The latex-lab-firstaid package
Temporary patches to external packages needed for the tagging project

LaTeX Project*

v0.85s 2026-01-19

Abstract

1 Introduction

The followings 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

```
⟨*package⟩
⟨@@=tag⟩
```

```
\ProvidesPackage {latex-lab-testphase-firstaid} [%
   \ltlabfirstaiddate\space v\ltlabfirstaiddate\space
   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.

```
\ExplSyntaxOn
\providecommand\FirstAidNeededT[3]{
  \exp_args:Ncx\str_if_eq:onF{ver@#1.#2}{#3}
      { \typeout{==>~ First~ Aid~ for~ #1.#2~ no~ longer~ applied!^^J
          \@spaces Expected:^^J
          \@spaces\@spaces #3^^J
          \@spaces but~ found:^^J
          \@spaces\@spaces \use:c{ver@#1.#2}^^J
          \@spaces so~ I'm~ assuming~ it~ got~ fixed.
      } }
```

*Initial implementation done by Ulrike Fischer

```
  \exp_args:Ncx\str_if_eq:onT{ver@#1.#2}{#3}
}
```

(*End of definition for* `\FirstAidNeededT`.)

2.1 tikz/pgf

ti*kz* 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 LaTeX `\input` command. Check https://github.com/pgf-tikz/pgf/issues/1424 for changes.

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

2.2 ams classes

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.

```
\AddToHook{class/amsart/after}
 {\def\@author{\authors}}
\AddToHook{class/amsbook/after}
 {\def\@author{\authors}}
\AddToHook{class/amsproc/after}
 {\def\@author{\authors}}
```

The classes redefine `\@startsection` and define their own heading commands which are not tagging compatible. The template code will reinstate the new `\@startsection` and so handle the headings using it. But we also need to redefine `\chapter` and `\part`. The following does not try to create exactly the same design, only near enough. This will not properly handle `\specialsection`.

```
\cs_new_protected:Npn \__tag_firstaid_amsbook_heading:
 {
  \DeclareDocumentCommand \chapter {s ={shorttitle}o m}
      {\ParseLaTeXeHeading {chapter}{##1} {##2} {##3}}
  \DeclareInstance{heading}{chapter}{display}
    {
      , name          = chapter
      , level         = 0
      , placement         = top
      , after-penalty-sep = 32pt
      , after-sep     = 22pt
      , number-format = \MakeUppercase{\chaptername}\enspace \thechapter
      , decls         = \centering
      , number-decls  = \normalsize\mdseries
      , title-decls   = \fontsize{\@xivpt}{18}\bfseries
      , headformat-instance = chapter
      , mark-cmd = \chaptermark{##1}
      , para-indent = true
```

```
      , contents-extra=  \addtocontents{lof}{\protect\addvspace{10\p@}}%
                         \addtocontents{lot}{\protect\addvspace{10\p@}}%
    }
  \DeclareInstance{headformat}{chapter}{display}
    {
      , indent        = 0pt
      , before-code   =
      , after-code    =
      , number-title-sep = 18pt
    }
  \DeclareDocumentCommand \part {s ={shorttitle}o m}
      { \ParseLaTeXeHeading {part} {##1} {##2} {##3} }
  \DeclareInstance{heading}{part}{display}
   {
     , name          = part
     , level         = -1
     , placement         = page
     , start-code= {\cleardoublepage\thispagestyle{empty}%
                    \null\vfil\markboth{}{}}
     , final-code=  \vfil\vfil\newpage\newpage\thispagestyle{empty}
     , number-format = \partname\nobreakspace\thepart
     , decls         = \centering\bfseries
     , number-decls  = \huge
     , title-decls   = \Huge
     , headformat-instance = display
     , mark-cmd      = \partmark {##1}
   }
 }
\cs_new_protected:Npn\__tag_firstaid_amsart_heading:
 {
   \DeclareInstance{heading}{part-@startsection}{display}
     {
      name=part,
      level=-1,
      mark-cmd=\partmark {##1},
      para-indent  = true,
      before-sep  = 12.0pt plus 12.0pt,
      after-penalty-sep  =  0pt,
      after-sep  =  6.0pt,
      decls  = \normalfont \bfseries \raggedright,
      headformat-instance  =  part-@startsection,
      number-format  =  \partname\space\theheading.
     }
   \DeclareInstance{headformat}{part-@startsection}{hang}
    {indent  =  \z@,  number-title-sep  =  0.5em}
 }
```

The redefinitions for amsbook:

```
\AddToHook{class/amsbook/after}[latex-lab-testphase-firstaid/amschap]
  {\__tag_firstaid_amsbook_heading:}
```

amsart and amsproc can use the same definition:

```
\AddToHook{class/amsproc/after}[latex-lab-testphase-firstaid/amschap]
  {\__tag_firstaid_amsart_heading:}
\AddToHook{class/amsart/after}[latex-lab-testphase-firstaid/amschap]
```

```
  {\__tag_firstaid_amsart_heading:}
```

2.3 ams classes and amsthm

The amsart, amsbook and amsproc classes redefine the theorem code and this breaks the tagging added by the block code. The following reenables tagging. It does *not* give a completly identical output (similar to the new theorem code, see https://github.com/latex3/tagging-project/issues/715). The code also does not try to use sockets yet, as the theorem definitions in the block code don't do that yet either.

