

The `latex-lab-tikz` package

Support for the tagging of `TikZ` pictures

L^AT_EX Project*

v0.80e 2026-01-20

Abstract

1 Introduction

This implements the tagging of `TikZ` picture along the ideas described in `latex-lab-graphic`, please check its documentation for the background!

Resulting structures and contents should be checked!

The code doesn't use the generic sockets documented in `latex-lab-graphic` but instead sets up sockets and keys that are `TikZ`-specific. One reason for this approach is to avoid that the tagging `TikZ` pictures has side effect on pictures but more importantly because as this implementation patches into `tikz` internals special code in the plugs is required.

The module is automatically loaded if `tagging=on` or `testphase=latest` is used. The patches of `TikZ` commands can be suppressed by removing the code with the label `latex-lab-testphase-tikz` from the hook `package/tikz/after` before loading `TikZ`.

2 Implementation

```
1 <*package>
2 <@@=tag>
3 \ProvidesExplPackage {latex-lab-testphase-tikz} {\ltlabtikzdate} {\ltlabtikzversion}
4 {Code related to the tagging of tikz pictures}
```

2.1 Sockets

`support/tikz/picture/init` (*socket*) Sockets at the begin and the end of a `tikzpicture`. The argument in the `init` should process the keys of the picture and switch the plugs if needed. Unlike the generic sockets `support/tikz/picture/end` (*socket*) the begin and end sockets here do not need arguments.

```
5 \NewTaggingSocket{tikz/picture/init}{1}
6 \NewTaggingSocket{tikz/picture/begin}{0}
7 \NewTaggingSocket{tikz/picture/end}{0}
```

`tikz/picture/text/begin` (*socket*) Sockets at the end and begin of text parts.
`/tikz/picture/text/end` (*socket*)

```
8 \NewSocket{tagsupport/tikz/picture/text/begin}{0}
9 \NewSocket{tagsupport/tikz/picture/text/end}{0}
```

*Initial implementation done by Ulrike Fischer

2.2 Plugs

`\support/tikz/picture/init`) (*plug*) The `init` socket takes a list of keys, processes the known keys to setup tagging options and then assigns the plugs. TODO: correct module name? TODO: `tikz` should probably use `pgfkeys`, but as `l3keys` are easier for me ...

```

10 \NewTaggingSocketPlug{tikz/picture/init}{default}
11 {
12     \keys_set_known:nn { tikz / tagging } {#1}
13 }
14 \AssignTaggingSocketPlug{tikz/picture/init}{default}

```

`\support/tikz/picture/begin`) (*plug*) This plug handles the `TikZ` picture as a text object. So the graphical parts are tagged as artifact, but when we encounter a node we activate tagging there. There is no `BBox`. A `TikZ` picture does not necessarily starts with a `\leavevmode` so we test for `vmode`.

```

15 \NewTaggingSocketPlug{tikz/picture/begin}{text}
16 {
17     \ifvmode
18     {
19         \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
20     }
21     \fi
22     \tag_mc_end_push:
23     \tagmcbegin{artifact}

```

We hook into two `pgf` commands to add the tagging code. They are only used for `postscript` and `svg` so it should be safe inside a tagging socket for now. TODO: ask for an interface.

```

24 \def\pgfsys@begin@text
25 {
26     \tag_resume:n{\pgfpicture}
27     \tag_socket_use:n{tikz/picture/text/begin}
28 }
29 \def\pgfsys@end@text
30 {
31     \tag_socket_use:n{tikz/picture/text/end}
32     \tag_suspend:n{\pgfpicture}
33 }
34 }
35 \NewTaggingSocketPlug{tikz/picture/end}{text}
36 {
37     \tagmccend
38     \tag_mc_begin_pop:n{ }
39 }

```

`\support/tikz/picture/begin`) (*plug*) This plug handles the `TikZ` picture as a figure. Around the graphic is a `Figure` environment which will use an alt text given in the optional argument and internally tagging is suspended. The `Bbox` will be set (after the second compilation) to the size of the bounding box.

```

40 \NewTaggingSocketPlug{tikz/picture/begin}{alt}
41 {
42     \ifvmode
43     {
44         \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi

```

```

45     }
46     \fi
47     \tag_mc_end_push:

```

TODO: The structure name should be adaptable, similar to the generic implementation

```

48     \tag_struct_begin:n
49     {
50         tag=Figure,
51         alt=\l__tikz_tagging_alt_tl
52     }
53     \pgfrememberpicturepositiononpagetrue
54     \tag_mc_begin:n{tag=Figure}
55 }
56
57 \NewTaggingSocketPlug{tikz/picture/end}{alt}
58 {
59     \tag_mc_end:

