

The latex-lab-graphic package

Tagging of included graphics

L^AT_EX Project*

v0.80b 2023-10-13

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} {\ltxlabgraphicdate} {\ltxlabgraphicversion}
4 {Code related to the tagging of graphics}
```

We load `l3opacity` for the debug code

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

Needed during switch to e-type:

```
6 \cs_generate_variant:Nn \__tag_prop_gput:Nnn {cne}
```

`__tag_graphic_savepos:n` this is the command which stores the position. Similar to `zref-savepos` it uses two `savepos` commands for the case that `bidi` changes the processing order.

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

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

4.1 Variables

`\l__tag_graphic_debug_bool` A boolean for debug code

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

(End of definition for `\l__tag_graphic_debug_bool`.)

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

```
19 \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` 20 `\tl_new:N \g__tag_graphic_lx_tl`

`\g__tag_graphic_uy_tl` 21 `\tl_new:N \g__tag_graphic_ly_tl`

`\l__tag_graphic_bboxcorr_bool` 22 `\tl_new:N \g__tag_graphic_ux_tl`

23 `\tl_new:N \g__tag_graphic_uy_tl`

24 `\seq_new:N \l__tag_graphic_bboxcorr_seq`

25 `\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.

26 `\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` 27 `\tl_new:N \l__tag_graphic_alt_tl`

`\l__tag_graphic_actual_tl` 28 `\tl_new:N \l__tag_graphic_alt_dflt_tl`

`\l__tag_graphic_struct_tl` 29 `\tl_set:Nn \l__tag_graphic_alt_dflt_tl {\Gin@base\Gin@ext}`

`\l__tag_graphic_artifact_bool` 30 `\tl_new:N \l__tag_graphic_actual_tl`

`\l__tag_graphic_BBox_bool` 31 `\tl_new:N \l__tag_graphic_struct_tl`

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

33 `\bool_new:N \l__tag_graphic_artifact_bool`

34 `\bool_new:N \l__tag_graphic_BBox_bool`

35 `\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` 36 `\fp_new:N \l__tag_graphic_sin_fp`

`\l__tag_graphic_lxly_fp` 37 `\fp_new:N \l__tag_graphic_cos_fp`

`\l__tag_graphic_lxuy_fp` 38 `\fp_new:N \l__tag_graphic_lxly_fp`

`\l__tag_graphic_uxly_fp` 39 `\fp_new:N \l__tag_graphic_lxuy_fp`

`\l__tag_graphic_uxuy_fp` 40 `\fp_new:N \l__tag_graphic_uxly_fp`

`\l__tag_graphic_ux_fp` 41 `\fp_new:N \l__tag_graphic_uxuy_fp`

`\l__tag_graphic_ly_fp` 42 `\fp_new:N \l__tag_graphic_ux_fp`

`\l__tag_graphic_lx_fp` 43 `\fp_new:N \l__tag_graphic_ly_fp`

`\l__tag_graphic_uy_fp` 44 `\fp_new:N \l__tag_graphic_lx_fp`

45 `\fp_new:N \l__tag_graphic_uy_fp`

`\l__tag_graphic_trim_ux_fp` this holds the scale value. Either `\Gin@scalex` or (if that is !) `\Gin@scaley`

`\l__tag_graphic_trim_ly_fp` 46 `\fp_new:N \l__tag_graphic_scale_fp`

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

`\l__tag_graphic_trim_uy_fp` 47 `\fp_new:N \l__tag_graphic_trim_ux_fp`

48 `\fp_new:N \l__tag_graphic_trim_ly_fp`

49 `\fp_new:N \l__tag_graphic_trim_lx_fp`

50 `\fp_new:N \l__tag_graphic_trim_uy_fp`

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

4.2 Tagging commands

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

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

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

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

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

`\Gin@tag@struct@end`

```
91 \cs_new_protected:Npn\Gin@tag@struct@end
92 {
93   \tag_if_active:T
94   {
95     \tag_mc_end:
96     \bool_if:NF\l__tag_graphic_artifact_bool
```

```

97     {
98       \tag_struct_end:
99     }
100   \tag_mc_begin_pop:n{
101 }
102 }

