The latex-lab-floats package
Tagging of floats

LaTeX Project[*]

v0.8 2023-05-17

Abstract

The following code implements a first draft for the tagging of float environments

1 Introduction

The code here handle the tagging of float environments.

Figures (and tables) are in LaTeX typically typeset in float environments. These are boxes which can *float* away to special float areas on the pages, e.g., to the top or the bottom of a page or to special float pages. If the rules allow it they can also be placed in the main text stream ("here"). Floats can also be collected at the end of the document. In either case the order within each type of floats (e.g., figures, tables, algorithms, etc.) is preserved.

A special type, called a H-float, (provided by the float package) is always placed in the main text stream and does not necessarly preserve the order with normal floats of the same type: It is basically a minipage with a caption.

Floats typically contain a figure (or a table, etc.) and a caption, but more complex constructions with subfigures, copyright statements, sources or additional description are possible too.

In the LaTeX source a float is normally more or less at the place of the first call-out, but when preparing a document for print the code is sometimes moved to place floats in a more visually pleasing way.

2 Tagging

Floats (with the exception of H-floats) do not belong into the text stream, they are "consultation objects": Readers must be able to choose if and when they read the float. Floats have captions, the PDF rules require that a `Caption` is the first or last structure in its parent structure. This poses some challenges on a good tagging.

In PDF 2.0 there is the suitable `Aside` tag which hopefully will be handled correctly regarding the reading order once processor actually support PDF 2.0. But in PDF 1.7 we rolemap it to `Note` and this doesn't lead to a good reading order. The code therefore collect the float structures and moves them to a `Sect` the end of the document or the chapter (H-floats once they are handled will not be moved).

[*] Initial implementation done by Ulrike Fischer

To fulfill the requirement that a `Caption` should be at the begin or end, we always move it to the begin of the structure. If a float has two captions the author has to insert a command which splits the float in two.

Subfigures and subcaptions are currently not handled, but will be implemented as simple `Part` with their own `Caption`.

3 Links

The code disable the caption patches from hyperref. It will add an anchor at the begin of the float or a split. It changes caption so that a link to a caption label will go to the begin of the float.

4 Tools

The code add two keys for the `\tagtool` command

flush-floats This will flush out the collected floats sofar (currently table and figure. The value is a sectioning level, e.g. `section` or `chapter`, the floats will then inserted as a `Sect` of this level (all `Sect` of smaller or equal level are closed). The key then starts a new container for following floats. If no value is given, the `Sect` is at the document level. The code automatically flush all open floats at the end of the document.

split-float This can be used inside a float if there are two captions. It will only work reasonably well if the content of the float parts are in a sensible order and can be separated by this command. More complex setups with tabulars will need more thoughts

```
⟨@@=tag⟩
⟨*package⟩
```

5 Implementation

```
\ProvidesExplPackage {latex-lab-testphase-float} {2023-04-28} {0.8}
  {Code related to the tagging of floats}
```

5.1 Variables

We rolemap to Aside, and float sections to Sect

```
\tagpdfsetup{add-new-tag=float/Aside}
\tagpdfsetup{add-new-tag=figures/Sect}
\tagpdfsetup{add-new-tag=tables/Sect}
```

`\g__tag_float_sect_prop`
`\g__tag_float_types_seq`

These variables will hold the structure number for the float container and the list of float types. Currently only figure and table are supported TODO: interface to declare new float types.

```
\prop_new:N \g__tag_float_sect_prop
\seq_new:N  \g__tag_float_types_seq
\seq_gput_right:Nn \g__tag_float_types_seq {figure}
\seq_gput_right:Nn \g__tag_float_types_seq {table}
```

(*End definition for* `\g__tag_float_sect_prop` *and* `\g__tag_float_types_seq`.)

\g__tag_float_sect_bool With this boolean float collection is switched on and off. Currently it is always on and set globally. TODO: think if an interface is needed. TODO: would a local variable make more sense?

```
\bool_new:N       \g__tag_float_sect_bool
\bool_gset_true:N  \g__tag_float_sect_bool
```

(*End definition for* `\g__tag_float_sect_bool`.)

5.2 Moving float structures

Currently it is for all float types or none. Probably we will need some more options here to select some float types.

__tag_float_init_collect: This initializes a container structure for every float type. It can be used more than once in a document, this allows to have e.g. chapter wise containers.

```
\cs_new_protected:Npn\__tag_float_init_collect:
  {
    \bool_if:NT\g__tag_float_sect_bool
      {
        \seq_map_inline:Nn\g__tag_float_types_seq
          {
            \tag_struct_begin:n{tag=##1s,stash}
            \prop_gput:Nnx\g__tag_float_sect_prop {##1-struct}{\int_use:N\c@g__tag_struct_abs
            \tag_struct_end:
          }
      }
  }
```

(*End definition for* `\__tag_float_init_collect:`.)