```
\AddToHook{class/amsart/after}[latex-lab-testphase-firstaid/abstract]
          {
            \__tag_firstaid_ams_abstract:
          }
\AddToHook{class/amsbook/after}[latex-lab-testphase-firstaid/abstract]
          {
            \__tag_firstaid_ams_abstract:
          }

\cs_new_protected:Npn \__tag_firstaid_ams_abstract:
  {
\renewenvironment{abstract}{%
  \ifx\maketitle\relax
    \ClassWarning{\@classname}{Abstract~ should~ precede~
      \protect\maketitle\space in~ AMS~ document~ classes;~ reported}%
  \fi
  \global\setbox\abstractbox=\vtop \bgroup
    \normalfont\Small
    \list{}{\labelwidth\z@
      \leftmargin3pc \rightmargin\leftmargin
      \listparindent\normalparindent \itemindent\z@
      \parsep\z@ \@plus\p@
      \let\fullwidthdisplay\relax
    }%
    \item[\hskip\labelsep\scshape\abstractname.]%
}{%
  \endlist
  \par              % <--- added
  \egroup
  \ifx\@setabstract\relax \@setabstracta \fi
}
  }
\ExplSyntaxOff
```

2.4 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.

```
\AddToHook{package/verse/after}[latex-lab-firstaid]{%
  \FirstAidNeededT{verse}{sty}{2014/05/10 v2.4b verse typesetting}%
   {%
     \AtBeginDocument{%
```

```
        \renewenvironment{verse}[1][\linewidth]{%
          \stepcounter{verse@envctr}%
          \setcounter{poemline}{0}\refstepcounter{poemline}%
          \setcounter{vslineno}{1}%
          \let\\=\@vscentercr
          \list{}{\itemsep \z@
                  \itemindent  -\vindent
                  \listparindent\itemindent
                  \parsep       \stanzaskip
                  \ifdim #1 < \linewidth
                    \rightmargin        \z@
                    \setlength{\leftmargin}{\linewidth}%
                    \addtolength{\leftmargin}{-#1}%
                    \addtolength{\leftmargin}{-0.5\leftmargin}%
                  \else
                    \rightmargin        \leftmargin
                  \fi
                  \addtolength{\leftmargin}{\vindent}}%
          \item[]%
        }%
        {\endlist}%
      }%
    }%
}
```

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 elabroate 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}!O{\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    = 0pt,%
            left-margin  = (\linewidth-\@vswidth)/2+\vindent,%
            right-margin = \ifdim\@vswidth<\linewidth 0pt
                           \else (\linewidth-\@vswidth)/2\fi,%
            ##1%
           }}%
      \ExpandArgs{o}\next\@vsremainingkvlist
      \item\relax
    }{\endblockenv}%
  }%
}
\makeatother
```

2.5 cleveref

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

```
\AddToHook{package/cleveref/after}
 {
   \let\@makefntext\cref@old@makefntext
   \AddToHook{cmd/@makefntext/before}{%
    \cref@constructprefix{footnote}{\cref@result}%
    \protected@edef\cref@currentlabel{%
     [footnote][\arabic{footnote}][\cref@result]%
    \p@footnote\@thefnmark}}
 }
```

2.6 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

```
\ExplSyntaxOn
\AddToHook{package/booktabs/after}
 {
  \def\@cmidrulea{
   \multispan\@cmidla
   &\multispan\@cmidlb
   \unskip\hskip\cmrkern@l
  {
   \tag_mc_begin:n{artifact}
   \CT@arc@\leaders\hrule \@height\@thisrulewidth\hfill\kern\z@}
   \hskip\cmrkern@r
   \tag_mc_end: \int_gdecr:N \g__tbl_row_int
   \cr}

  \def\@cmidruleb{%
    \multispan\@cmidlb
    \unskip\hskip \cmrkern@l%
   {
    \tag_mc_begin:n{artifact}
    \CT@arc@\leaders\hrule \@height\@thisrulewidth\hfill\kern\z@}
    \hskip\cmrkern@r
    \tag_mc_end: \int_gdecr:N \g__tbl_row_int
    \cr}
  }
\ExplSyntaxOff
```

2.7 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 (this requires the block code, but firstaid should be used only with phase-III anyway). 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 text-unit structure, if we are in hmode we have to set para mode to flattened before the fancyhdr code issues the`\par`. The closing of the text-unit structure is handled by the doendpe code in the block code.

```
\ExplSyntaxOn
\AddToHook{package/fancyvrb/after}
{
    \def\FV@LeaveVMode{%
      \if@noskipsec
        \leavevmode
      \else
        \if@FV@ResetMargins\if@inlabel\leavevmode\fi\fi
```

```
    \fi
    \ifvmode
      \@noparlisttrue
      \__tag_gincr_para_main_begin_int:
      \tag_struct_begin:n{tag=\l__tag_para_main_tag_tl}
    \else
      \bool_set_true:N\l__tag_para_flattened_bool
      \@noparlistfalse
      \unskip\par
    \fi
    }
```

