

# The `latex-lab-floats` package

## Tagging of floats

L<sup>A</sup>T<sub>E</sub>X Project\*

v0.81e 2024-03-23

### 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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 structure number of current float structure.

---

---

`\@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 to float 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.
```

```
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
```

*(End of definition for \g\_\_tag\_float\_sect\_prop, \g\_\_tag\_float\_types\_seq, and \@current@float@struct. 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?
```

```
10 \bool_new:N \g__tag_float_sect_bool
11 \bool_gset_true:N \g__tag_float_sect_bool
```

*(End of definition for \g\_\_tag\_float\_sect\_bool.)*

### 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.
```

```
12 \cs_new_protected:Npn\__tag_float_init_collect:
13 {
14   \bool_if:NT\g__tag_float_sect_bool
15   {
16     \seq_map_inline:Nn\g__tag_float_types_seq
17     {
18       \tag_struct_begin:n{tag=##1s,stash}
19       \prop_gput:Nne\g__tag_float_sect_prop {##1-struct}{\int_use:N\c@g__tag_struct_abs}
20       \tag_struct_end:
21     }
22   }
23 }
```

*(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.

```

24 \cs_new_protected:Npn \_tag_float_stop_sect:
25 {
26   \bool_if:NT\g__tag_float_sect_bool
27   {
28     \seq_map_inline:Nn\g__tag_float_types_seq
29     {
30       \prop_get:NnNT\g__tag_float_sect_prop{##1-used}\l__tag_tmpa_tl
31       {
32         \exp_args:Ne
33         \tag_struct_use_num:n{\prop_item:Nn\g__tag_float_sect_prop{##1-struct}}
34         \prop_gremove:Nn \g__tag_float_sect_prop{##1-used}
35       }
36     }
37   }
38 }

```

(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.

```

39 \keys_define:nm { tag / tool}
40 {
41   flush-floats .code:n =
42   {
43     \keys_set:nm {tag / tool} {sec-stop=#1}
44     \_tag_float_stop_sect:
45     \_tag_float_init_collect:
46   },
47   flush-float .default:n = Document
48 }

```

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

We need at least one pair

```

49 \AddToHook{begindocument/end}[latex-lab/float]
50 {\_tag_float_init_collect:}
51 \AddToHook{tagpdf/finish/before}[latex-lab/float]
52 {\par\_tag_sec_end:n{-10}\_tag_float_stop_sect:}
53 \DeclareHookRule{tagpdf/finish/before}{latex-lab/float}{before}{tagpdf}

```

### 6.3 Splitting floats

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

```

54 \keys_define:nm { tag / tool}
55 {
56   split-float .code:n =
57   {
58     \_tag_float_end:
59     \_tag_float_begin:
60     \MakeLinkTarget*{floatstructure.\int_use:N\c@g__tag_struct_abs_int}
61   }
62 }

```

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

## 6.4 Patching

`\\_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:`

```

63 \\cs_new_protected:Npn \\_tag_float_stop_par:
64   {
65     \\tag_mc_end:
66     \\bool_if:NF \\g__tag_float_sect_bool
67     {
68       \\tag_struct_end:
69     }
70   }
71 \\cs_new_protected:Npn \\_tag_float_start_par:
72   {
73     \\bool_if:NF \\g__tag_float_sect_bool
74     {
75       \\tag_struct_begin:n{tag=text}%
76     }
77     \\tag_mc_begin:n{tag=P}
78   }
79

```

(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.

```

80 \\cs_new_protected:Npn \\_tag_float_begin:
81   {%

```

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

```

82   \\bool_if:NTF\\g__tag_float_sect_bool
83   {
84     \\exp_args:Ne
85     \\tag_struct_begin:n{tag=float,parent=0\\prop_item:No\\g__tag_float_sect_prop{\\@capttype-str
86     \\prop_gput:Nee \\g__tag_float_sect_prop {\\@capttype-used}{true}
87   }
88   {
89     \\tag_struct_begin:n{tag=float}
90   }
91     \\tl_set:Ne\\@current@float@struct{\\tag_get:n{struct_num}}%
92     \\typeout{Float structure: \\@current@float@struct}
93   }
94
95 \\cs_new_protected:Npn \\_tag_float_end:{\\tag_struct_end:} %end Aside
96

```

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

```

97 \\ExplSyntaxOff
98 \\def\\@xfloat #1[#2]{%
99   \\@nodocument
100   \\def \\@capttype {#1}%
101   \\def \\@fps {#2}%
102   \\@onelevel@sanitize \\@fps
103   \\def \\reserved@b {!}%

```

```

104 \ifx \reserved@b \@fps
105   \@fpsaddefault
106 \else
107   \ifx \@fps \@empty
108     \@fpsaddefault
109   \fi
110 \fi
111 \ifhmode
112   \@bsphack

```

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

```

113   \@nameuse{_\tag_float_stop_par:}% <---end P
114   \@floatpenalty -\@Mii
115 \else
116   \@floatpenalty-\@Miii
117 \fi
118 \ifinner
119   \@parmoderr\@floatpenalty\z@
120 \else
121   \@next\@currbox\@freelist
122   {%
123     \@tempcnta \sixt@n
124     \expandafter \@tfor \expandafter \reserved@a
125       \expandafter : \expandafter = \@fps
126     \do
127     {%
128       \if \reserved@a h%
129         \ifodd \@tempcnta
130       \else
131         \advance \@tempcnta \@ne
132       \fi
133       \else\if \reserved@a t%
134         \@setfpsbit \tw@
135       \else\if \reserved@a b%
136         \@setfpsbit 4%
137       \else\if \reserved@a p%
138         \@setfpsbit 8%
139       \else\if \reserved@a !%
140         \ifnum \@tempcnta>15
141           \advance \@tempcnta -\sixt@n\relax
142         \fi
143       \else
144         \@latex@error{Unknown float option '\reserved@a'}%
145         {Option '\reserved@a' ignored and 'p' used.}%
146         \@setfpsbit 8%
147       \fi\fi\fi\fi\fi
148     }%
149     \@tempcntb \csname ftype@\@capytype \endcsname
150     \multiply \@tempcntb \@xxxii
151     \advance \@tempcnta \@tempcntb
152     \global \count\@currbox \@tempcnta
153   }%
154   \@fltovf
155 \fi

```

This starts the structure for the float.

