

The latex-lab-graphic package

Tagging of included graphics

L^AT_EX Project*

v0.8 2023-04-07

Abstract

The following code implements a first draft for the tagging of graphics included with `\includegraphics`.

1 Introduction

The code here handle the tagging of pictures included with `\includegraphics` and the picture environment. Pictures drawn with `l3draw` or `tikz` or similar packages aren't handled yet.

Tagging of graphics included with `\includegraphics` is at a first glance trivial: They are either only decorations, in which case they should be in a **artifact** MC-chunk or (in pdf 2.0) tagged as an **Artifact** structure, or they are meaningful and then they should be tagged as a **Figure**. Such a graphic is a simple box and no other content can interfere so adding the structure commands shouldn't pose much problems.

But things are actually not so easy.

At first there are two ways to add a graphic to a structure: similar to text as a marked content item (by surrounding it with `\tagmcbegin` and `\tagmcbend`) or by referencing the XObject with an OBJR object (similar to a link annotation). Which method is more sensible (and if it actually matters) is unknown but should be tested. Currently the first method is used as the second require changes in the backend files.

At second—and this is actually a *much* larger problem—a **Figure** structure should have an attribute with an **BBox** entry. The value of a **BBox** is an array of four numbers that gives the coordinates of the left, bottom, right, and top edges of the structure element's bounding box. That is the rectangle that completely encloses its *visible* content so not necessarily the TeX bounding box: if `viewport` or `trim` is used and the graphic is not clipped, the visible content can be larger.

Getting the **BBox** is quite straightforward for a graphic that is used once as is. But graphics can be trimmed, scaled, reflected, rotated and reused in various ways. This transformations typically involve a mix of TeX commands like shifting a box or changing the bounding box and backend commands like inserting a pdfliteral with a transformation matrix and and not in all cases getting the **BBox** is possible without rewriting large parts of the graphics/x packages. Problematic are

*Initial implementation done by Ulrike Fischer

- manipulations through external box commands (`\rotatebox`, `\reflectbox`, `\scalebox`). The current implementation in the `graphics/x` packages do not pass the transformation matrix in way that allows to track the changes for the `BBox` of an included graphic: sometimes the values are set to late (after the box is already stored), and often the values are not grouped and can leak out from earlier uses of the commands.
- some combination of keys in the optional argument of `\includegraphics`. Examples are `origin` and multiple calls to `scale` and `angle`) as they internally call the box commands. Examples of failing combinations can be found in the test file `graphic-faults`.
- graphics that are stored in a box and reused: to get the `BBox` one has to set a label that stores the position with `\pdfsavepos`, and if a box is reused one gets multiply defined labels. One possible solution here is to make use of the new delayed `\pdfliteral`. It allows to change the label names in the shipout, but this requires careful tracking the box usages and so various kernel changes.

2 Restrictions and Todos

Correct tagging is currently implemented only for simple `\includegraphics` and the keys `viewport`, `trim`, `scale` and `angle` (used at most once).

Not supported

- graphics inside `\rotatebox`, `\reflectbox`, `\scalebox`.
 TODO: A new implementation with `l3graphics` and `l3box` is probably needed here.
- multiple uses of the `scale` and `angle` keys
- multiple use of graphics stored in boxes. For such graphics automated tagging should be probably deactivated when storing the content and tagging should be added around the `\usebox`. (How to proceed when content is saved in boxes needs generally more testing).

3 Additional keys

The code defines additional keys for `\includegraphics`:

tag with the values

artifact When used the graphic will be tagged as artifact. This doesn't require a `BBox` and so works also in some of the not yet supported cases described above.

false When used tagging will be stopped completely. It is then the responsibility of the surrounding code to add appropriate tagging commands.

<name> Other values will be used as tag names in the structure. If the tag is not known as a structure tag you will get an warning from `tagpdf`. The default name is currently **Figure**

actualtext This allows to add an `/ActualText` to the structure. This is useful for small graphics that represent single chars or a short word like a logo. If `actualtext` is used, the graphics is not enclosed in `Figure` structure but in a `Span` structure and no `/BBox` attribute is added. This in accordance with (the draft of) PDF/UA-2 but violates perhaps PDF/UA-1.

correct-BBox If the calculated `/BBox` values are wrong they can be correct with this key. It expects four dimensions that are added to the `/BBox` values.

debug The value `BBox` will show the calculated `/BBox` as a half transparent red rectangle.