```

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

4.3 Patching graphics commands

All changes are currently done in \Gin@setfile.

```

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

```

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

```

134   \Gin@tag@struct@begin %NEW
135   \ifGin@draft
136     \hb@xt@\Gin@req@width{%
137       \vrule\hss
138       \vbox to \Gin@req@height{%
139         \hrule \@width \Gin@req@width
140         \vss
141       \edef\@tempa{#3}%

```

```

142         \rlap{ \ttfamily\expandafter\strip@prefix\meaning\@tempa}%
143         \vss
144         \hrule}%
145     \hss\vrule}%
146 \else
147     \@addtofilelist{#3}%
148     \ProvidesFile{#3}[Graphic file (type #1)]%
149     \setbox\z@\hbox{\csname Ginclude@#1\endcsname{#3}}%
150     \dp\z@\z@
151     \ht\z@\Gin@req@height
152     \wd\z@\Gin@req@width

```

This the main command to calculate the BBox values.

```

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

```

and here the tagging stops.

```

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

```

4.4 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.

```

158 \AddToHook{package/graphicx/after}[latex-lab]
159 {
160     \define@key{Gin}{alt}          {\t1_set:N\l__tag_graphic_alt_tl{\text_purify:n{#1}}}
161     \define@key{Gin}{artifact} []
162     {
163         \bool_set_true:N \l__tag_graphic_artifact_bool
164         \bool_set_false:N \l__tag_graphic_BBox_bool
165     }
166     \define@key{Gin}{actualtext}
167     {
168         \t1_set:N\l__tag_graphic_actual_tl{\text_purify:n{#1}}
169         \bool_set_false:N \l__tag_graphic_BBox_bool
170     }
171     \define@key{Gin}{correct-BBox}
172     {
173         \bool_set_true:N \l__tag_graphic_bboxcorr_bool
174         \seq_set_split:Nnn\l__tag_graphic_bboxcorr_seq{~}{#1~Opt~Opt~Opt~Opt}
175     }
176     \define@key{Gin}{tag}
177     {
178         \str_case:nnF {#1}
179         {
180             {artifact}
181             {
182                 \bool_set_true:N \l__tag_graphic_artifact_bool
183                 \bool_set_false:N \l__tag_graphic_BBox_bool
184             }
185             {false}{\tag_stop:}
186         }

```

```

187         {\tl_set:Nn\l__tag_graphic_struct_tl{#1}}
188     }
189 }
190 \AddToHook{package/graphics/after}[latex-lab]
191 {\RequirePackage{graphicx}}

```

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

```

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

```

4.5 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.

```

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

```

Setting to 0 is not strictly needed but looks cleaner.

```

228     {
229         \fp_zero:N\l__tag_graphic_trim_lx_fp

```



```

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

```

(End of definition for __tag_graphic_get_trim:.)

__tag_graphic_get_scale:

```

246 \cs_new_protected:Npn \__tag_graphic_get_scale:
247 {
248 \fp_set:Nn \l__tag_graphic_scale_fp
249 {
250 \str_if_eq:eeTF {\Gin@scalex} { ! }
251 { \Gin@scaley }
252 { \Gin@scalex }
253 }
254 }

```

(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 ...

```

255 \cs_new_protected:Npn \__tag_graphic_applyangle:nmmn #1#2#3#4 %lx,ly,ux,uy
256 {
257 \bool_lazy_and:nmT
258 {\cs_if_exist_p:N \Grot@angle }
259 {!\int_compare_p:nNn { \Grot@angle }={0}}
260 {
261 \fp_set:Nn \l__tag_graphic_sin_fp { sind(\Grot@angle) }
262 \fp_set:Nn \l__tag_graphic_cos_fp { cosd(\Grot@angle) }
263 \fp_set:Nn \l__tag_graphic_lx_fp {#1}
264 \fp_set:Nn \l__tag_graphic_ly_fp {#2}
265 \fp_set:Nn \l__tag_graphic_ux_fp {#3}
266 \fp_set:Nn \l__tag_graphic_uy_fp {#4}

