First aid for external files and packages that need updating

Frank Mittelbach, LaTeX Project

June 30, 2021

Abstract

This file contains some first aid for packages or classes that require updates because of internal changes to LaTeX but that aren't yet reflected in the package/class code.

Contents

1 Introduction

Over the years package writers have hooked into various parts of internal LaTeX commands (largely because proper interfaces were missing in important places) and if we are now gradually adding such interfaces these internal commands do change and as a result patching into them stops working.

As part of making such internal changes the LaTeX Project team attempts to check for such usage in packages, alert the package maintainers and ensures that the packages get updated alongside the core LaTeX system. However it is not always possible to get packages that will fail with a new kernel updated in time and if that is the case we try to provide a temporary fix in this file for them. Once the package gets updated the fix will then be removed again.

For that reason, it is put into a separate bundle so that we can update it easily without requiring the CTAN maintainers to install a new full LaTeX system just because we take out (or add) a fix for a package here.

In the best case scenario the file documented here should be empty. In practice it will probably always contain one or the other fix while we are waiting for the package to get updated.

> **Important notice:** The fixes provided here are not meant to be a permanent solution, but are only provided to support the transition period. They are (usually) neither complete nor necessarily the best solution! Furthermore, as they are done from the "outside", they usually add some burden and slow down LaTeX processing, even if the package/class is not used in the document.
>
> We will therefore remove such code as soon as possible again. In practice this means that if some package never gets updated/corrected, then it will eventually fail to work, because after one or at most two LaTeX releases we will take out the transition code to ensure that this "first aid patching" doesn't get out of bounds.

1.1 Minor kernel fixes

If we encounter issues with the kernel code that should get fixed before the next main release we normally generate a patch release for LaTeX. However, depending on the complexity of the fix we might first add the fix here and generate a full patch release only when a number of such issues have accumulated. This way we lessen the impact on CTAN maintainers because for each tach release we have to make and distribute also a matching development release.

2 The Implementation

This file is meant to be loaded during format generation which is why we give it the extension `.ltx`.

```
1 ⟨*kernel⟩
2 \def\LaTeXFirstAidDate{2021/06/28}
3 \def\LaTeXFirstAidVersion{v1.0o}
4 \ProvidesFile{latex2e-first-aid-for-external-files.ltx}
5               [\LaTeXFirstAidDate\space \LaTeXFirstAidVersion\space
6                 LaTeX kernel fixes to external files and packages]
```

`\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.

```
7 \ExplSyntaxOn
8 \cs_new:Npn\FirstAidNeededT#1#2#3{
9   \exp_args:Ncx\str_if_eq:onF{ver@#1.#2}{#3}
10      { \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.
      } }
  \exp_args:Ncx\str_if_eq:onT{ver@#1.#2}{#3}
}
\ExplSyntaxOff
```
⟨/kernel⟩

2.1 The `filehook` package first aid

The `filehook` package implements hooks into file loading commands. These days this is already provided by the kernel albeit not with the same user interface. Until that package gets updated (to use the kernel interfaces) we provide a substitution. This does not offer all hooks from `filehook` but all that have been used in packages available in TEX Live.

Note that this doesn't fix `currfile` because that package uses `filehook` but relies on the internals of the old implementation.

The package has now got an update so we aren't activating the first aid. However, at the moment it basically bypasses the new hook mechanism and puts the old hooks in thereby disabling, for example, the possibility to re-order code through rules.

We therefore keep `filehook-ltx.sty` around as a guideline for further updates.