```

this is the code that calculates and sets the BBox attribute

```

60     \cs_set:Npn\pgfqpoint##1##2
61     {
62         \tl_set:N\l_tmpa_tl
63         {
64             \dim_to_decimal_in_bp:n {##1+ \pgf@picminx}
65             \c_space_tl
66             \dim_to_decimal_in_bp:n {##2+ \pgf@picminy}
67             \c_space_tl
68             \dim_to_decimal_in_bp:n {##1+ \pgf@picmaxx}
69             \c_space_tl
70             \dim_to_decimal_in_bp:n {##2+ \pgf@picmaxx}
71         }
72     }

```

The following command executes a `\pgfqpoint` command with the origin coordinates, and so set the `\l_tmpa_tl` used then in the `BBox` command. With `pgfplots` the command is not expandable. So we execute it before the structure command.

```

73     \cs_if_exist:cT { pgf@sys@pdf@mark@pos@pgfid\the\pgf@picture@serial@count }
74     {
75         \use:c { pgf@sys@pdf@mark@pos@pgfid\the\pgf@picture@serial@count }
76         \tag_struct_gput:ene
77         {\tag_get:n{struct_num}}
78         {attribute}
79         {
80             /O /Layout /BBox~
81             [
82                 \l_tmpa_tl
83             ]
84         }
85     }
86     \tag_struct_end:
87     \tag_mc_begin_pop:n{}
88 }

```

`\port/tikz/picture/begin`) (*plug*) This plug handles the `TikZ` picture as a symbol with an `actualtext`. It tags the content
`\support/tikz/picture/end`) (*plug*) as a `Span` and expects an `actualtext`. Internally tagging is suspended.

```

89 \NewTaggingSocketPlug{tikz/picture/begin}{actualtext}

```

```

90 {
91   \ifvmode
92   {
93     \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
94   }
95   \fi
96   \tag_mc_end_push:
97   \tag_struct_begin:n{tag=Span,actualtext=\l__tikz_tagging_actualtext_tl}
98   \tag_mc_begin:n{ }
99 }
100
101 \NewTaggingSocketPlug{tikz/picture/end}{actualtext}
102 {
103   \tag_mc_end:
104   \tag_struct_end:
105   \tag_mc_begin_pop:n{ }
106 }

```

`\port/tikz/picture/begin` (*plug*) This plug handles the TikZ picture as an artifact, as decoration. So it is surrounded by
`\support/tikz/picture/end` (*plug*) an artifact MC and internal text does not restart tagging.

```

107 \NewTaggingSocketPlug{tikz/picture/begin}{artifact}
108 {
109   \ifvmode
110   {
111     \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
112   }
113   \fi
114   \tag_mc_end_push:
115   \tag_mc_begin:n{artifact}
116 }
117
118 \NewTaggingSocketPlug{tikz/picture/end}{artifact}
119 {
120   \tag_mc_end:
121   \tag_mc_begin_pop:n{ }
122 }

```

By default we use the text plugs. This allow packages like todonotes to get a sensible tagging without changes.

```

123 \AssignTaggingSocketPlug{tikz/picture/begin}{text}
124 \AssignTaggingSocketPlug{tikz/picture/end}{text}

```

`\tikz/picture/text/begin` (*plug*) These sockets are used inside the text plugs and ends the previous mc and restarts it
`\t/tikz/picture/text/end` (*plug*) after the text.

```

125 \NewTaggingSocketPlug{tikz/picture/text/begin}{default}
126 {
127   \tag_mc_end:
128   \tag_mc_begin:n{ }
129 }
130 \NewTaggingSocketPlug{tikz/picture/text/end}{default}
131 {
132   \tag_mc_end:
133   \tag_mc_begin:n{artifact}

```

```

134 }
135 \AssignTaggingSocketPlug{tikz/picture/text/begin}{default}
136 \AssignTaggingSocketPlug{tikz/picture/text/end}{default}

```

2.3 Patching tagging into tikz

We add the begin socket to the `\tikz@picture` command (it is also used by `\tikz`). This allows us to process the keys of the picture, assign the plugs and then to suspend tagging.

```

137 \AddToHook{package/tikz/after}
138 {
139   \AddToHookWithArguments{cmd/tikz@picture/before}
140   {
141     \tag_socket_use:nn{tikz/picture/init}{#1}
142     \tag_socket_use:n {tikz/picture/begin}
143   }

```

The suspend command is in the `cmd/pgfpicture/before` hook so that it works also if `pgfpicture` is used internally, e.g. by the `forest` package.

```

144   \AddToHookWithArguments{cmd/pgfpicture/before}
145   {
146     \tag_suspend:n{\pgfpicture}
147   }