The code also redefines the `alt` key to actually add its values as an alternative text. If no `alt` value is given, a warning is issued and the file name of the graphic is used.

```
1 <@=tag>
2 <*package>
```

4 Implementation

```
3 \ProvidesExplPackage {latex-lab-testphase-graphic} {2023-04-07} {0.8}
4 {Code related to the tagging of graphics}
```

We load `l3opacity` for the debug code

```
5 \RequirePackage{l3opacity}
```

4.1 Saving the position

We need a replacement for `zref-savepos`. This uses `l3ref` which loaded by the `pdfmanagement/tagpdf`.

TODO: this is perhaps needed in other places too and should be moved to `l3ref`!!

```
6 \ref_attribute_gset:nmn{xpos}{0}{shipout}{\int_use:N\tex_lastxpos:D}
7 \ref_attribute_gset:nmn{ypos}{0}{shipout}{\int_use:N\tex_lastypos:D}
```

`__tag_graphic_savepos:n` this is the command which stores the position. It uses two `savepos` in case `bidi` needs it (see `zref-savepos`).

```
8 \cs_new_protected:Npn\__tag_graphic_savepos:n #1
9 {
10 \tex_savepos:D
11 \ref_label:nn{#1}{xpos,ypos,abspage}
12 \tex_savepos:D
13 }
14 \cs_generate_variant:Nn \__tag_graphic_savepos:n {e}
```

(End of definition for `__tag_graphic_savepos:n`.)

4.2 Variables

`\l__tag_graphic_debug_bool` A boolean for debug code

```
15 \bool_new:N \l__tag_graphic_debug_bool
16 \keys_define:nn { document / metadata }
17 {
18 debug / BBox .code:n = { \bool_set_true:N \l__tag_graphic_debug_bool }
19 }
```

(End of definition for `\l__tag_graphic_debug_bool`.)

`\g__tag_graphic_int` This is used to get unique labels in the savepos code.

20 `\int_new:N\g__tag_graphic_int`

(End of definition for `\g__tag_graphic_int`.)

`\g__tag_graphic_lx_tl` This commands will hold the calculated BBox values. Local variables would probably work too, but global variables can be easier retrieved in tests and debugging code ...

`\g__tag_graphic_ly_tl`

`\g__tag_graphic_ux_tl` 21 `\tl_new:N \g__tag_graphic_lx_tl`

`\g__tag_graphic_uy_tl` 22 `\tl_new:N \g__tag_graphic_ly_tl`

`\bboxcorr_seq\l__tag_graphic_bboxcorr_bool` 23 `\tl_new:N \g__tag_graphic_ux_tl`

24 `\tl_new:N \g__tag_graphic_uy_tl`

25 `\seq_new:N\l__tag_graphic_bboxcorr_seq`

26 `\bool_new:N\l__tag_graphic_bboxcorr_bool`

(End of definition for `\g__tag_graphic_lx_tl` and others.)

`\l__tag_graphic_currentlabel_tl` This holds the label name of the savepos.

27 `\tl_new:N \l__tag_graphic_currentlabel_tl`

(End of definition for `\l__tag_graphic_currentlabel_tl`.)

`\l__tag_graphic_alt_tl` Variables for the alt text, the actualtext and the structure tag.

`\l__tag_graphic_alt_dflt_tl` 28 `\tl_new:N \l__tag_graphic_alt_tl`

29 `\tl_new:N \l__tag_graphic_alt_dflt_tl`

30 `\tl_set:Nn \l__tag_graphic_alt_dflt_tl {\Gin@base\Gin@ext}`

31 `\tl_new:N \l__tag_graphic_actual_tl`

32 `\tl_new:N \l__tag_graphic_struct_tl`

33 `\tl_set:Nn\l__tag_graphic_struct_tl {Figure}`

34 `\bool_new:N\l__tag_graphic_artifact_bool`

35 `\bool_new:N\l__tag_graphic_BBox_bool`

36 `\bool_set_true:N\l__tag_graphic_BBox_bool`

(End of definition for `\l__tag_graphic_alt_tl` and others.)

`\l__tag_graphic_sin_fp` A bunch of fp-variables (we don't use tl-vars, to avoid to have to take care about minus signs everywhere)

`\l__tag_graphic_cos_fp`

`\l__tag_graphic_scale_fp` 37 `\fp_new:N\l__tag_graphic_sin_fp`

`\l__tag_graphic_lxly_fp` 38 `\fp_new:N\l__tag_graphic_cos_fp`

`\l__tag_graphic_lxuy_fp` 39 `\fp_new:N\l__tag_graphic_lxly_fp`

`\l__tag_graphic_uxly_fp` 40 `\fp_new:N\l__tag_graphic_lxuy_fp`

`\l__tag_graphic_uxuy_fp` 41 `\fp_new:N\l__tag_graphic_uxly_fp`

`\l__tag_graphic_ux_fp` 42 `\fp_new:N\l__tag_graphic_uxuy_fp`

`\l__tag_graphic_ly_fp` 43 `\fp_new:N\l__tag_graphic_ux_fp`

`\l__tag_graphic_lx_fp` 44 `\fp_new:N\l__tag_graphic_ly_fp`

`\l__tag_graphic_uy_fp` 45 `\fp_new:N\l__tag_graphic_lx_fp`

`\l__tag_graphic_trim_ux_fp` 46 `\fp_new:N\l__tag_graphic_uy_fp`

`\l__tag_graphic_trim_ly_fp` this holds the scale value. Currently this only the one from `\Gin@scalex`