__tag_float_stop_sect: This pushes out the floats. For every type is checks if there is actually a float of this type and then writes out the container structure.

```
\cs_new_protected:Npn \__tag_float_stop_sect:
  {
    \bool_if:NT\g__tag_float_sect_bool
     {
      \seq_map_inline:Nn\g__tag_float_types_seq
        {
          \prop_get:NnNT\g__tag_float_sect_prop{##1-used}\l__tag_tmpa_tl
            {
              \exp_args:Ne
              \tag_struct_use_num:n{\prop_item:Nn\g__tag_float_sect_prop{##1-struct}}
              \prop_gremove:Nn \g__tag_float_sect_prop{##1-used}
            }
        }
     }
  }
```

(*End definition for* `\__tag_float_stop_sect:`.)

`flush-floats` This is a key for `\tagtool` to flush out the collected floats. The value allows to set to which level the create Sect contains. So `section` will close all previous Sect until the section level and create a new section.

```
\keys_define:nn { tag / tool}
 {
   flush-floats .code:n =
     {
       \keys_set:nn {tag / tool} {sec-stop=#1}
       \__tag_float_stop_sect:
       \__tag_float_init_collect:
     },
   flush-float .default:n = Document
 }
```

(*End definition for* `flush-floats`. *This function is documented on page* **??**.)

We need at least one pair

```
\AddToHook{begindocument/end}[latex-lab/float]
  {\__tag_float_init_collect:}
\AddToHook{tagpdf/finish/before}[latex-lab/float]
  {\par\__tag_sec_end:n{-10}\__tag_float_stop_sect:}
\DeclareHookRule{tagpdf/finish/before}{latex-lab/float}{before}{tagpdf}
```

5.3 Splitting floats

`split-float` TODO: check if the target affect spacing!!

```
\keys_define:nn { tag / tool}
  {
   split-float .code:n =
    {
      \__tag_float_end:
      \__tag_float_begin:
      \MakeLinkTarget[tagstructure]{g__tag_struct_abs_int}
    }
 }
```

(*End definition for* `split-float`. *This function is documented on page* **??**.)

5.4 Patching

`\__tag_float_stop_par:`
`\__tag_float_start_par:`

if a float is in a par, we need commands to stop and restart the P-mc

```
\cs_new_protected:Npn \__tag_float_stop_par:
  {
    \tag_mc_end:
    \bool_if:NF \g__tag_float_sect_bool
     {
      \tag_struct_end:
     }
  }
\cs_new_protected:Npn \__tag_float_start_par:
  {
    \bool_if:NF \g__tag_float_sect_bool
     {
      \tag_struct_begin:n{tag=text}%
     }
   \tag_mc_begin:n{tag=P}
  }

```

(*End definition for* `\__tag_float_stop_par:` *and* `\__tag_float_start_par:`*.*)

These commands are the main commands to start and end the float tagging.

```
\cs_new_protected:Npn \__tag_float_begin:
 {%
```

We test if the float structure should be included directly or move to a dedicated section.

```
  \bool_if:NTF\g__tag_float_sect_bool
   {
     \exp_args:Ne
     \tag_struct_begin:n{tag=float,parent=0\prop_item:No\g__tag_float_sect_prop{\@captype-str
     \prop_gput:Nxx \g__tag_float_sect_prop {\@captype-used}{true}
   }
   {
     \tag_struct_begin:n{tag=float}
   }
     \tl_set:Nx\@current@float@struct{\tag_get:n{struct_num}}%
     \typeout{Float structure: \@current@float@struct}
 }

\cs_new_protected:Npn\__tag_float_end:{\tag_struct_end:} %end Aside

```

This patches the main command `\@xfloat`. There is a : in the code, so we disable expl3 syntax

```
\ExplSyntaxOff
\def\@xfloat #1[#2]{%
  \@nodocument
  \def \@captype {#1}%
   \def \@fps {#2}%
   \@onelevel@sanitize \@fps
   \def \reserved@b {!}%
   \ifx \reserved@b \@fps
     \@fpsadddefault
   \else
     \ifx \@fps \@empty
       \@fpsadddefault
     \fi
   \fi
   \ifhmode
     \@bsphack
```

