

The latex-lab-firstaid package

Temporary patches to external packages needed for the tagging project

L^AT_EX Project*

v0.85y 2026-07-29

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 `cname` `\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          %% <-- needed?
118   , number-title-sep = 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   , number-title-sep = 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   , after-vspace      = 0.5em
150   , heading-decls     = \normalfont \bfseries
151   , headformat-instance = subsection
152   , number-format     = \theheading .
153   , number-decls      = \upshape
154 }
155 \DeclareInstance{headformat}{subsection}{runin}
156 {
157   , number-title-sep = 0.5em
158   , title-format     = #1 \AddPunctuation{.}
159   , heading-indent   = \normalparindent
160 }

161 \DeclareInstance{heading}{subsubsection}{runin}
162 {
163   , name              = subsubsection
164   , level             = 3

```

```

165     , mark-cmd           = \subsubsectionmark{#1}
166     , before-vspace      = 0pt
167     , after-vspace       = 0.5em
168     , heading-decls      = \normalfont \itshape
169     , headformat-instance = subsubsection
170     , number-format      = \theheading .
171     , number-decls       = \upshape
172   }
173 \DeclareInstance{headformat}{subsubsection}{runin}
174 {
175     , number-title-sep = 0.5em
176     , title-format     = #1 \AddPunctuation{.}
177     , heading-indent   = \normalparindent
178   }

179 \DeclareInstance{heading}{paragraph}{runin}
180 {
181     , name             = paragraph
182     , level            = 4
183     , mark-cmd         = \paragraphmark{#1}
184     , before-vspace    = 0pt
185     , after-vspace     = \fontdimen2 \font
186     , heading-decls    = \normalfont
187     , headformat-instance = paragraph
188     , number-format    = \theheading .
189     , number-decls     = \upshape
190   }
191 \DeclareInstance{headformat}{paragraph}{runin}
192 {
193     , number-title-sep = 0.5em
194     , title-format     = #1 \AddPunctuation{.}
195     , heading-indent   = \normalparindent
196   }

197 \DeclareInstance{heading}{subparagraph}{runin}
198 {
199     , name             = subparagraph
200     , level            = 5
201     , mark-cmd         = \subparagraphmark{#1}
202     , before-vspace    = 0pt
203     , after-vspace     = \fontdimen2\font
204     , heading-decls    = \normalfont
205     , headformat-instance = subparagraph
206     , number-format    = \theheading .
207     , number-decls     = \upshape
208   }
209 \DeclareInstance{headformat}{subparagraph}{runin}
210 {
211     , number-title-sep = 0.5em
212     , title-format     = #1.
213   }

214 \</amsbook>
215 \<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.

```

216 \DeclareInstance{heading}{part}{display}
217 {
218   , name                = part
219   , level                = -1                % -1 or 0 ?
220   , mark-cmd            = \partmark{#1}
221   , para-indent         = true
222   , before-vspace       = \linespacing plus \linespacing
223   , after-vspace        = 0.5 \linespacing
224   , heading-decls       = \normalfont \bfseries \raggedright
225   , headformat-instance = part
226   , prefix              = \partname
227   , number-format       = \theheading .
228   , number-decls        = \mdseries \upshape % proc: has just \upshape
229 }
230 \DeclareInstance{headformat}{part}{hang}
231 {
232   , number-title-sep = 0.5em
233 }
234 \DeclareInstance{heading}{section}{display}
235 {
236   , name                = section
237   , level                = 1
238   , mark-cmd            = \sectionmark{#1}
239   , para-indent         = true
240   , before-vspace       = 0.7\linespacing plus \linespacing
241   , after-vspace        = 0.5\linespacing
242   , heading-decls       = \normalfont \scshape \centering
243   , headformat-instance = section
244   , number-format       = \theheading .
245   , number-decls        = \mdseries \upshape
246 }
247 \DeclareInstance{headformat}{section}{hang}
248 {
249   , number-title-sep = 0.5em
250 }

```

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

```

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

