

The l3backend-pdf-extra package

Additional backend PDF features

The L^AT_EX3 Project*

Released XXXX-XX-XX

1 l3backend-pdf-extra Implementation

1.1 Crossreferences

This uses the temporary l3ref-tmp.sty. It will be replaced by kernel code later. It is only needed to get a reference for the absolute page counter. This uses the counter from the new lthooks/ltshipout package.

```
1 <@@=pdf>
2 <*drivers>
3 \RequirePackage{l3ref-tmp}
4 \cs_generate_variant:Nn \ref_label:nn {en}
5 \cs_generate_variant:Nn \ref_value:nn {en}
6 \cs_new_protected:Npn \__pdf_backend_ref_label:nn #1 #2
7 {
8   \@bsphack
9   \ref_label:nn{#1}{abspage}
10  \@esphack
11 }
12 \cs_new:Npn \__pdf_backend_ref_value:nn #1 #2
13 {
14   \ref_value:nn{#1}{#2}
15 }
16 \cs_generate_variant:Nn \__pdf_backend_ref_label:nn {en}
17 \cs_generate_variant:Nn \__pdf_backend_ref_value:nn {en}
18 </drivers>
19 <*dvipdfmx | xdvipdfmx>
20 % avoid that destinations names are optimized.
21 % is this still needed??
22 % see https://tug.org/pipermail/dvipdfmx/2019-May/000002.html
23 \__kernel_backend_literal:x { dvipdfmx:config-C~ 0x0010 }
24 </dvipdfmx | xdvipdfmx>
```

```
\g__pdf_tmpa_prop Some scratch variables
\l__pdf_tmpa_tl
\l__pdf_backend_tmpa_box
25 <*drivers>
26 \prop_new:N \g__pdf_tmpa_prop
27 \tl_new:N \l__pdf_tmpa_tl
```

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

```

28 \box_new:N \l__pdf_backend_tmpa_box
29 </drivers>

```

(End definition for `\g__pdf_tmpa_prop`, `\l__pdf_tmpa_tl`, and `\l__pdf_backend_tmpa_box`.)

`\g__pdf_backend_resourceid_int` a counter to create labels for the resources, a counter to number properties in bdc marks,
`\g__pdf_backend_name_int` a counter for the `\pdfpageref` implementation.
`\g__pdf_backend_page_int`

```

30 <*drivers>
31 \int_new:N \g__pdf_backend_resourceid_int
32 \int_new:N \g__pdf_backend_name_int
33 \int_new:N \g__pdf_backend_page_int
34 </drivers>

```

(End definition for `\g__pdf_backend_resourceid_int`, `\g__pdf_backend_name_int`, and `\g__pdf_backend_page_int`.)

1.2 luacode

Load the lua code.

```

35 <*luatex>
36 \directlua { require("l3backend-pdf.lua") }
37 </luatex>

```

1.3 Hooks

1.3.1 Add the “end run” hooks

Here we add the end run hook to suitable end hooks.

```

38 <*pdftex | luatex>
39 \hook_gput_code:nnn {enddocument/afterlastpage}
40 {pdf/endrun}
41 {
42   \hook_use:n {pdf/management/end_run}
43 }
44 </pdftex | luatex>
45 <*dvipdfmx | xdvipdfmx>
46 \hook_gput_code:nnn {shipout/lastpage}
47 {pdf/endrun}
48 {
49   \hook_use:n {pdf/management/end_run}
50 }
51 \hook_gset_rule:nnnn {shipout/lastpage}{pdf}{before}{pdf/endrun}
52 </dvipdfmx | xdvipdfmx>
53 <*dvips>
54 \hook_gput_code:nnn {shipout/lastpage}
55 {pdf/endrun}
56 {
57   \hook_use:n {pdf/management/end_run}
58 }
59 \hook_gset_rule:nnnn {shipout/lastpage}{pdf}{before}{pdf/endrun}
60 </dvips>

```

1.3.2 Add the “shipout” hooks

Now we add to the shipout hooks the relevant token lists. We also push the page resources in shipout/firstpage (AtBeginDvi) as the backend code sets color stack there. The xetex driver needs a rule here. If it clashes on the first page, we will need a test ...

```

61 <*drivers>
62 \tl_if_exist:NTF \@kernel@after@shipout@background
63 {
64   \g@addto@macro \@kernel@before@shipout@background{\relax}
65   \g@addto@macro \@kernel@after@shipout@background
66     {
67       \hook_use:n {pdf/management/thispage_shipout}
68     }
69   \g@addto@macro \@kernel@after@shipout@lastpage
70     {
71       \hook_use:n {pdf/management/lastpage_shipout}
72     }
73 }
74 {
75   \hook_gput_code:nnn{shipout/background}{pdf}
76   {
77     \hook_use:n {pdf/management/thispage_shipout}
78   }
79   \hook_gput_code:nnn {shipout/lastpage} {pdf}
80   {
81     \hook_use:n {pdf/management/lastpage_shipout}
82   }
83 }
84
85 </drivers>

```

1.4 The /Pages dictionary (pdfpagesattr)

`_pdf_backend_Pages_primitive:n` This is the primitive command to add something to the /Pages dictionary. It works differently for the backends: pdftex and luatex overwrite existing content, dvips and dvipdfmx are additive. luatex sets it in lua. The higher level code has to take this into account.

```

86 <*pdftex>
87 \cs_new_protected:Npn \_pdf_backend_Pages_primitive:n #1
88 {
89   \tex_global:D \tex_pdfpagesattr:D { #1 }
90 }
91 </pdftex>
92 <*luatex>
93 %luatex: does it in lua
94 \sys_if_engine luatex:T
95 {
96   \cs_new_protected:Npn \_pdf_backend_Pages_primitive:n #1
97   {
98     \tex_directlua:D
99     {
100       pdf.setpagesattributes( \_pdf_backend_luastring:n { #1 } )
101     }

```

```

102     }
103   }
104   </luatex>
105   <*dvips>
106   \cs_new_protected:Npx \__pdf_backend_Pages_primitive:n #1
107   {
108     \tex_special:D{ps:~[#1~/PAGES~pdfmark} %]
109   }
110   </dvips>
111   <*dvipdfmx | xdvipdfmx>
112   \cs_new_protected:Npn \__pdf_backend_Pages_primitive:n #1
113   {
114     \__pdf_backend:n{put~@pages~<<#1>>}
115   }
116   </dvipdfmx | xdvipdfmx>
117   <*dvisvgm>
118   \cs_new_protected:Npn \__pdf_backend_Pages_primitive:n #1
119   {}
120   </dvisvgm>

```

(End definition for `__pdf_backend_Pages_primitive:n`.)

1.5 “Page” and “ThisPage” attributes (pdfpageattr)

`__pdf_backend_Page_primitive:n` `__pdf_backend_Page_primitive:n` is the primitive command to add something to the /Page dictionary. It works differently for the backends: pdftex and luatex overwrite existing content, dvips and dvipdfmx are additive. luatex sets it in lua. The higher level code has to take this into account. `__pdf_backend_Page_gput:nn` stores default values. `__pdf_backend_Page_gremove:n` allows to remove a value. `__pdf_backend_Page_gpush:n` adds a value to the current page. `__pdf_backend_Page_gpush:n` merges the default and the current page values and add them to the dictionary of the current page in `\g__pdf_backend_thispage_shipout_tl`.

```

121 % backend commands
122 <*pdftex>
123 %the primitive
124 \cs_new_protected:Npn \__pdf_backend_Page_primitive:n #1
125 {
126   \tex_global:D \tex_pdfpageattr:D { #1 }
127 }
128 % the command to store default values.
129 % Uses a prop with pdflatex + dvi,
130 % sets a lua table with luatex
131 \cs_new_protected:Npn \__pdf_backend_Page_gput:nn #1 #2 %key,value
132 {
133   \pdfdict_gput:nnn {g__pdf_Core/Page}{ #1 }{ #2 }
134 }
135 % the command to remove a default value.
136 % Uses a prop with pdflatex + dvi,
137 % changes a lua table with luatex
138 \cs_new_protected:Npn \__pdf_backend_Page_gremove:n #1
139 {
140   \pdfdict_gremove:nn {g__pdf_Core/Page}{ #1 }
141 }

