

The **tagpdf-user** module
Code related to L^AT_EX2e user commands and document
commands
part of the tagpdf package

Ulrike Fischer*

Version 0.82, released 2021-06-14

1 Setup commands

<code>\tagpdfsetup</code>	<code>\tagpdfsetup{<key val list>}</code>
---------------------------	---

This is the main setup command to adapt the behaviour of tagpdf. It can be used in the preamble and in the document (but not all keys make sense there).

2 Commands related to mc-chunks

<code>\tagmcbegin</code>	<code>\tagmcbegin {<key-val>}</code>
<code>\tagmcend</code>	<code>\tagmcend</code>
<code>\tagmcuse</code>	<code>\tagmcuse{<label>}</code>

These are wrappers around `\tag_mc_begin:n`, `\tag_mc_end:` and `\tag_mc_use:n`. The commands and their argument are documented in the `tagpdf-mc` module. In difference to the expl3 commands, `\tagmcbegin` issues also an `\ignorespaces`, and `\tagmcend` will issue in horizontal mode an `\unskip`.

<code>\tagmcifin</code>	<code>\tagmcifin {<true code>}{<false code>}</code>
-------------------------	---

This is a wrapper around `\tag_mc_if_in:TF` and tests if an mc is open or not. It is mostly of importance for pdf_lat_ex as lua_lat_ex doesn't mind much if a mc tag is not correctly closed. Unlike the expl3 command it is not expandable.

The command is probably not of much use and will perhaps disappear in future versions. It normally makes more sense to push/pop an mc-chunk.

*E-mail: fischer@troubleshooting-tex.de

3 Commands related to structures

<code>\tagstructbegin</code>	<code>\tagstructbegin {⟨key-val⟩}</code>
<code>\tagstructend</code>	<code>\tagstructend</code>
<code>\tagstructuse</code>	<code>\tagstructuse{⟨label⟩}</code>

These are direct wrappers around `\tag_struct_begin:n`, `\tag_struct_end:` and `\tag_struct_use:n`. The commands and their argument are documented in the `tagpdf-struct` module.

4 Debugging

<code>\ShowTagging</code>	<code>\ShowTagging {⟨key-val⟩}</code>
---------------------------	---------------------------------------

This is a generic function to output various debugging helps. It not necessarily stops the compilation. The keys and their function are described below.

<code>mc-data</code>	<code>mc-data = ⟨number⟩</code>
----------------------	---------------------------------

This key is (currently?) relevant for lua mode only. It shows the data of all mc-chunks created so far. It is accurate only after shipout (and perhaps a second compilation), so typically should be issued after a newpage. The value is a positive integer and sets the first mc-shown. If no value is given, 1 is used and so all mc-chunks created so far are shown.

<code>mc-current</code>	<code>mc-current</code>
-------------------------	-------------------------

This key shows the number and the tag of the currently open mc-chunk. If no chunk is open it shows only the state of the abs count. It works in all mode, but the output in luamode looks different.

<code>struct-stack</code>	<code>struct-stack = log show</code>
---------------------------	--------------------------------------

This key shows the current structure stack. With `log` the info is only written to the log-file, `show` stops the compilation and shows on the terminal. If no value is used, then the default is `show`.

5 Extension commands

The following commands and code parts are not core command of tagpdf. They either provide work-arounds for missing functionality elsewhere, or do a first step to apply tagpdf commands to document commands.

The commands and keys should be view as experimental!

This part will be regularly revisited to check if the code should go to a better place or can be improved and so can change easily.

5.1 Fake space

<code>\pdffakespace</code>	(lua-only) This provides a lua-version of the <code>\pdffakespace</code> primitive of pdftex.
----------------------------	---

5.2 Paratagging

This is a first try to make use of the new paragraph hooks in a current LaTeX to automate the tagging of paragraph. It requires sane paragraph nesting, faulty code, e.g. a missing `\par` at the end of a low-level vbox can highly confuse the tagging. The tags should be carefully checked if this is used.

<code>paratagging</code>	<code>paratagging = true false</code>
<code>paratagging-show</code>	<code>paratagging-show = true false</code>

This keys can be used in `\tagpdfsetup` and enable/disable paratagging. `paratagging-show` puts small red numbers at the begin and end of a paragraph. This is meant as a debugging help. The number are boxes and have a (tiny) height, so they can affect typesetting.

<code>\tagpdfparaOn</code>	These commands allow to enable/disable para tagging too and are a bit faster then <code>\tagpdfsetup</code> . But I'm not sure if the names are good.
<code>\tagpdfparaOff</code>	

5.3 Link tagging

Links need a special structure and cross reference system. This is added through hooks of the `l3pdfannot` module and will work automatically if tagging is activated.