```

The end socket is in the `\endpgfpicture` command.

```

148   \AddToHook{cmd/endpgfpicture/after}
149   {
150     \tag_resume:n{\pgfpicture}
151     \tag_socket_use:n{tikz/picture/end}
152   }
153 }

```

2.4 User interface: Keys to change the tagging behaviour

These keys will be processed directly at the begin of the picture commands to change the tagging behaviour. They are specific to `tikz` and should probably better be implemented with `pgfkeys`

```

154 \tl_new:N \l__tikz_tagging_alt_tl
155 \tl_set:Nn \l__tikz_tagging_alt_tl {Alternative~text~missing!}
156 \tl_new:N \l__tikz_tagging_actualtext_tl
157
158 \keys_define:nn { tikz / tagging }
159 {
160   alt .code:n =
161   {
162     \tl_set:Nn\l__tikz_tagging_alt_tl {\text_purify:n{#1}}
163     \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
164     \AssignTaggingSocketPlug{tikz/picture/end}{alt}
165     \def\pgfsys@begin@text{}
166     \def\pgfsys@end@text{}
167   },
168   actualtext .code:n =
169   {

```

```

170     \tl_set:N\l__tikz_tagging_actualtext_tl {\text_purify:n{#1}}
171     \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
172     \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
173     \def\pgfsys@begin@text{}
174     \def\pgfsys@end@text{}
175   },
176   artifact .code:n =
177   {
178     \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
179     \AssignTaggingSocketPlug{tikz/picture/end}{artifact}
180     \def\pgfsys@begin@text{}
181     \def\pgfsys@end@text{}
182   },
183   tagging-setup .multichoice:,
184   tagging-setup / text .code:n =
185   {
186     \AssignTaggingSocketPlug{tikz/picture/begin}{text}
187     \AssignTaggingSocketPlug{tikz/picture/end}{text}
188   },
189   tagging-setup / unknown .code:n=
190   {
191     \keys_set:nn { tag/graphic }{#1}
192   }
193 }

```

2.5 Key definitions for TikZ

```

194 \AddToHook{package/tikz/after}
195 {

```

we add a dummy definition for alt, actualtext and artifact key to avoid errors, and to allow to set the variables outside the tikzpicture.

```

196   \tikzset
197   {
198     alt/.code=
199     {
200       \tl_set:N\l__tikz_tagging_alt_tl {\text_purify:n{#1}}
201       \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
202       \AssignTaggingSocketPlug{tikz/picture/end}{alt}
203       \def\pgfsys@begin@text{}
204       \def\pgfsys@end@text{}
205     },
206     actualtext/.code=
207     {
208       \tl_set:N\l__tikz_tagging_actualtext_tl {\text_purify:n{#1}}
209       \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
210       \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
211       \def\pgfsys@begin@text{}
212       \def\pgfsys@end@text{}
213     },
214     artifact/.code=
215     {
216       \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
217       \AssignTaggingSocketPlug{tikz/picture/end}{artifact}

```

```

218         \def\pgfsys@begin@text{}
219         \def\pgfsys@end@text{}
220     }
221 }

```

2.6 TikZ keys to switch the recipes

Todo: alt and actualtext should be able to handle a value, but how is that done in pgfkeys??

```

222 \tikzset
223 {
224     tagging-setup/.is-choice,
225     tagging-setup/text/.code =
226     {
227         \AssignTaggingSocketPlug{tikz/picture/begin}{text}
228         \AssignTaggingSocketPlug{tikz/picture/end}{text}
229     },
230     tagging-setup/alt/.code =
231     {
232         \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
233         \AssignTaggingSocketPlug{tikz/picture/end}{alt}
234         \def\pgfsys@begin@text{}
235         \def\pgfsys@end@text{}
236     },
237     tagging-setup/actualtext/.code =
238     {
239         \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
240         \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
241         \def\pgfsys@begin@text{}
242         \def\pgfsys@end@text{}
243     },
244     tagging-setup/artifact/.code =
245     {
246         \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
247         \AssignTaggingSocketPlug{tikz/picture/end}{artifact}
248         \def\pgfsys@begin@text{}
249         \def\pgfsys@end@text{}
250     },
251 }
252 }

```

Todonotes works with the default text recipe quite ok, details can be handled later.

```

253 \package
254 \*latex-lab
255 \ProvidesFile{tikz-latex-lab-testphase.ltx}
256     [\ltmlabtikzdate\space v\ltmlabtikzversion\space
257     latex-lab wrapper tikz]
258
259 \RequirePackage{latex-lab-testphase-tikz}
260
261 \latex-lab

```