```

get the x coordinates (cos,-sin)

```

267 \fp_set:Nn\l__tag_graphic_lxly_fp
268 {
269 -\l__tag_graphic_trim_lx_fp * \l__tag_graphic_cos_fp

```

```

270     +\l__tag_graphic_trim_ly_fp * \l__tag_graphic_sin_fp
271   }
272 \fp_set:Nn\l__tag_graphic_lxuy_fp
273 {
274   (-\l__tag_graphic_trim_lx_fp) * \l__tag_graphic_cos_fp
275   +
276   (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
277   * (-\l__tag_graphic_sin_fp)
278 }
279 \fp_set:Nn\l__tag_graphic_uxly_fp
280 {
281   (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
282   * \l__tag_graphic_cos_fp
283   +
284   (\l__tag_graphic_trim_ly_fp) * (\l__tag_graphic_sin_fp)
285 }
286 \fp_set:Nn\l__tag_graphic_uxuy_fp
287 {
288   (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
289   * \l__tag_graphic_cos_fp
290   +
291   (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
292   * (-\l__tag_graphic_sin_fp)
293 }
294 \tl_gset:Ne\g__tag_graphic_lx_tl
295 {
296   \fp_eval:n
297   {
298     min
299     (
300       \l__tag_graphic_lxly_fp,
301       \l__tag_graphic_lxuy_fp,
302       \l__tag_graphic_uxly_fp,
303       \l__tag_graphic_uxuy_fp,
304     )
305     +\l__tag_graphic_lx_fp
306     +\l__tag_graphic_trim_lx_fp
307   }
308 }
309 \tl_gset:Ne\g__tag_graphic_ux_tl
310 {
311   \fp_eval:n
312   {
313     max
314     (
315       \l__tag_graphic_lxly_fp,
316       \l__tag_graphic_lxuy_fp,
317       \l__tag_graphic_uxly_fp,
318       \l__tag_graphic_uxuy_fp
319     )
320     +\l__tag_graphic_lx_fp
321     +\l__tag_graphic_trim_lx_fp
322   }
323 }

```

get the y coordinates (sin,cos)

```
324 \fp_set:Nn\l__tag_graphic_lxly_fp
325 {
326   -\l__tag_graphic_trim_lx_fp * \l__tag_graphic_sin_fp
327   -\l__tag_graphic_trim_ly_fp * \l__tag_graphic_cos_fp
328 }
329 \fp_set:Nn\l__tag_graphic_lxuy_fp
330 {
331   - \l__tag_graphic_trim_lx_fp * \l__tag_graphic_sin_fp
332   +
333   (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
334   * \l__tag_graphic_cos_fp
335 }
336 \fp_set:Nn\l__tag_graphic_uxly_fp
337 {
338   (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
339   * \l__tag_graphic_sin_fp
340   - \l__tag_graphic_trim_ly_fp * \l__tag_graphic_cos_fp
341 }
342 \fp_set:Nn\l__tag_graphic_uxuy_fp
343 {
344   (\l__tag_graphic_ux_fp-\l__tag_graphic_lx_fp-\l__tag_graphic_trim_lx_fp)
345   * \l__tag_graphic_sin_fp
346   +
347   (\l__tag_graphic_uy_fp-\l__tag_graphic_ly_fp-\l__tag_graphic_trim_ly_fp)
348   * \l__tag_graphic_cos_fp
349 }
350 \tl_gset:Ne\g__tag_graphic_ly_tl
351 {
352   \fp_eval:n
353   {
354     min
355     (
356       \l__tag_graphic_lxly_fp,
357       \l__tag_graphic_lxuy_fp,
358       \l__tag_graphic_uxly_fp,
359       \l__tag_graphic_uxuy_fp
360     )
361     + \l__tag_graphic_ly_fp + \l__tag_graphic_trim_ly_fp
362   }
363 }
364 \tl_gset:Ne\g__tag_graphic_uy_tl
365 {
366   \fp_eval:n
367   {
368     max
369     (
370       \l__tag_graphic_lxly_fp,
371       \l__tag_graphic_lxuy_fp,
372       \l__tag_graphic_uxly_fp,
373       \l__tag_graphic_uxuy_fp,
374     )
375     + \l__tag_graphic_ly_fp + \l__tag_graphic_trim_ly_fp
376   }
}
```

```

377     }
378   }
379 }
380 \cs_generate_variant:Nn\__tag_graphic_applyangle:nmmm {VVVV}