Links should (probably) have an alternative text in the Contents key. It is unclear which text this should be and how to get it. Currently the code simply adds the fix texts `url` and `ref`. Another text can be added by changing the dictionary value:

```
\pdfannot_dict_put:nnn
{ link/GoTo }
{ Contents }
{ (ref) }
```

6 User commands and extensions of document commands

```
1 <@=tag>
2 <*user>
3 \ProvidesExplPackage {tagpdf-user} {2021-06-14} {0.82}
4 {tagpdf - user commands}
```

7 Setup and preamble commands

`\tagpdfsetup`

```
5 \NewDocumentCommand \tagpdfsetup { m }
6 {
7   \keys_set:nn { __tag / setup } { #1 }
8 }
```

(End definition for `\tagpdfsetup`. This function is documented on page 1.)

8 Commands for the mc-chunks

```

\tagmcbegin
\tagmcend
\tagmcuse
9 \NewDocumentCommand \tagmcbegin { m }
10 {
11   \tag_mc_begin:n {#1}\ignorespaces
12 }
13
14
15 \NewDocumentCommand \tagmcend { }
16 {
17   \if_mode_horizontal: \unskip \fi: %
18   \tag_mc_end:
19 }
20
21 \NewDocumentCommand \tagmcuse { m }
22 {
23   \tag_mc_use:n {#1}
24 }
25

```

(End definition for `\tagmcbegin`, `\tagmcend`, and `\tagmcuse`. These functions are documented on page 1.)

`\tagmcifinTF` This is a wrapper around `\tag_mc_if_in:` and tests if an mc is open or not. It is mostly of importance for pdf_latex as lua_latex doesn't mind much if a mc tag is not correctly closed. Unlike the expl3 command it is not expandable.

```

26 \NewDocumentCommand \tagmcifinTF { m m }
27 {
28   \tag_mc_if_in:TF { #1 } { #2 }
29 }

```

(End definition for `\tagmcifinTF`. This function is documented on page ??.)

9 Commands for the structure

`\tagstructbegin` `\tagstructend` `\tagstructuse` These are structure related user commands. There are direct wrapper around the expl3 variants.

```

30 \NewDocumentCommand \tagstructbegin { m }
31 {
32   \tag_struct_begin:n {#1}
33 }
34
35 \NewDocumentCommand \tagstructend { }
36 {
37   \tag_struct_end:
38 }
39
40 \NewDocumentCommand \tagstructuse { m }
41 {
42   \tag_struct_use:n {#1}
43 }

```

(End definition for `\tagstructbegin`, `\tagstructend`, and `\tagstructuse`. These functions are documented on page 2.)

```
\tagpdfifluatexTF I should deprecate them ...
\tagpdfifluatexT 44 \cs_set_eq:NN\tagpdfifluatexTF \sys_if_engine luatex:TF
\tagpdfifpdfTeXTF 45 \cs_set_eq:NN\tagpdfifluatexT \sys_if_engine luatex:T
46 \cs_set_eq:NN\tagpdfifpdfTeXT \sys_if_engine pdfTeX:T
```

(End definition for `\tagpdfifluatexTF`, `\tagpdfifluatexT`, and `\tagpdfifpdfTeXTF`. These functions are documented on page ??.)

10 Debugging

\ShowTagging This is a generic command for various show commands. It takes a keyval list, the various keys are implemented below.

```
47 \NewDocumentCommand\ShowTagging { m }
48 {
49   \keys_set:nn { __tag / show }{ #1 }
50
51 }
```

(End definition for `\ShowTagging`. This function is documented on page 2.)

mc-data This key is (currently?) relevant for lua mode only. It shows the data of all mc-chunks created so far. It is accurate only after shipout, so typically should be issued after a newpage. With the optional argument the minimal number can be set.

```
52 \keys_define:nn { __tag / show }
53 {
54   mc-data .code:n =
55   {
56     \sys_if_engine luatex:T
57     {
58       \lua_now:e{!tx.__tag.trace.show_all_mc_data(#1,\__tag_get_mc_abs_cnt:,0)}
59     }
60   }
61   ,mc-data .default:n = 1
62 }
63
```

(End definition for `mc-data`. This function is documented on page 2.)

