The l3pdfannot module
Commands for PDF annotations
LaTeX PDF management testphase bundle

The LaTeX Project*

Version 0.95a, released 2021-02-21

1 l3pdfannot documentation

This module contains a number of commands to create PDF annotations. The commands are *not* always simple wrappers around primitive commands. To allow external packages to configure links and other annotations, some of the commands have hooks and use shared attribute dictionaries. For these commands the hooks and dictionaries are selected depending on the ⟨*type*⟩ of the annotation. Currently the module only supports some general commands and link annotations. Commands for other annotations like widgets will be added later.

1.1 General annotation commands

`\pdfannot_box:nnnn` `\pdfannot_box:nnnn {⟨width⟩} {⟨height⟩} {⟨depth⟩} {⟨annot spec⟩}`

New: 2019-09-05
Updated: 2020-04-14

This creates an `/Type/Annot` object with the given dimensions. It doesn't use hooks or dictionaries.

`\pdfannot_box:nnnnn` `\pdfannot_box:nnnnn {⟨type⟩} {⟨width⟩} {⟨height⟩} {⟨depth⟩} {⟨annot spec⟩}`

New: 2020-03-30

This creates an `/Type/Annot` object with the given dimensions. ⟨*type*⟩ should be currently one of `link/URI`, `link/GoToR`, `link/Launch`, `link/GoTo` or `link/Named` or `widget`, it will then insert the attribute dictionary of this type additionally to the manually given ⟨*annot spec*⟩. The attribute dictionaries can be filled with commands described below. Hooks are not used.

`\pdfannot_box_ref_last:` `\pdfannot_box_ref_last:`

New: 2019-09-05

This retrieves the object reference of the last box annotation created.

*E-mail: latex-team@latex-project.org

1.2 Link annotations

Link annotations are special cases of annotations. In the PDF they are identified by an `/Subtype/Link` entry in the dictionary. Link annotations are quite important as many documents contain links, both internal and external. They need a set of special commands for two reasons:

At first the content of links are not only boxes. Links can contain line and page breaks (this is normally implemented by the primitive command by creating a set of annotations).

At second link annotations are objects that need some "management" as more than one package wants to configure their look and behaviour. For example hyperref, ocgx2 and the code for tagged PDF (currently in tagpdf) all want to add keys and values to the dictionaries of link annotation and code around links. So commands to create link annotations should offer suitable hooks. There are three standard places in a link where such hooks are needed: At the begin (for example for a structure command or color), in the *attr spec* dictionary of the link (for example for the border), and at the end of the link (to close a structure or the color group). For the begin and end hooks of the LaTeX hook management are predefined and used. To add and remove values from the *attr spec* dictionary special commands described below are provided. The link commands switch to horizontal mode as the commands of pdftex and luatex can't be used in vertical mode.

`\c_pdfannot_link_types_seq` There are currently five link types, `URI`, `GoToR`, `Launch`, `GoTo` or `Named`, and there are store in this constant.

`pdfannot/link/TYPE/before`
`pdfannot/link/TYPE/begin`
`pdfannot/link/TYPE/end`
`pdfannot/link/TYPE/after`

These are the hooks used by the following commands. TYPE can be one of `URI`, `GoToR`, `Launch`, `GoTo` or `Named`

`link/TYPE` These is the name of the dictionary used by the following commands. TYPE can be one of `URI`, `GoToR`, `Launch`, `GoTo` or `Named`. The dictionary can be changed by the commands `\pdfannot_dict_put:nnn` and friends described below.

\pdfannot_link:nnn `\pdfannot_link:nnn {⟨type⟩} {⟨user action spec⟩} {⟨link text⟩}`

New: 2020-03-12
Updated: 2020-12-06

This creates a link around the ⟨*link text*⟩ with the specified ⟨*user action spec*⟩[1]. `/Subtype/Link` is added automatically. ⟨*type*⟩ should be one of `URI`, `GoToR`, `Launch`, `GoTo` or `Named`. The `GoTo` variant does *not* complain if the destination name is not known like `\pdfannot_link_goto_begin:nw`. The attributes stored in the local dictionary `link/`⟨*type*⟩ are inserted as *attr spec* and the code in the begin and end hook `pdfannot/link/`⟨*type*⟩`/before` and `pdfannot/link/`⟨*type*⟩`/after` is executed before and after the link (outside the link command) while `pdfannot/link/`⟨*type*⟩`/begin` and `pdfannot/link/`⟨*type*⟩`/end` are directly around the link text. None of the hooks introduce a group. ⟨*type*⟩ should normally be identical to the value of the `/S` key in the action dictionary. As example

```
\pdfannot_dict_put:nnn
  {link/URI} { C } {[1~0~0]} %red border
\pdfannot_link:nnn { URI }
 {
   /A
     <<
       /Type/Action
       /S/URI
       /URI(https://www.latex-project.org)
     >>
 }
 { link text }
```

