also issue <https://github.com/latex3/tagging-project/issues/69>

```

533 \ExplSyntaxOn
534 \AddToHook{package/booktabs/after}
535 {
536   \def\@cmidrulea{
537     \multispan\@cmidla
538     &\multispan\@cmidlb
539     \unskip\hskip\cmrkern@l
540   {
541     \UseTaggingSocket{tbl/leaders/begin}
542     \CT@arc@leaders\hrule \@height\@thisrulewidth\hfill\kern\z@}
543     \hskip\cmrkern@r
544     \UseTaggingSocket{tbl/leaders/end}
545     \tbl_gdecr_row_count:
546     \cr}
547
548   \def\@cmidruleb{%
549     \multispan\@cmidlb
550     \unskip\hskip \cmrkern@l%
551   {
552     \UseTaggingSocket{tbl/leaders/begin}
553     \CT@arc@leaders\hrule \@height\@thisrulewidth\hfill\kern\z@}
554     \hskip\cmrkern@r
555     \UseTaggingSocket{tbl/leaders/end}
556     \tbl_gdecr_row_count:
557     \cr}
558   }
559 \ExplSyntaxOff

```

3.4 fancyvrb

The firstaid adds first partial tagging support to the environments of fancyvrb (inline verbatim is untested). This supports then also packages like minted which internally uses fancyvrb and classes like l3doc (where currently the verbatim environment based on fancyvrb is overwritten by the block code). The environments are surrounded by a `verbatim` structure, every line by a `codeline` structure. Line numbers are tagged as `Lbl`, inside of the `codeline` structure. The frame lines are marked as artifact.

`\FV@LeaveVMode` If we are in vmode we have to open a semantic-para structure, if we are in hmode we have to set para mode to flattened before the fancyvrb code issues the `\par`. The closing

of the semantic-para structure is handled by the doendpe code in the block code.

```

560 \ExplSyntaxOn
561 \AddToHook{package/fancyvrb/after}
562 {
563   \def\FV@LeaveVMode{%
564     \if@noskipsec
565       \leavevmode
566     \else
567       \if\FV@ResetMargins\if@inlabel\leavevmode\fi\fi
568     \fi
569     \ifvmode
570       \@nparlisttrue
571       \UseTaggingSocket{para/semantic/begin}{}
572     \else
573       \tagpdfsetup{para/flattened=true}
574       \@nparlistfalse
575       \unskip\par
576     \fi
577   }

```

(End of definition for \FV@LeaveVMode.)

\FV@List At the begin of the list code we have to tag the frame as artifact and start the `verbatim` structure

```

578 \def\FV@List#1{%
579   \begingroup
580   \FV@UseKeyValues
581   \FV@LeaveVMode
582   \if@inlabel\else\setbox\@labels=\box\voidb@x\fi
583   \FV@ListNesting{#1}%
584   \FV@ListParameterHook
585   \FV@ListVSpace
586   \FV@SetLineWidth
587   \FV@InterLinePenalty
588   \let\FV@ProcessLine\FV@ListProcessLine@i
589   \FV@CatCodes
590   \FV@FormattingPrep
591   \FV@ObeyTabsInit
592   \cs_if_exist:NT \FV@BeginListFrame
593   {
594     \UseTaggingSocket{mc}{artifact}{\FV@BeginListFrame}
595   }
596   \UseTaggingSocket{struct/begin}{tag=\UseStructureName{block/verbatim}}
597 }

```

(End of definition for \FV@List.)

\FV@EndList At the end of the list code we close the `verbatim` structure and tag the frame as artifact.

```

598 \def\FV@EndList{%
599   \FV@ListProcessLastLine
600   \UseTaggingSocket{struct/end}
601   \cs_if_exist:NT \FV@EndListFrame
602   {
603     \UseTaggingSocket{mc}{artifact}{\FV@EndListFrame}

```

```

604     }
605     \@endparenv
606     \endgroup
607     \@endpetrue
608   }

```

(End of definition for `\FV@EndList`.)

