The latex-lab-tikz package
Support for the tagging of Ti*k*Z pictures

LaTeX Project*

v0.80j 2026-09-22

Abstract

1 Introduction

This implements the tagging of Ti*k*Z 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 Ti*k*Z-specific. One reason for this approach is to avoid that the tagging Ti*k*Z 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 Ti*k*Z commands can be suppressed by removing the code with the label `latex-lab-testphase-tikz` from the hook `package/tikz/after` before loading Ti*k*Z.

The module implements a number of keys that can be used both in `\tikzset` and in the optional argument of `tikzpicture` and `\tikz`.

`alt` This key switches to the illustrative mode. Its value is the alternative text. If no value is given the alternative text variable is not changed, `alt=` will empty the text variable.

`actualtext` This switches to the actualtext mode. 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 a `Figure` structure but in a `Span` structure and no `/BBox` attribute is added. [1]

`artifact` This tags the graphic as an artifact.

`tagging-setup` This key takes as argument a key-list, which can contain the following keys:

`alt=`⟨***`text`***⟩ This is a second way to tag the graphic as a figure with alternative text. `alt` without value will tag as a figure but not change the text variable, `alt=` will empty the text variable.

*Initial implementation done by Ulrike Fischer

[1] This is in accordance with PDF/UA-2 but violates perhaps PDF/UA-1.

actualtext=⟨*text*⟩ This is a second way to tag the graphic as a symbol with actualtext.

artifact This is a second way to tag the graphic as artifact.

text This switches to text mode.

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

tag=⟨*name*⟩ This switches to the illustrative mode but uses ⟨*name*⟩ as tag name in the structure instead of the default `Figure`. This can for example be used to tag an image of a formula with `Formula`. The key typically should be combined with a suitable alt key:

```
\begin{tikzpicture}[tagging-setup={alt=math,tag=Formula}]
```

It is possible to define new keys to change the tagging setup. As they should be processed earlier than normal tikz keys, they must be part of a special family:

```
\tikzset
 {
   my alt/.style 2 args={alt=`#1' is for #2,},
   my alt/.belongs to family=/tikz/tagging-setup,
 }
 \begin{tikzpicture}[my alt={A}{ardvaark}]
 ...
 \end{tikzpicture}
```

2 Implementation

```
⟨*package⟩
⟨@@=tag⟩

\ProvidesExplPackage {latex-lab-testphase-tikz} {\ltlabtikzdate} {\ltlabtikzversion}
  {Code related to the tagging of tikz pictures}
```

2.1 Sockets

`pport/tikz/picture/init` (*socket*)
`port/tikz/picture/begin` (*socket*)
`upport/tikz/picture/end` (*socket*)
`support/tikz/para/begin` (*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 the begin and end sockets here do not need arguments. The `tikz/para/begin` tagging socket is issued at the begin if the tikz picture is in vmode. Its default plug inserts a `\leavevmode`, but can be changed if this lead to problems in some places; plugs which remove the `\leavevmode` need to take care of dangling labels, so should test for if `\if@inlabel` is true, and also need to create the para structures.

```
\NewTaggingSocket{tikz/picture/init}{1}
\NewTaggingSocket{tikz/picture/begin}{0}
\NewTaggingSocket{tikz/picture/end}{0}
\NewTaggingSocket{tikz/para/begin}{0}
```

`ikz/picture/text/begin` (*socket*)
`/tikz/picture/text/end` (*socket*)

Sockets at the end and begin of text parts.

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

2.2 Plugs

support/tikz/para/begin) (*plug*)

```
\NewTaggingSocketPlug{tikz/para/begin}{default}{\leavevmode}
\AssignTaggingSocketPlug{tikz/para/begin}{default}
```

pport/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.

```
\NewTaggingSocketPlug{tikz/picture/init}{default}
  {
    \pgfkeyssavekeyfilterstateto\tikz@temp@tagging@saved@filterstate
    \pgfkeysinstallkeyfilter{/pgf/key~filters/active~families~or~no~family}
      {{/pgf/key~filters/false}{/pgf/key~filters/false}}
    \pgfqkeysactivatesinglefamilyandfilteroptions{/tikz/tagging-setup}{/tikz}{#1}
    \tikz@temp@tagging@saved@filterstate
  }
\AssignTaggingSocketPlug{tikz/picture/init}{default}
```

port/tikz/picture/begin) (*plug*) upport/tikz/picture/end) (*plug*) This plug handles the Ti*k*Z 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 Ti*k*Z picture does not necessarily start with a `\leavevmode` so we test for vmode and insert a socket that by default issues a `\leavevmode`.

```
\NewTaggingSocketPlug{tikz/picture/begin}{text}
  {
   \mode_if_vertical:T
    {
     \tag_socket_use:n{tikz/para/begin}
    }
   \tag_mc_end_push:
   \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.

