

The latex-lab-firstaid package

Temporary patches to external packages needed for the tagging project

L^AT_EX Project*

v0.86g 2026-09-22

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   \ltxlabfirstaiddatetime\space v\ltxlabfirstaidversion\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 $\text{\LaTeX}$ `\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 listings

The `listings` package needs various adaptations to enable tagging support. They are in an external file.

```

20 \AddToHook{package/listings/after}[latex-lab-firstaid]{\RequirePackage{latex-lab-
listings}}

```

2.3 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.

```

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

24 </package>
25 <*amscls>
26 \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.

```

27 \def\@author{\authors}

```

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

```

28 \renewenvironment{abstract}{%
29 \ifx\maketitle\relax
30 \ClassWarning{\@classname}{Abstract~ should~ precede~
31 \protect\maketitle\space in~ AMS~ document~ classes;~ reported}%
32 \fi
33 \global\setbox\abstractbox=\vtop \bgroup
34 \normalfont\Small
35 \list{}{\labelwidth\z@

```

```

36         \leftmargin3pc \rightmargin\leftmargin
37         \listparindent\normalparindent \itemindent\z@
38         \parsep\z@ \@plus\p@
39         \let\fullwidthdisplay\relax
40     }%
41     \item[\hskip\labelsep\scshape\abstractname.]%
42 }{%
43     \endlist
44     \par          % <--- added
45     \egroup
46     \ifx\@setabstract\relax \@setabstracta \fi
47 }

```

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.

```

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

```

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

```

51 \cs_set_eq:NN \newtheoremstyle \newtheoremstyle@saved
52 \tl_set:Nn \newtheoremstyle@vspace@default {
53   .5\baselineskip \@plus .2\baselineskip \@minus .2\baselineskip
54 }

```

2.3.1 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).

```

55 \DeclareDocumentCommand \part {s ={\shorttitle}o m}
56 { \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.

```

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

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

```

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

```

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

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

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

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

75 \DeclareDocumentCommand \subparagraph {s ={\shorttitle}o m}
76 { \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.

```

77 <amsbook>

78 \DeclareInstance{heading}{part}{display}
79 {
80   , name                = part
81   , level               = -1
82   , placement           = page
83   , placement-start-code = \cleardoublepage \thispagestyle{empty}
84                           \null \vfil \markboth{}{}
85   , placement-end-code  = \vfil \vfil \newpage \thispagestyle{empty}
86   , mark-cmd            = \partmark {#1}
87   , heading-decls       = \centering\fontsize{\@xvipt}{22}\bfseries
88   , headformat-instance = part
89   , prefix              = \partname
90   , number-format       = #1
91   , title-decls         = \fontsize{\@xxpt}{25}\bfseries
92 }

```

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*`.

```

93 \DeclareInstance{heading}{chapter}{display}
94 {
95   , name                = chapter
96   , level               = 0
97   , mark-cmd            = \chaptermark {#1}
98   , para-indent         = true
99   , placement           = top
100   , after-penalty-vspace = 32pt
101   , after-vspace        = 22pt
102   , heading-decls       = \normalsize\mdseries\centering
103   , headformat-instance = chapter
104   , prefix              = \MakeUppercase{\chaptername}
105   , number-format       = #1

```

```

106     , title-decls          = \fontsize{\@xivpt}{18} \bfseries
107     , contents-extra      = \addtocontents{lof}{\protect\addvspace{10\p@}}
108                           \addtocontents{lot}{\protect\addvspace{10\p@}}
109   }

110 \DeclareInstanceCopy{heading}{chapter*}{chapter}
111 \EditInstance{heading}{chapter*}{ after-penalty-vspace = 62pt }

112 \DeclareInstance{headformat}{chapter}{display}
113   {
114     , heading-indent = 0pt
115     , separator-b    = \par \nobreak \vspace{18pt}
116   }