```

```

142 % the command used in the document.
143 % direct call of the primitive special with dvips/dvipdfmx
144 % \latelua: fill a page related table with luatex, merge it with the page
145 % table and push it directly
146 % write to aux and store in prop with pdflatex
147 \cs_new_protected:Npn \__pdf_backend_ThisPage_gput:nn #1 #2
148 {
149     %we need to know the page the resource should be added too.
150     \int_gincr:N\g__pdf_backend_resourceid_int
151     %\zref@labelbylist {l3pdf\int_use:N\g__pdf_backend_resourceid_int} {l3pdf}
152     %\ref_label:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
153     \__pdf_backend_ref_label:en { l3pdf\int_use:N\g__pdf_backend_resourceid_int }{abspage}
154     \tl_set:Nx \l__pdf_tmpa_tl
155     {
156         %\zref@extractdefault
157         {l3pdf\int_use:N\g__pdf_backend_resourceid_int}
158         {pdf@abspage}
159         {0}
160         % \ref_value:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
161         \__pdf_backend_ref_value:en {l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
162     }
163     \pdfdict_if_exist:nF { g__pdf_Core/backend_Page\l__pdf_tmpa_tl}
164     {
165         \pdfdict_new:n { g__pdf_Core/backend_Page\l__pdf_tmpa_tl}
166     }
167     %backend_Page has no handler.
168     \pdfdict_gput:nnn {g__pdf_Core/backend_Page\l__pdf_tmpa_tl}{ #1 }{ #2 }
169 }
170 %the code to push the values, used in shipout
171 %merges the two props and then fills the register in pdflatex
172 %merges the two tables and then fills (in lua) in luatex
173 %issues the values stored in the global prop with dvi
174 \cs_new_protected:Npn \__pdf_backend_ThisPage_gpush:n #1
175 {
176     \prop_gset_eq:Nc \g__pdf_tmpa_prop { \pdfdict_name:n { g__pdf_Core/Page } }
177     \prop_if_exist:cT { \pdfdict_name:n { g__pdf_Core/backend_Page#1 } }
178     {
179         \prop_map_inline:cn { \pdfdict_name:n { g__pdf_Core/backend_Page#1 } }
180         {
181             \prop_gput:Nnn \g__pdf_tmpa_prop { ##1 }{ ##2 }
182         }
183     }
184     \exp_args:Nx \__pdf_backend_Page_primitive:n
185     {
186         \prop_map_function:NN \g__pdf_tmpa_prop \pdfdict_item:ne
187     }
188 }
189 </pdfTeX>
190 <*luatex>
191 % do we need to use some escaping for the values????
192 \cs_new:Npn \__pdf_backend_luastring:n #1
193 {
194     "\tex_luaescapestring:D { \tex_unexpanded:D { #1 } }"
195 }

```

```

196 %not used, only there for consistency
197 \cs_new_protected:Npn \__pdf_backend_Page_primitive:n #1
198 {
199   \tex_latelua:D
200   {
201     pdf.setpageattributes(\__pdf_backend_luastring:n { #1 })
202   }
203 }
204 % the command to store default values.
205 % Uses a prop with pdflatex + dvi,
206 % sets a lua table with luatex
207 \cs_new_protected:Npn \__pdf_backend_Page_gput:nn #1 #2
208 {
209   \tex_directlua:D
210   {
211     ltx.__pdf.backend_Page_gput
212     (
213       \__pdf_backend_luastring:n { #1 },
214       \__pdf_backend_luastring:n { #2 }
215     )
216   }
217 }
218 % the command to remove a default value.
219 % Uses a prop with pdflatex + dvi,
220 % changes a lua table with luatex
221 \cs_new_protected:Npn \__pdf_backend_Page_gremove:n #1
222 {
223   \tex_directlua:D
224   {
225     ltx.__pdf.backend_Page_gremove (\__pdf_backend_luastring:n { #1 })
226   }
227 }
228 % the command used in the document.
229 % direct call of the primitive special with dvips/dvipdfmx
230 % \latelua: fill a page related table with luatex, merge it with the page
231 % table and push it directly
232 % write to aux and store in prop with pdflatex
233 \cs_new_protected:Npn \__pdf_backend_ThisPage_gput:nn #1 #2
234 {
235   \tex_latelua:D
236   {
237     ltx.__pdf.backend_ThisPage_gput
238     (
239       tex.count["g_shipout_readonly_int"],
240       \__pdf_backend_luastring:n { #1 },
241       \__pdf_backend_luastring:n { #2 }
242     )
243     ltx.__pdf.backend_ThisPage_gpush (tex.count["g_shipout_readonly_int"])
244   }
245 }
246 %the code to push the values, used in shipout
247 %merges the two props and then fills the register in pdflatex
248 %merges the two tables (the one is probably still empty) and then fills (in lua) in luatex
249 %issues the values stored in the global prop with dvi

```

```

250 \cs_new_protected:Npn \__pdf_backend_ThisPage_gpush:n #1
251 {
252   \tex_latelua:D
253   {
254     ltx.__pdf.backend_ThisPage_gpush (tex.count["g_shipout_readonly_int"])
255   }
256 }
257
258 </luatex>
259 <*dvipdfmx | xdvipdfmx>
260 %the primitive
261 \cs_new_protected:Npn \__pdf_backend_Page_primitive:n #1
262 {
263   \tex_special:D{pdf:-put~@thispage-<<#1>>}
264 }
265 % the command to store default values.
266 % Uses a prop with pdflatex + dvi,
267 % sets a lua table with luatex
268 \cs_new_protected:Npn \__pdf_backend_Page_gput:nn #1 #2
269 {
270   \pdfdict_gput:nnn {g__pdf_Core/Page}{ #1 }{ #2 }
271 }
272 % the command to remove a default value.
273 % Uses a prop with pdflatex + dvi,
274 % changes a lua table with luatex
275 \cs_new_protected:Npn \__pdf_backend_Page_gremove:n #1
276 {
277   \pdfdict_gremove:nn {g__pdf_Core/Page}{ #1 }
278 }
279 % the command used in the document.
280 % direct call of the primitive special with dvips/dvipdfmx
281 % \latelua: fill a page related table with luatex, merge it with the page
282 % table and push it directly
283 % write to aux and store in prop with pdflatex
284 \cs_new_protected:Npn \__pdf_backend_ThisPage_gput:nn #1 #2
285 {
286   \__pdf_backend_Page_primitive:n { /#1~#2 }
287 }
288 %the code to push the values, used in shipout
289 %merges the two props and then fills the register in pdflatex
290 %merges the two tables (the one is probably still empty)
291 % and then fills (in lua) in luatex
292 %issues the values stored in the global prop with dvi
293 \cs_new_protected:Npn \__pdf_backend_ThisPage_gpush:n #1
294 {
295   \exp_args:Nx \__pdf_backend_Page_primitive:n
296   { \pdfdict_use:n { g__pdf_Core/Page} }
297 }
298 </dvipdfmx | xdvipdfmx>
299 <*dvips>
300 \cs_new_protected:Npn \__pdf_backend_Page_primitive:n #1
301 {
302   \tex_special:D{ps:-[ThisPage]<<#1>>~/PUT~pdfmark} %]
303 }