`\l__tag_graphic_trim_lx_fp` 47 `\fp_new:N\l__tag_graphic_scale_fp`

`\l__tag_graphic_trim_uy_fp`

the follow variables hold the four trim values (or the equivalent calculated values if viewport is used).

```

48 \fp_new:N\l__tag_graphic_trim_ux_fp
49 \fp_new:N\l__tag_graphic_trim_ly_fp
50 \fp_new:N\l__tag_graphic_trim_lx_fp
51 \fp_new:N\l__tag_graphic_trim_uy_fp

```

(End of definition for `\l__tag_graphic_sin_fp` and others.)

4.3 Tagging commands

`\Gin@tag@struct@begin` The command to start the tagging.

```

52 \msg_new:nnn {tag}{alt-text-missing}
53 {
54   Alternative-text-for-graphic-is-missing.\
55   Using~'#1'~instead
56 }
57 \cs_new_protected:Npn\Gin@tag@struct@begin
58 {
59   \tag_if_active:T
60   {
61     \tag_mc_end_push:

```

we don't open a structure for artifacts to make it easier to use graphics in saveboxes.

```

62     \bool_if:NTF\l__tag_graphic_artifact_bool
63     {
64       \tag_mc_begin:n{artifact}
65     }
66     {
67       \tl_if_empty:NTF\l__tag_graphic_actual_tl
68       {
69         \tl_if_empty:NT\l__tag_graphic_alt_tl
70         {
71           \msg_warning:nx{tag}{alt-text-missing}{\l__tag_graphic_alt_dflt_tl}
72           \tl_set:Nx\l__tag_graphic_alt_tl {\l__tag_graphic_alt_dflt_tl}
73         }
74         \tag_struct_begin:n
75         {
76           tag=\l__tag_graphic_struct_tl,
77           alt=\l__tag_graphic_alt_tl,
78         }
79       }
80       {
81         \tag_struct_begin:n
82         {
83           tag=Span,
84           actualtext=\l__tag_graphic_actual_tl,
85         }
86         \bool_set_false:N\l__tag_graphic_BBox_bool
87       }
88       \tag_mc_begin:n{}
89     }
90   }
91 }

```

(End of definition for \Gin@tag@struct@begin. This function is documented on page ??.)

\Gin@tag@struct@end

```
92 \cs_new_protected:Npn\Gin@tag@struct@end
93 {
94   \tag_if_active:T
95   {
96     \tag_mc_end:
97     \bool_if:NF\l__tag_graphic_artifact_bool
98     {
99       \tag_struct_end:
100     }
101     \tag_mc_begin_pop:n{ }
102   }
103 }
```

(End of definition for \Gin@tag@struct@end. This function is documented on page ??.)

4.4 Patching graphics commands

All changes are currently done in \Gin@setfile.

```
104 \AddToHook{package/graphics/after}
105 {
106   \def\Gin@setfile#1#2#3{%
107     \ifx\#2\\\Gread@false\fi
108     \ifGin@bbox\else
109       \ifGread@
110         \csname Gread@%
111           \expandafter\ifx\csname Gread@#1\endcsname\relax
112             eps%
113           \else
114             #1%
115           \fi
116         \endcsname{\Gin@base#2}%
117       \else
118         \Gin@nosize{#3}%
119       \fi
120     \fi
121     \Gin@viewport@code
122     \Gin@nat@height\Gin@ury bp%
123     \advance\Gin@nat@height-\Gin@lly bp%
124     \Gin@nat@width\Gin@urx bp%
125     \advance\Gin@nat@width-\Gin@llx bp%
126     \Gin@req@sizes
127     \expandafter\ifx\csname Ginclude@#1\endcsname\relax
128       \Gin@drafttrue
129       \expandafter\ifx\csname Gread@#1\endcsname\relax
130         \@latex@error{Can not include graphics of type: #1}\@ehc
131         \global\expandafter\let\csname Gread@#1\endcsname\@empty
132       \fi
133     \fi
134     \leavevmode
```