```

(End of definition for `__tag_graphic_applyangle:nmmm`.)

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

```

381 \cs_new_protected:Npn \__tag_graphic_applycorr:NNNN #1 #2 #3 #4
382 {
383   \bool_if:NT\l__tag_graphic_bboxcorr_bool
384   {
385     \tl_set:Ne #1
386     {
387       \fp_eval:n
388       {
389         #1
390         +
391         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {1} }
392       }
393     }
394     \tl_set:Ne #2
395     {
396       \fp_eval:n
397       {
398         #2
399         +
400         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {2} }
401       }
402     }
403     \tl_set:Ne #3
404     {
405       \fp_eval:n
406       {
407         #3
408         +
409         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {3} }
410       }
411     }
412     \tl_set:Ne #4
413     {
414       \fp_eval:n
415       {
416         #4
417         +
418         \dim_to_decimal_in_bp:n {\seq_item:Nn \l__tag_graphic_bboxcorr_seq {4} }
419       }
420     }
421   }
422 }

```

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

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

```

423 \cs_new_protected:Npn \Gin@tag@bbox@attribute
424 {

```

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

```

425 \bool_lazy_all:nT
426 {
427   {\tag_if_active_p:}
428   {\!l__tag_graphic_artifact_bool}
429   {\l__tag_graphic_BBox_bool}
430 }
431 {
432   \__tag_graphic_get_scale:
433   \__tag_graphic_get_trim:
434   \int_gincr:N\g__tag_graphic_int
435   \tl_set:N\l__tag_graphic_currentlabel_tl {\__tag_graphic\_int_use:N \g__tag_graphic_int}
436   \__tag_graphic_savepos:e { \l__tag_graphic_currentlabel_tl }
437   \tl_gset:N\g__tag_graphic_lx_tl
438   {
439     \dim_to_decimal_in_bp:n
440     { \property_ref:een {\l__tag_graphic_currentlabel_tl}{xpos}{0}sp }
441   }
442   \tl_gset:N\g__tag_graphic_ly_tl
443   {
444     \dim_to_decimal_in_bp:n
445     { \property_ref:een {\l__tag_graphic_currentlabel_tl}{ypos}{0}sp }
446   }
447   \tl_gset:N\g__tag_graphic_ux_tl
448   {
449     \fp_eval:n
450     {
451       \g__tag_graphic_lx_tl
452       +
453       \dim_to_decimal_in_bp:n { \Gin@req@width }
454     }
455   }
456   \tl_gset:N\g__tag_graphic_uy_tl
457   {
458     \fp_eval:n
459     {
460       \g__tag_graphic_ly_tl
461       +
462       \dim_to_decimal_in_bp:n { \Gin@req@height }
463     }
464   }

```

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

```

465   \legacy_if:nF {\Gin@clip}
466   {
467     \tl_gset:N\g__tag_graphic_ux_tl
468     {
469       \fp_eval:n
470       {
471         \g__tag_graphic_ux_tl
472         +
473         \l__tag_graphic_trim_ux_fp