Replacing `filehook` with a leaner version would then work like this:

```
⟨*kernel⟩
%\declare@file@substitution{filehook.sty}{filehook-ltx.sty}
⟨/kernel⟩
```

What follows is a simplified (partial) implementation of the `filehook` interfaces. Not implemented are:

```
\AtBeginOfFiles      \AtEndOfFiles
\AtBeginOfInputs     \AtEndOfInputs
\AtBeginOfInputFile  \AtEndOfInputFile
```

```
⟨*filehook-ltx⟩
\newcommand\AtBeginOfEveryFile [1]
  {\AddToHook{file/before}{#1}}
\newcommand\AtEndOfEveryFile [1]
  {\AddToHook{file/after}{#1}}
\newcommand\AtBeginOfIncludes [1]
  {\AddToHook{include/before}{#1}}
\newcommand\AtEndOfIncludes [1]
  {\AddToHook{include/end}{#1}}
\newcommand\AfterIncludes [1]
  {\AddToHook{include/after}{#1}}
\newcommand\AtBeginOfPackages [1]
  {\AddToHook{package/before}{#1}}
\newcommand\AtEndOfPackages [1]
  {\AddToHook{package/after}{#1}}
```

```
\newcommand\AtBeginOfClasses [1]
  {\AddToHook{class/before}{#1}}
\newcommand\AtEndOfClasses [1]
  {\AddToHook{class/after}{#1}}
\newcommand\AtBeginOfFile [2]
  {\AddToHook{file/before/#1}{#2}}
\newcommand\AtEndOfFile [2]
  {\AddToHook{file/after/#1}{#2}}
```

Some commands offered a starred form

```
\DeclareDocumentCommand \AtBeginOfPackageFile {smm}
   {\IfBooleanTF{#1}%
     {\@ifpackageloaded{#2}%
         {#3}%
         {\AddToHook{package/before/#2}{#3}}}%
     {\AddToHook{package/before/#2}{#3}}%
   }
\DeclareDocumentCommand \AtEndOfPackageFile {smm}
   {\IfBooleanTF{#1}%
     {\@ifpackageloaded{#2}%
         {#3}%
         {\AddToHook{package/after/#2}{#3}}}%
     {\AddToHook{package/after/#2}{#3}}%
   }
```

Are the * forms here of any use? I know they are use 3–4 times on CTAN but I wonder if those are real or mistaken usages.

```
\DeclareDocumentCommand \AtBeginOfClassFile {smm}
   {\IfBooleanTF{#1}%
     {\@ifclassloaded{#2}%
         {#3}%
         {\AddToHook{class/before/#2}{#3}}}%
     {\AddToHook{class/before/#2}{#3}}%
   }
\DeclareDocumentCommand \AtEndOfClassFile {smm}
   {\IfBooleanTF{#1}%
     {\@ifclassloaded{#2}%
         {#3}%
         {\AddToHook{class/after/#2}{#3}}}%
     {\AddToHook{class/after/#2}{#3}}%
   }
\newcommand\AtBeginOfIncludeFile [2]
  {\AddToHook{include/before/#1}{#2}}
\newcommand\AtEndOfIncludeFile [2]
  {\AddToHook{include/end/#1}{#2}}
\newcommand\AfterIncludeFile [2]
  {\AddToHook{include/after/#1}{#2}}
```

⟨/filehook-ltx⟩

2.2 The `bidi` package first aid

The `bidi` package adds a lot of hooks in various places and those added to `\document` and `\enddocument` are now no longer necessary as the kernel already

provides the right hooks there.

However, we aren't trying to change that but instead only make sure that the existing patches still work by adding some first aid after `biditools` has been loaded.

If the package gets updated one can easily take that out simply through

```
\RemoveFromHook{file/after/biditools.sty}[firstaid]
```

This makes it easy to test new bidi code while the first aid code is still in the kernel.

```
⟨*kernel⟩
\AddToHook{file/after/biditools.sty}[firstaid]{%
  \FirstAidNeededT{biditools}{sty}%
                  {2020/05/13 v2 Programming tools for bidi package}%
  {%
```

`bidi` adds some code to the beginning of `\document` which contains `\endgroup` and `\begingroup` which is no longer correct.