Here the tagging begins. We want to catch also the draft box, and for luatex tagging must be started before the `\setbox`.

```

135 \Gin@tag@struct@begin %NEW
136 \ifGin@draft
137     \hb@xt@\Gin@req@width{%
138         \vrule\hss
139         \vbox to \Gin@req@height{%
140             \hrule \@width \Gin@req@width
141             \vss
142             \edef\@tempa{#3}%
143             \rlap{ \ttfamily\expandafter\strip@prefix\meaning\@tempa}%
144             \vss
145             \hrule}%
146         \hss\vrule}%
147 \else
148     \@addtofilelist{#3}%
149     \ProvidesFile{#3}[Graphic file (type #1)]%
150     \setbox\z@\hbox{\csname Ginclude@#1\endcsname{#3}}%
151     \dp\z@\z@
152     \ht\z@\Gin@req@height
153     \wd\z@\Gin@req@width

```

This the main command to calculate the BBox values.

```

154 \Gin@tag@bbox@attribute %new
155 \box\z@

```

and here the tagging stops.

```

156 \Gin@tag@struct@end %new
157 \fi}
158 }

```

4.5 Additional keys for the graphics command

TODO: this is a bit temporary and will perhaps need more refinement. we also ensure that graphicx is loaded for the keyval support.

```

159 \AddToHook{package/graphicsx/after}[latex-lab]
160 {
161     \define@key{Gin}{alt}          {\t1_set:Nx\l__tag_graphic_alt_t1{\text_purify:n{#1}}}
162     \define@key{Gin}{artifact}[]
163     {
164         \bool_set_true:N \l__tag_graphic_artifact_bool
165         \bool_set_false:N \l__tag_graphic_BBox_bool
166     }
167     \define@key{Gin}{actualtext}
168     {
169         \t1_set:Nx\l__tag_graphic_actual_t1{\text_purify:n{#1}}
170         \bool_set_false:N \l__tag_graphic_BBox_bool
171     }
172     \define@key{Gin}{correct-BBox}
173     {
174         \bool_set_true:N \l__tag_graphic_bboxcorr_bool
175         \seq_set_split:Nnn\l__tag_graphic_bboxcorr_seq{~}{#1~0pt~0pt~0pt}
176     }
177     \define@key{Gin}{tag}

```

```

178     {
179       \str_case:nnF {#1}
180       {
181         {artifact}
182         {
183           \bool_set_true:N \l__tag_graphic_artifact_bool
184           \bool_set_false:N \l__tag_graphic_BBox_bool
185         }
186         {false}{\tag_stop:}
187       }
188       {\tl_set:Nn\l__tag_graphic_struct_tl{#1}}
189     }
190   }
191 \AddToHook{package/graphics/after}[latex-lab]
192   {\RequirePackage{graphicx}}

```

For picture and other environments we need a similar set of keys. TODO: redefine `\includegraphics` to make use of these here??

```

193 \keys_define:nn{tag/picture}
194   {
195     ,alt .code:n =
196       {\tl_set:Nx\l__tag_graphic_alt_tl{\text_purify:n{#1}}}
197     ,artifact .code:n =
198       {
199         \bool_set_true:N \l__tag_graphic_artifact_bool
200         \bool_set_false:N \l__tag_graphic_BBox_bool
201       }
202     ,actualtext .code:n =
203       {
204         \tl_set:Nx\l__tag_graphic_actual_tl{\text_purify:n{#1}}
205         \bool_set_false:N \l__tag_graphic_BBox_bool
206       }
207     ,correct-BBox .code:n =
208       {
209         \bool_set_true:N \l__tag_graphic_bboxcorr_bool
210         \seq_set_split:Nnn\l__tag_graphic_bboxcorr_seq{~}{#1~0pt~0pt~0pt}
211       }
212     ,tag .code:n =
213       {
214         \str_case:nnF {#1}
215         {
216           {artifact}
217           {
218             \bool_set_true:N \l__tag_graphic_artifact_bool
219             \bool_set_false:N \l__tag_graphic_BBox_bool
220           }
221           {false}{\tag_stop:}
222         }
223         {\tl_set:Nn\l__tag_graphic_struct_tl{#1}}
224       }
225   }

```


4.6 Calculating the BBox

`__tag_graphic_get_trim:` Graphics can be trimmed with the trim and the viewport key. If the graphic is not clipped the values must be taken into account when rotating. If viewport is used we have to calculate the trim.

```
226 \cs_new_protected:Npn \__tag_graphic_get_trim:
227 {
228   \legacy_if:nTF {Gin@clip}
```