```
156 \nameuse{__tag_float_begin:}%
157 \global \setbox\@currbox
158 \color@vbox
159 \normalcolor
160 \vbox \bgroup
161 \hsize\columnwidth
162 \@parboxrestore
163 \@floatboxreset
```

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

```
164 \MakeLinkTarget*{floatstructure.\number\value{g__tag_struct_abs_int}}%
165 }%
```

The end code of the float ...

```
166 \def\end@float{%
167 \endfloatbox
168 \nameuse{__tag_float_end:}%
169 \ifnum\@floatpenalty <\z@
170 \@largefloatcheck
171 \@cons\@currlist\@currbox
172 \ifnum\@floatpenalty <-\@Mii
173 \penalty -\@Miv
174 \@tempdima\prevdepth
175 \vbox{}%
176 \prevdepth\@tempdima
177 \penalty\@floatpenalty
178 \else
179 \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
180 \nameuse{__tag_float_start_par:}% restart P safe here??
181 \fi
182 \fi
183 }
```

and similar for double floats:

```
184 \def\end@dblfloat{%
185 \if@twocolumn
186 \endfloatbox
187 \nameuse{__tag_float_end:}%
188 \ifnum\@floatpenalty <\z@
189 \@largefloatcheck
190 \global\dp\@currbox1sp %
191 \@cons\@currlist\@currbox
192 \ifnum\@floatpenalty <-\@Mii
193 \penalty -\@Miv
194 \@tempdima\prevdepth
195 \vbox{}%
196 \prevdepth\@tempdima
197 \penalty\@floatpenalty
198 \else
199 \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
200 \nameuse{__tag_float_start_par:}% restart P safe here??
201 \fi
202 \fi
203 \else
```

```

204 \end@float
205 \fi
206 }%
207 \ExplSyntaxOn

```

## 6.5 Handling captions

To avoid that hyperref interferes we disable its patches:

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

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

```

209 \let\@kernel@refstepcounter\refstepcounter %as long it is not in the kernel
210 \def\caption{%
211   \ifx\@capttype\@undefined
212     \@latex@error{\noexpand\caption\c_space_tl outside~float}\@ehd
213     \expandafter\@gobble
214   \else

```

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

```

215   \tl_if_empty:NTF\@current@float@struct
216   {
217     \refstepcounter\@capttype
218   }
219   {
220     \@kernel@refstepcounter\@capttype

```

we need to reset the target for `\addcontentsline`.

```

221     \xdef\@currentHref{floatstructure.\@current@float@struct}%
222   }
223   \expandafter\@firstofone
224 \fi
225 {\@dblarg{\@caption\@capttype}}%
226 }

```

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`)

```
227 \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.

```

228 \AddToHook{begindocument}
229 {
230   \long\def\@makecaption#1#2{%
231     \vskip\abovecaptionskip

```

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

```

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

```

we stop paratagging. TODO: check

```
235     \tagtool{para=false}
```



if caption is used outside a float there is perhaps no number, then we use the parent structure and hope ...

```

236     \tl_if_empty:NT \@current@float@struct
237     { \tl_set:Ne \@current@float@struct {\tag_get:n{struct_num}} }
238     \tag_struct_begin:n{tag=Caption,parent=\@current@float@struct}

```

move the caption to the begin of the float structure:

```

239     \tag_if_active:T
240     {
241         \seq_gpop_right:cN {g__tag_struct_kids_\@current@float@struct _seq}\l__tag_tmpa_tl
242         \seq_gput_left:cV {g__tag_struct_kids_\@current@float@struct _seq}\l__tag_tmpa_tl
243     }
244     \ifdim \wd\@tempboxa >\hsize
245         \tag_struct_begin:n{tag=Lbl}
246         \tag_mc_begin:n{}
247         #1:~
248         \tag_mc_end:
249         \tag_struct_end:
250         \tag_mc_begin:n{}
251         #2\par
252         \tag_mc_end:
253     \else

```

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

```

254         \global \@minipagefalse
255         \hb@xt@\hsize{\hfil
256         \tag_struct_begin:n{tag=Lbl}
257         \tag_mc_begin:n{}
258         #1:~
259         \tag_mc_end:
260         \tag_struct_end:
261         \tag_mc_begin:n{}
262         #2
263         \tag_mc_end:\hfil}%
264     \fi
265     \tag_struct_end: %caption
266     \vskip\belowcaptionskip}
267 }

```

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

```

268 \end{package}
269 \end{*latex-lab}
270 \ProvidesFile{float-latex-lab-testphase.ltx}
271 [\l\tlabfloatdate\space v\l\tlabfloatversion\space latex-lab wrapper float]
272 \RequirePackage{latex-lab-testphase-float}
273 \end{*latex-lab}

```