```
   \def\pgfsys@begin@text
    {
      \tag_resume:n{\pgfpicture}
      \tag_socket_use:n{tikz/picture/text/begin}
    }
   \def\pgfsys@end@text
    {
      \tag_socket_use:n{tikz/picture/text/end}
      \tag_suspend:n{\pgfpicture}
    }
  }
\NewTaggingSocketPlug{tikz/picture/end}{text}
  {
   \tagmcend
   \tag_mc_begin_pop:n{}
  }
```

port/tikz/picture/begin) (*plug*) upport/tikz/picture/end) (*plug*) This plug handles the Ti*k*Z 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.

```
\NewTaggingSocketPlug{tikz/picture/begin}{alt}
  {
   \mode_if_vertical:T
    {
     \tag_socket_use:n{tikz/para/begin}
    }
    \tag_mc_end_push:
    \tag_struct_begin:n
     {
      tag=\UseStructureName{graphic},
      alt=\l__tikz_tagging_alt_tl
     }
    \pgfrememberpicturepositiononpagetrue
    \tag_mc_begin:n{tag=Figure}
  }

\NewTaggingSocketPlug{tikz/picture/end}{alt}
  {
   \tag_mc_end:
```

this is the code that calculates and sets the BBox attribute

```
   \cs_set:Npn\pgfqpoint##1##2
    {
      \tl_set:Ne\l_tmpa_tl
       {
         \dim_to_decimal_in_bp:n {##1+ \pgf@picminx}
         \c_space_tl
         \dim_to_decimal_in_bp:n {##2+ \pgf@picminy}
         \c_space_tl
         \dim_to_decimal_in_bp:n {##1+ \pgf@picmaxx}
         \c_space_tl
         \dim_to_decimal_in_bp:n {##2+ \pgf@picmaxx}
       }
    }
```

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.

```
   \cs_if_exist:cT { pgf@sys@pdf@mark@pos@pgfid\the\pgf@picture@serial@count }
    {
      \use:c { pgf@sys@pdf@mark@pos@pgfid\the\pgf@picture@serial@count }
      \tag_struct_gput:ene
        {\tag_get:n{struct_num}}
        {attribute}
        {
          /O /Layout /BBox~
          [
            \l_tmpa_tl
          ]
        }
     }
   \tag_struct_end:
```

```
    \tag_mc_begin_pop:n{}
  }
```

port/tikz/picture/begin) (*plug*) This plug handles the Ti*k*Z picture as a symbol with an actualtext. It tags the content
upport/tikz/picture/end) (*plug*) as a Span and expects an actualtext. Internally tagging is suspended.

```
\NewTaggingSocketPlug{tikz/picture/begin}{actualtext}
  {
   \mode_if_vertical:T
    {
     \tag_socket_use:n{tikz/para/begin}
    }
   \tag_mc_end_push:
   \tag_struct_begin:n
     {tag=\UseStructureName{graphic/symbol},
      actualtext=\l__tikz_tagging_actualtext_tl}
   \tag_mc_begin:n{}
  }

\NewTaggingSocketPlug{tikz/picture/end}{actualtext}
  {
   \tag_mc_end:
   \tag_struct_end:
   \tag_mc_begin_pop:n{}
  }
```

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

```
\NewTaggingSocketPlug{tikz/picture/begin}{artifact}
  {
    \mode_if_vertical:T
     {
      \tag_socket_use:n{tikz/para/begin}
     }
    \tag_mc_end_push:
    \tag_mc_begin:n {artifact}
  }

\NewTaggingSocketPlug{tikz/picture/end}{artifact}
  {
    \tag_mc_end:
    \tag_mc_begin_pop:n{}
  }
```

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

```
\AssignTaggingSocketPlug{tikz/picture/begin}{text}
\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.

```
\NewTaggingSocketPlug{tikz/picture/text/begin}{default}
  {
   \tag_mc_end:
```

```
    \tag_mc_begin:n{}
  }
\NewTaggingSocketPlug{tikz/picture/text/end}{default}
  {
    \tag_mc_end:
    \tag_mc_begin:n {artifact}
  }
\AssignTaggingSocketPlug{tikz/picture/text/begin}{default}
\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.