Setting to 0 is not strictly needed but looks cleaner.

```
229   {
230     \fp_zero:N\l__tag_graphic_trim_lx_fp
231     \fp_zero:N\l__tag_graphic_trim_ly_fp
232     \fp_zero:N\l__tag_graphic_trim_ux_fp
233     \fp_zero:N\l__tag_graphic_trim_uy_fp
234   }
235   {
236     \fp_set:Nn \l__tag_graphic_trim_lx_fp {\l__tag_graphic_scale_fp*\Gin@vllx}
237     \fp_set:Nn \l__tag_graphic_trim_ly_fp {\l__tag_graphic_scale_fp*\Gin@vlyy}
238     \fp_set:Nn \l__tag_graphic_trim_ux_fp {\l__tag_graphic_scale_fp*\Gin@vurx}
239     \fp_set:Nn \l__tag_graphic_trim_uy_fp {\l__tag_graphic_scale_fp*\Gin@vury}
240     \cs_if_exist:NT \Gin@ollx
241     {
242       \fp_set:Nn \l__tag_graphic_trim_ux_fp {\l__tag_graphic_scale_fp* (\Gin@ourx-(\Gin@ourl)}
243       \fp_set:Nn \l__tag_graphic_trim_uy_fp {\l__tag_graphic_scale_fp* (\Gin@oury-(\Gin@ourb)}
244     }
245   }
246 }
```

(End of definition for __tag_graphic_get_trim:.)

`__tag_graphic_get_scale:`

```
247 \cs_new_protected:Npn \__tag_graphic_get_scale:
248 {
249   \fp_set:Nn \l__tag_graphic_scale_fp { \Gin@scalex }
250 }
```

(End of definition for __tag_graphic_get_scale:.)

`__tag_graphic_applyangle:nmmn`

This takes the current BBox and rotates it according to the use angle. This is the most laborious code, as we have to take also the trim values into account. We have to compare the values after the rotation to find the right corners for the BBox. Not sure, if this is the most effective code, the l3draw package has similar code to calculate a rotation, this can perhaps be reused ...

```
251 \cs_new_protected:Npn \__tag_graphic_applyangle:nmmn #1#2#3#4 %lx,ly,ux,uy
252 {
253   \bool_lazy_and:nnT
254     {\cs_if_exist_p:N \Grot@angle }
255     {\int_compare_p:nNn { \Grot@angle }={0}}
256   {
257     \fp_set:Nn \l__tag_graphic_sin_fp { sind(\Grot@angle) }
258     \fp_set:Nn \l__tag_graphic_cos_fp { cosd(\Grot@angle) }
259     \fp_set:Nn \l__tag_graphic_lx_fp {#1}
260     \fp_set:Nn \l__tag_graphic_ly_fp {#2}
```

```

261     \fp_set:Nn \l__tag_graphic_ux_fp {#3}
262     \fp_set:Nn \l__tag_graphic_uy_fp {#4}
get the x coordinates (cos,-sin)
263     \fp_set:Nn \l__tag_graphic_lxly_fp
264     {
265         -\l__tag_graphic_trim_lx_fp * \l__tag_graphic_cos_fp
266         +\l__tag_graphic_trim_ly_fp * \l__tag_graphic_sin_fp
267     }
268     \fp_set:Nn \l__tag_graphic_lxuy_fp
269     {
270         (-\l__tag_graphic_trim_lx_fp) * \l__tag_graphic_cos_fp
271         +
272         (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
273         * (-\l__tag_graphic_sin_fp)
274     }
275     \fp_set:Nn \l__tag_graphic_uxly_fp
276     {
277         (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
278         * \l__tag_graphic_cos_fp
279         +
280         (\l__tag_graphic_trim_ly_fp) * (\l__tag_graphic_sin_fp)
281     }
282     \fp_set:Nn \l__tag_graphic_uxuy_fp
283     {
284         (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
285         * \l__tag_graphic_cos_fp
286         +
287         (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
288         * (-\l__tag_graphic_sin_fp)
289     }
290     \tl_gset:Nx \g__tag_graphic_lx_tl
291     {
292         \fp_eval:n
293         {
294             min
295             (
296                 \l__tag_graphic_lxly_fp,
297                 \l__tag_graphic_lxuy_fp,
298                 \l__tag_graphic_uxly_fp,
299                 \l__tag_graphic_uxuy_fp,
300             )
301             +\l__tag_graphic_lx_fp
302             +\l__tag_graphic_trim_lx_fp
303         }
304     }
305     \tl_gset:Nx \g__tag_graphic_ux_tl
306     {
307         \fp_eval:n
308         {
309             max
310             (
311                 \l__tag_graphic_lxly_fp,
312                 \l__tag_graphic_lxuy_fp,
313                 \l__tag_graphic_uxly_fp,

```

```

314         \l__tag_graphic_uxuy_fp
315     )
316     +\l__tag_graphic_lx_fp
317     +\l__tag_graphic_trim_lx_fp
318 }
319 }