```

```

304 % the command to store default values.
305 % Uses a prop with pdflatex + dvi,
306 % sets a lua table with luatex
307 \cs_new_protected:Npn \__pdf_backend_Page_gput:nn #1 #2
308 {
309     \pdfdict_gput:nnn {g__pdf_Core/Page}{ #1 }{ #2 }
310 }
311 % the command to remove a default value.
312 % Uses a prop with pdflatex + dvi,
313 % changes a lua table with luatex
314 \cs_new_protected:Npn \__pdf_backend_Page_gremove:n #1
315 {
316     \pdfdict_gremove:nn {g__pdf_Core/Page}{ #1 }
317 }
318 % the command used in the document.
319 % direct call of the primitive special with dvips/dvipdfmx
320 % \latelua: fill a page related table with luatex, merge it with the page
321 % table and push it directly
322 % write to aux and store in prop with pdflatex
323 \cs_new_protected:Npn \__pdf_backend_ThisPage_gput:nn #1 #2
324 {
325     \__pdf_backend_Page_primitive:n { /#1~#2 }
326 }
327 %the code to push the values, used in shipout
328 %merges the two props and then fills the register in pdflatex
329 %merges the two tables (the one is probably still empty)
330 %and then fills (in lua) in luatex
331 %issues the values stored in the global prop with dvi
332 \cs_new_protected:Npn \__pdf_backend_ThisPage_gpush:n #1
333 {
334     \exp_args:Nx \__pdf_backend_Page_primitive:n
335         { \pdfdict_use:n { g__pdf_Core/Page} }
336 }
337 </dvips>
338 <*dvisvgm>
339 % mostly only dummies ...
340 \cs_new_protected:Npn \__pdf_backend_Page_primitive:n #1
341 {}
342 % Uses a prop with pdflatex + dvi,
343 \cs_new_protected:Npn \__pdf_backend_Page_gput:nn #1 #2
344 {
345     \pdfdict_gput:nnn {g__pdf_Core/Page}{ #1 }{ #2 }
346 }
347 % the command to remove a default value.
348 % Uses a prop with pdflatex + dvi,
349 \cs_new_protected:Npn \__pdf_backend_Page_gremove:n #1
350 {
351     \pdfdict_gremove:nn {g__pdf_Core/Page}{ #1 }
352 }
353 % the command used in the document.
354 \cs_new_protected:Npn \__pdf_backend_ThisPage_gput:nn #1 #2
355 {}
356 %the code to push the values, used in shipout
357 \cs_new_protected:Npn \__pdf_backend_ThisPage_gpush:n #1

```



```

358 {}
359 </dvisvgm>

```

(End definition for `_pdf_backend_Page_primitive:n` and others.)

1.6 “Page/Resources”: ExtGState, ColorSpace, Shading, Pattern

Path: Page/Resources/ExtGState etc. The actual output of the resources is handled together with the bdc/Properties. Here is only special code.

`_pdf_backend_PageResources_gput:nnn`

stores values for the page resources.

#1 : name of the resource (ExtGState, ColorSpace, Shading, Pattern)

#2 : a pdf name without slash

#3 : value

This pushes out the objects. It is a no-op with xdvipdfmx and dvips.

`_pdf_backend_PageResources_obj_gpush:`

```

360 % backend commands the command to fill the register
361 % and to push the values.
362 %
363 % The names are quite often needed
364 % a similar list is now in l3pdfmanagement. Perhaps it should be merged.
365 <*drivers>
366 \clist_const:Nn \c__pdf_backend_PageResources_clist
367 {
368     ExtGState,
369     ColorSpace,
370     Pattern,
371     Shading,
372 }
373 </drivers>
374 % pdftex and luatex
375 <*pdftex | luatex>
376 %create the backend objects:
377 \clist_map_inline:Nn \c__pdf_backend_PageResources_clist
378 {
379     \_pdf_backend_object_new:nn {Page/Resources/#1} {dict}
380     \cs_if_exist:NT \tex_directlua:D
381     {
382         \tex_directlua:D
383         {
384             ltx.__pdf.object["Page/Resources/#1"]
385             =
386             "\_pdf_backend_object_ref:n{Page/Resources/#1}"
387         }
388     }
389 }
390 </pdftex | luatex>
391 <*luatex>
392 %values are only stored in a prop and will be output at end document.
393 \cs_new_protected:Npn \_pdf_backend_PageResources_gput:nnn #1 #2 #3
394 {
395     \pdfdict_gput:nnn {g__pdf_Core/Page/Resources/#1} { #2 }{ #3 }
396     % luatex must also trigger the lua side

```

```

397 \tex_latelua:D{ltx.__pdf.Page.Resources.#1=true}
398 \tex_latelua:D
399 {
400   ltx.pdf.Page.Resources_gpush(tex.count["g_shipout_readonly_int"])
401 }
402 }
403 </luatex>
404 <*pdfTeX>
405 \cs_new_protected:Npn \__pdf_backend_PageResources_gput:nnn #1 #2 #3
406 {
407   \pdfdict_gput:nnn {g__pdf_Core/Page/Resources/#1} { #2 }{ #3 }
408 }
409 </pdfTeX>
410 <*pdfTeX | luatex>
411 %code for end of document code
412 \cs_new_protected:Npn \__pdf_backend_PageResources_obj_gpush:
413 {
414   \clist_map_inline:Nn \c__pdf_backend_PageResources_clist
415   {
416     \prop_if_empty:cF
417     { \pdfdict_name:n { g__pdf_Core/Page/Resources/##1} }
418     {
419       \__pdf_backend_object_write:nx
420       { Page/Resources/##1 }
421       { \pdfdict_use:n { g__pdf_Core/Page/Resources/##1} }
422     }
423   }
424 }
425 </pdfTeX | luatex>
426 % xdvipdfmx
427 % \special{pdf:pageresources<<#1>>} doesn't work correctly with object names ...
428 % https://tug.org/pipermail/dvipdfmx/2019-August/000021.html,
429 % so we use \special{pdf:put @resources}
430 % this must be issued on every page!
431 <*dvipdfmx | xdvipdfmx>
432 %objects should not only be created but also "initialized"
433 % initialization should be done before anyone tries to write
434 % so we add rules for the backend.
435 <xdvipdfmx>\hook_gset_rule:nnnn{shipout/firstpage}{l3backend-xetex}{after}{pdf}
436 <dvipdfmx>\hook_gset_rule:nnnn{shipout/firstpage}{l3backend-dvipdfmx}{after}{pdf}
437 %
438 \clist_map_inline:Nn \c__pdf_backend_PageResources_clist
439 {
440   \__pdf_backend_object_new:nn { Page/Resources/#1 } { dict }
441   \hook_gput_code:nnn{shipout/firstpage}{pdf}{\__pdf_backend_object_write:nn { Page/Resour
442 }
443 \cs_new_protected:Npn \__pdf_backend_PageResources:n #1
444 {
445   \__pdf_backend:n {put~@resources~<<#1>>}
446 }
447 \cs_new_protected:Npn \__pdf_backend_PageResources_gput:nnn #1 #2 #3
448 {
449   % this is not used for output, but there is a test if the resource is empty
450   \prop_gput:cnn { \pdfdict_name:n { g__pdf_Core/Page/Resources/#1} } { #2 }{ #3 }

```

```

451 %objects are not filled with \pdf_object_write as this is not additive!
452 \__pdf_backend:x
453 {
454     put~\__pdf_backend_object_ref:n {Page/Resources/#1}<</#2~#3>>
455 }
456 }
457
458 \cs_new_protected:Npn \__pdf_backend_PageResources_obj_gpush: {}
459 </dvipdfmx | xdvipdfmx>
460 <*dvips>
461 % dvips unneeded, or no-op
462 \cs_new_protected:Npn \__pdf_backend_PageResources:n #1 {}
463 \cs_new_protected:Npn \__pdf_backend_PageResources_gput:nnn #1 #2 #3
464 { %only for the show command TEST!!
465     \pdfdict_gput:nnn {g__pdf_Core/Page/Resources/#1} { #2 }{ #3 }
466 }
467 \cs_new_protected:Npn \__pdf_backend_PageResources_obj_gpush: {}
468 </dvips>
469 <*dvisvgm>
470 % dvips unneeded, or no-op
471 \cs_new_protected:Npn \__pdf_backend_PageResources:n #1 {}
472 \cs_new_protected:Npn \__pdf_backend_PageResources_gput:nnn #1 #2 #3
473 { %only for the show command TEST!!
474     \pdfdict_gput:nnn {g__pdf_Core/Page/Resources/#1} { #2 }{ #3 }
475 }
476 \cs_new_protected:Npn \__pdf_backend_PageResources_obj_gpush: {}
477 </dvisvgm>

```

(End definition for __pdf_backend_PageResources_gput:nnn and __pdf_backend_PageResources_obj_gpush:.)