Patching `\document` using `\bidi@patchcmd` doesn't work so we take the extra groups out by hand:

```
  \def\firstaid@bidi@document@patch
          \endgroup#1\begingroup#2\firstaid@bidi@document@patch
          {\unexpanded{#1#2}}%
  \edef\document{\expandafter\firstaid@bidi@document@patch\document
            \firstaid@bidi@document@patch}%
```

There are also some patches into `\enddocument`, some continue to go in but one fails, so we add that now into the right place.

```
  \AddToHook{enddocument/info}%
            {\let\bidi@AfterEndDocumentCheckLabelsRerun\@firstofone
              \bidi@afterenddocumentchecklabelsrerunhook}%
  }%
}
```

2.3 The `dinbrief` class first aid

Again a case of a no longer correct `\endgroup` in document. Here the fix is simply though.

```
\AddToHook{file/after/dinbrief.cls}[firstaid]{%
  \FirstAidNeededT{dinbrief}{cls}{2000/03/02 LaTeX2e class}%
                  {\AddToHook{env/document/begin}{\begingroup}}%
}
```

2.4 The `pgfpages` and `pgfmorepages` first aid

`pgfpages` alters the `\shipout` primitive to support multiple page up scenarios. If used together with `atbegshi` that worked because the alterations done by `atbegshi` came later and so used the new definition provide by `pgfpages`. Now that the code from `atbegshi` is already in the kernel this further redefinition doesn't happen with the result that the change to `\shipout` comes to late and breaks the kernel processes.

```
\ExplSyntaxOn
\AddToHook{file/after/pgfpages.sty}[firstaid]{%
```

Undo overwriting `\shipout`:

```
  \cs_gset_eq:NN \shipout \pgfpages@originalshipout
```

Instead overwrite the L3 programming layer name of the primitive. This is really an absolute no-go, but for now the simplest solution to keep the original code running.

It will be replaced when the "configuration points" interface for LaTeX becomes available. At that point the package will be able to set up a different strategy for doing shipouts and without the need to overrite a primitive (which it did in the past and which we do below) and then this code here can be taken out again.

```
  \cs_set_eq:NN \pgfpages@originalshipout \tex_shipout:D
  \cs_set_eq:NN \tex_shipout:D \pgfpages@interceptshipout
}
\ExplSyntaxOff
```

Same issue with `pgfmorepages` but slightly different implementation (sigh).

```
\ExplSyntaxOn
\AddToHook{file/after/pgfmorepages.sty}[firstaid]{
  \cs_set_nopar:Npn \pgfhookintoshipout {
    \cs_set_eq:NN \pgfpages@originalshipout \tex_shipout:D
    \cs_set_eq:NN \tex_shipout:D \pgfpages@interceptshipout
  }
}
\ExplSyntaxOff
```

⟨/kernel⟩

2.5 The `everysel` package first aid

The `\selectfont` command got a hook (with the 2021/05 release) which was originally provided by the `everysel` package. Now that it is in the kernel this package is no longer needed (or only in a simplified manner).

If it is requested we replace it with a simplified package (until) it gets updated at which point this line can be removed.

```
⟨*kernel⟩
% this has been updated
%\declare@file@substitution{everysel.sty}{everysel-ltx.sty}
⟨/kernel⟩
```

```
⟨*everysel-ltx⟩
\ProvidesPackage{everysel-ltx}
   [2020/12/04 v1.0a
     Emulation of the original everysel^^Jpackage with kernel methods]
\newcommand*{\EverySelectfont}[1]
   {\AddToHook{selectfont}{#1}}
\newcommand*{\AtNextSelectfont}[1]
   {\AddToHookNext{selectfont}{#1}}
⟨/everysel-ltx⟩
```