117 \DeclareInstance{heading}{section}{display}
118   {
119     , name           = section
120     , level          = 1
121     , mark-cmd       = \sectionmark{#1}
122     , para-indent    = true
123     , before-vspace  = 0.7\linespacing plus \linespacing
124     , after-vspace   = 0.5\linespacing
125     , heading-decls  = \normalfont \bfseries \centering
126     , headformat-instance = section
127     , number-format  = #1.
128     , number-decls   = \upshape
129   }
130 \DeclareInstance{headformat}{section}{hang}
131   {
132     separator-b = \hspace{0.5em}
133   }

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

134 \DeclareInstanceCopy{heading}{specialsection}{section}
135 \EditInstance{heading}{specialsection}
136   {
137     , before-vspace = 2\linespacing plus \linespacing
138     , heading-decls = \large \scshape \centering
139   }

140 \DeclareInstance{heading}{subsection}{runin}
141   {
142     , name           = subsection
143     , level          = 2
144     , mark-cmd       = \subsectionmark{#1}
145     , before-vspace  = 0.5\linespacing plus 0.7\linespacing
146     , after-hspace   = \fontdimen2\font
147     , heading-decls  = \normalfont \bfseries
148     , headformat-instance = subsection
149     , number-format  = #1.
150     , number-decls   = \upshape
151     , title-format   = #1 \AddPunct{.}
152   }
153 \DeclareInstance{headformat}{subsection}{runin}
154   {

```

```

155     , separator-b      = \hspace{0.5em}
156     , heading-indent = \normalparindent
157   }

158 \DeclareInstance{heading}{subsubsection}{runin}
159 {
160   , name                = subsubsection
161   , level               = 3
162   , mark-cmd            = \subsubsectionmark{#1}
163   , before-vspace       = 0pt
164   , after-hspace        = \fontdimen2\font
165   , heading-decls       = \normalfont \itshape
166   , headformat-instance = subsubsection
167   , number-format       = #1.
168   , number-decls        = \upshape
169   , title-format        = #1 \AddPunct{.}
170 }

171 \DeclareInstance{headformat}{subsubsection}{runin}
172 {
173   , separator-b      = \hspace{0.5em}
174   , heading-indent = \normalparindent
175 }

176 \DeclareInstance{heading}{paragraph}{runin}
177 {
178   , name                = paragraph
179   , level               = 4
180   , mark-cmd            = \paragraphmark{#1}
181   , before-vspace       = 0pt
182   , after-hspace        = \fontdimen2 \font
183   , heading-decls       = \normalfont
184   , headformat-instance = paragraph
185   , number-format       = #1.
186   , number-decls        = \upshape
187   , title-format        = #1 \AddPunct{.}
188 }

189 \DeclareInstance{headformat}{paragraph}{runin}
190 {
191   , separator-b      = \hspace{0.5em}
192   , heading-indent = \normalparindent
193 }

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

```

```

207 \DeclareInstance{headformat}{subparagraph}{runin}
208 {
209   , separator-b = \hspace{0.5em}
210 }

```

The running header setup from amsbook doesn't correctly function and needs adjustments. For now we provide some simple redefinitions which do not necessarily produce exactly the same results.

```

211 \def\partmark#1{%
212   \markboth{}{}
213 }
214 \def\chaptermark#1{%
215   \markboth {\MakeUppercase{%
216     \ifnum \c@secnumdepth >\m@ne
217       \thechapter. \ %
218     \fi
219     #1}}{}
220 }
221 \def\sectionmark#1{%
222   \markright {\MakeUppercase{%
223     \ifnum \c@secnumdepth >\z@
224       \thesection. \ %
225     \fi
226     #1}}
227 }
228 \</amsbook>
229 \<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.

```

230 \DeclareInstance{heading}{part}{display}
231 {
232   , name                = part
233   , level                = -1                % -1 or 0 ?
234   , mark-cmd            = \partmark{#1}
235   , para-indent         = true
236   , before-vspace       = \linespacing plus \linespacing
237   , after-vspace        = 0.5 \linespacing
238   , heading-decls       = \normalfont \bfseries \raggedright
239   , headformat-instance = part
240   , prefix              = \partname
241   , number-format       = \theheading .
242   , number-decls        = \mdseries \upshape % proc: has just \upshape
243 }
244 \DeclareInstance{headformat}{part}{hang}
245 {
246   , separator-b = \hspace{0.5em}
247 }
248 \DeclareInstance{heading}{section}{display}
249 {