1.6.1 Page resources /Properties + BDC operators

```

\__pdf_backend_bdc:nn \__pdf_backend_bdc:nn, \__pdf_backend_bdc_obj:nn, \__pdf_backend_bdc_obj:n,
\__pdf_backend_bdc_obj:nn \__pdf_backend_bmc:n and \__pdf_backend_emc: are the backend command that cre-
\__pdf_backend_bdc_obj:n ate the bdc/emc marker and store the properties. \__pdf_backend_PageResources_-
\__pdf_backend_bmc:n gpush:n outputs the /Properties and/or the other resources for the current page.
\__pdf_backend_emc:
\__pdf_backend_PageResources_gpush:n
478 % pdftex and luatex (and perhaps dvips ...) need to know if there are in a
479 % xform stream ...
480 <*drivers>
481 \bool_new:N \l__pdf_backend_xform_bool
482 </drivers>
483 <*dvips>
484 % dvips is easy: create an object, and reference it in the bdc
485 % ghostscript will then automatically replace it by a name
486 % and add the name to the /Properties dict
487 % special variant von accsupp
488 % https://chat.stackexchange.com/transcript/message/50831812#50831812
489 %
490 \cs_set_protected:Npn \__pdf_backend_bdc:nn #1 #2 % #1 eg. Span, #2: dict_content
491 {
492     \__pdf_backend_pdfmark:x{/#1~<<#2>>~/BDC}
493 }
494 \cs_set_protected:Npn \__pdf_backend_bdc_obj:nn #1 #2 % #1 eg. Span, #2: object name

```

```

495 {
496   \__pdf_backend_pdfmark:x{/#1~\__pdf_backend_object_ref:n{#2}~/BDC}
497 }
498 \cs_set_protected:Npn \__pdf_backend_bdc_obj:n #1 % #1 eg. Span,
499 {
500   \__pdf_backend_pdfmark:x{/#1~\__pdf_backend_object_last:~/BDC}
501 }
502 \cs_set_protected:Npn \__pdf_backend_emc:
503 {
504   \__pdf_backend_pdfmark:n{/EMC} %
505 }
506 \cs_set_protected:Npn \__pdf_backend_bmc:n #1
507 {
508   \__pdf_backend_pdfmark:n{/#1~/BMC} %
509 }
510 \cs_new_protected:Npn \__pdf_backend_PageResources_gpush:n #1 {}
511
512 </dvips>
513 <*dvisvgm>
514 % dvisvgm should do nothing
515 %
516 \cs_set_protected:Npn \__pdf_backend_bdc:nn #1 #2 % #1 eg. Span, #2: dict_content
517 {}
518 \cs_set_protected:Npn \__pdf_backend_bdc_obj:nn #1 #2 % #1 eg. Span, #2: object name
519 {}
520 \cs_set_protected:Npn \__pdf_backend_bdc_obj:n #1 % #1 eg. Span,
521 {}
522 \cs_set_protected:Npn \__pdf_backend_emc:
523 {}
524 \cs_set_protected:Npn \__pdf_backend_bmc:n #1
525 {}
526 \cs_new_protected:Npn \__pdf_backend_PageResources_gpush:n #1 {}
527
528 </dvisvgm>
529
530 % xetex has to create the entries in the /Properties manually
531 % (like the other backends)
532 % use pdfbase special
533 % https://chat.stackexchange.com/transcript/message/50832016#50832016
534 % the property is added to xform resources automatically,
535 % no need to worry about it.
536 <*dvipdfmx | xdvipdfmx>
537 \cs_set_protected:Npn \__pdf_backend_bdc_obj:nn #1 #2 % #1 eg. Span, #2: object name
538 {
539   \int_gincr:N \g__pdf_backend_name_int
540   \__kernel_backend_literal:x
541   {
542     pdf:code~/#1/l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl BDC
543   }
544   \__kernel_backend_literal:x
545   {
546     pdf:put~@resources~
547     <<
548     /Properties~

```

```

549         <<
550         /l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl
551         \__pdf_backend_object_ref:n { #2 }
552         >>
553     >>
554 }
555 }
556 \cs_set_protected:Npn \__pdf_backend_bdc_obj:n #1 % #1 eg. Span
557 {
558     \int_gincr:N \g__pdf_backend_name_int
559     \__kernel_backend_literal:x
560     {
561         pdf:code~/#1/l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl BDC
562     }
563     \__kernel_backend_literal:x
564     {
565         pdf:put~@resources~
566         <<
567         /Properties~
568         <<
569         /l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl
570         \__pdf_backend_object_last:
571         >>
572     >>
573 }
574 }
575 \cs_set_protected:Npn \__pdf_backend_bmc:n #1
576 {
577     \__kernel_backend_literal:n {pdf:code~/#1~BMC} %pdfbase
578 }
579
580 %this require management
581 \cs_set_protected:Npn \__pdf_backend_bdc_contobj:nn #1 #2
582 {
583     \pdf_object_now:nn { dict }{ #2 }
584     \__pdf_backend_bdc_obj:n { #1 }
585 }
586
587 \cs_set_protected:Npn \__pdf_backend_bdc_contstream:nn #1 #2
588 {
589     \__kernel_backend_literal:n {pdf:code~ /#1~<<#2>>~BDC }
590 }
591
592 \cs_set_protected:Npn \__pdf_backend_bdc:nn #1 #2
593 {
594     \bool_if:NTF \g__pdfmanagement_active_bool
595     {\cs_gset_eq:NN \__pdf_backend_bdc:nn \__pdf_backend_bdc_contobj:nn}
596     {\cs_gset_eq:NN \__pdf_backend_bdc:nn \__pdf_backend_bdc_contstream:nn}
597     \__pdf_backend_bdc:nn {#1}{#2}
598 }
599 \cs_set_protected:Npn \__pdf_backend_emc:
600 {
601     \__kernel_backend_literal:n {pdf:code~EMC} %pdfbase
602 }