2.6 The `CJK` package first aid

```
⟨*kernel⟩
```

The package redefines `\selectfont` to add some code but otherwise uses the old definition. So we make a copy of the newer kernel definition and restore it after the package got loaded. The extra code that the package needs can go into the newly provided hook instead.

```
\DeclareCommandCopy\CJK@selectfont\selectfont
\AddToHook{file/after/CJK.sty}[firstaid]{%
  \FirstAidNeededT{CJK}{sty}%
                   {2015/04/18 4.8.4}%
  {%
    \DeclareCommandCopy\selectfont\CJK@selectfont
    \AddToHook{selectfont}[CJK]{%
        \expandafter\ifx\csname CJK@\curr@fontshape\endcsname \relax
        \else
          \CJK@bold@false
          \csname CJK@\curr@fontshape\endcsname
          \fi
    }%
  }%
}
```

2.7 `\footref` first aid

A few classes unconditionally define `\footref`. Until that has changed we provide some first aid to let them do this.

— This seems to be resolved now —

```
%\AddToHook{file/after/scrkbase.sty}[firstaid]{%
%  \FirstAidNeededT{scrkbase}{sty}%
%       {2020/09/21 v3.32 KOMA-Script package (KOMA-Script-dependent basics and keyval usage)
%    {\let\footref\@undefined}
%   }
%\AddToHook{class/before/memoir}[firstaid]{%
% % for version {2020/10/04 v3.7n configurable book, report, article document class}%
%  \let\footref\@undefined
%}
```

2.8 The `bigfoot` first aid

The `bigfoot` packages makes the assumption that two `\newinsert` allocations have a recognisable order in their numbers, the second one has a lower number. This was correct in the classic TEX implementation but with the extended allocation possibilities of all modern engines is no longer the case and there is a point where the allocations take a "jump" breaking the odering assumption. These days we are fairly close to that point and depending on how many packages are loaded before `bigfoot` the package breaks.

This firstaid therefore jumps over the problematical point by pushing the count allocation to a safe value if necessary.

```
\AddToHook{file/after/bigfoot.sty}{%
   \ifnum\count10<\insc@unt
     \global\count10=\insc@unt
   \fi
```

We also correct a bug that `bigfoot` tries to shift mark registers, but in LaTeX (at least since 2015) the allocation number is not 266, so it does that to a random number of mark registers (which sometimes blows up depending on the value in 266).

```
\def\FN@allmarks#1{\@elt{#1}%
  \ifnum#1<\count256 %<--- problem: 266 isn't the counter for marks
    \expandafter\FN@allmarks\expandafter{\number\numexpr#1+\@ne}%
    \fi}%
}
```

2.9 `ulem` first aid

In 2020 we fixed various kernel commands to accept `calc` syntax. The `ulem` package redefines some internals and that now conflicts with the new definitions as they involve an extra group. So we alter the definition of `\@hspace` if `ulem` was loaded. This is not perfect, obviously, so it will go out the moment `ulem` gets adjusted.

```
\AddToHook{file/after/ulem.sty}[firstaid]{%
  \def\@hspace#1{\begingroup\setlength\skip@{#1}%
                 \edef\x{\endgroup\hskip\the\skip@\relax}\x}%
  }
```

2.10 `varwidth` first aid

The `varwidth` package does a lot of low-level paragraph manipulation assuming traditional TeX paragraphs. However, with the paragraph hooks we end up with one extra glue 0pt item on the vertical list and if that isn't removed then the package doesn't find its penalties.