\pdfannot_link_begin:nnw `\pdfannot_link_begin:nnw {⟨type⟩} {⟨user action spec⟩} ⟨content⟩`
\pdfannot_link_end:n `\pdfannot_link_end:n {⟨type⟩}`

Updated: 2020-12-06

This creates a link around the ⟨*content*⟩ with the specified ⟨*user action spec*⟩ (e.g. an /A dictionary with an URI) or ⟨*destination*⟩ (a name as defined with the first argument of `\pdf_destination:nn`). `/Subtype/Link` is added automatically. In contrast to `\pdfannot_link:nnn` this function does not absorb the argument when finding the ⟨*content*⟩, and so can be used in circumstances where the ⟨*content*⟩ may not be a simple argument. But beside this, it works similar and use the same hooks. As example

```
\pdfannot_link_begin:nnw { URI }
 {
   /A<<
     /Type/Action
     /S/URI
     /URI(https://www.latex-project.org)
   >>
 }
 link text
\pdfannot_link_end:n { URI }
```

\pdfannot_link_goto_begin:nw `\pdfannot_link_goto_begin:nw {⟨destination⟩} ⟨content⟩`
\pdfannot_link_goto_end: `\pdfannot_link_goto_end:`

Updated: 2020-12-06

This is a special, shorter version for links to internal destinations. It always uses the hooks and dictionary of the `GoTo` link type. ⟨*destination*⟩ is a destination name. In difference to `\pdfannot_link:nnn { GoTo }` it will complain if ⟨*destination*⟩ is an unknown destination and give the message

```
name{ZZZZ} has been referenced but does not exist, replaced by a fixed one
```

\pdfannot_link_ref_last: This retrieves the object reference a link created previously with the commands above. This doesn't work currently with xelatex but a feature request has been made. see https://tug.org/pipermail/dvipdfmx/2020-December/000134.html

New: 2021-02-14

\pdfannot_ref_last: This retrieves the object reference a previously annotation created either with a link or a general box command. When the last was a link it won't work with xelatex. see https://tug.org/pipermail/dvipdfmx/2020-December/000134.html

New: 2021-02-14

\pdfannot_link_margin:n `\pdfannot_link_margin:n {⟨dimen⟩}`

New: 2020-03-12 This sets the dimension of the link margin.

\pdfannot_dict_put:nnn `\pdfannot_dict_put:nnn {⟨dictionary name⟩} {⟨key⟩} {⟨value⟩}`

New: 2020-12-04 This adds (locally) a key-value to the internal annot dictionaries used by the link commands above. ⟨*dictionary name*⟩ should be currently one of `link/URI`, `link/URI`,`link/GoToR`, `link/Launch`, `link/GoTo`, `link/Named`.

\pdfannot_dict_remove:nn `\pdfannot_dict_remove:nn {⟨dictionary name⟩} {⟨key⟩}`

New: 2020-12-04 This removes a key-value from the internal annot dictionary ⟨*dictionary name*⟩ should be currently one of `link/URI`, `link/GoToR`, `link/Launch`, `link/GoTo`, `link/Named`.

\pdfannot_dict_show:n `\pdfannot_dict_show:n {⟨dictionary name⟩}`

New: 2020-12-04 This shows the content of the internal annot dictionary. ⟨*dictionary name*⟩ should be currently one of `link/URI`, `link/URI`, `link/GoToR`, `link/Launch`, `link/GoTo`, `link/Named`.

\l_pdfannot_F_bitset This is a bitset variable, with the named index names suitable for the /F flag in an annotation. It can be used for example like this:

New: 2020-12-28

```
\pdfannot_dict_put:nnn {link/URI} {F}
   { \bitset_to_arabic:N \l_pdfannot_F_bitset }
\bitset_set_true:Nn \l_pdfannot_F_bitset {Print}
```

The known keys for the bitset are `Invisible`, `Hidden`, `Print`, `NoZoom`, `NoRotate`, `NoView`, `ReadOnly`, `Locked`, `ToggleNoView`, `LockedContents` which correspond to the names used in the PDF references.