```

```

263     , after-vspace      = 0.5em
264     , heading-decls    = \normalfont \bfseries
265     , headformat-instance = subsection
266     , number-format     = \theheading .
267     , number-decls      = \mdseries \upshape
268   }
269 \DeclareInstance{headformat}{subsection}{runin}
270 {
271     , number-title-sep = 0.5em
272     , title-format     = #1 \AddPunctuation{.}
273 }

274 \DeclareInstance{heading}{subsubsection}{runin}
275 {
276     , name              = subsubsection
277     , level              = 3
278     , mark-cmd           = \subsubsectionmark{#1}
279     , before-vspace     = 0.5\linespacing plus 0.7\linespacing
280     , after-vspace      = 0.5em
281     , heading-decls     = \normalfont \itshape
282     , headformat-instance = subsubsection
283     , number-format     = \theheading .
284     , number-decls      = \mdseries \upshape
285 }
286 \DeclareInstance{headformat}{subsubsection}{runin}
287 {
288     , number-title-sep = 0.5em
289     , title-format     = #1 \AddPunctuation{.}
290 }

291 \DeclareInstance{heading}{paragraph}{runin}
292 {
293     , name              = paragraph
294     , level              = 4
295     , mark-cmd           = \paragraphmark{#1}
296     , before-vspace     = 0pt
297     , after-vspace      = \fontdimen2\font
298     , heading-decls     = \normalfont
299     , headformat-instance = paragraph
300     , number-format     = \theheading .
301     , number-decls      = \mdseries \upshape
302 }
303 \DeclareInstance{headformat}{paragraph}{runin}
304 {
305     , number-title-sep = 0.5em
306     , title-format     = #1 \AddPunctuation{.}
307 }

308 \DeclareInstance{heading}{subparagraph}{runin}
309 {
310     , name              = subparagraph
311     , level              = 5
312     , mark-cmd           = \subparagraphmark{#1}
313     , before-vspace     = 0pt
314     , after-vspace      = \fontdimen2\font

```

```

315     , heading-decls      = \normalfont
316     , headformat-instance = subparagraph
317     , number-format      = \theheading .
318     , number-decls       = \mdseries \upshape
319   }
320 \DeclareInstance{headformat}{subparagraph}{runin}
321 {
322     , number-title-sep = 0.5em
323     , title-format     = #1 \AddPunctuation{.}
324 }

325 \</amsart>

326 \<*amsproc>

327 \DeclareInstance{heading}{part}{display}
328 {
329     , name                = part
330     , level                = -1                % -1 or 0 ?
331     , mark-cmd             = \partmark{#1}
332     , para-indent         = true
333     , before-vspace        = \linespacing plus \linespacing
334     , after-vspace         = 0.5 \linespacing
335     , heading-decls        = \normalfont \bfseries \raggedright
336     , headformat-instance = part
337     , prefix               = \partname
338     , number-format        = \theheading .
339     , number-decls         = \upshape
340 }
341 \DeclareInstance{headformat}{part}{hang}
342 {
343     , number-title-sep = 0.5em
344 }

345 \DeclareInstance{heading}{section}{display}
346 {
347     , name                = section
348     , level                = 1
349     , mark-cmd             = \sectionmark{#1}
350     , para-indent         = true
351     , before-vspace        = 0.7\linespacing plus \linespacing
352     , after-vspace         = 0.5\linespacing
353     , heading-decls        = \normalfont \bfseries \centering
354     , headformat-instance = section
355     , number-format        = \theheading .
356     , number-decls         = \upshape
357 }
358 \DeclareInstance{headformat}{section}{hang}
359 {
360     , number-title-sep = 0.5em
361 }

```

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

```

362 \DeclareInstanceCopy{heading}{specialsection}{section}
363 \EditInstance{heading}{specialsection}

```

```

364 {
365   , before-vspace = \linespacing plus \linespacing
366   , heading-decls = \large \scshape \centering
367 }

368 \DeclareInstance{heading}{subsection}{runin}
369 {
370   , name           = subsection
371   , level          = 2
372   , mark-cmd       = \subsectionmark{#1}
373   , before-vspace  = 0.5\linespacing plus 0.7\linespacing
374   , after-vspace   = 0.5em
375   , heading-decls  = \normalfont \bfseries
376   , headformat-instance = subsection
377   , number-format  = \theheading .
378   , number-decls   = \upshape
379 }
380 \DeclareInstance{headformat}{subsection}{runin}
381 {
382   , number-title-sep = 0.5em
383   , title-format     = #1 \AddPunctuation{.}
384   , heading-indent   = \normalparindent
385 }

386 \DeclareInstance{heading}{subsubsection}{runin}
387 {
388   , name           = subsubsection
389   , level          = 3
390   , mark-cmd       = \subsubsectionmark{#1}
391   , before-vspace  = 0.5\linespacing plus 0.7\linespacing
392   , after-vspace   = 0.5em
393   , heading-decls  = \normalfont \itshape
394   , headformat-instance = subsubsection
395   , number-format  = \theheading .
396   , number-decls   = \upshape
397 }
398 \DeclareInstance{headformat}{subsubsection}{runin}
399 {
400   , number-title-sep = 0.5em
401   , title-format     = #1 \AddPunctuation{.}
402   , heading-indent   = \normalparindent
403 }

404 \DeclareInstance{heading}{paragraph}{runin}
405 {
406   , name           = paragraph
407   , level          = 4
408   , mark-cmd       = \paragraphmark{#1}
409   , before-vspace  = 0pt
410   , after-vspace   = \fontdimen2\font
411   , heading-decls  = \normalfont
412   , headformat-instance = paragraph
413   , number-format  = \theheading .
414   , number-decls   = \upshape
415 }