`\FV@ListProcessLine` At last the tagging of the code lines. Here we have to tag also numbers and frame parts if they exist.

```

609   \def\FV@ListProcessLine#1{%
610     \hbox to \hsize{%
611       \kern\leftmargin
612       \hbox to \linewidth{%
613         \UseTaggingSocket{struct/begin}{tag=\UseStructureName{block/verbatim/codeline}}
614         \cs_if_exist:NT \FV@LeftListNumber
615         {
616           \UseTaggingSocket{struct-mc}{tag=\UseStructureName{block/verbatim/linenumber}}
617           {
618             \FV@LeftListNumber
619           }
620         }
621         \cs_if_exist:NT \FV@LeftListFrame
622         {
623           \UseTaggingSocket{mc}{artifact}{\FV@LeftListFrame}
624         }
625         \UseTaggingSocket{mc}{}{\FancyVerbFormatLine{#1}}%
626         \UseTaggingSocket{struct/end}\hss
627         \cs_if_exist:NT \FV@RightListFrame
628         {
629           \UseTaggingSocket{mc}{artifact}{\FV@RightListFrame}
630         }
631         \cs_if_exist:NT \FV@RightListNumber
632         {
633           \UseTaggingSocket{struct-mc}{tag=\UseStructureName{block/verbatim/linenumber}}
634           {
635             \FV@RightListNumber
636           }
637         }
638       }
639     \hss}}
640   }
641   \ExplSyntaxOff

```

(End of definition for `\FV@ListProcessLine`.)

3.5 The float package first aid

The float package has a number of incompatibilites:

- Restyling floats with `\restylefloat` breaks the tagging support.
- New float types created with `\newfloat` are not properly tagged (tagging issue 890)
- Non-floating floats using the H placement are not properly tagged.

The following firstaid addresses the last problem. A flaw is that if the float is inside a paragraph and followed by text there the text is tagged as a new semantic-para structure. TODO

```

642 \ExplSyntaxOn
643 \AddToHook{package/float/after}[firstaid]
644 {
645   \__tag_firstaid_float_patch_H:
646 }
647 \cs_new_protected:Npn \__tag_firstaid_float_patch_H:
648 {
649   \NewTaggingSocket{float/H/begin}{1}
650   \NewTaggingSocketPlug{float/H/begin}{default}
651   {
652     \mode_if_horizontal:T {\tag_mc_end_push: \tag_struct_end:}
653     \tag_struct_begin:n {tag=\UseStructureName{float/##1}}
654   }
655   \AssignTaggingSocketPlug{float/H/begin}{default}
656
657   \NewTaggingSocket{float/H/end}{0}
658   \NewTaggingSocketPlug{float/H/end}{default}
659   {
660     \tag_struct_end:
661     \mode_if_horizontal:T
662     {
663       \tag_struct_begin:n {tag=\UseStructureName{para/textblock}}
664       \tag_mc_begin_pop:n {}
665     }
666   }
667   \AssignTaggingSocketPlug{float/H/end}{default}

```

and here the redefinitions of the commands of float

```

668 \def\@float@HH##1[H]{\UseTaggingSocket{float/H/begin}{##1}%
669   \expandafter\let\csname end##1\endcsname\float@endH
670   \let\@currbox\float@box
671   \def\@capttype{##1}\setbox\@floatcapt=\vbox{}%
672   \expandafter\ifx\csname fst##1\endcsname\relax
673     \@flstylefalse\else\@flstyletrue\fi
674   \setbox\@currbox\color\@vbox\normalcolor
675   \vbox\bgroup \hsize\columnwidth \@parboxrestore
676   \@floatboxreset \setnobreak
677   \ignorespaces}
678 \renewcommand\float@endH{\@endfloatbox\UseTaggingSocket{float/H/end}
679   \vskip\intextsep
680   \if@flstyle\setbox\@currbox\float@makebox\columnwidth\fi
681   \box\@currbox\vskip\intextsep\relax}
682 }
683 \ExplSyntaxOff

```

3.6 The apacite firstaid

The apacite package redefines \@lbibitem and removes the arguments. This means that the patching with a generic hook with arguments fails. So we readd the argument.

```

684 \AddToHook{package/apacite/after}

```

```
685   {\if@APAC@natbib@apa\def\@lbibitem[#1]#2{\NAT@swattrue\orig@nat@\lbibitem[#1]{#2}}\fi}  
686 \end{package}
```