```

```

603 % properties are handled automatically, but the other resources should be added
604 % at shipout
605 \cs_new_protected:Npn \__pdf_backend_PageResources_gpush:n #1
606 {
607   \clist_map_inline:Nn \c__pdf_backend_PageResources_clist
608   {
609     \prop_if_empty:cF { \pdfdict_name:n { g__pdf_Core/Page/Resources/##1} }
610     {
611       \__kernel_backend_literal:x
612       {
613         pdf:put~@resources~
614         <</##1~\__pdf_backend_object_ref:n {Page/Resources/##1}>>
615       }
616     }
617   }
618 }
619 </dvipdfmx | xdvipdfmx>
620 % luatex + pdftex
621 < *luatex>
622 \cs_set_protected:Npn \__pdf_backend_bdc_obj:nn #1 #2 % #1 eg. Span, #2: object name
623 {
624   \int_gincr:N \g__pdf_backend_name_int
625   \exp_args:Nx \__kernel_backend_literal_page:n
626   { /#1 ~ /l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl BDC }
627   \bool_if:NTF \l__pdf_backend_xform_bool
628   {
629     \exp_args:Nnx\pdfdict_gput:nnn
630     { g__pdf_Core/Xform/Resources/Properties }
631     { l3pdf\int_use:N\g__pdf_backend_name_int }
632     { \__pdf_backend_object_ref:n { #2 } }
633   }
634   {
635     \exp_args:Nx \tex_latelua:D
636     {
637       ltx.pdf.Page_Resources_Properties_gput
638       (
639         tex.count["g_shipout_readonly_int"],
640         "l3pdf\int_use:N\g__pdf_backend_name_int",
641         "\__pdf_backend_object_ref:n { #2 }"
642       )
643     }
644   }
645 }
646 \cs_set_protected:Npn \__pdf_backend_bdc_obj:n #1% #1 eg. Span
647 {
648   \int_gincr:N \g__pdf_backend_name_int
649   \exp_args:Nx \__kernel_backend_literal_page:n
650   { /#1 ~ /l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl BDC }
651   \bool_if:NTF \l__pdf_backend_xform_bool
652   {
653     \exp_args:Nnx\pdfdict_gput:nnn %no handler needed
654     { g__pdf_Core/Xform/Resources/Properties }
655     { l3pdf\int_use:N\g__pdf_backend_name_int }
656     { \__pdf_backend_object_last: }

```

```

657     }
658     {
659         \exp_args:Nx \tex_latelua:D
660         {
661             ltx.pdf.Page_Resources_Properties_gput
662             (
663                 tex.count["g_shipout_readonly_int"],
664                 "l3pdf\int_use:N\g__pdf_backend_name_int",
665                 "\__pdf_backend_object_last:"
666             )
667         }
668     }
669 }
670 \cs_set_protected:Npn \__pdf_backend_bmc:n #1
671 {
672     \__kernel_backend_literal_page:n { /#1~BMC }
673 }
674 \cs_set_protected:Npn \__pdf_backend_bdc_contobj:nn #1 #2
675 {
676     \pdf_object_now:nn { dict } { #2 }
677     \__pdf_backend_bdc_obj:n { #1 }
678 }
679 \cs_set_protected:Npn \__pdf_backend_bdc_contstream:nn #1 #2
680 {
681     \__kernel_backend_literal_page:n { /#1~<<#2>>~BDC }
682 }
683 \cs_set_protected:Npn \__pdf_backend_bdc:nn #1 #2
684 {
685     \bool_if:NTF \g__pdfmanagement_active_bool
686     {\cs_gset_eq:NN \__pdf_backend_bdc:nn \__pdf_backend_bdc_contobj:nn}
687     {\cs_gset_eq:NN \__pdf_backend_bdc:nn \__pdf_backend_bdc_contstream:nn}
688     \__pdf_backend_bdc:nn {#1}{#2}
689 }
690 \cs_set_protected:Npn \__pdf_backend_emc:
691 {
692     \__kernel_backend_literal_page:n { EMC }
693 }
694
695 \cs_new_protected:Npn \__pdf_backend_PageResources_gpush:n #1 {}
696 </luatex>
697 <*pdfTeX>
698 % pdfLaTeX is the most complicated as it has to go through the aux ...
699 % the push command is extended to take other resources too
700 \cs_set_protected:Npn \__pdf_backend_bdc_obj:nn #1 #2 % #1 eg. Span, #2: object name
701 {
702     \int_gincr:N \g__pdf_backend_name_int
703     \exp_args:Nx \__kernel_backend_literal_page:n
704     { /#1 ~ /l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl BDC }
705     % code to set the property ...
706     \int_gincr:N \g__pdf_backend_resourceid_int
707     \bool_if:NTF \l__pdf_backend_xform_bool
708     {
709         \exp_args:Nxxx \pdfdict_gput:nnn %no handler needed
710         { g__pdf_Core/Xform/Resources/Properties }

```

```

711         { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
712         { \__pdf_backend_object_ref:n { #2 } }
713     }
714     {
715         %\zref@labelbylist
716         { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
717         { l3pdf }
718         %\ref_label:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
719         \__pdf_backend_ref_label:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
720         \tl_set:Nx \l__pdf_tmpa_tl
721         {
722             %\zref@extractdefault
723             { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
724             %\pdf@abspage}
725             {0}
726             %\ref_value:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
727             \__pdf_backend_ref_value:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
728         }
729         \pdfdict_if_exist:nF { g__pdf_Core/backend_Page\l__pdf_tmpa_tl/Resources/Properties
730         {
731             \pdfdict_new:n { g__pdf_Core/backend_Page\l__pdf_tmpa_tl/Resources/Properties }
732         }
733         \exp_args:Nnxx\pdfdict_gput:nnn
734         { g__pdf_Core/backend_Page\l__pdf_tmpa_tl/Resources/Properties }
735         { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
736         { \__pdf_backend_object_ref:n{#2} }
737     }
738 }
739 \cs_set_protected:Npn \__pdf_backend_bdc_obj:n #1% #1 eg. Span
740 {
741     \int_gincr:N \g__pdf_backend_name_int
742     \exp_args:Nx\__kernel_backend_literal_page:n
743     { /#1 ~ /l3pdf\int_use:N\g__pdf_backend_name_int\c_space_tl BDC }
744     % code to set the property ....
745     \int_gincr:N\g__pdf_backend_resourceid_int
746     \bool_if:NTF \l__pdf_backend_xform_bool
747     {
748         \exp_args:Nnxx\pdfdict_gput:nnn
749         { g__pdf_Core/Xform/Resources/Properties }
750         { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
751         { \__pdf_backend_object_last: }
752     }
753     {
754         %\zref@labelbylist
755         { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
756         { l3pdf }
757         %\ref_label:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
758         \__pdf_backend_ref_label:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
759         \tl_set:Nx \l__pdf_tmpa_tl
760         {
761             %\zref@extractdefault
762             { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
763             %\pdf@abspage}
764             {0}

```



```

765         % \ref_value:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
766         \__pdf_backend_ref_value:en{l3pdf\int_use:N\g__pdf_backend_resourceid_int}{abspage}
767     }
768     \pdfdict_if_exist:nF { g__pdf_Core/backend_Page\l__pdf_tmpa_tl/Resources/Properties
769     {
770         \pdfdict_new:n { g__pdf_Core/backend_Page\l__pdf_tmpa_tl/Resources/Properties }
771     }
772     \exp_args:Nnxx\pdfdict_gput:nnn
773     { g__pdf_Core/backend_Page\l__pdf_tmpa_tl/Resources/Properties }
774     { l3pdf\int_use:N\g__pdf_backend_resourceid_int }
775     { \__pdf_backend_object_last: }
776     %\pdfdict_show:n { g_backend_Page\l__pdf_tmpa_tl/Resources/Properties }
777 }
778 }
779 \cs_set_protected:Npn \__pdf_backend_bmc:n #1
780 {
781     \__kernel_backend_literal_page:n { /#1~BMC }
782 }
783 \cs_set_protected:Npn \__pdf_backend_bdc_contobj:nn #1 #2
784 {
785     \pdf_object_now:nn { dict } { #2 }
786     \__pdf_backend_bdc_obj:n { #1 }
787 }
788 \cs_set_protected:Npn \__pdf_backend_bdc_contstream:nn #1 #2
789 {
790     \__kernel_backend_literal_page:n { /#1~<<#2>>~BDC }
791 }
792 \cs_set_protected:Npn \__pdf_backend_bdc:nn #1 #2
793 {
794     \bool_if:NTF \g__pdfmanagement_active_bool
795     {\cs_gset_eq:NN \__pdf_backend_bdc:nn \__pdf_backend_bdc_contobj:nn}
796     {\cs_gset_eq:NN \__pdf_backend_bdc:nn \__pdf_backend_bdc_contstream:nn}
797     \__pdf_backend_bdc:nn {#1}{#2}
798 }
799 \cs_set_protected:Npn \__pdf_backend_emc:
800 {
801     \__kernel_backend_literal_page:n { EMC }
802 }
803
804 \cs_new:Npn \__pdf_backend_PageResources_gpush_aux:n #1 %#1 ExtGState etc
805 {
806     \prop_if_empty:cF
807     { \pdfdict_name:n { g__pdf_Core/Page/Resources/#1} }
808     {
809         \pdfdict_item:ne { #1 }{\pdf_object_ref:n {Page/Resources/#1}}
810     }
811 }
812
813 \cs_new_protected:Npn \__pdf_backend_PageResources_gpush:n #1
814 {
815     \exp_args:NNx \tex_global:D \tex_pdfpageresources:D
816     {
817         \prop_if_exist:cT
818         { \pdfdict_name:n { g__pdf_Core/backend_Page#1/Resources/Properties } }