```

```

416 \DeclareInstance{headformat}{paragraph}{runin}
417 {
418   , number-title-sep = 0.5em
419   , title-format      = #1 \AddPunctuation{.}
420   , heading-indent    = \normalparindent
421 }

422 \DeclareInstance{heading}{subparagraph}{runin}
423 {
424   , name              = subparagraph
425   , level              = 5
426   , mark-cmd          = \subparagraphmark{#1}
427   , before-vspace     = 0pt
428   , after-vspace      = \fontdimen2\font
429   , heading-decls     = \normalfont
430   , headformat-instance = subparagraph
431   , number-format     = \theheading .
432   , number-decls      = \upshape
433 }
434 \DeclareInstance{headformat}{subparagraph}{runin}
435 {
436   , number-title-sep = 0.5em
437   , title-format      = #1 \AddPunctuation{.}
438 }

439 \</amsproc>

440 \ExplSyntaxOff
441 \</amscls>
442 \<*package>

443 \ExplSyntaxOff

```

2.3 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.

```

444 \AddToHook{package/verse/after}[latex-lab-firstaid]{%
445   \FirstAidNeededT{verse}{sty}{2014/05/10 v2.4b verse typesetting}%
446   {%
447     \AtBeginDocument{%
448       \renewenvironment{verse}[1][\linewidth]{%
449         \stepcounter{verse@envctr}%
450         \setcounter{poemline}{0}\refstepcounter{poemline}%
451         \setcounter{vslineno}{1}%
452         \let\=\@vscentercr
453         \list{}{\itemsep \z@
454           \itemindent -\vindent
455           \listparindent\itemindent
456           \parsep      \stanzaskip
457           \ifdim #1 < \linewidth
458             \rightmargin \z@

```

```

459             \setlength{\leftmargin}{\linewidth}%
460             \addtolength{\leftmargin}{-#1}%
461             \addtolength{\leftmargin}{-0.5\leftmargin}%
462         \else
463             \rightmargin      \leftmargin
464         \fi
465         \addtolength{\leftmargin}{\vindent}}%
466     \item[]%
467 }%
468 {\endlist}%
469 }%
470 }%
471 }

```

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

```

2.4 cleveref

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

```

472 \AddToHook{package/cleveref/after}
473 {
474   \let\@makefnstext\cref@old@makefnstext
475   \AddToHook{cmd/@makefnstext/before}{%
476     \cref@constructprefix{footnote}{\cref@result}%
477     \protected@edef\cref@currentlabel{%
478       [footnote][\arabic{footnote}][\cref@result]%
479     \p@footnote\@thefnmark}}
480 }

```

2.5 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>

```

481 \ExplSyntaxOn
482 \AddToHook{package/booktabs/after}
483 {
484   \def\@cmidrulea{
485     \multispan\@cmidla
486     &\multispan\@cmidlb
487     \unskip\hskip\cmrkern@l
488   }

```

```

489 \UseTaggingSocket{tbl/leaders/begin}
490 \CT@arc@{leaders/hrule \@height\@thisrulewidth\hfill\kern\z@}
491 \hskip\cmrkern@r
492 \UseTaggingSocket{tbl/leaders/end}
493 \tbl_gdecr_row_count:
494 \cr}
495
496 \def\@cmidruleb{%
497   \multispan\@cmidlb
498   \unskip\hskip \cmrkern@l%
499   {
500     \UseTaggingSocket{tbl/leaders/begin}
501     \CT@arc@{leaders/hrule \@height\@thisrulewidth\hfill\kern\z@}
502     \hskip\cmrkern@r
503     \UseTaggingSocket{tbl/leaders/end}
504     \tbl_gdecr_row_count:
505     \cr}
506   }
507 \ExplSyntaxOff

```

2.6 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.

