

# The `latex-lab-toc-tagging-functions` package

## Tagpdf functions related to the tagging of toc and similar lists

L<sup>A</sup>T<sub>E</sub>X Project\*

Version 0.82 2023-05-20

### Abstract

Header for the testphase package

```
1 <*header>
2 \ProvidesExplPackage {latex-lab-testphase-toc} {2023-05-20} {0.82}
3   { commands related to the tagging of toc-like lists}
4 </header>
```

## 1 Introduction

The followings contains various functions related to the tagging of the table of contents and similar list. The functions will at some time be moved inside the tagpdf code.

```
5 <*package>
6 <@@=tag>
```

## 2 Temporary variables

`\l__tag_toc_tmpa_tl`

```
7 \tl_new:N \l__tag_toc_tmpa_tl
```

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

## 3 General struct commands

The following variables and commands are not restricted to toc, but probably will be need in other places too.

---

`\g__tag_struct_dest_num_prop`

---

This variable records for (some or all, not clear yet) destination names the related structure number to allow to reference them in a Ref. The key is the destination. Moved into tagpdf!

---

\*Initial implementation done by Ulrike Fischer

We use `\refstepcounter` to store the relation between destination names/`\@currentHref` and structure numbers

TODO: the functions should be moved into `tagpdf` so that one doesn't has to test if the prop exists or not.

```

8 \AddToHook{cmd/refstepcounter/after}
9 {
10   \tl_if_blank:VF \@currentHref
11   {
12     \prop_if_exist:NT \g__tag_struct_dest_num_prop
13     {
14       \prop_gput:Nxx \g__tag_struct_dest_num_prop {\@currentHref}{\tag_get:n{struct_num}}
15     }
16   }
17 }
18 \AddToHook{cmd/H@refstepcounter/after}
19 {
20   \tl_if_blank:VF \@currentHref
21   {
22     \prop_if_exist:NT \g__tag_struct_dest_num_prop
23     {
24       \prop_gput:Nxx \g__tag_struct_dest_num_prop {\@currentHref}{\tag_get:n{struct_num}}
25     }
26   }
27 }

```

---

#### `\g__tag_struct_ref_by_dest_prop`

This variable contains structures whose Ref key should be updated at the end to point to structured related with this destination. As this is probably need in other places too, it is not only a toc-variable. Moved into `tagpdf`!

`\g__tag_struct_ref_by_dest:` This command is executed and update the Ref keys of the structures listed in `\g__@@_struct_ref_by_dest_prop`. It is currently only relevant for the TOCI. But other structures could need that later too. The command is executed in the `tagpdf/finish/before` hook.

```

28 \msg_new:nnn { tag } {struct-dest-unknown}
29 {
30   Destination~#1~has~no~related~structure.\\
31   /Ref~for~structure~#2~not~updated
32 }
33
34 \cs_new_protected:Npn \g__tag_struct_ref_by_dest:
35 {
36   \prop_map_inline:Nn\g__tag_struct_ref_by_dest_prop
37   {
38     \prop_get:NnNTF \g__tag_struct_dest_num_prop {##2} \l__tag_tmpa_tl
39     {
40       \__tag_struct_gput_data_ref:ee
41       { ##1 }
42       { \tag_struct_object_ref:e{ \l__tag_tmpa_tl } }
43     }
44     {
45       \msg_warning:nnnn {tag}{struct-dest-unknown}{##2}{ ##1}

```

```

46         }
47     }
48 }
49 \hook_gput_code:nnn {tagpdf/finish/before}{tagpdf/struct/Ref}{\g__tag_struct_ref_by_dest:}

```

(End definition for \g\_\_tag\_struct\_ref\_by\_dest:.)

## 4 Toc code

---

\g\_\_tag\_toc\_level\_int \g\_\_tag\_toc\_level\_int records in a toc the current absolute level. We must close open structures at the end of the toc, for this we maintain a stack \g\_\_tag\_toc\_stack\_seq.