```

```

819         {
820             /Properties~
821             <<
822                 \prop_map_function:cN
823                 { \pdfdict_name:n { g__pdf_Core/backend_Page#1/Resources/Properties } }
824                 \pdfdict_item:ne
825             >>
826         }
827         %% add ExtGState etc
828         \clist_map_function:NN
829         \c__pdf_backend_PageResources_clist
830         \__pdf_backend_PageResources_gpush_aux:n
831     }
832 }
833
834 </pdftex>

```

(End definition for __pdf_backend_bdc:nn and others.)

1.7 “Catalog” & subdirectories (pdfcatalog)

The backend command is already in the driver: __pdf_backend_catalog_gput:nn

1.7.1 Special case: the /Names/EmbeddedFiles dictionary

Entries to /Names are handled differently, in part (/Desc) it is automatic, for other special commands like \pdfnames must be used. For EmbeddedFiles we need some code to push the tree if files have been added. dvips wants code for every file and then creates the Name tree automatically.

```

835 % pdflatex
836 <*pdftex>
837 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_gpush:n #1 %array content
838 {
839     \pdf_object_now:nn {dict} {/Names [#1] }
840     \tex_pdfnames:D {/EmbeddedFiles~\pdf_object_last:}
841 }
842 </pdftex>
843 <*luatex>
844 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_gpush:n #1 %array content
845 {
846     \pdf_object_now:nn {dict} {/Names [#1] }
847     \tex_pdfextension:D~names~{/EmbeddedFiles~\pdf_object_last:}
848 }
849 </luatex>
850 <*dvi.pdfmx | xdvipdfmx>
851 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_gpush:n #1 %array content
852 {
853     \pdf_object_now:nn {dict} { /Names [#1] }
854     %n or x?
855     \__pdf_backend:x {put~@names~<</EmbeddedFiles~\pdf_object_last: >>}
856 }
857 </dvi.pdfmx | xdvipdfmx>
858
859 %dvips: noop

```

```

860 <*dvips>
861 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_gpush:n #1 {}
862 </dvips>
863 %dvisvgm: noop
864 <*dvisvgm>
865 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_gpush:n #1 {}
866 </dvisvgm>
867

```

Names in the EmbeddedFiles name tree must sorted alphabetically, so we need commands to create this names. And we need a sequence to store the names and the objects. We use the prefix l3ef, and we assume that at most 9999 files will be used.

\g__pdf_backend_EmbeddedFiles_int

(End definition for \g__pdf_backend_EmbeddedFiles_int.)

__pdf_backend_EmbeddedFiles_name:

```

868 <*drivers>
869 \int_new:N \g__pdf_backend_EmbeddedFiles_int
870 \cs_new:Npn \__pdf_backend_EmbeddedFiles_name:
871 {
872 (
873   l3ef
874   \int_compare:nNnT {\g__pdf_backend_EmbeddedFiles_int} < {10}
875   {0}
876   \int_compare:nNnT {\g__pdf_backend_EmbeddedFiles_int} < {100}
877   {0}
878   \int_compare:nNnT {\g__pdf_backend_EmbeddedFiles_int} < {1000}
879   {0}
880   \int_use:N \g__pdf_backend_EmbeddedFiles_int
881 )
882 }
883 </drivers>

```

(End definition for __pdf_backend_EmbeddedFiles_name:.)

\g__pdf_backend_EmbeddedFiles_seq
\g__pdf_backend_EmbeddedFiles_named_prop

The sequence will hold the content of the array that is pushed out at then end (not with dvips), the prop holds the obj names-names relation.

(End definition for \g__pdf_backend_EmbeddedFiles_seq and \g__pdf_backend_EmbeddedFiles_named_prop.)

```

884 <*drivers>
885 \seq_new:N \g__pdf_backend_EmbeddedFiles_seq
886 \prop_new:N \g__pdf_backend_EmbeddedFiles_named_prop
887 </drivers>

```

__pdf_backend_NamesEmbeddedFiles_add:n

This command saves an object reference of a filespec dictionary in the EmbeddedFiles name tree. We define a prop to store the relation between object name and name in the name tree.

```

888 <*pdftex | luatex | dvipdfmx | xdvi pdfmx>
889 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_add:n #1
890   %#1 object ref
891   {
892     \int_gincr:N \g__pdf_backend_EmbeddedFiles_int

```

```

893 \prop_gput:Nnx \g__pdf_backend_EmbeddedFiles_named_prop
894 { #1 }
895 { \__pdf_backend_EmbeddedFiles_name: }
896 \seq_gput_right:Nx \g__pdf_backend_EmbeddedFiles_seq
897 { \__pdf_backend_EmbeddedFiles_name: \c_space_tl #1 }
898 }
899
900 </pdfTeX | luatex | dvipdfmx | xdviPDFmx>
901 <*dvips>
902 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_add:n #1
903 {
904 \int_gincr:N \g__pdf_backend_EmbeddedFiles_int
905 \prop_gput:Nnx \g__pdf_backend_EmbeddedFiles_named_prop
906 { #1 }
907 { \__pdf_backend_EmbeddedFiles_name: }
908 \__pdf_backend_pdfmark:x
909 {
910 /Name~\__pdf_backend_EmbeddedFiles_name:~
911 /FS~#1~
912 /EMBED
913 }
914 }
915 </dvips>
916 <*dvisvgm>
917 %no op. Or is there any sensible use for it?
918 \cs_new_protected:Npn \__pdf_backend_NamesEmbeddedFiles_add:n #1
919 {}
920 </dvisvgm>

```

(End definition for __pdf_backend_NamesEmbeddedFiles_add:n.)

1.7.2 FormXObject / backend

__pdf_backend_xform_new:nnnn #1 : name
 #2 : attributes
 #3 : resources needed?? or are all resources autogenerated?
 #4 : content, this doesn't need to be a box!

```

\__pdf_backend_xform_use:n 921 <*pdfTeX>
\__pdf_backend_xform_ref:n 922 \cs_new_protected:Npn \__pdf_backend_xform_new:nnnn #1 #2 #3 #4
923 % #1 name
924 % #2 attributes
925 % #3 resources
926 % #4 content, not necessarily a box!
927 {
928 \hbox_set:Nn \l__pdf_backend_tmpa_box
929 {
930 \bool_set_true:N \l__pdf_backend_xform_bool
931 \prop_gc_clear:c {\pdfdict_name:n { g__pdf_Core/Xform/Resources/Properties }}
932 #4
933 }
934 %store the dimensions
935 \tl_const:cx