```

```

250     , name                = section
251     , level               = 1
252     , mark-cmd            = \sectionmark{#1}
253     , para-indent         = true
254     , before-vspace       = 0.7\linespacing plus \linespacing
255     , after-vspace        = 0.5\linespacing
256     , heading-decls       = \normalfont \scshape \centering
257     , headformat-instance = section
258     , number-format       = \theheading .
259     , number-decls        = \mdseries \upshape
260   }
261 \DeclareInstance{headformat}{section}{hang}
262 {
263   , separator-b = \hspace{0.5em}
264 }

```

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

```

265 \DeclareInstanceCopy{heading}{specialsection}{section}
266 \EditInstance{heading}{specialsection}
267 {
268   , before-vspace = \linespacing plus \linespacing
269   , heading-decls = \normalfont \centering
270 }
271 \DeclareInstance{heading}{subsection}{runin}
272 {
273   , name                = subsection
274   , level               = 2
275   , mark-cmd            = \subsectionmark{#1}
276   , before-vspace       = 0.5\linespacing plus 0.7\linespacing
277   , after-hspace        = \fontdimen2\font
278   , heading-decls       = \normalfont \bfseries
279   , headformat-instance = subsection
280   , number-format       = \theheading .
281   , number-decls        = \mdseries \upshape
282   , title-format        = #1 \AddPunct{.}
283 }
284 \DeclareInstance{headformat}{subsection}{runin}
285 {
286   , separator-b = \hspace{0.5em}
287 }
288 \DeclareInstance{heading}{subsubsection}{runin}
289 {
290   , name                = subsubsection
291   , level               = 3
292   , mark-cmd            = \subsubsectionmark{#1}
293   , before-vspace       = 0.5\linespacing plus 0.7\linespacing
294   , after-hspace        = \fontdimen2\font
295   , heading-decls       = \normalfont \itshape
296   , headformat-instance = subsubsection
297   , number-format       = \theheading .
298   , number-decls        = \mdseries \upshape
299   , title-format        = #1 \AddPunct{.}

```

```

300 }
301 \DeclareInstance{headformat}{subsubsection}{runin}
302 {
303   , separator-b = \hspace{0.5em}
304 }

305 \DeclareInstance{heading}{paragraph}{runin}
306 {
307   , name           = paragraph
308   , level          = 4
309   , mark-cmd       = \paragraphmark{#1}
310   , before-vspace  = 0pt
311   , after-hspace   = \fontdimen2\font
312   , heading-decls  = \normalfont
313   , headformat-instance = paragraph
314   , number-format  = \theheading .
315   , number-decls   = \mdseries \upshape
316   , title-format   = #1 \AddPunct{.}
317 }
318 \DeclareInstance{headformat}{paragraph}{runin}
319 {
320   , separator-b = \hspace{0.5em}
321 }

322 \DeclareInstance{heading}{subparagraph}{runin}
323 {
324   , name           = subparagraph
325   , level          = 5
326   , mark-cmd       = \subparagraphmark{#1}
327   , before-vspace  = 0pt
328   , after-hspace   = \fontdimen2\font
329   , heading-decls  = \normalfont
330   , headformat-instance = subparagraph
331   , number-format  = \theheading .
332   , number-decls   = \mdseries \upshape
333   , title-format   = #1 \AddPunct{.}
334 }
335 \DeclareInstance{headformat}{subparagraph}{runin}
336 {
337   , separator-b = \hspace{0.5em}
338 }

339 \</amsart>

340 \<amsproc>

341 \DeclareInstance{heading}{part}{display}
342 {
343   , name           = part
344   , level          = -1 % -1 or 0 ?
345   , mark-cmd       = \partmark{#1}
346   , para-indent    = true
347   , before-vspace  = \linespacing plus \linespacing
348   , after-vspace   = 0.5 \linespacing
349   , heading-decls  = \normalfont \bfseries \raggedright