2 l3pdfannot implementation

```
⟨@@=pdfannot⟩
⟨*header⟩
\ProvidesExplPackage {l3pdfannot} {2021-02-22} {0.95a}
  {PDF-annotations}
\RequirePackage{l3pdfdict}
⟨/header⟩
```

Annotations have a /F flag, we provide a public bitset for it.

```
⟨*package⟩
\RequirePackage{l3bitset}
\bitset_new:Nn \l_pdfannot_F_bitset
  {
    Invisible      = 1,
    Hidden         = 2,
    Print          = 3,
    NoZoom         = 4,
    NoRotate       = 5,
    NoView         = 6,
    ReadOnly       = 7,
    Locked         = 8,
    ToggleNoView   = 9,
    LockedContents = 10
  }
```

2.1 General Annotations

\g__pdfannot_use_lastlink_bool

The pdf engines have two different primitive commands to refer to the last created annotation: one for links, one for boxed annotation. We use a boolean to decide which one should be used, so that only one user command is needed.

```
\bool_new:N \g__pdfannot_use_lastlink_bool
```

(*End definition for* `\g__pdfannot_use_lastlink_bool`.)

```
\cs_new_protected:Npn \pdfannot_box:nnnn #1 #2 #3 #4
  {
    \__pdf_backend_annotation:nnnn {#1}{#2}{#3}{#4}
    \bool_gset_false:N\g__pdfannot_use_lastlink_bool
  }

\cs_new:Npn \pdfannot_box_ref_last:
  {
    \__pdf_backend_annotation_last:
  }

\cs_new_protected:Npn \pdfannot_box:nnnnn #1 #2 #3 #4 #5
  {
    \exp_args:Nx
    \__pdf_backend_annotation:nnnn {#2}{#3}{#4}
      {
        \pdfdict_if_exist:nT { l__pdfannot/#1 }
          {
            \pdfdict_use:n { l__pdfannot/#1}
          }
        #5
```

```
    }
  \bool_gset_false:N\g__pdfannot_use_lastlink_bool
}
```

2.2 Annotations, subtype Widget

Currently no code is provided here. The local dictionary `l_@@/Widget` is a skeleton dictionary for this subtype. It currently contains as only entry the subtype setting (the /Type is added by the backend).

```
\pdfdict_new:n { l__pdfannot/widget }
\pdfdict_put:nnn { l__pdfannot/widget }{ Subtype }{ /Widget }
```

2.3 Annotations, subtype Link

The code assumes that there will be different link types (currently URI, GoToR, Launch, GoTo, Named, hyperref uses the names url,file,run,link,menu) and that links of the same type share the *attr spec* and also the same begin/end code. The list of link types need to stay restricted and well documented so that all packages know which types they have to handle. It is stored in a constant seq.

`\c_pdfannot_link_types_seq` This constant sequence contains the list of currently supported link types for which hooks and dictionaries exist.

(*End definition for* `\c_pdfannot_link_types_seq`. *This variable is documented on page* 2.)

`link/TYPE`
`pdfannot/link/TYPE/before`
`pdfannot/link/TYPE/begin`
`pdfannot/link/TYPE/end`
`pdfannot/link/TYPE/after`

These setup the dictionary and the hook pairs.

```
\seq_const_from_clist:Nn \c_pdfannot_link_types_seq { URI , GoToR , Launch , GoTo, Named }
\seq_map_inline:Nn \c_pdfannot_link_types_seq
  {
    \pdfdict_new:n { l__pdfannot/link/#1 }
    \hook_new_pair:nn
      {pdfannot/link/#1/before}
      {pdfannot/link/#1/after}
    \hook_new_pair:nn
      {pdfannot/link/#1/begin}
      {pdfannot/link/#1/end}
  }
```

(*End definition for* `link/TYPE` *and others. These variables are documented on page* 2.)

2.3.1 Annotations, subtype Link /management

`\pdfannot_link:nnn`
`\pdfannot_link:nxn`

```
\cs_new_protected:Nn \pdfannot_link:nnn %#1 type (URI, GoTo etc),
                                        %#2 action spec, #3 link text
  {
    \hook_use:n { pdfannot/link/#1/before}
    \mode_leave_vertical:
    \exp_args:Nxx %xetex needs expansion
    \__pdf_backend_link_begin_user:nnw
      {
         \pdfdict_if_exist:nT { l__pdfannot/link/#1 }
          {
```

```
            \pdfdict_use:n { l__pdfannot/link/#1}
          }
      }
      {
        /Subtype/Link
        #2 %exp_not?
      }
    \bool_gset_true:N \g__pdfannot_use_lastlink_bool
    \hook_use:n { pdfannot/link/#1/begin}
    #3
    \hook_use:n { pdfannot/link/#1/end}
    \__pdf_backend_link_end:
    \bool_gset_true:N \g__pdfannot_use_lastlink_bool
    \hook_use:n { pdfannot/link/#1/after}
  }
\cs_generate_variant:Nn \pdfannot_link:nnn {nxn}
```

