The latex-lab-floats package
Tagging of floats

LaTeX Project*

v0.811 2025-10-17

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 necessarily 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 collects the float structures and moves them to a `Sect` at 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`
`split-float`

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.

5 Kernel commands

`\@current@float@struct` This variable holds the number of the current float structure. With tagging this is the structure number, without tagging a unique counter. A float can contain more than one float structure (e.g. if there is more than one caption).

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

```
1 ⟨@@=tag⟩
2 ⟨*package⟩
```

6 Implementation

```
\ProvidesExplPackage {latex-lab-testphase-float} {\ltlabfloatdate} {\ltlabfloatversion}
  {Code related to the tagging of floats}
```

6.1 Variables

We rolemap floats to Aside, and float sections to Sect.

\g__tag_float_sect_prop
\g__tag_float_types_seq
\@current@float@struct
\g__tag_float_int

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. To set the target for links we need also a unique counter. With tagging we could use the structure number, but the structure commands now are hidden inside tagging sockets so we use a dedicated counter.

```
\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}
\tl_new:N\@current@float@struct
\int_new:N\g__tag_float_int
```

(*End of definition for* `\g__tag_float_sect_prop` *and others. This variable is documented on page 2.*)

\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 of definition for* `\g__tag_float_sect_bool`.)

__tag_float_init:

To be able to set unique targets for links, we need a counter outside the tagging sockets. TODO: check if this command should be public or a socket or a hook.

```
\cs_new_protected:Npn \__tag_float_init:
 {
   \int_gincr:N \g__tag_float_int
   \tl_set:Ne \@current@float@struct { \int_use:N \g__tag_float_int}
 }
```

(*End of definition for* `\__tag_float_init:`.)

6.2 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:Nne\g__tag_float_sect_prop {##1-struct}{\int_use:N\c@g__tag_struct_abs_in
            \tag_struct_end:
          }
      }
  }
```

(*End of 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 of 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 of definition for* `flush-floats`. *This function is documented on page 2.*)

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}
```

6.3 Splitting floats

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

```
\keys_define:nn { tag / tool}
  {
   split-float .code:n =
    {
      \UseTaggingSocket{float/end}
      \__tag_float_init:
      \UseTaggingSocket{float/begin}
      \MakeLinkTarget*{floatstructure.\@current@float@struct}
    }
 }
```

(*End of definition for* `split-float`*. This function is documented on page 2.*)

6.4 Tagging sockets

For now we test if the sockets are already defined

```
\socket_if_exist:nF {tagsupport/float/begin}
  {
    \NewTaggingSocket{float/hmode/begin}{0}
    \NewTaggingSocket{float/hmode/end}{0}
    \NewTaggingSocket{float/begin}{0}
    \NewTaggingSocket{float/end}{0}
  }
```

pport/float/hmode/begin) (*plug*) This plug should be used if a float is called in hmode. It then closes the MC-chunks and starts the structure.

```
\NewTaggingSocketPlug{float/hmode/begin}{default}
  {
    \__tag_float_stop_par:
  }
\AssignTaggingSocketPlug{float/hmode/begin}{default}
```

gsupport/float/hmode/end) (*plug*) This plug should be used if a float is called in hmode and the end of the float it then restarts the MC.

```
\NewTaggingSocketPlug{float/hmode/end}{default}
  {
    \__tag_float_start_par:
  }
\AssignTaggingSocketPlug{float/hmode/end}{default}
```

(tagsupport/float/begin) (*plug*)

```
\NewTaggingSocketPlug{float/begin}{default}
  {
    \__tag_float_begin:
  }
\AssignTaggingSocketPlug{float/begin}{default}
```

t (tagsupport/float/end) (*plug*)

```
\NewTaggingSocketPlug{float/end}{default}
  {
    \__tag_float_end:
  }
\AssignTaggingSocketPlug{float/end}{default}
```

`\__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 of 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-
struct}}%
     \prop_gput:Nee \g__tag_float_sect_prop {\@captype-used}{true}
   }
   {
     \tag_struct_begin:n{tag=float}
   }
     \tl_set:Ne\@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

```

6.5 Patching

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

```
     \UseTaggingSocket{float/hmode/begin}%
     \@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.

```
  \csname __tag_float_init:\endcsname
  \UseTaggingSocket{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*{\@captype.struct.\@current@float@struct}%
}%
```

The end code of the float ...

```
\def\end@float{%
  \@endfloatbox
  \UseTaggingSocket{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
      \UseTaggingSocket{float/hmode/end}%
    \fi
  \fi
}
```

and similar for double floats:

```
\def\end@dblfloat{%
  \if@twocolumn
    \@endfloatbox
```

```
    \UseTaggingSocket{float/end}%
    \ifnum\@floatpenalty <\z@
      \@largefloatcheck
      \global\dp\@currbox1sp %
      \@cons\@currlist\@currbox
      \ifnum\@floatpenalty <-\@Mii
        \penalty -\@Miv
        \@tempdima\prevdepth
        \vbox{}%
        \prevdepth\@tempdima
        \penalty\@floatpenalty
      \else
        \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
        \UseTaggingSocket{float/hmode/end}%
      \fi
    \fi
  \else
    \end@float
  \fi
}%
\ExplSyntaxOn
```

6.6 Handling captions

To avoid that hyperref interferes we disable its patches:

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

6.6.1 (Tagging) sockets

First some temporary sockets. These sockets are in lttagging.

```
\socket_if_exist:nF {tagsupport/caption/begin}
 {
  \NewTaggingSocket{caption/begin}{1}
  \NewTaggingSocket{caption/end}{0}
  \NewTaggingSocket{caption/label/begin}{0}
  \NewTaggingSocket{caption/label/end}{0}
 }
```

These socket are currently defined in tagpdf.

```
\socket_if_exist:nF {tagsupport/para/begin}
 {
   \NewTaggingSocket{para/begin}{0}
   \NewTaggingSocket{para/end}{0}
 }
```

`caption/label` (*socket*) This socket is a lightweight start for some interface to format the label or add a font command. The argument is the label text. The default plug `kernel` adds a colon and a space. TODO: revisit after checking float and caption packages to identify which sockets and hooks are needed.

```
\NewSocket{caption/label}{1}
```

kernel (caption/label) (*plug*) The standard label formatting from the kernel.

```
\NewSocketPlug{caption/label}{kernel}
 {
   #1:~
 }
\AssignSocketPlug{caption/label}{kernel}
```

default (*plug*) The caption begin socket takes an argument: the structure number of the parent float. If the argument is empty, the current structure is used. TODO: a tagpdf key that moves a structure to the begin of the parent. The caption is moved to the first position with the firstkid option.

```
\NewTaggingSocketPlug{caption/begin}{default}
  {
     \tl_if_empty:eTF {#1}
      {
        \tag_struct_begin:n{tag=Caption,firstkid}
      }
      {
        \tag_struct_begin:n{tag=Caption,parent=#1,firstkid}
      }
     \bool_set_true:N \l__tag_para_flattened_bool
  }
\AssignTaggingSocketPlug{caption/begin}{default}
```

default (*plug*)

```
\NewTaggingSocketPlug{caption/end}{default}
  {
    \tag_struct_end:
  }
\AssignTaggingSocketPlug{caption/end}{default}
```

ort/caption/label/begin) (*plug*)

```
\NewTaggingSocketPlug{caption/label/begin}{default}
  {
```

suppress para tagging at the begin.

```
     \tagpdfparaOff
     \tag_struct_begin:n{tag=Lbl}
     \tag_mc_begin:n{}
  }
\AssignTaggingSocketPlug{caption/label/begin}{default}
```

pport/caption/label/end) (*plug*)

```
\NewTaggingSocketPlug{caption/label/end}{default}
  {
    \tag_mc_end:
    \tag_struct_end:
    \tagpdfparaOn
  }
\AssignTaggingSocketPlug{caption/label/end}{default}
```

6.6.2 Redefinitions

With hyperref that means that the `\refstepcounter` now can affect spacing so we change that to the kernel refstepcounter:

```
\def\caption{%
   \ifx\@captype\@undefined
     \@latex@error{\noexpand\caption\c_space_tl outside~float}\@ehd
     \expandafter\@gobble
   \else
```

if a caption is used outside a float no target has been set and `\@current@float@struct` is empty

```
     \tl_if_empty:NTF\@current@float@struct
      {
        \refstepcounter\@captype
      }
      {
        \@kernel@refstepcounter\@captype
```

we need to reset the target for `\addcontentsline`. We use `\@captype` to support autoref.

```
        \xdef\@currentHref{\@captype.struct.\@current@float@struct}%
      }
     \expandafter\@firstofone
   \fi
   {\@dblarg{\@caption\@captype}}%
}
```

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

```
\NewHookWithArguments{cmd/@makecaption/before}{2}
\AddToHook{begindocument}
  {
    \long\def\@makecaption#1#2{%
      \UseHookWithArguments{cmd/@makecaption/before}{2}{#1}{#2}%
      \vskip\abovecaptionskip
```

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

```
      \SuspendTagging{\@makecaption}
      \sbox\@tempboxa{\UseSocket{caption/label}{#1}#2}%
      \ResumeTagging{\@makecaption}
```

We pass `\@current@float@struct` as parent structure number. If that is empty the socket will use the parent structure and hope ...

```
      \UseTaggingSocket{caption/begin}{\@current@float@struct}
      \ifdim \wd\@tempboxa >\hsize
      \UseTaggingSocket{caption/label/begin}
      \UseSocket{caption/label}{#1}
      \UseTaggingSocket{caption/label/end}
      \UseTaggingSocket{para/begin}
          #2
      \par
      \else
```

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

```
          \global \@minipagefalse
        \hb@xt@\hsize{\hfil
         \UseTaggingSocket{caption/label/begin}
         \UseSocket{caption/label}{#1}
         \UseTaggingSocket{caption/label/end}
         \UseTaggingSocket{para/begin}
          #2
         \UseTaggingSocket{para/end}
         \hfil}%
       \fi
      \UseTaggingSocket{caption/end}
      \vskip\belowcaptionskip}
 }
```

(*End of definition for* `\@makecaption`. *This function is documented on page 2.*)

```
⟨/package⟩

⟨*latex-lab⟩
\ProvidesFile{float-latex-lab-testphase.ltx}
        [\ltlabfloatdate\space v\ltlabfloatversion\space latex-lab wrapper float]
\RequirePackage{latex-lab-testphase-float}
⟨/latex-lab⟩
```