```

```

936     { c__pdf_backend_xform_wd_ \tl_to_str:n {#1} _tl }
937     { \tex_the:D \box_wd:N \l__pdf_backend_tmpa_box }
938 \tl_const:cx
939     { c__pdf_backend_xform_ht_ \tl_to_str:n {#1} _tl }
940     { \tex_the:D \box_ht:N \l__pdf_backend_tmpa_box }
941 \tl_const:cx
942     { c__pdf_backend_xform_dp_ \tl_to_str:n {#1} _tl }
943     { \tex_the:D \box_dp:N \l__pdf_backend_tmpa_box }
944 %% do we need to test if #2 and #3 are empty??
945 \tex_immediate:D \tex_pdfxform:D
946 ~ attr ~ { #2 }
947 %% which other resources should be default? Is an argument actually needed?
948 ~ resources ~
949 {
950     #3
951     \int_compare:nNnT
952     { \prop_count:c { \pdfdict_name:n { g__pdf_Core/Xform/Resources/Properties } } }
953     >
954     { 0 }
955     {
956         /Properties~
957         <<
958         \pdfdict_use:n { g__pdf_Core/Xform/Resources/Properties }
959         >>
960     }
961
962     \prop_if_empty:cF
963     { \pdfdict_name:n { g__pdf_Core/Page/Resources/ExtGState } }
964     {
965         /ExtGState~ \pdf_object_ref:n { Page/Resources/ExtGState }
966     }
967     \prop_if_empty:cF
968     { \pdfdict_name:n { g__pdf_Core/Page/Resources/Pattern } }
969     {
970         /Pattern~ \pdf_object_ref:n { Page/Resources/Pattern }
971     }
972     \prop_if_empty:cF
973     { \pdfdict_name:n { g__pdf_Core/Page/Resources/Shading } }
974     {
975         /Shading~ \pdf_object_ref:n { Page/Resources/Shading }
976     }
977     \prop_if_empty:cF
978     { \pdfdict_name:n { g__pdf_Core/Page/Resources/ColorSpace } }
979     {
980         /ColorSpace~ \pdf_object_ref:n { Page/Resources/ColorSpace }
981     }
982 }
983 \l__pdf_backend_tmpa_box
984 \int_const:cn
985 { c__pdf_backend_xform_ \tl_to_str:n {#1} _int }
986 { \tex_pdflastxform:D }
987 }
988
989 \cs_new_protected:Npn \__pdf_backend_xform_use:n #1

```

```

990 {
991   \tex_pdfrefxform:D
992   \int_use:c { c__pdf_backend_xform_ \tl_to_str:n {#1} _int }
993   \scan_stop:
994 }
995
996 \cs_new:Npn \__pdf_backend_xform_ref:n #1
997 {
998   \int_use:c { c__pdf_backend_xform_ \tl_to_str:n {#1} _int } ~ 0 ~ R
999 }
1000 </pdfTeX>
1001 <*luatex>
1002 %luatex
1003 %nearly identical but not completely ...
1004 \cs_new_protected:Npn \__pdf_backend_xform_new:nnnn #1 #2 #3 #4
1005 % #1 name
1006 % #2 attributes
1007 % #3 resources
1008 % #4 content, not necessarily a box!
1009 {
1010   \hbox_set:Nn \l__pdf_backend_tmpa_box
1011   {
1012     \bool_set_true:N \l__pdf_backend_xform_bool
1013     \prop_gc_clear:c { \pdfdict_name:n { g__pdf_Core/Xform/Resources/Properties } }
1014     #4
1015   }
1016   \tl_const:cx
1017   { c__pdf_backend_xform_wd_ \tl_to_str:n {#1} _tl }
1018   { \tex_the:D \box_wd:N \l__pdf_backend_tmpa_box }
1019   \tl_const:cx
1020   { c__pdf_backend_xform_ht_ \tl_to_str:n {#1} _tl }
1021   { \tex_the:D \box_ht:N \l__pdf_backend_tmpa_box }
1022   \tl_const:cx
1023   { c__pdf_backend_xform_dp_ \tl_to_str:n {#1} _tl }
1024   { \tex_the:D \box_dp:N \l__pdf_backend_tmpa_box }
1025   %% do we need to test if #2 and #3 are empty??
1026   \tex_immediate:D \tex_pdfxform:D
1027   ~ attr ~ { #2 }
1028   %% which resources should be default? Is an argument actually needed?
1029   ~ resources ~
1030   {
1031     #3
1032     \int_compare:nNnT
1033     { \prop_count:c { \pdfdict_name:n { g__pdf_Core/Xform/Resources/Properties } } }
1034     >
1035     { 0 }
1036     {
1037       /Properties~
1038       <<
1039       \pdfdict_use:n { g__pdf_Core/Xform/Resources/Properties }
1040       >>
1041     }
1042     \prop_if_empty:cF
1043     { \pdfdict_name:n { g__pdf_Core/Page/Resources/ExtGState } }

```

```

1044     {
1045         /ExtGState~ \pdf_object_ref:n { Page/Resources/ExtGState }
1046     }
1047     \prop_if_empty:cF
1048     { \pdffdict_name:n { g__pdf_Core/Page/Resources/Pattern } }
1049     {
1050         /Pattern~ \pdf_object_ref:n { Page/Resources/Pattern }
1051     }
1052     \prop_if_empty:cF
1053     { \pdffdict_name:n { g__pdf_Core/Page/Resources/Shading } }
1054     {
1055         /Shading~ \pdf_object_ref:n { Page/Resources/Shading }
1056     }
1057     \prop_if_empty:cF
1058     { \pdffdict_name:n { g__pdf_Core/Page/Resources/ColorSpace } }
1059     {
1060         /ColorSpace~ \pdf_object_ref:n { Page/Resources/ColorSpace }
1061     }
1062 }
1063 \l__pdf_backend_tmpa_box
1064 \int_const:cn
1065 { c__pdf_backend_xform_ \tl_to_str:n {#1} _int }
1066 { \tex_pdflastxform:D }
1067 }
1068
1069 \cs_new_protected:Npn \__pdf_backend_xform_use:n #1 %protected as with xelatex
1070 {
1071     \tex_pdfrefxform:D \int_use:c
1072     {
1073         c__pdf_backend_xform_ \tl_to_str:n {#1} _int
1074     }
1075     \scan_stop:
1076 }
1077
1078 \cs_new:Npn \__pdf_backend_xform_ref:n #1
1079 { \int_use:c { c__pdf_backend_xform_ \tl_to_str:n {#1} _int } ~ 0 ~ R }
1080
1081 </luatex>
1082 <*dvipdfmx | xdvipdfmx>
1083 % xetex
1084 % it needs a bit testing if it really works to set the box to 0 before the special ...
1085 % does it disturb viewing the xobject?
1086 % what happens with the resources (bdc)? (should work as they are specials too)
1087 % xetex requires that the special is in horizontal mode. This means it affects
1088 % typesetting. But we can no delay the whole form code to shipout
1089 % as the object reference and the size is often wanted on the current page.
1090 % so we need to allocate a box - but probably they won't be thousands xform
1091 % in a document so it shouldn't matter.
1092 \cs_new_protected:Npn \__pdf_backend_xform_new:nnnn #1 #2 #3 #4
1093 % #1 name
1094 % #2 attributes
1095 % #3 resources
1096 % #4 content, not necessarily a box!
1097 {