(*End definition for* `\pdfannot_link:nnn`. *This function is documented on page 3.*)

`\pdfannot_link_begin:nnw`
`\pdfannot_link_begin:nxw`
`\pdfannot_link_end:n`

```
\cs_new_protected:Npn \pdfannot_link_begin:nnw #1 #2 %#1 type, #2 action spec
  {
    \hook_use:n { pdfannot/link/#1/before}
    \mode_leave_vertical:
    \exp_args:Nxx %xetex needs expansion
      \__pdf_backend_link_begin_user:nnw
        {
           \pdfdict_if_exist:nT { l__pdfannot/link/#1 }
            {
              \pdfdict_use:n { l__pdfannot/link/#1}
            }
        }
        { #2 }
      \bool_gset_true:N \g__pdfannot_use_lastlink_bool
      \hook_use:n { pdfannot/link/#1/begin}
  }

\cs_new_protected:Nn \pdfannot_link_end:n %#1 type, e.g. url
  {
    \hook_use:n { pdfannot/link/#1/end}
    \__pdf_backend_link_end:
    \bool_gset_true:N \g__pdfannot_use_lastlink_bool
    \hook_use:n { pdfannot/link/#1/after}
  }
\cs_generate_variant:Nn \pdfannot_link_begin:nnw {nxw}
```

(*End definition for* `\pdfannot_link_begin:nnw` *and* `\pdfannot_link_end:n`. *These functions are documented on page 3.*)

`\pdfannot_link_goto_begin:nw`
`\pdfannot_link_goto_end:`

```
\cs_new_protected:Npn \pdfannot_link_goto_begin:nw #1 %#1 destination
  {
    \hook_use:n { pdfannot/link/GoTo/before}
    \mode_leave_vertical:
    \exp_args:Nxx %xetex needs expansion
```

```
    \__pdf_backend_link_begin_goto:nnw
      {
        \pdfdict_use:n { l__pdfannot/link/GoTo}
      }
      { #1 }
     \bool_gset_true:N \g__pdfannot_use_lastlink_bool
     \hook_use:n { pdfannot/link/GoTo/begin}
  }

\cs_new_protected:Nn \pdfannot_link_goto_end:
  {
    \hook_use:n { pdfannot/link/GoTo/end}
    \__pdf_backend_link_end:
    \bool_gset_true:N \g__pdfannot_use_lastlink_bool
     \hook_use:n { pdfannot/link/GoTo/after}
  }
```

(*End definition for* `\pdfannot_link_goto_begin:nw` *and* `\pdfannot_link_goto_end:`*. These functions are documented on page 4.*)

`\pdfannot_link_ref_last:`
`\pdfannot_ref_last:`

```
\cs_new:Nn \pdfannot_link_ref_last: { \__pdf_backend_link_last: }
\cs_new:Npn \pdfannot_ref_last:
  {
    \bool_if:NTF \g__pdfannot_use_lastlink_bool
      {
        \__pdf_backend_link_last:
      }
      {
        \__pdf_backend_annotation_last:
      }
  }
```

(*End definition for* `\pdfannot_link_ref_last:` *and* `\pdfannot_ref_last:`*. These functions are documented on page 4.*)

`\pdfannot_link_margin:n`

```
\cs_new_protected:Npn \pdfannot_link_margin:n #1
  {
    \__pdf_backend_link_margin:n { #1 }
  }
```

(*End definition for* `\pdfannot_link_margin:n`*. This function is documented on page 4.*)

`\pdfannot_dict_put:nnn`
`\pdfannot_dict_put:nnx`
`\pdfannot_dict_remove:nn`
`\pdfannot_dict_show:n`

```
\cs_new_protected:Npn \pdfannot_dict_put:nnn #1 #2 #3
  {
    \pdfdict_put:nnn { l__pdfannot/#1 } { #2 }{ #3 }
  }
\cs_generate_variant:Nn \pdfannot_dict_put:nnn {nnx}
\cs_new_protected:Npn \pdfannot_dict_remove:nn #1 #2
  {
    \pdfdict_remove:nn { l__pdfannot/#1 } { #2 }
  }
```

```
\cs_new_protected:Npn \pdfannot_dict_show:n #1
  {
    \pdfdict_show:n { l__pdfannot/#1 }
  }
⟨/package⟩
```

(*End definition for* `\pdfannot_dict_put:nnn`, `\pdfannot_dict_remove:nn`, *and* `\pdfannot_dict_show:n`. *These functions are documented on page 4.*)

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

R

S