If the float is in hmode we have to interrupt the P

```
     \@nameuse{__tag_float_stoppar:} %<---end P
     \@floatpenalty -\@Mii
   \else
     \@floatpenalty-\@Miii
   \fi
  \ifinner
     \@parmoderr\@floatpenalty\z@
  \else
```

```
    \@next\@currbox\@freelist
      {%
       \@tempcnta \sixt@@n
       \expandafter \@tfor \expandafter \reserved@a
         \expandafter :\expandafter =\@fps
         \do
          {%
           \if \reserved@a h%
             \ifodd \@tempcnta
             \else
               \advance \@tempcnta \@ne
             \fi
           \else\if \reserved@a t%
             \@setfpsbit \tw@
           \else\if \reserved@a b%
             \@setfpsbit 4%
           \else\if \reserved@a p%
             \@setfpsbit 8%
           \else\if \reserved@a !%
             \ifnum \@tempcnta>15
               \advance\@tempcnta -\sixt@@n\relax
             \fi
           \else
             \@latex@error{Unknown float option `\reserved@a'}%
             {Option `\reserved@a' ignored and `p' used.}%
             \@setfpsbit 8%
           \fi\fi\fi\fi\fi
           }%
       \@tempcntb \csname ftype@\@captype \endcsname
       \multiply \@tempcntb \@xxxii
       \advance \@tempcnta \@tempcntb
       \global \count\@currbox \@tempcnta
       }%
    \@fltovf
  \fi
```

This starts the structure for the float.

```
  \@nameuse{__tag_float_begin:}%
  \global \setbox\@currbox
    \color@vbox
      \normalcolor
      \vbox \bgroup
        \hsize\columnwidth
        \@parboxrestore
        \@floatboxreset
```

We add a target for links. TODO: check that it doesn't affect spacing!!

```
        \MakeLinkTarget[tagstructure]{g__tag_struct_abs_int}%
}%
```

The end code of the float ...

```
\def\end@float{%
  \@endfloatbox
  \@nameuse{__tag_float_end:}%
  \ifnum\@floatpenalty <\z@
```

```
    \@largefloatcheck
    \@cons\@currlist\@currbox
    \ifnum\@floatpenalty <-\@Mii
      \penalty -\@Miv
      \@tempdima\prevdepth
      \vbox{}%
      \prevdepth\@tempdima
      \penalty\@floatpenalty
    \else
      \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
      \@nameuse{__tag_float_start_par:} %restart P safe here??
    \fi
  \fi
}
\ExplSyntaxOn
```

5.5 Handling captions

To avoid that hyperref interferes we disable its patches:

```
\def\hyper@nopatch@caption{}
```

As we will use the structure number in the target, we need to provide a theH-representation. (Once the kernel will create theH-representation generally this will be provided automatically, as tagpdf uses `\newcounter`)

```
\providecommand\theHg__tag_struct_abs_int{\int_use:N\c@g__tag_struct_abs_int}
```

`\@makecaption` `\@makecaption` is defined by the classes so we overwrite it for now at begin document.

```
\AddToHook{begindocument}
  {
    \long\def\@makecaption#1#2{%
      \vskip\abovecaptionskip
      \xdef\@currentHref{tagstructure.\@current@float@struct}%
```

we don't want tagging when storing the caption for the singleline check

```
      \tag_stop:n{caption}
      \sbox\@tempboxa{#1:~#2}%
      \tag_start:n{caption}
```

we stop paratagging. TODO: check

```
      \tagtool{para=false}
      \tag_struct_begin:n{tag=Caption,parent=\@current@float@struct}
```

move the caption to the begin of the float structure:

```
     \seq_gpop_right:cN {g__tag_struct_kids_\@current@float@struct _seq}\l__tag_tmpa_tl
     \seq_gput_left:cV  {g__tag_struct_kids_\@current@float@struct _seq}\l__tag_tmpa_tl
      \ifdim \wd\@tempboxa >\hsize
        \tag_struct_begin:n{tag=Lbl}
        \tag_mc_begin:n{}
          #1:~
        \tag_mc_end:
        \tag_struct_end:
        \tag_mc_begin:n{}
          #2\par
        \tag_mc_end:
      \else
```

we don’t reuse the box as it doesn’t contain tagging, but set the text explicitly.

```
          \global \@minipagefalse
        \hb@xt@\hsize{\hfil
         \tag_struct_begin:n{tag=Lbl}
          \tag_mc_begin:n{}
           #1:~
          \tag_mc_end:
         \tag_struct_end:
         \tag_mc_begin:n{}
          #2\par
         \tag_mc_end:\hfil}%
       \fi
       \tag_struct_end: %caption
      \vskip\belowcaptionskip}
  }
```

(*End definition for* `\@makecaption`*. This function is documented on page* **??**.)

```
⟨/package⟩
```

```
⟨*latex-lab⟩
\ProvidesFile{float-latex-lab-testphase.ltx}
        [2023-04-30 v0.8 code related to the tagging of floats]
\RequirePackage{latex-lab-testphase-float}
⟨/latex-lab⟩
```