mc-current This shows some info about the current mc-chunk. It works in generic and lua-mode.

```
64 \keys_define:nn { __tag / show }
65 { mc-current .code:n =
66   {
67     \bool_if:NTF \g__tag_mode_lua_bool
68     {
69       \sys_if_engine luatex:T
70       {
71         \int_compare:nNnTF
72         { -2147483647 }
73         =
74         {
```

```

75         \lua_now:e
76         {
77             tex.print
78             (tex.getattribute
79              (luatexbase.attributes.g__tag_mc_cnt_attr))
80         }
81     }
82     {
83         \lua_now:e
84         {
85             ltx.__tag.trace.log
86             (
87                 "mc-current:~no~MC~open,~current~absent
88                 =\__tag_get_mc_abs_cnt:"
89                 ,0
90             )
91             texio.write_nl("")
92         }
93     }
94     {
95         \lua_now:e
96         {
97             ltx.__tag.trace.log
98             (
99                 "mc-current:~absent=\__tag_get_mc_abs_cnt:=="
100                 ..
101                 tex.getattribute(luatexbase.attributes.g__tag_mc_cnt_attr)
102                 ..
103                 "~=>tag="
104                 ..
105                 tostring
106                 (ltx.__tag.func.get_tag_from
107                  (tex.getattribute
108                   (luatexbase.attributes.g__tag_mc_type_attr)))
109                 ..
110                 "="
111                 ..
112                 tex.getattribute
113                 (luatexbase.attributes.g__tag_mc_type_attr)
114                 ,0
115             )
116             texio.write_nl("")
117         }
118     }
119 }
120 {
121     \msg_note:nn{ tag }{ mc-current }
122 }
123 }
124 }
125 }

```

(End definition for `mc-current`. This function is documented on page 2.)

struct-stack

```
126 \keys_define:nn { __tag / show }
127 {
128     struct-stack .choice:
129     ,struct-stack / log .code:n = \seq_log:N \g__tag_struct_tag_stack_seq
130     ,struct-stack / show .code:n = \seq_show:N \g__tag_struct_tag_stack_seq
131     ,struct-stack .default:n = show
132 }
```

(End definition for struct-stack. This function is documented on page 2.)

11 Commands to extend document commands

The following commands and code parts are not core command of tagpdf. The either provide work arounds for missing functionality elsewhere, or do a first step to apply tagpdf commands to document commands. This part should be regularly revisited to check if the code should go to a better place or can be improved.

11.1 Fake space

\pdffakespace We need a luatex variant for \pdffakespace. This should probably go into the kernel at some time.

```
133 \sys_if_engine_luatex:T
134 {
135     \NewDocumentCommand\pdffakespace { }
136     {
137         \__tag_fakespace:
138     }
139 }
```

(End definition for \pdffakespace. This function is documented on page 2.)

11.2 Paratagging

The following are some simple commands to enable/disable paratagging. Probably one should add some checks if we are already in a paragraph.

```
\l__tag_para_bool    At first some variables.
\l__tag_para_show_bool
\g__tag_para_int
140 \bool_new:N \l__tag_para_bool
141 \bool_new:N \l__tag_para_show_bool
142 \int_new:N \g__tag_para_int
```

(End definition for \l__tag_para_bool, \l__tag_para_show_bool, and \g__tag_para_int.)

paratagging
paratagging-show

These keys enable/disable locally paratagging, and the debug modus. It can affect the typesetting if paratagging-show is used. The small numbers are boxes and they have a (small) height.

```
143 \keys_define:nn { __tag / setup }
144 {
145     paratagging .bool_set:N = \l__tag_para_bool,
146     paratagging-show .bool_set:N = \l__tag_para_show_bool,
147 }
148
```

(End definition for paratagging and paratagging-show. These functions are documented on page 3.)

This fills the para hooks with the needed code.

```

149 \AddToHook{para/begin}
150 {
151   \int_gincr:N \g__tag_para_int
152   \bool_if:NT \l__tag_para_bool
153   {
154     \tag_struct_begin:n {tag=P}
155     \bool_if:NT \l__tag_para_show_bool
156     { \tag_mc_begin:n{artifact}
157       \llap{\color_select:n{red}\tiny\int_use:N\g__tag_para_int\ }
158       \tag_mc_end:
159     }
160     \tag_mc_begin:n {tag=P}
161   }
162 }
163 \AddToHook{para/end}
164 {
165   \bool_if:NT \l__tag_para_bool
166   {
167     \tag_mc_end:
168     \bool_if:NT \l__tag_para_show_bool
169     { \tag_mc_begin:n{artifact}
170       \rlap{\color_select:n{red}\tiny\int_use:N\g__tag_para_int}
171       \tag_mc_end:
172     }
173     \tag_struct_end:
174   }
175 }

```

`\tagpdfparaOn` This two command switch para mode on and off. `\tagpdfsetup` could be used too but is longer.

```

176 \newcommand\tagpdfparaOn {\bool_set_true:N \l__tag_para_bool}
177 \newcommand\tagpdfparaOff{\bool_set_false:N \l__tag_para_bool}

```

(End definition for `\tagpdfparaOn` and `\tagpdfparaOff`. These functions are documented on page 3.)