```

get the y coordinates (sin,cos)

```

320 \fp_set:Nn\l__tag_graphic_lxly_fp
321 {
322     -\l__tag_graphic_trim_lx_fp * \l__tag_graphic_sin_fp
323     -\l__tag_graphic_trim_ly_fp * \l__tag_graphic_cos_fp
324 }
325 \fp_set:Nn\l__tag_graphic_lxuy_fp
326 {
327     - \l__tag_graphic_trim_lx_fp * \l__tag_graphic_sin_fp
328     +
329     (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
330     * \l__tag_graphic_cos_fp
331 }
332 \fp_set:Nn\l__tag_graphic_uxly_fp
333 {
334     (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
335     * \l__tag_graphic_sin_fp
336     - \l__tag_graphic_trim_ly_fp * \l__tag_graphic_cos_fp
337 }
338 \fp_set:Nn\l__tag_graphic_uxuy_fp
339 {
340     (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
341     * \l__tag_graphic_sin_fp
342     +
343     (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
344     * \l__tag_graphic_cos_fp
345 }
346 \tl_gset:Nx\g__tag_graphic_ly_tl
347 {
348     \fp_eval:n
349     {
350         min
351         (
352             \l__tag_graphic_lxly_fp,
353             \l__tag_graphic_lxuy_fp,
354             \l__tag_graphic_uxly_fp,
355             \l__tag_graphic_uxuy_fp
356         )
357         + \l__tag_graphic_ly_fp + \l__tag_graphic_trim_ly_fp
358     }
359 }
360 \tl_gset:Nx\g__tag_graphic_uy_tl
361 {
362     \fp_eval:n
363     {
364         max
365         (
366             \l__tag_graphic_lxly_fp,

```

```

367         \l__tag_graphic_lxuy_fp,
368         \l__tag_graphic_uxly_fp,
369         \l__tag_graphic_uxuy_fp,
370     )
371     + \l__tag_graphic_ly_fp + \l__tag_graphic_trim_ly_fp
372 }
373 }
374 }
375 }
376 \cs_generate_variant:Nn\__tag_graphic_applyangle:nnnn {VVVV}

```

(End of definition for __tag_graphic_applyangle:nnnn.)

__tag_graphic_applycorr:NNNN This command is used to add at the end the correction values. Quite dump ...

```

377 \cs_new_protected:Npn \__tag_graphic_applycorr:NNNN #1 #2 #3 #4
378 {
379   \bool_if:NT\l__tag_graphic_bboxcorr_bool
380   {
381     \tl_set:Nx #1
382     {
383       \fp_eval:n
384       {
385         #1
386         +
387         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {1} }
388       }
389     }
390     \tl_set:Nx #2
391     {
392       \fp_eval:n
393       {
394         #2
395         +
396         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {2} }
397       }
398     }
399     \tl_set:Nx #3
400     {
401       \fp_eval:n
402       {
403         #3
404         +
405         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {3} }
406       }
407     }
408     \tl_set:Nx #4
409     {
410       \fp_eval:n
411       {
412         #4
413         +
414         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {4} }
415       }
416     }

```

```

417     }
418 }

```

(End of definition for `__tag_graphic_applycorr:NNNN`.)

`\Gin@tag@bbox@attribute` This is the main command to calculate and set the Bbox attribute

```

419 \cs_new_protected:Npn \Gin@tag@bbox@attribute
420 {

```

the attribute is only needed if tagging is active and there is not artifact.

```

421   \bool_lazy_all:nT
422   {
423     {\tag_if_active_p:}
424     {\!\__tag_graphic_artifact_bool}
425     {\!\__tag_graphic_BBox_bool}
426   }
427   {
428     \fp_set:Nn \l__tag_graphic_scale_fp { \Gin@scalex }
429     \__tag_graphic_get_trim:
430     \int_gincr:N\g__tag_graphic_int
431     \tl_set:Nx\l__tag_graphic_currentlabel_tl {\__tag_graphic\_int_use:N \g__tag_graphic_int}
432     \__tag_graphic_savepos:e { \l__tag_graphic_currentlabel_tl }
433     \tl_gset:Nx\g__tag_graphic_lx_tl
434     {
435       \dim_to_decimal_in_bp:n
436       { \__tag_ref_value:enn {\l__tag_graphic_currentlabel_tl}{xpos}{0}sp }
437     }
438     \tl_gset:Nx\g__tag_graphic_ly_tl
439     {
440       \dim_to_decimal_in_bp:n
441       { \__tag_ref_value:enn {\l__tag_graphic_currentlabel_tl}{ypos}{0}sp }
442     }
443     \tl_gset:Nx\g__tag_graphic_ux_tl
444     {
445       \fp_eval:n
446       {
447         \g__tag_graphic_lx_tl
448         +
449         \dim_to_decimal_in_bp:n { \Gin@req@width }
450       }
451     }
452     \tl_gset:Nx\g__tag_graphic_uy_tl
453     {
454       \fp_eval:n
455       {
456         \g__tag_graphic_ly_tl
457         +
458         \dim_to_decimal_in_bp:n { \Gin@req@height }
459       }
460     }