So this needs to be removed as well by adding an additional `\unskip`.

```
\AddToHook{file/after/varwidth.sty}[firstaid]{%
  \FirstAidNeededT{varwidth}{sty}%
       {2009/03/30 ver 0.92; \space Variable-width minipages}%
    {%
\def\@vwid@sift{%
  \skip@\lastskip\unskip
  \ifdim\lastskip=\z@\unskip\fi   % <---- the first aid here (not just unskip)
  \dimen@\lastkern\unkern
  \count@\lastpenalty\unpenalty
  \setbox\z@\lastbox
  \ifvoid\z@ \advance\sift@deathcycles\@ne \else \sift@deathcycles\z@ \fi
  \ifnum\sift@deathcycles>33
    \let\@vwid@sift\relax
    \PackageWarning{varwidth}{Failed to reprocess entire contents}%
  \fi
  \ifnum\count@=\@vwid@preeqp \@vwid@eqmodefalse\fi
  \ifnum\count@=\@vwid@posteqp \@vwid@eqmodetrue\fi
  \ifnum\count@=\@vwid@toppen % finished
    \let\@vwid@sift\relax
  \else\ifnum\count@=\@vwid@offsets
    \@vwid@setoffsets
  \else
    \ifnum\count@=\@vwid@postw
```

```
    \else
      \@vwid@resetb % reset box \z@ or measure it
    \fi
    \@vwid@append
  \fi\fi
  \@vwid@sift}%
     }%
   }
```

2.11 Temporary fixes for the kernel (until the next patch-level release)

This fixes GitHub issue 591. It is only needed in LuaTEX and replaces just one instance of `\tex_par:D` with the following version which removes other nodes in the current list first.

```
\ExplSyntaxOn
\sys_if_engine_luatex:T
  {
    \newluafunction \g__para_end_empty_par_id_int
    \exp_args:Nx \everyjob {
      \exp_not:V \everyjob
      \exp_not:N \lua_now:n {
        local~texnest, flush_list, par_token = tex.nest, node.flush_list, token.create'tex_pa
        lua.get_functions_table()[\int_use:N \g__para_end_empty_par_id_int] = function()
          local~nest_level = texnest.top~
          local~cur_head = nest_level.head~
          flush_list(cur_head.next)
          nest_level.tail, cur_head.next = cur_head, nil~
          token.put_next(par_token)
        end
      }
    }
    \protected \luadef \__para_end_empty_par: \g__para_end_empty_par_id_int
    \group_begin:
    \cs_set:Npn \__para_extract_everypar:w #1 \the \toks #2 \s_stop
      {
        \tl_gset:Nn \g__para_standard_everypar_tl {
          \box_gset_to_last:N \g_para_indent_box
          \group_begin:
            \__para_end_empty_par:
          \group_end:
          \tex_everypar:D { \msg_error:nnnn { hooks }{ para-mode }{before}{vertical} }
          \@kernel@before@para@before
          \hook_use:n {para/before}
          \group_begin:
            \tex_everypar:D {}
            \skip_zero:N \tex_parskip:D
            \tex_noindent:D
          \group_end:
          \tex_everypar:D{\g__para_standard_everypar_tl}
          \@kernel@before@para@begin
          \hook_use:n {para/begin}
          \if_mode_horizontal: \else:
```

```
            \msg_error:nnnn { hooks }{ para-mode }{begin}{vertical} \fi:
          \__para_handle_indent:
          \the \toks #2
        }
      }
    \exp_last_unbraced:No \__para_extract_everypar:w \g__para_standard_everypar_tl \s_stop
    \group_end:
  }
\ExplSyntaxOff
```

The next fixes Github issue 605 (by inserting a `\scan_stop:` before doing the check for hmode).

```
\ExplSyntaxOn
\cs_set_protected:Npn \para_end: {
  \scan_stop:
  \mode_if_horizontal:TF {
    \mode_if_inner:F {
         \tex_unskip:D
         \hook_use:n{para/end}
         \@kernel@after@para@end
         \mode_if_horizontal:TF {
           \if_int_compare:w 0 < \tex_lastnodetype:D
             \tex_kern:D \c_zero_dim
           \fi:
           \tex_par:D
           \hook_use:n{para/after}
           \@kernel@after@para@after
         }
         { \msg_error:nnnn { hooks }{ para-mode }{end}{horizontal} }
    }
  }
  \tex_par:D
}
\cs_set_eq:NN \par     \para_end:
\cs_set_eq:NN \@@par   \para_end:
\cs_set_eq:NN \endgraf \para_end:
\ExplSyntaxOff
```

⟨/kernel⟩

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.