```

```

474         }
475     }
476     \tl_gset:Ne\g__tag_graphic_lx_tl
477     {
478         \fp_eval:n
479         {
480             \g__tag_graphic_lx_tl
481             -
482             \l__tag_graphic_trim_lx_fp
483         }
484     }
485     \tl_gset:Ne\g__tag_graphic_uy_tl
486     {
487         \fp_eval:n
488         {
489             \g__tag_graphic_uy_tl
490             +
491             \l__tag_graphic_trim_uy_fp
492         }
493     }
494     \tl_gset:Ne\g__tag_graphic_ly_tl
495     {
496         \fp_eval:n
497         {
498             \g__tag_graphic_ly_tl
499             -
500             \l__tag_graphic_trim_ly_fp
501         }
502     }
503 }

```

If there is an angle we now rotate the values.

```

504     \__tag_graphic_applyangle:VVVV
505     \g__tag_graphic_lx_tl
506     \g__tag_graphic_ly_tl
507     \g__tag_graphic_ux_tl
508     \g__tag_graphic_uy_tl

```

At last we have to add the correction values

```

509     \__tag_graphic_applycorr:NNNN
510     \g__tag_graphic_lx_tl
511     \g__tag_graphic_ly_tl
512     \g__tag_graphic_ux_tl
513     \g__tag_graphic_uy_tl

514     \bool_if:NT\l__tag_graphic_debug_bool
515     {
516         \__tag_graphic_show_bbox:VVVVne
517         \g__tag_graphic_lx_tl
518         \g__tag_graphic_ly_tl
519         \g__tag_graphic_ux_tl
520         \g__tag_graphic_uy_tl
521         {red}
522         {\int_use:N\g__tag_graphic_int}
523     }

```

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.

```

524     \__tag_prop_gput:cne
525     { g__tag_struct_\int_eval:n {\c@g__tag_struct_abs_int}_prop }
526     { A }
527     {
528         <<
529         /O /Layout /BBox~
530         [
531             \g__tag_graphic_lx_tl\c_space_tl
532             \g__tag_graphic_ly_tl\c_space_tl
533             \g__tag_graphic_ux_tl\c_space_tl
534             \g__tag_graphic_uy_tl
535         ]
536         >>
537     }
538 }
539 }

```

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

4.6 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.

```

540 \newcommand\picture@tag@bbox@attribute
541 {
542     \bool_lazy_all:nT
543     {
544         {\tag_if_active_p:}
545         {\!\__tag_graphic_artifact_bool}
546         {\!\__tag_graphic_BBox_bool}
547     }
548     {
549         \int_gincr:N\g__tag_graphic_int
550         \tl_set:Ne\l__tag_graphic_currentlabel_tl {\__tag_graphic_\int_use:N \g__tag_graphic_int}
551         \__tag_graphic_savepos:e { \l__tag_graphic_currentlabel_tl }
552         \tl_gset:Ne \g__tag_graphic_lx_tl
553         {
554             \dim_to_decimal_in_bp:n
555             { \property_ref:een {\l__tag_graphic_currentlabel_tl}{xpos}{0}sp }
556         }
557         \tl_gset:Ne \g__tag_graphic_ly_tl
558         {
559             \dim_to_decimal_in_bp:n
560             { \property_ref:een {\l__tag_graphic_currentlabel_tl}{ypos}{0}sp - \dp\@picbox }
561         }
562         \tl_gset:Ne \g__tag_graphic_ux_tl
563         {
564             \dim_to_decimal_in_bp:n
565             {
566                 \g__tag_graphic_lx_tl bp + \wd\@picbox