(*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

```
  \def\FV@List#1{%
    \begingroup
    \FV@UseKeyValues
    \FV@LeaveVMode
    \if@inlabel\else\setbox\@labels=\box\voidb@x\fi
    \FV@ListNesting{#1}%
    \FV@ListParameterHook
    \FV@ListVSpace
    \FV@SetLineWidth
    \FV@InterLinePenalty
    \let\FV@ProcessLine\FV@ListProcessLine@i
    \FV@CatCodes
    \FV@FormattingPrep
    \FV@ObeyTabsInit
    \cs_if_exist:NT \FV@BeginListFrame
      {
       \tag_mc_begin:n{artifact}
       \FV@BeginListFrame
       \tag_mc_end:
      }
    \tag_struct_begin:n{tag=verbatim}
    }
```

(*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.

```
  \def\FV@EndList{%
    \FV@ListProcessLastLine
    \tag_struct_end:
    \cs_if_exist:NT \FV@EndListFrame
      {
       \tag_mc_begin:n{artifact}
       \FV@EndListFrame
       \tag_mc_end:
      }
    \@endparenv
    \endgroup
    \@endpetrue
    }
```

(*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.

```
\def\FV@ListProcessLine#1{%
  \hbox to \hsize{%
    \kern\leftmargin
    \hbox to \linewidth{%
     \tag_struct_begin:n{tag=codeline}
      \cs_if_exist:NT \FV@LeftListNumber
       {
         \tag_struct_begin:n{tag=Lbl}
         \tag_mc_begin:n{}
         \FV@LeftListNumber
         \tag_mc_end:
         \tag_struct_end:
       }
      \cs_if_exist:NT \FV@LeftListFrame
       {
         \tag_mc_begin:n{artifact}
         \FV@LeftListFrame
         \tag_mc_end:
       }
      \tag_mc_begin:n{}%
      \FancyVerbFormatLine{#1}%
      \tag_mc_end:
      \tag_struct_end:\hss
      \cs_if_exist:NT \FV@RightListFrame
       {
         \tag_mc_begin:n{artifact}
         \FV@RightListFrame
         \tag_mc_end:
       }
      \cs_if_exist:NT \FV@RightListNumber
       {
         \tag_struct_begin:n{tag=Lbl}
         \tag_mc_begin:n{}
         \FV@RightListNumber
         \tag_mc_begin:n{}
         \tag_struct_end:
       }
      }
      \hss}}
 }
\ExplSyntaxOff
```

2.8 The **microtype** package first aid

The microtype package fails in kernel 2025/11 if tagging is active as due to the change to `\partokencontext` para tagging is triggered and then is unbalanced in an unwanted place. We disable the tagging in this command.

```
\AddToHook{package/microtype/after}[firstaid]{%
  \FirstAidNeededT{microtype}{sty}{2025/07/09 v3.2b
```

```
  Micro-typographical refinements
  (RS)}
 {%
   \def\MT@get@prot#1#2{%
     \begingroup
       \SuspendTagging{\MT@get@prot}%%added by firstaid
       \setbox\MT@tempbox\vbox{%
         \everypar{}%
         \parfillskip=\z@skip
         \hbadness\@M
         \clubpenalty\z@
         \widowpenalty\z@
         \interlinepenalty\z@
         \@newlistfalse
         \MT@prot@hook
         \begingroup
           \MT@noindent #1\relax\MT@csq@eqgroup
         \endgroup}%
       \vbadness=\@M
       \splittopskip=\z@
       \vfuzz=\maxdimen
       \setbox\MT@tempbox\vbox{%
         \ifvbox\MT@tempbox
           \global\setbox\MT@tempbox=\vsplit\MT@tempbox to \normalbaselineskip
           \unvbox\MT@tempbox
           \global\setbox\MT@tempbox=\lastbox
         \fi
       }%
     \endgroup
     \ifhbox\MT@tempbox
       \@tempdima=\@nameuse{#2marginkern}\MT@tempbox\relax
       \expandafter\ifdim\@tempdima=\z@ \else
         \leavevmode
         \MT@vinfo{|<< adding #2 margin kern for `#1':\MessageBreak
           \the\@tempdima \on@line}%
         \kern\@tempdima
       \fi
     \fi
   }}
 }
```

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

```
⟨/package⟩
⟨*latex-lab⟩
\ProvidesFile{firstaid-latex-lab-testphase.ltx}
        [\ltlabfirstaiddate\space v\ltlabfirstaidversion\space
         latex-lab wrapper firstaid]

\RequirePackage{latex-lab-testphase-firstaid}

⟨/latex-lab⟩
```