

The latex-lab-floats package

Tagging of floats

L^AT_EX 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 L^AT_EX 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 L^AT_EX 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

```
3 \ProvidesExplPackage {latex-lab-testphase-float} {\ltlabfloatdate} {\ltlabfloatversion}  
4 {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 These variables will hold the structure number for the float container and the list of float  
\g__tag_float_types_seq types. Currently only figure and table are supported TODO: interface to declare new  
\@current@float@struct float types. To set the target for links we need also a unique counter. With tagging  
  \g__tag_float_int we could use the structure number, but the structure commands now are hidden inside  
 tagging sockets so we use a dedicated counter.
```

```
5 \prop_new:N \g__tag_float_sect_prop  
6 \seq_new:N \g__tag_float_types_seq  
7 \seq_gput_right:Nn \g__tag_float_types_seq {figure}  
8 \seq_gput_right:Nn \g__tag_float_types_seq {table}  
9 \tl_new:N\@current@float@struct  
10 \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?
```

```
11 \bool_new:N \g__tag_float_sect_bool  
12 \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.
```

```
13 \cs_new_protected:Npn \__tag_float_init:  
14 {  
15   \int_gincr:N \g__tag_float_int  
16   \tl_set:Ne \@current@float@struct { \int_use:N \g__tag_float_int}  
17 }
```

(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.
```

```

18 \cs_new_protected:Npn \__tag_float_init_collect:
19 {
20   \bool_if:NT\g__tag_float_sect_bool
21   {
22     \seq_map_inline:Nn\g__tag_float_types_seq
23     {
24       \tag_struct_begin:n{tag=##1s,stash}
25       \prop_gput:Nne\g__tag_float_sect_prop {##1-struct}{\int_use:N\c@g__tag_struct_abs_in
26       \tag_struct_end:
27     }
28   }
29 }

```

(End of definition for __tag_float_init_collect:.)

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

```

30 \cs_new_protected:Npn \__tag_float_stop_sect:
31 {
32   \bool_if:NT\g__tag_float_sect_bool
33   {
34     \seq_map_inline:Nn\g__tag_float_types_seq
35     {
36       \prop_get:NnNT\g__tag_float_sect_prop{##1-used}\l__tag_tmpa_tl
37       {
38         \exp_args:Ne
39         \tag_struct_use_num:n{\prop_item:Nn\g__tag_float_sect_prop{##1-struct}}
40         \prop_gremove:Nn \g__tag_float_sect_prop{##1-used}
41       }
42     }
43   }
44 }

```

(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.

```

45 \keys_define:nn { tag / tool}
46 {
47   flush-floats .code:n =
48   {
49     \keys_set:nn {tag / tool} {sec-stop=#1}
50     \__tag_float_stop_sect:
51     \__tag_float_init_collect:
52   },
53   flush-float .default:n = Document
54 }

```

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

We need at least one pair

```

55 \AddToHook{begindocument/end}[latex-lab/float]
56   {\_tag_float_init_collect:}
57 \AddToHook{tagpdf/finish/before}[latex-lab/float]
58   {\par\_tag_sec_end:n{-10}\_tag_float_stop_sect:}
59 \DeclareHookRule{tagpdf/finish/before}[latex-lab/float]{before}{tagpdf}

```

6.3 Splitting floats

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

```

60 \keys_define:nn { tag / tool}
61   {
62     split-float .code:n =
63     {
64       \UseTaggingSocket{float/end}
65       \_tag_float_init:
66       \UseTaggingSocket{float/begin}
67       \MakeLinkTarget*{floatstructure.\@current@float@struct}
68     }
69   }

```

(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

```

70 \socket_if_exist:nF {tagssupport/float/begin}
71   {
72     \NewTaggingSocket{float/hmode/begin}{0}
73     \NewTaggingSocket{float/hmode/end}{0}
74     \NewTaggingSocket{float/begin}{0}
75     \NewTaggingSocket{float/end}{0}
76   }

```

`support/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.

```

77 \NewTaggingSocketPlug{float/hmode/begin}{default}
78   {
79     \_tag_float_stop_par:
80   }
81 \AssignTaggingSocketPlug{float/hmode/begin}{default}

```

`support/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.

```

82 \NewTaggingSocketPlug{float/hmode/end}{default}
83   {
84     \_tag_float_start_par:
85   }
86 \AssignTaggingSocketPlug{float/hmode/end}{default}

```

`(tagssupport/float/begin)` (*plug*)

```

87 \NewTaggingSocketPlug{float/begin}{default}
88 {
89   \_tag_float_begin:
90 }
91 \AssignTaggingSocketPlug{float/begin}{default}

```

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

```

92 \NewTaggingSocketPlug{float/end}{default}
93 {
94   \_tag_float_end:
95 }
96 \AssignTaggingSocketPlug{float/end}{default}

```

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

```

\_tag_float_start_par:
97 \cs_new_protected:Npn \_tag_float_stop_par:
98 {
99   \tag_mc_end:
100   \bool_if:NF \g__tag_float_sect_bool
101   {
102     \tag_struct_end:
103   }
104 }
105 \cs_new_protected:Npn \_tag_float_start_par:
106 {
107   \bool_if:NF \g__tag_float_sect_bool
108   {
109     \tag_struct_begin:n{tag=text}%
110   }
111   \tag_mc_begin:n{tag=P}
112 }

```

(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.