```

508 \ExplSyntaxOn
509 \AddToHook{package/fancyvrb/after}
510 {
511   \def\FV@LeaveVMode{%
512     \if@noskipsec
513       \leavevmode
514     \else
515       \if\FV@ResetMargins\if@inlabel\leavevmode\fi\fi
516     \fi
517     \ifvmode
518       \@nolisttrue
519       \UseTaggingSocket{para/semantic/begin}{}
520     \else
521       \tagpdfsetup{para/flattened=true}
522       \@nolistfalse
523       \unskip\par
524     \fi
525   }

```

(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

```

526 \def\FV@List#1{%
527 \begingroup
528 \FV@UseKeyValues
529 \FV@LeaveVMode
530 \if@inlabel\else\setbox\@labels=\box\voidb@x\fi
531 \FV@ListNesting{#1}%
532 \FV@ListParameterHook
533 \FV@ListVSpace
534 \FV@SetLineWidth
535 \FV@InterLinePenalty
536 \let\FV@ProcessLine\FV@ListProcessLine@i
537 \FV@CatCodes
538 \FV@FormattingPrep
539 \FV@ObeyTabsInit
540 \cs_if_exist:NT \FV@BeginListFrame
541 {
542 \UseTaggingSocket{mc}{artifact}{\FV@BeginListFrame}
543 }
544 \UseTaggingSocket{struct/begin}{tag=\UseStructureName{block/verbatim}}
545 }
```

(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.

```

546 \def\FV@endList{%
547 \FV@ListProcessLastLine
548 \UseTaggingSocket{struct/end}
549 \cs_if_exist:NT \FV@endListFrame
550 {
551 \UseTaggingSocket{mc}{artifact}{\FV@endListFrame}
552 }
553 \@endparenv
554 \endgroup
555 \@endpetrue
556 }
```

(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.

```

557 \def\FV@ListProcessLine#1{%
558 \hbox to \hsize{%
559 \kern\leftmargin
560 \hbox to \linewidth{%
561 \UseTaggingSocket{struct/begin}{tag=\UseStructureName{block/verbatim/codeline}}
562 \cs_if_exist:NT \FV@LeftListNumber
563 {
564 \UseTaggingSocket{struct-mc}{tag=\UseStructureName{block/verbatim/linenumber}}
565 {
566 \FV@LeftListNumber
567 }
568 }
```

```

569     \cs_if_exist:NT \FV@LeftListFrame
570     {
571         \UseTaggingSocket{mc}{artifact}{\FV@LeftListFrame}
572     }
573     \UseTaggingSocket{mc}{}{\FancyVerbFormatLine{#1}}%
574     \UseTaggingSocket{struct/end}\hss
575     \cs_if_exist:NT \FV@RightListFrame
576     {
577         \UseTaggingSocket{mc}{artifact}{\FV@RightListFrame}
578     }
579     \cs_if_exist:NT \FV@RightListNumber
580     {
581         \UseTaggingSocket{struct-mc}{tag=\UseStructureName{block/verbatim/linenumber}}
582         {
583             \FV@RightListNumber
584         }
585     }
586     }
587     \hss}}
588 }
589 \ExplSyntaxOff

```

(End of definition for \FV@ListProcessLine.)

2.7 The float package first aid

The float package has a number of incompatibilities:

- 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**

```

590 \ExplSyntaxOn
591 \AddToHook{package/float/after}[firstaid]
592 {
593     \__tag_firstaid_float_patch_H:
594 }
595 \cs_new_protected:Npn \__tag_firstaid_float_patch_H:
596 {
597     \NewTaggingSocket{float/H/begin}{1}
598     \NewTaggingSocketPlug{float/H/begin}{default}
599     {
600         \mode_if_horizontal:T {\tag_mc_end_push: \tag_struct_end:}
601         \tag_struct_begin:n {tag=\UseStructureName{float/##1}}
602     }
603     \AssignTaggingSocketPlug{float/H/begin}{default}
604
605     \NewTaggingSocket{float/H/end}{0}
606     \NewTaggingSocketPlug{float/H/end}{default}
607     {

```

```

608     \tag_struct_end:
609     \mode_if_horizontal:T
610     {
611         \tag_struct_begin:n {tag=\UseStructureName{para/textblock}}
612         \tag_mc_begin_pop:n { }
613     }
614 }
615 \AssignTaggingSocketPlug{float/H/end}{default}

```

and here the redefinitions of the commands of float

```

616 \def\@float@HH##1[H]{\UseTaggingSocket{float/H/begin}{##1}%
617   \expandafter\let\csname end##1\endcsname\float@endH
618   \let\@currbox\float@box
619   \def\@capttype{##1}\setbox\@floatcapt=\vbox{ }%
620   \expandafter\ifx\csname fst##1\endcsname\relax
621   \@flstylefalse\else\@flstyletrue\fi
622   \setbox\@currbox\color@vbox\normalcolor
623   \vbox\bgroup \hsize\columnwidth \@parboxrestore
624   \@floatboxreset \@setnobreak
625   \ignorespaces}
626 \renewcommand\float@endH{\@endfloatbox\UseTaggingSocket{float/H/end}
627   \vskip\intextsep
628   \if@flstyle\setbox\@currbox\float@makebox\columnwidth\fi
629   \box\@currbox\vskip\intextsep\relax}
630 }
631 \ExplSyntaxOff

```

2.8 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.

```

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

```