```

If the graphics is not clipped we must add the trim values.

```

461     \legacy_if:nF {Gin@clip}
462     {
463       \tl_gset:Nx\g__tag_graphic_ux_tl

```

```

464     {
465         \fp_eval:n
466         {
467             \g__tag_graphic_ux_tl
468             +
469             \l__tag_graphic_trim_ux_fp
470         }
471     }
472     \tl_gset:Nx\g__tag_graphic_lx_tl
473     {
474         \fp_eval:n
475         {
476             \g__tag_graphic_lx_tl
477             -
478             \l__tag_graphic_trim_lx_fp
479         }
480     }
481     \tl_gset:Nx\g__tag_graphic_uy_tl
482     {
483         \fp_eval:n
484         {
485             \g__tag_graphic_uy_tl
486             +
487             \l__tag_graphic_trim_uy_fp
488         }
489     }
490     \tl_gset:Nx\g__tag_graphic_ly_tl
491     {
492         \fp_eval:n
493         {
494             \g__tag_graphic_ly_tl
495             -
496             \l__tag_graphic_trim_ly_fp
497         }
498     }
499 }

```

If there is an angle we now rotate the values.

```

500     \__tag_graphic_applyangle:VVVV
501     \g__tag_graphic_lx_tl
502     \g__tag_graphic_ly_tl
503     \g__tag_graphic_ux_tl
504     \g__tag_graphic_uy_tl

```

At last we have to add the correction values

```

505     \__tag_graphic_applycorr:NNNN
506     \g__tag_graphic_lx_tl
507     \g__tag_graphic_ly_tl
508     \g__tag_graphic_ux_tl
509     \g__tag_graphic_uy_tl
510     \bool_if:NT\l__tag_graphic_debug_bool
511     {
512         \__tag_graphic_show_bbox:VVVVne
513         \g__tag_graphic_lx_tl

```

```

514     \g__tag_graphic_ly_tl
515     \g__tag_graphic_ux_tl
516     \g__tag_graphic_uy_tl
517     {red}
518     {\int_use:N\g__tag_graphic_int}
519 }

```

Now we add the attribute. We do it manually as it had to be delayed until now. The structure and the mc must be open earlier, before the `\setbox` (at least for luatex it has to). TODO: think about interface if more attributes are needed.

```

520     \__tag_prop_gput:cnx
521     { g__tag_struct_\int_eval:n {\c@g__tag_struct_abs_int}_prop }
522     { A }
523     {
524         <<
525         /O /Layout /BBox~
526         [
527             \g__tag_graphic_lx_tl\c_space_tl
528             \g__tag_graphic_ly_tl\c_space_tl
529             \g__tag_graphic_ux_tl\c_space_tl
530             \g__tag_graphic_uy_tl
531         ]
532         >>
533     }
534 }
535 }

```

(End of definition for `\Gin@tag@bbox@attribute`. This function is documented on page ??.)

4.7 Support for the picture environment

`\picture@tag@bbox@attribute` Picture needs a similar command to calculate the bbox. But here we stay simple and use simply the size of the picbox.

```

536 \newcommand\picture@tag@bbox@attribute
537 {
538     \bool_lazy_all:nT
539     {
540         {\tag_if_active_p:}
541         {\!\l__tag_graphic_artifact_bool}
542         {\!\l__tag_graphic_BBox_bool}
543     }
544     {
545         \int_gincr:N\g__tag_graphic_int
546         \tl_set:Nx\l__tag_graphic_currentlabel_tl {\__tag_graphic_\int_use:N \g__tag_graphic_int}
547         \__tag_graphic_savepos:e { \l__tag_graphic_currentlabel_tl }
548         \tl_gset:Nx \g__tag_graphic_lx_tl
549         {
550             \dim_to_decimal_in_bp:n
551             { \__tag_ref_value:enn {\l__tag_graphic_currentlabel_tl}{xpos}{0}sp }
552         }
553         \tl_gset:Nx \g__tag_graphic_ly_tl
554         {
555             \dim_to_decimal_in_bp:n
556             { \__tag_ref_value:enn {\l__tag_graphic_currentlabel_tl}{ypos}{0}sp - \dp\@picbox }