```

113 \cs_new_protected:Npn \_tag_float_begin:
114 {%

```

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

```

115   \bool_if:NTF\g__tag_float_sect_bool
116   {
117     \exp_args:Ne
118     \tag_struct_begin:n{tag=float,parent=0\prop_item:No\g__tag_float_sect_prop{\@capttype-
struct}}%
119     \prop_gput:Nee \g__tag_float_sect_prop {\@capttype-used}{true}
120   }
121   {
122     \tag_struct_begin:n{tag=float}
123   }
124   \tl_set:Ne\@current@float@struct{\tag_get:n{struct_num}}%
125   \typeout{Float structure: \@current@float@struct}
126 }
127
128 \cs_new_protected:Npn\_tag_float_end:{\tag_struct_end:} %end Aside
129

```

6.5 Patching

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

```
130 \ExplSyntaxOff
131 \def\xfloat #1[#2]{%
132   \nodocument
133   \def \capttype {#1}%
134   \def \@fps {#2}%
135   \@onelevel@sanitize \@fps
136   \def \reserved@b {!}%
137   \ifx \reserved@b \@fps
138     \fpsaddddefault
139   \else
140     \ifx \@fps \empty
141       \fpsaddddefault
142     \fi
143   \fi
144   \ifhmode
145     \bsphack
```

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

```
146   \UseTaggingSocket{float/hmode/begin}%
147   \floatpenalty -\@Mii
148   \else
149     \floatpenalty-\@Miii
150   \fi
151 \ifinner
152   \@parmoderr\@floatpenalty\z@
153 \else
154   \@next\@currbox\@freelist
155   {%
156     \@tempcnta \sist@n
157     \expandafter \@tfor \expandafter \reserved@a
158     \expandafter :\expandafter =\@fps
159     \do
160     {%
161       \if \reserved@a h%
162         \ifodd \@tempcnta
163         \else
164           \advance \@tempcnta \@ne
165         \fi
166       \else\if \reserved@a t%
167         \@setfpsbit \tw@
168       \else\if \reserved@a b%
169         \@setfpsbit 4%
170       \else\if \reserved@a p%
171         \@setfpsbit 8%
172       \else\if \reserved@a !%
173         \ifnum \@tempcnta>15
174           \advance\@tempcnta -\sist@n\relax
175         \fi
176       \else
```

```

177         \@latex@error{Unknown float option `reserved@a'}%
178         {Option `reserved@a' ignored and `p' used.}%
179         \@setfpsbit 8%
180         \fi\fi\fi\fi\fi
181     }%
182     \@tempcntb \csname ftype@\@capytype \endcsname
183     \multiply \@tempcntb \@xxxii
184     \advance \@tempcnta \@tempcntb
185     \global \count\@currbox \@tempcnta
186     }%
187     \@fltovf
188 \fi

```

This starts the structure for the float.

```

189 \csname __tag_float_init:\endcsname
190 \UseTaggingSocket{float/begin}%
191 \global \setbox\@currbox
192     \color@vbox
193     \normalcolor
194     \vbox \bgroup
195         \hsize\columnwidth
196         \@parboxrestore
197         \@floatboxreset

```

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

```

198     \MakeLinkTarget*{\@capytype.struct.\@current@float@struct}%
199 }%

```

The end code of the float ...

```

200 \def\end@float{%
201     \endfloatbox
202     \UseTaggingSocket{float/end}%
203     \ifnum\@floatpenalty <\z@
204         \@largefloatcheck
205         \@cons\@currlist\@currbox
206         \ifnum\@floatpenalty <-\@Mii
207             \penalty -\@Miv
208             \@tempdima\prevdepth
209             \vbox{}}%
210         \prevdepth\@tempdima
211         \penalty\@floatpenalty
212     \else
213         \vadjust{\penalty -\@Miv \vbox{}}\penalty\@floatpenalty\@Esphack
214         \UseTaggingSocket{float/hmode/end}%
215     \fi
216 \fi
217 }

```

and similar for double floats:

```

218 \def\end@dblfloat{%
219     \if@twocolumn
220         \endfloatbox

```



```

221 \UseTaggingSocket{float/end}%
222 \ifnum\@floatpenalty <\z@
223   \@largefloatcheck
224   \global\dp\@currbox1sp %
225   \@cons\@currlist\@currbox
226   \ifnum\@floatpenalty <-\@Mii
227     \penalty -\@Miv
228     \@tempdima\prevdepth
229     \vbox{}%
230     \prevdepth\@tempdima
231     \penalty\@floatpenalty
232   \else
233     \vadjust{\penalty -\@Miv \vbox{\penalty\@floatpenalty}\@Esphack
234     \UseTaggingSocket{float/hmode/end}%
235   \fi
236 \fi
237 \else
238   \end@float
239 \fi
240 }%
241 \ExplSyntaxOn

```

6.6 Handling captions

To avoid that hyperref interferes we disable its patches:

```

242 \def\hyper@nopatch@caption{}

```

6.6.1 (Tagging) sockets

First some temporary sockets. These sockets are in ltagging.