11.3 Links

We need to close and reopen mc-chunks around links. Currently we handle URI and GoTo (internal) links. Links should have an alternative text in the Contents key. It is unclear which text this should be and how to get it.

```

178 \hook_gput_code:nnn
179 {pdfannot/link/URI/before}
180 {tagpdf}
181 {
182   \tag_mc_end_push:
183   \tag_struct_begin:n { tag=Link }
184   \tag_mc_begin:n { tag=Link }
185   \pdfannot_dict_put:nnx
186   { link/URI }
187   { StructParent }
188   { \tag_struct_parent_int: }

```



```

189     }
190
191     \hook_gput_code:nnn
192     {pdfannot/link/URI/after}
193     {tagpdf}
194     {
195         \tag_struct_insert_annot:xx {\pdfannot_link_ref_last:}{\tag_struct_parent_int:}
196         \tag_mc_end:
197         \tag_struct_end:
198         \tag_mc_begin_pop:n{ }
199     }
200
201     \hook_gput_code:nnn
202     {pdfannot/link/GoTo/before}
203     {tagpdf}
204     {
205         \tag_mc_end_push:
206         \tag_struct_begin:n{tag=Link}
207         \tag_mc_begin:n{tag=Link}
208         \pdfannot_dict_put:nnx
209         { link/GoTo }
210         { StructParent }
211         { \tag_struct_parent_int: }
212     }
213
214     \hook_gput_code:nnn
215     {pdfannot/link/GoTo/after}
216     {tagpdf}
217     {
218         \tag_struct_insert_annot:xx {\pdfannot_link_ref_last:}{\tag_struct_parent_int:}
219         \tag_mc_end:
220         \tag_struct_end:
221         \tag_mc_begin_pop:n{ }
222     }
223
224
225     % "alternative descriptions " for PAX3. How to get better text here??
226     \pdfannot_dict_put:nnn
227     { link/URI }
228     { Contents }
229     { (url) }
230
231     \pdfannot_dict_put:nnn
232     { link/GoTo }
233     { Contents }
234     { (ref) }
235
</user>

```

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.

Symbols	
<code>_</code>	157, 170
A	
<code>\AddToHook</code>	149, 163
B	
bool commands:	
<code>\bool_if:NTF</code> ...	67, 152, 155, 165, 168
<code>\bool_new:N</code>	140, 141
<code>\bool_set_false:N</code>	177
<code>\bool_set_true:N</code>	176
C	
color commands:	
<code>\color_select:n</code>	157, 170
cs commands:	
<code>\cs_set_eq:NN</code>	44, 45, 46
F	
fi commands:	
<code>\fi:</code>	17
H	
hook commands:	
<code>\hook_gput_code:nnn</code>	178, 191, 201, 214
I	
if commands:	
<code>\if_mode_horizontal:</code>	17
<code>\ignorespaces</code>	1, 11
int commands:	
<code>\int_compare:nNnTF</code>	71
<code>\int_gincr:N</code>	151
<code>\int_new:N</code>	142
<code>\int_use:N</code>	157, 170
K	
keys commands:	
<code>\keys_define:nn</code>	52, 64, 126, 143
<code>\keys_set:nn</code>	7, 49
L	
<code>\llap</code>	157
lua commands:	
<code>\lua_now:n</code>	58, 75, 83, 95
M	
<code>mc-current</code>	2, 64
<code>mc-data</code>	2, 52
msg commands:	
<code>\msg_note:nn</code>	122
N	
<code>\newcommand</code>	176, 177
<code>\NewDocumentCommand</code>
5, 9, 15, 21, 26, 30, 35, 40, 47, 135	
P	
<code>paratagging</code>	3, 143
<code>paratagging-show</code>	3, 143
pdfannot commands:	
<code>\pdfannot_dict_put:nnn</code>
185, 208, 226, 231	
<code>\pdfannot_link_ref_last:</code> ...	195, 218
<code>\pdffakespace</code>	2, 133
<code>\ProvidesExplPackage</code>	3
R	
<code>\rlap</code>	170
S	
seq commands:	
<code>\seq_log:N</code>	129
<code>\seq_show:N</code>	130
<code>\ShowTagging</code>	2, 47
<code>struct-stack</code>	2, 126
sys commands:	
<code>\sys_if_engine luatex:TF</code>
44, 45, 56, 69, 133	
<code>\sys_if_engine pdftex:TF</code>	46
T	
tag commands:	
<code>\tag_mc_begin:n</code>
11, 156, 160, 169, 184, 207	
<code>\tag_mc_begin_pop:n</code>	198, 221
<code>\tag_mc_end:</code> 18, 158, 167, 171, 196, 219	
<code>\tag_mc_end_push:</code>	182, 205
<code>\tag_mc_if_in:TF</code>	28
<code>\tag_mc_use:n</code>	23
<code>\tag_struct_begin:n</code> .	32, 154, 183, 206
<code>\tag_struct_end:</code>	37, 173, 197, 220
<code>\tag_struct_insert_annot:nn</code>	195, 218
<code>\tag_struct_parent_int:</code>
188, 195, 211, 218	
<code>\tag_struct_use:n</code>	42