```

```

350     , headformat-instance = part
351     , prefix              = \partname
352     , number-format       = \theheading .
353     , number-decls        = \upshape
354   }
355 \DeclareInstance{headformat}{part}{hang}
356 {
357   , separator-b = \hspace{0.5em}
358 }

359 \DeclareInstance{heading}{section}{display}
360 {
361   , name              = section
362   , level             = 1
363   , mark-cmd          = \sectionmark{#1}
364   , para-indent       = true
365   , before-vspace     = 0.7\linespacing plus \linespacing
366   , after-vspace      = 0.5\linespacing
367   , heading-decls     = \normalfont \bfseries \centering
368   , headformat-instance = section
369   , number-format     = \theheading .
370   , number-decls      = \upshape
371 }
372 \DeclareInstance{headformat}{section}{hang}
373 {
374   , separator-b = \hspace{0.5em}
375 }

```

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

```

376 \DeclareInstanceCopy{heading}{specialsection}{section}
377 \EditInstance{heading}{specialsection}
378 {
379   , before-vspace = \linespacing plus \linespacing
380   , heading-decls = \large \scshape \centering
381 }

382 \DeclareInstance{heading}{subsection}{runin}
383 {
384   , name              = subsection
385   , level             = 2
386   , mark-cmd          = \subsectionmark{#1}
387   , before-vspace     = 0.5\linespacing plus 0.7\linespacing
388   , after-hspace      = \fontdimen2\font
389   , heading-decls     = \normalfont \bfseries
390   , headformat-instance = subsection
391   , number-format     = \theheading .
392   , number-decls      = \upshape
393   , title-format      = #1 \AddPunct{.}
394 }
395 \DeclareInstance{headformat}{subsection}{runin}
396 {
397   , separator-b      = \hspace{0.5em}
398   , heading-indent   = \normalparindent
399 }

```

```

400 \DeclareInstance{heading}{subsubsection}{runin}
401 {
402   , name                = subsubsection
403   , level                = 3
404   , mark-cmd             = \subsubsectionmark{#1}
405   , before-vspace        = 0.5\linespacing plus 0.7\linespacing
406   , after-hspace         = \fontdimen2\font
407   , heading-decls        = \normalfont \itshape
408   , headformat-instance = subsubsection
409   , number-format        = \theheading .
410   , number-decls         = \upshape
411   , title-format         = #1 \AddPunct{.}
412 }
413 \DeclareInstance{headformat}{subsubsection}{runin}
414 {
415   , separator-b          = \hspace{0.5em}
416   , heading-indent       = \normalparindent
417 }

418 \DeclareInstance{heading}{paragraph}{runin}
419 {
420   , name                = paragraph
421   , level                = 4
422   , mark-cmd             = \paragraphmark{#1}
423   , before-vspace        = 0pt
424   , after-hspace         = \fontdimen2\font
425   , heading-decls        = \normalfont
426   , headformat-instance = paragraph
427   , number-format        = \theheading .
428   , number-decls         = \upshape
429   , title-format         = #1 \AddPunct{.}
430 }
431 \DeclareInstance{headformat}{paragraph}{runin}
432 {
433   , separator-b          = \hspace{0.5em}
434   , heading-indent       = \normalparindent
435 }

436 \DeclareInstance{heading}{subparagraph}{runin}
437 {
438   , name                = subparagraph
439   , level                = 5
440   , mark-cmd             = \subparagraphmark{#1}
441   , before-vspace        = 0pt
442   , after-hspace         = \fontdimen2\font
443   , heading-decls        = \normalfont
444   , headformat-instance = subparagraph
445   , number-format        = \theheading .
446   , number-decls         = \upshape
447   , title-format         = #1 \AddPunct{.}
448 }
449 \DeclareInstance{headformat}{subparagraph}{runin}
450 {
451   , separator-b          = \hspace{0.5em}
452 }

