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

LaTeX Project[*]

v0.80c 2025-05-10

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.

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*)

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.

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

`tikz/picture/text/begin` (*socket*)
`t/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}
```

[*] Initial implementation done by Ulrike Fischer

2.2 Plugs

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. TODO: correct module name? TODO: tikz should probably use pgfkeys, but as l3keys are easier for me …

```
\NewTaggingSocketPlug{tikz/picture/init}{default}
  {
    \keys_set_known:nn { tikz / tagging } {#1}
  }
\AssignTaggingSocketPlug{tikz/picture/init}{default}
```

pport/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 necessary starts with a `\leavevmode` so we test for vmode.

```
\NewTaggingSocketPlug{tikz/picture/begin}{text}
  {
   \ifvmode
    {
     \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
    }
   \fi
   \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{\tikz/picture}
      \tag_socket_use:n{tikz/picture/text/begin}
    }
   \def\pgfsys@end@text
    {
      \tag_socket_use:n{tikz/picture/text/end}
      \tag_suspend:n{\tikzpicture}
    }
  }
\NewTaggingSocketPlug{tikz/picture/end}{text}
  {
   \tagmcend
   \tag_mc_begin_pop:n{}
  }
```

pport/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}
  {
    \ifvmode
     {
      \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
     }
    \fi
    \tag_mc_end_push:
```

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

```
    \tag_struct_begin:n
     {
      tag=Figure,
      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
    {
      \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}
    }
   \cs_if_exist:cT { pgf@sys@pdf@mark@pos@pgfid\the\pgf@picture@serial@count }
    {
      \tag_struct_gput:ene
        {\tag_get:n{struct_num}}
        {attribute}
        {
          /O /Layout /BBox~
          [
            \use:c
             { pgf@sys@pdf@mark@pos@pgfid\the\pgf@picture@serial@count }
          ]
        }
     }
   \tag_struct_end:
   \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 symbol with an actualtext. It tags the content as a Span and expects an actualtext. Internally tagging is suspended.

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

```
  {
   \ifvmode
    {
     \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
    }
   \fi
   \tag_mc_end_push:
   \tag_struct_begin:n{tag=Span,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*)
upport/tikz/picture/end) (*plug*)

This plug handles the Ti*k*Z picture as an artifact, as decoration. So it is surrounded by an artifact MC and internal text does not restart tagging.

```
\NewTaggingSocketPlug{tikz/picture/begin}{artifact}
  {
    \ifvmode
     {
      \if@inlabel \leavevmode \else \tag_socket_use:n{para/begin}\fi
     }
    \fi
    \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*)
t/tikz/picture/text/end) (*plug*)

These sockets are used inside the text plugs and ends the previous mc and restarts it 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}
       \tag_suspend:n{\tikzpicture}
     }
```

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

```
   \AddToHook{cmd/endpgfpicture/after}
     {
       \tag_resume:n{\tikzpicture}
       \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 should probably better be implemented with pgfkeys ….

```
\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

\keys_define:nn { tikz / tagging }
  {
    alt        .code:n =
     {
       \tl_if_empty:nF{#1}
         { \tl_set:Ne\l__tikz_tagging_alt_tl {\text_purify:n{#1}}}
       \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
       \AssignTaggingSocketPlug{tikz/picture/end}{alt}
       \def\pgfsys@begin@text{}
       \def\pgfsys@end@text{}
     },
    actualtext .code:n =
     {
       \tl_if_empty:nF{#1}
```

```
         { \tl_set:Ne\l__tikz_tagging_actualtext_tl {\text_purify:n{#1}} }
       \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
       \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
       \def\pgfsys@begin@text{}
       \def\pgfsys@end@text{}
     },
    artifact   .code:n =
     {
       \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
       \AssignTaggingSocketPlug{tikz/picture/end}{artifact}
       \def\pgfsys@begin@text{}
       \def\pgfsys@end@text{}
     },
    tagging-setup .multichoice:,
    tagging-setup / text .code:n =
     {
       \AssignTaggingSocketPlug{tikz/picture/begin}{text}
       \AssignTaggingSocketPlug{tikz/picture/end}{text}
     },
    tagging-setup / unknown .code:n=
      {
        \keys_set:nn { tag/graphic }{#1}
      }
  }
```

2.5 Key definitions for Ti*k*Z

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

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

```
   \tikzset
     {
      alt/.code=
       {
         \tl_set:Ne\l__tikz_tagging_alt_tl {\text_purify:n{#1}}
         \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
         \AssignTaggingSocketPlug{tikz/picture/end}{alt}
         \def\pgfsys@begin@text{}
         \def\pgfsys@end@text{}
       },
      actualtext/.code=
       {
         \tl_set:Ne \l__tikz_tagging_actualtext_tl {\text_purify:n{#1}}
         \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
         \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
         \def\pgfsys@begin@text{}
         \def\pgfsys@end@text{}
       },
      artifact/.code=
       {
         \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
         \AssignTaggingSocketPlug{tikz/picture/end}{artifact}
         \def\pgfsys@begin@text{}
         \def\pgfsys@end@text{}
       }
     }
```

2.6 Ti*k*Z keys to switch the recipes

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

```
   \tikzset
     {
       tagging-setup/.is~choice,
       tagging-setup/text/.code =
        {
          \AssignTaggingSocketPlug{tikz/picture/begin}{text}
          \AssignTaggingSocketPlug{tikz/picture/end}{text}
        },
       tagging-setup/alt/.code =
        {
          \AssignTaggingSocketPlug{tikz/picture/begin}{alt}
          \AssignTaggingSocketPlug{tikz/picture/end}{alt}
          \def\pgfsys@begin@text{}
          \def\pgfsys@end@text{}
        },
       tagging-setup/actualtext/.code =
        {
           \AssignTaggingSocketPlug{tikz/picture/begin}{actualtext}
           \AssignTaggingSocketPlug{tikz/picture/end}{actualtext}
           \def\pgfsys@begin@text{}
           \def\pgfsys@end@text{}
        },
       tagging-setup/artifact/.code =
        {
          \AssignTaggingSocketPlug{tikz/picture/begin}{artifact}
          \AssignTaggingSocketPlug{tikz/picture/end}{artifact}
          \def\pgfsys@begin@text{}
          \def\pgfsys@end@text{}
        },
     }
 }
```

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

```
⟨/package⟩
```

```
⟨*latex-lab⟩
\ProvidesFile{tikz-latex-lab-testphase.ltx}
        [\ltlabtikzdate\space v\ltlabtikzversion\space
         latex-lab wrapper tikz]

\RequirePackage{latex-lab-testphase-tikz}

⟨/latex-lab⟩
```