```

```

567     }
568   }
569   \tl_gset:Ne \g__tag_graphic_uy_tl
570   {
571     \dim_to_decimal_in_bp:n
572     {
573       \g__tag_graphic_ly_tl bp + \ht\@picbox + \dp\@picbox
574     }
575   }
576   \__tag_graphic_applycorr:NNNN
577   \g__tag_graphic_lx_tl
578   \g__tag_graphic_ly_tl
579   \g__tag_graphic_ux_tl
580   \g__tag_graphic_uy_tl
581   \bool_if:NT\l__tag_graphic_debug_bool
582   {
583     \__tag_graphic_show_bbox:VVVVne
584     \g__tag_graphic_lx_tl
585     \g__tag_graphic_ly_tl
586     \g__tag_graphic_ux_tl
587     \g__tag_graphic_uy_tl
588     {red}
589     {\int_use:N\g__tag_graphic_int}
590   }
591   \__tag_prop_gput:cne
592   { g__tag_struct_\int_eval:n {\c@g__tag_struct_abs_int}_prop }
593   { A }
594   {
595     <<
596     /O /Layout /BBox~
597     [
598       \g__tag_graphic_lx_tl\c_space_tl
599       \g__tag_graphic_ly_tl\c_space_tl
600       \g__tag_graphic_ux_tl\c_space_tl
601       \g__tag_graphic_uy_tl
602     ]
603     >>
604   }
605 }
606 }
607

```

(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.

```

608 \RenewDocumentCommand\picture{0{m}
609 {
610   \leavevmode
611   \keys_set:nn{tag/picture}{#1} %
612   \tl_set:Nn\l__tag_graphic_alt_dflt_tl {picture-environment}
613   \pictur@#2
614 }

```


inside the picture box we stop tagging.

```

615 \def\@picture(#1,#2)(#3,#4){%
616   \@defaultunitsset\@picht{#2}\unitlength
617   \@defaultunitsset\@tempdimc{#1}\unitlength
618   \Gin@tag@struct@begin
619   \setbox\@picbox\hb@xt@\@tempdimc\bgroup
620   \tag_stop: %do not tag inside the picture box
621   \@defaultunitsset\@tempdimc{#3}\unitlength
622   \hskip -\@tempdimc
623   \@defaultunitsset\@tempdimc{#4}\unitlength
624   \lower\@tempdimc\hbox\bgroup
625   \ignorespaces}

626 \def\endpicture{%
627   \egroup\hss\egroup
628   \ht\@picbox\@picht\dp\@picbox\z@
629   \picture@tag@bbox@attribute
630   \mbox{\box\@picbox}
631   \Gin@tag@struct@end}

```

4.7 Debugging code

`_tag_graphic_show_bbox:nnnnn`

```

632 \cs_new_protected:Npn \_tag_graphic_show_bbox:nnnnn #1#2#3#4#5#6%#5 color, #6 graphic
633 {
634   \iow_log:n {tag/graphic~debug:~BBox~of~graphics~#6~is~#1~#2~#3~#4}
635   \hook_gput_code:nnn
636   {shipout/foreground}
637   {tag/graphic}
638   {
639     \int_compare:nNnT
640     {\g_shipout_readonly_int}
641     =
642     {\property_ref:een{\_tag_graphic_#6}{abspage}{0}}
643     {
644       \put
645       (#1 bp,\dim_eval:n{-\paperheight + \dim_eval:n{#2 bp}})
646       {
647         \opacity_select:n{0.5}\color_select:n{#5}
648         \rule
649         {\dim_eval:n {#3 bp-\dim_eval:n{#1 bp}}}
650         {\dim_eval:n {#4 bp-\dim_eval:n{#2 bp}}}
651       }
652     }
653   }
654 }
655 \cs_generate_variant:Nn \_tag_graphic_show_bbox:nnnnn {VVVVne}

```

(End of definition for _tag_graphic_show_bbox:nnnnn.)

```

656 </package>
657 <*latex-lab>
658 \ProvidesFile{graphic-latex-lab-testphase.ltx}
659 [\ltxlabgraphicdate\space v\ltxlabgraphicversion\space latex-lab wrapper graphic]

```

```
660 \RequirePackage{latex-lab-testphase-graphic}  
661 \end{document}
```