```

453 `\amsproc`

454 `*amsbook`

The AMS classes do a lot of low-level overwrites to macros building out the table of contents structures. `\starttoc` being one example which is partially fixed here. This isn't the full fix: various `\l@...` macros also need changes, but that has to wait for another day ... for now we only make sure that the code doesn't break badly, but they layout is not matching what the class normally produces!

```

455 \def\starttoc#1#2{%
456   \begingroup
457   \setTrue{#1}%
458   \ifx\contentsname#2%
459     \chapter[unnumbered,toc=]{\contentsname}%
460   \else
461     \chapter[=,unnumbered]{#2}%
462   \fi
463   \typeout{#2}%
464   \makeatletter
465   \@input{\jobname.#1}%
466   \if@filesw
467     \exp\newwrite\csname tf@#1\endcsname
468     \immediate\exp\openout\csname tf@#1\endcsname \jobname.#1\relax
469   \fi
470   \global\@nobreakfalse
471   \endgroup
472   \newpage
473 }

474 \AddToHook{begindocument}{
475   \namedef{r@tocindent-1}{1em}
476   \namedef{r@tocindent0}{1em}
477   \namedef{r@tocindent1}{1em}
478 }

479 \amsbook

480 \ExplSyntaxOff
481 \amscls
482 *package

483 \ExplSyntaxOff

```

2.4 fncychap

First aid for fncychap is kept in a separate file.

```

484 \AddToHook{package/fncychap/after}[latex-lab-firstaid]{\input{latex-lab-firstaid-
fncychap.ltx}}

485 \package
486 *fncychap

487 \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.

```
488 \EditInstance{heading}{chapter}{
489     after-vspace      = 0pt
490     , headformat-instance = fncychap
491 }
```

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

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

```

The code for this template is fairly simple.

```
493 \DeclareTemplateCode{headformat}{fncychap}{6}{}
494 {

```

Output some debugging info at the start.

```
495 \_head_show_arguments:nnnnnn {#1}{#2}{#3}{#4}{#5}{#6}

```

Everything else happens in a group to keep changes local.

```
496 \group_begin:

```

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

```
497 \bool_if:NTF \l__head_unnumbered_bool
498 { \MakeLinkTarget [ \l__head_name_t1 ] {} }
499 { \MakeLinkTarget { \l__head_name_t1 } }
```

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

```
500 \parindent 0pt
501 \raggedright
502 \interlinepenalty 10000
503 \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.

```
504 \AssignTaggingSocketPlug{parbox/before}{heading}
505 \AssignTaggingSocketPlug{minipage/before}{heading}
506 \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.

```
507 % \leavevmode % this can wait ...
508 \bool_if:NF \l__head_unnumbered_bool
509 { \DOCH }
```

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

```
510 \legacy_if:nTF { @mainmatter }
511 { \DOTI{#4} }
512 { \DOTIS{#4} }
```

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

```
513 \par
514 \group_end:
515 }
```

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:

```
516 \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.

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

518 \def\appendix{\par
519 \setcounter{chapter}{0}
520 \setcounter{section}{0}
521 \gdef\@chapapp{\appendixname}
522 \protected\gdef\thechapter{\thespecialchapter\Alph}
523 }
```

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`

```
524 \protected\def\thespecialchapter#1{
525 \leavevmode
526 \tag_if_in:nTF{Lb1}
527 { #1 {chapter} }
528 {
529 \UseTaggingSocket{inline/begin}{tag=heading-number}
530 #1 {chapter}
531 \UseTaggingSocket{inline/end}
532 }
533 }
```

`\mghrulefill`

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

```

534 \protected\def\mghrulefill#1{\leavevmode
535   \UseTaggingSocket{rule}{}{
536     \leaders\hrule\@height #1\hfill\kern\z@
537   }
538 }

```

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

```

539 \ExplSyntaxOff

540 \</fncychap>
541 \<*package>