---

```

50 \int_new:N \g__tag_toc_level_int
51 \seq_new:N \g__tag_toc_stack_seq

```

\\_tag\_toc\_starttoc\_init:n The init code clears the stack, and set the level to -100 and start to TOC structure. We also disable paratagging. The /Title is currently simply the type, but this could be done nicer.

```

52 \cs_new_protected:Npn \_tag_toc_starttoc_init:n #1
53 {
54     \bool_set_false:N \l__tag_para_bool
55     \seq_gclear:N \g__tag_toc_stack_seq
56     \int_gset:Nn \g__tag_toc_level_int {-100}
57     \tag_struct_begin:n{tag=TOC,title=#1}
58 }

```

Now map it into the config point:

```

59 \cs_set_protected:Npn \@starttoc@cfgpoint@before#1
60 {
61     \_tag_toc_starttoc_init:n{#1}
62 }

```

(End definition for \\_tag\_toc\_starttoc\_init:n. This function is documented on page ??.)

\\_tag\_toc\_starttoc\_finalize:

```

63 \cs_new_protected:Npn \_tag_toc_starttoc_finalize:
64 {
65     \int_step_inline:nn
66         {\seq_count:N \g__tag_toc_stack_seq }
67         {\tag_struct_end:}
68     \tag_struct_end:
69     \seq_gclear:N \g__tag_toc_stack_seq
70 }

```

Now map it into the config point:

```

71 \cs_set_protected:Npn \@starttoc@cfgpoint@after#1
72 {
73     \_tag_toc_starttoc_finalize:
74 }

```

(End definition for \\_tag\_toc\_starttoc\_finalize:.)

`\__tag_toc_end:n` This commands ends all TOC on the stack with a level higher than the argument

```

75 \cs_new_protected:Npn \__tag_toc_end:n #1
76 {
77   \seq_get:NNT\g__tag_toc_stack_seq \l__tag_toc_tmpa_tl
78   {
79     \bool_lazy_and:nnT
80     {
81       \str_if_eq_p:ee{\tl_head:N\l__tag_toc_tmpa_tl}{TOC}
82     }
83     {
84       \int_compare_p:nNn {#1}<{\tl_tail:N \l__tag_toc_tmpa_tl}
85     }
86     {
87       \seq_gpop:NN\g__tag_toc_stack_seq \l__tag_toc_tmpa_tl
88       \tag_struct_end:
89       \__tag_toc_end:n{#1}
90     }
91   }
92 }
93 \cs_generate_variant:Nn \__tag_toc_end:n {e}

```

(End definition for `\__tag_toc_end:n`.)

`\__tag_toc_contentsline_begin:nnn` This is main command executed at the begin of a `\contentsline`.

```

94 \cs_new_protected:Npn \__tag_toc_contentsline_begin:nnn #1 #2 #3 % #1 level, #2 content, #3 de
95 {
96   \tag_if_active:T
97   {

```

We detect the intended level by checking the value of `toclevel@...` (currently only provided by `hyperref`, but should be there always). To be on the safe side we set it to 1 if not defined.

```

98     \ExpandArgs{c}\providecommand { toplevel@#1 }{ 1 } % just in case ...
99     \int_compare:nNnF { \use:c{toclevel@#1} } > { \use:c{c@tocdepth} }
100     {

```

if level goes up, start new sub TOC unless we are at the begin

```

101     \bool_lazy_and:nnT
102     { \int_compare_p:nNn { \g__tag_toc_level_int } > {-100} }
103     { \int_compare_p:nNn { \use:c{toclevel@#1} } > { \g__tag_toc_level_int } }
104     {
105       \seq_gpush:Nx \g__tag_toc_stack_seq {{TOC}\use:c{toclevel@#1}}
106       \tag_struct_begin:n{tag=TOC}
107     }