```

```

557     }
558     \tl_gset:Nx \g__tag_graphic_ux_tl
559     {
560         \dim_to_decimal_in_bp:n
561         {
562             \g__tag_graphic_lx_tl bp + \wd\@picbox
563         }
564     }
565     \tl_gset:Nx \g__tag_graphic_uy_tl
566     {
567         \dim_to_decimal_in_bp:n
568         {
569             \g__tag_graphic_ly_tl bp + \ht\@picbox + \dp\@picbox
570         }
571     }
572     \__tag_graphic_applycorr:NNNN
573         \g__tag_graphic_lx_tl
574         \g__tag_graphic_ly_tl
575         \g__tag_graphic_ux_tl
576         \g__tag_graphic_uy_tl
577     \bool_if:NT\l__tag_graphic_debug_bool
578     {
579         \__tag_graphic_show_bbox:VVVVe
580         \g__tag_graphic_lx_tl
581         \g__tag_graphic_ly_tl
582         \g__tag_graphic_ux_tl
583         \g__tag_graphic_uy_tl
584         {red}
585         {\int_use:N\g__tag_graphic_int}
586     }
587     \__tag_prop_gput:cnx
588     { g__tag_struct_\int_eval:n {\c@g__tag_struct_abs_int}_prop }
589     { A }
590     {
591         <<
592         /O /Layout /BBox~
593         [
594             \g__tag_graphic_lx_tl\c_space_tl
595             \g__tag_graphic_ly_tl\c_space_tl
596             \g__tag_graphic_ux_tl\c_space_tl
597             \g__tag_graphic_uy_tl
598         ]
599         >>
600     }
601 }
602 }
603

```

(End of definition for `\picture@tag@bbox@attribute`. This function is documented on page ??.)

We redefine `\picture` to accept an optional argument and change the default alt text. We also ensure that we are in hmode, so that stopping tagging doesn't confuse the paratags.

```

604 \RenewDocumentCommand\picture{0{}m}
605 {

```



```

606 \leavevmode
607 \keys_set:nn{tag/picture}{#1} %
608 \tl_set:Nn\l__tag_graphic_alt_dflt_tl {picture-environment}
609 \pictur@#2
610 }

```

inside the picture box we stop tagging.

```

611 \def\@picture(#1,#2)(#3,#4){%
612 \@defaultunitsset\@picht{#2}\unitlength
613 \@defaultunitsset\@tempdimc{#1}\unitlength
614 \Gin@tag@struct@begin
615 \setbox\@picbox\hb@xt@\@tempdimc\bgroup
616 \tag_stop: %do not tag inside the picture box
617 \@defaultunitsset\@tempdimc{#3}\unitlength
618 \hskip -\@tempdimc
619 \@defaultunitsset\@tempdimc{#4}\unitlength
620 \lower\@tempdimc\hbox\bgroup
621 \ignorespaces}
622 \def\endpicture{%
623 \egroup\hss\egroup
624 \ht\@picbox\@picht\dp\@picbox\z@
625 \picture@tag@bbox@attribute
626 \mbox{\box\@picbox}
627 \Gin@tag@struct@end}

```

4.8 Debugging code

`_tag_graphic_show_bbox:nnnnnn`

```

628 \cs_new_protected:Npn \_tag_graphic_show_bbox:nnnnnn #1#2#3#4#5#6#%#5 color, #6 graphic
629 {
630 \iow_log:n {tag/graphic~debug:-BBox-of~graphics~#6-is~#1~#2~#3~#4}
631 \hook_gput_code:nnn
632 {shipout/foreground}
633 {tag/graphic}
634 {
635 \int_compare:nNnT
636 {\g_shipout_readonly_int}
637 =
638 {\_tag_ref_value:enn{\_tag_graphic_#6}{abspage}{0}}
639 {
640 \put
641 (#1 bp,\dim_eval:n{-\paperheight + \dim_eval:n{#2 bp}})
642 {
643 \opacity_select:n{0.5}\color_select:n{#5}
644 \rule
645 {\dim_eval:n {#3 bp-\dim_eval:n{#1 bp}}}
646 {\dim_eval:n {#4 bp-\dim_eval:n{#2 bp}}}
647 }
648 }
649 }
650 }
651 \cs_generate_variant:Nn \_tag_graphic_show_bbox:nnnnnn {VVVVne}

```

```
(End of definition for \_tag\_graphic\_show\_bbox:nnnnn.)  
652 </package>  
653 <*latex-lab>  
654 \ProvidesFile{graphic-latex-lab-testphase.ltx}  
655 [2023-04-07 v0.8 code related to the tagging included graphics]  
656 \RequirePackage{latex-lab-testphase-graphic}  
657 </latex-lab>
```