```

2.5 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.

```

542 \AddToHook{package/verse/after}[latex-lab-firstaid]{%
543   \FirstAidNeededT{verse}{sty}{2014/05/10 v2.4b verse typesetting}%
544   {%
545     \AtBeginDocument{%
546       \renewenvironment{verse}[1][\linewidth]{%
547         \stepcounter{verse@envctr}%
548         \setcounter{poemline}{0}\refstepcounter{poemline}%
549         \setcounter{vslineno}{1}%
550         \let\@vscentercr
551         \list{}\@itemsep \z@
552           \itemindent -\vindent
553           \listparindent\itemindent
554           \parsep \stanzaskip
555           \ifdim #1 < \linewidth
556             \rightmargin \z@
557             \setlength{\leftmargin}{\linewidth}%
558             \addtolength{\leftmargin}{-#1}%
559             \addtolength{\leftmargin}{-0.5\leftmargin}%
560           \else
561             \rightmargin \leftmargin
562           \fi
563           \addtolength{\leftmargin}{\vindent}}%
564       \item[]%
565     }%
566     {\endlist}%
567   }%
568 }%
569 }

```

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

```

2.6 cleveref

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

```

570 \AddToHook{package/cleveref/after}
571 {
572   \let\@makefnstext\cref@old@makefnstext
573   \AddToHook{cmd/\@makefnstext/before}{%
574     \cref@constructprefix{footnote}{\cref@result}%
575     \protected@edef\cref@currentlabel{%
576       [footnote][\arabic{footnote}][\cref@result]%
577       \p@footnote\@thefnmark}}
578 }

```

2.7 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>

```

579 \ExplSyntaxOn
580 \AddToHook{package/booktabs/after}
581 {
582   \def\@cmidrulea{
583     \multispan\@cmidla
584     &\multispan\@cmidlb
585     \unskip\hskip\cmrkern@l
586     {
587       \UseTaggingSocket{tbl/leaders/begin}
588       \CT@arc@\leaders\hrule \@height\@thisrulewidth\hfill\kern\z@}
589     \hskip\cmrkern@r
590     \UseTaggingSocket{tbl/leaders/end}
591     \tbl_gdecr_row_count:
592     \cr}
593
594   \def\@cmidruleb{%
595     \multispan\@cmidlb
596     \unskip\hskip \cmrkern@l%
597     {
598       \UseTaggingSocket{tbl/leaders/begin}
599       \CT@arc@\leaders\hrule \@height\@thisrulewidth\hfill\kern\z@}
600     \hskip\cmrkern@r
601     \UseTaggingSocket{tbl/leaders/end}
602     \tbl_gdecr_row_count:
603     \cr}
604 }

```

605 \ExplSyntaxOff

2.8 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.

```

606 \ExplSyntaxOn
607 \AddToHook{package/fancyvrb/after}
608 {
609   \def\FV@LeaveVMode{%
610     \if@noskipsec
611       \leavevmode
612     \else
613       \if\FV@ResetMargins\if@inlabel\leavevmode\fi\fi
614     \fi
615     \ifvmode
616       \@nolisttrue
617       \UseTaggingSocket{para/semantic/begin}{}
618     \else
619       \tagpdfsetup{para/flattened=true}
620       \@nolistfalse
621       \unskip\par
622     \fi
623   }

```

(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

```

624 \def\FV@List#1{%
625   \begingroup
626   \FV@UseKeyValues
627   \FV@LeaveVMode
628   \if@inlabel\else\setbox\@labels=\box\voidb@x\fi
629   \FV@ListNesting{#1}%
630   \FV@ListParameterHook
631   \FV@ListVSpace
632   \FV@SetLineWidth
633   \FV@InterLinePenalty
634   \let\FV@ProcessLine\FV@ListProcessLine@i
635   \FV@CatCodes
636   \FV@FormattingPrep
637   \FV@ObeyTabsInit
638   \cs_if_exist:NT \FV@BeginListFrame
639   {

```

```

640         \UseTaggingSocket{mc}{artifact}{\FV@BeginListFrame}
641     }
642     \UseTaggingSocket{struct/begin}{tag=\UseStructureName{block/verbatim}}
643 }

```

(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.

```

644     \def\FV@EndList{%
645         \FV@ListProcessLastLine
646         \UseTaggingSocket{struct/end}
647         \cs_if_exist:NT \FV@EndListFrame
648         {
649             \UseTaggingSocket{mc}{artifact}{\FV@EndListFrame}
650         }
651     \@endparenv
652     \endgroup
653     \@endpetrue
654 }

```

(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.