```

if level goes down close all TOC's with a higher level

```

108     \int_compare:nNnT
109     { \use:c{toclevel@#1} } < { \g__tag_toc_level_int }
110     {
111       \__tag_toc_end:e { \use:c{toclevel@#1} }
112     }

```

if same level do nothing update `toclevel` to the current level.

```

113     \int_gset:Nn \g__tag_toc_level_int { \use:c{toclevel@#1} }

```

now open the TOCI, the tagging of the inner structure is left to the \l@xxx commands. setting the title is not strictly necessary but looks nicer but we have to remove the \numberline

```

114         \group_begin:
115         \text_declare_expand_equivalent:Nn \numberline \use_none:n
116         \exp_args:Nx \tag_struct_begin:n{tag=TOCI,title={\text_purify:n {#2}}}
```

The TOCI structure should get a /Ref, so we put a request with its destination name into the prop.

```

117         \prop_gput:Nxx \g__tag_struct_ref_by_dest_prop
118         { \tag_get:n {struct_num} }{#3}
119         \seq_gpush:Nx \g__tag_toc_stack_seq {{TOCI}\use:c{toclevel@#1}}
120         \group_end:
121     }
122 }
123 }
```

Now map it into the config point:

```

124 \cs_set_protected:Npn \@contentsline@cfgpoint@before#1#2#3#4
125 {
126     \__tag_toc_contentsline_begin:nnn {#1}{#2}{#4}
127 }
```

(End definition for \\_\_tag\_toc\_contentsline\_begin:nnn.)

\\_\_tag\_toc\_contentsline\_end:n This is the closing code of a \contentsline. If the contentsline was actually printed, the code has to close the TOCI structure and to update the stack.

```

128 \msg_new:nnn {tag}{toc-no-TOCI}{Missing-TOCI-structure-on-toc-stack}
129
130 \cs_new_protected:Npn \__tag_toc_contentsline_end:n #1 %#1 level name
131 {
132     \int_compare:nNnF { \use:c{toclevel@#1} } > { \use:c{c@tocdepth} }
133     {
134         \seq_gpop:NNT \g__tag_toc_stack_seq \l__tag_tmpa_tl
135         {
136             \str_if_eq:eeTF{\tl_head:N\l__tag_tmpa_tl}{TOCI}
137             {
138                 \tag_struct_end:
139             }
140             {
141                 \msg_warning:nn{tag}{toc-no-TOCI}
142             }
143         }
144     }
145 }
```

Now we map it to the config point

```

146 \cs_set_protected:Npn \@contentsline@cfgpoint@after #1#2#3#4
147 {
148     \__tag_toc_contentsline_end:n {#1}
149 }
```

(End definition for \\_\_tag\_toc\_contentsline\_end:n.)

## 4.1 Tagging of the content

This need discussion.

```
150 \AddToHook{contentsline/text/before}[tagpdf]{%
151   \tag_struct_begin:n{tag=Reference}%
152   \tag_mc_begin:n{tag=Reference}}
153 \AddToHook{contentsline/text/after}[tagpdf]{%
154   \tag_mc_end:}
155 \AddToHook{contentsline/page/before}[tagpdf]{%
156   \tag_mc_begin:n{tag=Reference}}
157 \AddToHook{contentsline/page/after}[tagpdf]{%
158   \tag_mc_end:
159   \tag_struct_end:} %Reference
160 \AddToHook{contentsline/number/before}[tagpdf]{%
161   \tag_mc_end:
162   \tag_struct_begin:n{tag=Lbl}%
163   \tag_mc_begin:n{tag=Lbl}}
164 \AddToHook{contentsline/number/after}[tagpdf]{%
165   \tag_mc_end:
166   \tag_struct_end:
167   \tag_mc_begin:n{tag=Reference}}
168 \def\@dottedtocline@cfgpoint@leaders#1{%
169   \tag_mc_begin:n{artifact}\tag_stop:n{leaders}\nobreak#1\nobreak\tag_start:n{leaders}\tag_mc_
170 \}
```