```
\AddToHook{package/tikz/after}
 {
   \AddToHookWithArguments{cmd/tikz@picture/before}
     {
       \tag_socket_use:nn {tikz/picture/init} {#1}
       \tag_socket_use:n {tikz/picture/begin}
     }
```

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.

```
   \AddToHookWithArguments{cmd/pgfpicture/before}
     {
       \tag_suspend:n {\pgfpicture}
     }
```

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

```
   \AddToHook{cmd/endpgfpicture/after}
     {
       \tag_resume:n {\pgfpicture}
       \tag_socket_use:n {tikz/picture/end}
     }
  }
```

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 so are implemented with pgfkeys.

`\l__tikz_tagging_alt_tl`
`\l__tikz_tagging_actualtext_tl`

The variables that hold the text alternatives.

```
\tl_new:N  \l__tikz_tagging_alt_tl
\tl_set:Nn \l__tikz_tagging_alt_tl {Alternative~text~missing!}
\tl_new:N  \l__tikz_tagging_actualtext_tl
```

(*End of definition for* `\l__tikz_tagging_alt_tl` *and* `\l__tikz_tagging_actualtext_tl`.)

```
\AddToHook{package/tikz/after}
  {
```

The tagging keys must be set earlier than the normal processing of Ti*k*Z keys. We implement that with a family and filter the relevant keys in the init socket, this means that the keys will be processed again when tikz does its normal processing but there is no good way to avoid that without changing various tikz internals.

```
\pgfqkeys{/tikz}
 {
  tagging-setup/.is~family,
```

The short versions are forwarded to the longer one

```
  alt/.forward~to=/tikz/tagging-setup/alt,
  alt/.belongs~to~family=/tikz/tagging-setup,
  actualtext/.forward~to=/tikz/tagging-setup/actualtext,
  actualtext/.belongs~to~family=/tikz/tagging-setup,
  artifact/.forward~to=/tikz/tagging-setup/artifact,
  artifact/.belongs~to~family=/tikz/tagging-setup,
```

The tagging-setup key.

```
  tagging-setup/.code=\pgfqkeys{/tikz/tagging-setup}{#1},
  tagging-setup/.belongs~to~family=/tikz/tagging-setup,
  tagging-setup/text/.code =
   {
     \AssignTaggingSocketPlug{tikz/picture/begin}{text}
     \AssignTaggingSocketPlug{tikz/picture/end}{text}
   },
  tagging-setup/text/.belongs~to~family=/tikz/tagging-setup,
  tagging-setup/alt/.code =
   {
     \tl_if_eq:nnF{#1}{\pgfkeysnovalue}
      { \pdf_purify:nN {#1}\l__tikz_tagging_alt_tl }
     \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
     \AssignTaggingSocketPlug{tikz/picture/end}{alt}
     \def\pgfsys@begin@text{}
     \def\pgfsys@end@text{}
   },
  tagging-setup/alt/.belongs~to~family=/tikz/tagging-setup,
  tagging-setup/tag/.code =
   {
     \AssignStructureRole{graphic}{#1}
     \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
     \AssignTaggingSocketPlug{tikz/picture/end}{alt}
     \def\pgfsys@begin@text{}
     \def\pgfsys@end@text{}
   },
  tagging-setup/tag/.belongs~to~family=/tikz/tagging-setup,
  tagging-setup/actualtext/.code =
   {
      \tl_if_eq:nnF{#1}{\pgfkeysnovalue}
       { \pdf_purify:nN {#1} \l__tikz_tagging_actualtext_tl }
      \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
      \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
      \def\pgfsys@begin@text{}
      \def\pgfsys@end@text{}
   },
  tagging-setup/actualtext/.belongs~to~family=/tikz/tagging-setup,
```

```
      tagging-setup/artifact/.code =
       {
         \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
         \AssignTaggingSocketPlug{tikz/picture/end}{artifact}
         \def\pgfsys@begin@text{}
         \def\pgfsys@end@text{}
       },
      tagging-setup/artifact/.belongs~to~family=/tikz/tagging-setup,
      tagging-setup/off/.code =
        {
          \AssignTaggingSocketPlug{tikz/picture/begin}{noop}
          \AssignTaggingSocketPlug{tikz/picture/end}{noop}
          \def\pgfsys@begin@text{}
          \def\pgfsys@end@text{}
        },
      tagging-setup/off/.belongs~to~family=/tikz/tagging-setup,
    }
}
```

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

⟨/package⟩