```

655     \def\FV@ListProcessLine#1{%
656         \hbox to \hsize{%
657             \kern\leftmargin
658             \hbox to \linewidth{%
659                 \UseTaggingSocket{struct/begin}{tag=\UseStructureName{block/verbatim/codeline}}
660                 \cs_if_exist:NT \FV@LeftListNumber
661                 {
662                     \UseTaggingSocket{struct-mc}{tag=\UseStructureName{block/verbatim/linenumber}}
663                     {
664                         \FV@LeftListNumber
665                     }
666                 }
667                 \cs_if_exist:NT \FV@LeftListFrame
668                 {
669                     \UseTaggingSocket{mc}{artifact}{\FV@LeftListFrame}
670                 }
671                 \UseTaggingSocket{mc}{}{\FancyVerbFormatLine{#1}}%
672                 \UseTaggingSocket{struct/end}\hss
673                 \cs_if_exist:NT \FV@RightListFrame
674                 {
675                     \UseTaggingSocket{mc}{artifact}{\FV@RightListFrame}
676                 }
677                 \cs_if_exist:NT \FV@RightListNumber
678                 {
679                     \UseTaggingSocket{struct-mc}{tag=\UseStructureName{block/verbatim/linenumber}}
680                     {
681                         \FV@RightListNumber
682                     }
683                 }
684             }
685             \hss}}

```

```

686   }
687   \ExplSyntaxOff

```

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

2.9 The float package first aid

The float package has a number of incompatibilites:

- Restyling floats with `\restylefloat` breaks the tagging 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

```

688 \ExplSyntaxOn
689 \AddToHook{package/float/after}[firstaid]
690 {
691   \__tag_firstaid_float_patch_H:
692 }
693 \cs_new_protected:Npn \__tag_firstaid_float_patch_H:
694 {
695   \NewTaggingSocket{float/H/begin}{1}
696   \NewTaggingSocketPlug{float/H/begin}{default}
697   {
698     \mode_if_horizontal:T {\tag_mc_end_push: \tag_struct_end:}
699     \tag_struct_begin:n {tag=\UseStructureName{float/##1}}
700   }
701   \AssignTaggingSocketPlug{float/H/begin}{default}
702
703   \NewTaggingSocket{float/H/end}{0}
704   \NewTaggingSocketPlug{float/H/end}{default}
705   {
706     \tag_struct_end:
707     \mode_if_horizontal:T
708     {
709       \tag_struct_begin:n {tag=\UseStructureName{para/textblock}}
710       \tag_mc_begin_pop:n {}
711     }
712   }
713   \AssignTaggingSocketPlug{float/H/end}{default}

```

and here the redefinitions of the commands of float

```

714 \def\@float@HH##1[H]{\UseTaggingSocket{float/H/begin}{##1}%
715   \expandafter\let\csname end##1\endcsname\float@endH
716   \let\@currbox\float@box
717   \def\@capttype{##1}\setbox\@floatcapt=\vbox{}%
718   \expandafter\ifx\csname fst@##1\endcsname\relax
719     \@flstylefalse\else\@flstyletrue\fi
720   \setbox\@currbox\color@vbox\normalcolor
721     \vbox\bgroup \hsize\columnwidth \@parboxrestore
722     \@floatboxreset \setnobreak

```

```

723     \ignorespaces}
724 \renewcommand\float@endH{\@endfloatbox\UseTaggingSocket{float/H/end}
725     \vskip\intextsep
726     \if@fstyle\setbox\@currbox\float@makebox\columnwidth\fi
727     \box\@currbox\vskip\intextsep\relax}
728 }
729 \ExplSyntaxOff

```

2.10 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.

```

730 \AddToHook{package/apacite/after}
731   {\if@APAC@natbib@apa\def\@lbibitem[#1]#2{\NAT@swattrue\orig@nat@lbibitem[#1]{#2}}\fi}
732 \</package>

```