```

243 \socket_if_exist:nF {taggsupport/caption/begin}
244 {
245   \NewTaggingSocket{caption/begin}{1}
246   \NewTaggingSocket{caption/end}{0}
247   \NewTaggingSocket{caption/label/begin}{0}
248   \NewTaggingSocket{caption/label/end}{0}
249 }

```

These socket are currently defined in tagpdf.

```

250 \socket_if_exist:nF {taggsupport/para/begin}
251 {
252   \NewTaggingSocket{para/begin}{0}
253   \NewTaggingSocket{para/end}{0}
254 }

```

`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.

```

255 \NewSocket{caption/label}{1}

```

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

```
256 \NewSocketPlug{caption/label}{kernel}
257 {
258   #1:~
259 }
260 \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.

```
261 \NewTaggingSocketPlug{caption/begin}{default}
262 {
263   \tl_if_empty:eTF {#1}
264   {
265     \tag_struct_begin:n{tag=Caption,firstkid}
266   }
267   {
268     \tag_struct_begin:n{tag=Caption,parent=#1,firstkid}
269   }
270   \bool_set_true:N \l__tag_para_flattened_bool
271 }
272 \AssignTaggingSocketPlug{caption/begin}{default}
```

`default (plug)`

```
273 \NewTaggingSocketPlug{caption/end}{default}
274 {
275   \tag_struct_end:
276 }
277 \AssignTaggingSocketPlug{caption/end}{default}
```

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

```
278 \NewTaggingSocketPlug{caption/label/begin}{default}
279 {
280   suppress para tagging at the begin.
281   \tagpdfparaOff
282   \tag_struct_begin:n{tag=Lbl}
283   \tag_mc_begin:n{ }
284 }
285 \AssignTaggingSocketPlug{caption/label/begin}{default}
```

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

```
286 \NewTaggingSocketPlug{caption/label/end}{default}
287 {
288   \tag_mc_end:
289   \tag_struct_end:
290   \tagpdfparaOn
291 }
292 \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`:

```

292 \def\caption{%
293   \ifx\capttype\undefined
294     \latexerror{\noexpand\caption\c_space_tl outside~float}\@ehd
295     \expandafter\@gobble
296   \else

```

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

```

297   \tl_if_empty:NTF\@current@float@struct
298   {
299     \refstepcounter\@capttype
300   }
301   {
302     \@kernel@refstepcounter\@capttype

```

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

```

303     \xdef\@currentHref{\@capttype.struct.\@current@float@struct}%
304   }
305   \expandafter\@firstofone
306 \fi
307 {\@dblarg{\@caption\@capttype}}%
308 }

```

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

```

309 \NewHookWithArguments{cmd/@makecaption/before}{2}
310 \AddToHook{begindocument}
311 {
312   \long\def\@makecaption#1#2{%
313     \UseHookWithArguments{cmd/@makecaption/before}{2}{#1}{#2}%
314     \vskip\abovecaptionskip

```

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

```

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

```

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

```

318     \UseTaggingSocket{caption/begin}{\@current@float@struct}
319     \ifdim \wd\@tempboxa >\hsize
320     \UseTaggingSocket{caption/label/begin}
321     \UseSocket{caption/label}{#1}
322     \UseTaggingSocket{caption/label/end}
323     \UseTaggingSocket{para/begin}
324     #2
325     \par
326     \else

```

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

```
327     \global \@minipagefalse
328     \hb@xt@\hsize{\hfil
329     \UseTaggingSocket{caption/label/begin}
330     \UseSocket{caption/label}{#1}
331     \UseTaggingSocket{caption/label/end}
332     \UseTaggingSocket{para/begin}
333     #2
334     \UseTaggingSocket{para/end}
335     \hfil}%
336     \fi
337     \UseTaggingSocket{caption/end}
338     \vskip\belowcaptionskip}
339 }
```

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

```
340 </package>

341 <*latex-lab>
342 \ProvidesFile{float-latex-lab-testphase.ltx}
343     [\ltxlabfloatdate\space v\ltxlabfloatversion\space latex-lab wrapper float]
344 \RequirePackage{latex-lab-testphase-float}
345 </latex-lab>
```