```

```

1098 \int_gincr:N \g__pdf_backend_object_int
1099 \int_const:cn
1100 { c__pdf_backend_xform_ \tl_to_str:n {#1} _int }
1101 { \g__pdf_backend_object_int }
1102 \box_new:c { g__pdf_backend_xform_#1_box }
1103 \hbox_gset:cn { g__pdf_backend_xform_#1_box }
1104 {
1105     \bool_set_true:N \l__pdf_backend_xform_bool
1106     #4
1107 }
1108 \tl_const:cx
1109 { c__pdf_backend_xform_wd_ \tl_to_str:n {#1} _tl }
1110 { \tex_the:D \box_wd:c { g__pdf_backend_xform_#1_box } }
1111 \tl_const:cx
1112 { c__pdf_backend_xform_ht_ \tl_to_str:n {#1} _tl }
1113 { \tex_the:D \box_ht:c { g__pdf_backend_xform_#1_box } }
1114 \tl_const:cx
1115 { c__pdf_backend_xform_dp_ \tl_to_str:n {#1} _tl }
1116 { \tex_the:D \box_dp:c { g__pdf_backend_xform_#1_box } }
1117 \box_set_dp:cn { g__pdf_backend_xform_#1_box } { \c_zero_dim }
1118 \box_set_ht:cn { g__pdf_backend_xform_#1_box } { \c_zero_dim }
1119 \box_set_wd:cn { g__pdf_backend_xform_#1_box } { \c_zero_dim }
1120 \hook_gput_next_code:nn {shipout/background}
1121 {
1122     \mode_leave_vertical: %needed, the xform disappears without it.
1123     \__pdf_backend:x
1124     {
1125         bxobj ~ \__pdf_backend_xform_ref:n { #1 }
1126         \c_space_tl width ~ \pdf_xform_wd:n { #1 }
1127         \c_space_tl height ~ \pdf_xform_ht:n { #1 }
1128         \c_space_tl depth ~ \pdf_xform_dp:n { #1 }
1129     }
1130     \box_use_drop:c { g__pdf_backend_xform_#1_box }
1131     \__pdf_backend:x {put ~ @resources ~<<#3>> }
1132     \__pdf_backend:x
1133     {
1134         put~ @resources ~
1135         <<
1136         /ExtGState~ \pdf_object_ref:n { Page/Resources/ExtGState }
1137         >>
1138     }
1139     \__pdf_backend:x
1140     {
1141         put~ @resources ~
1142         <<
1143         /Pattern~ \pdf_object_ref:n { Page/Resources/Pattern }
1144         >>
1145     }
1146     \__pdf_backend:x
1147     {
1148         put~ @resources ~
1149         <<
1150         /Shading~ \pdf_object_ref:n { Page/Resources/Shading }
1151         >>

```



```

1152     }
1153     \__pdf_backend:x
1154     {
1155         put~ @resources ~
1156         <<
1157         /ColorSpace~
1158         \pdf_object_ref:n { Page/Resources/ColorSpace }
1159         >>
1160     }
1161     \exp_args:Nx
1162     \__pdf_backend:x {exobj ~<<#2>>}
1163 }
1164 }
1165
1166
1167
1168 \cs_new:Npn \__pdf_backend_xform_ref:n #1
1169 {
1170     @pdf.xform \int_use:c { c__pdf_backend_xform_ \tl_to_str:n {#1} _int }
1171 }
1172
1173 \cs_new_protected:Npn \__pdf_backend_xform_use:n #1
1174 {
1175     \hbox_set:Nn \l__pdf_backend_tmpa_box
1176     {
1177         \__pdf_backend:x
1178         {
1179             uxobj~ \__pdf_backend_xform_ref:n { #1 }
1180         }
1181     }
1182     \box_set_wd:Nn \l__pdf_backend_tmpa_box { \pdf_xform_wd:n { #1 } }
1183     \box_set_ht:Nn \l__pdf_backend_tmpa_box { \pdf_xform_ht:n { #1 } }
1184     \box_set_dp:Nn \l__pdf_backend_tmpa_box { \pdf_xform_dp:n { #1 } }
1185     \box_use_drop:N \l__pdf_backend_tmpa_box
1186 }
1187 </dvipdfmx | xdvipdfmx>
1188 <*dvisvgm>
1189 % unclear what it should do!!
1190 \cs_new_protected:Npn \__pdf_backend_xform_new:nnnn #1 #2 #3 #4 {}
1191 \cs_new_protected:Npn \__pdf_backend_xform_use:n #1 {}
1192 \cs_new:Npn \__pdf_backend_xform_ref:n {}
1193 </dvisvgm>
1194 <*drivers>
1195 %% all
1196 \prg_new_conditional:Npnn \__pdf_backend_xform_if_exist:n #1 { p , T , F , TF }
1197 {
1198     \int_if_exist:cTF { c__pdf_backend_xform_ \tl_to_str:n {#1} _int }
1199     { \prg_return_true: }
1200     { \prg_return_false: }
1201 }
1202 \prg_new_eq_conditional:NNn \pdf_xform_if_exist:n \__pdf_backend_xform_if_exist:n
1203 { TF , T , F , p }
1204 </drivers>

```

(End definition for __pdf_backend_xform_new:nnnn, __pdf_backend_xform_use:n, and __pdf_-

backend_xform_ref:n.)

1.8 lua code for lualatex

```
1205  $\langle *lua \rangle$ 
1206 ltx= ltx or {}
1207 ltx.__pdf      = ltx.__pdf or {}
1208 ltx.__pdf.Page = ltx.__pdf.Page or {}
1209 ltx.__pdf.Page.dflt = ltx.__pdf.Page.dflt or {}
1210 ltx.__pdf.Page.Resources = ltx.__pdf.Resources or {}
1211 ltx.__pdf.Page.Resources.Properties = ltx.__pdf.Page.Resources.Properties or {}
1212 ltx.__pdf.Page.Resources.List={"ExtGState","ColorSpace","Pattern","Shading"}
1213 ltx.__pdf.object = ltx.__pdf.object or {}
1214
1215 ltx.pdf= ltx.pdf or {} -- for "public" functions
1216
1217 local __pdf = ltx.__pdf
1218 local pdf = pdf
1219
1220 local function __pdf_backend_Page_gput (name,value)
1221   __pdf.Page.dflt[name]=value
1222 end
1223
1224 local function __pdf_backend_Page_gremove (name)
1225   __pdf.Page.dflt[name]=nil
1226 end
1227
1228 local function __pdf_backend_Page_gclear ()
1229   __pdf.Page.dflt={}
1230 end
1231
1232 local function __pdf_backend_ThisPage_gput (page,name,value)
1233   __pdf.Page[page] = __pdf.Page[page] or {}
1234   __pdf.Page[page][name]=value
1235 end
1236
1237 local function __pdf_backend_ThisPage_gpush (page)
1238   local token=""
1239   local t = {}
1240   local tkeys= {}
1241   for name,value in pairs(__pdf.Page.dflt) do
1242     t[name]=value
1243   end
1244   if __pdf.Page[page] then
1245     for name,value in pairs(__pdf.Page[page]) do
1246       t[name] = value
1247     end
1248   end
1249   -- sort the table to get reliable test files.
1250   for name,value in pairs(t) do
1251     table.insert(tkeys,name)
1252   end
1253   table.sort(tkeys)
1254   for _,name in ipairs(tkeys) do
```

```

1255     token = token .. "/"..name.." "..t[name]
1256   end
1257   return token
1258 end
1259
1260 function ltx.__pdf.backend_ThisPage_gput (page,name,value) -- tex.count["g_shipout_readonly_
1261   __pdf_backend_ThisPage_gput (page,name,value)
1262 end
1263
1264 function ltx.__pdf.backend_ThisPage_gpush (page)
1265   pdf.setpageattributes(__pdf_backend_ThisPage_gpush (page))
1266 end
1267
1268 function ltx.__pdf.backend_Page_gput (name,value)
1269   __pdf_backend_Page_gput (name,value)
1270 end
1271
1272 function ltx.__pdf.backend_Page_gremove (name)
1273   __pdf_backend_Page_gremove (name)
1274 end
1275
1276 function ltx.__pdf.backend_Page_gclear ()
1277   __pdf_backend_Page_gclear ()
1278 end
1279
1280
1281 local Properties = ltx.__pdf.Page.Resources.Properties
1282 local ResourceList= ltx.__pdf.Page.Resources.List
1283 local function __pdf_backend_PageResources_gpush (page)
1284   local token=""
1285   if Properties[page] then
1286     -- we sort the table, so that the pdf test works
1287     local t = {}
1288     for name,value in pairs (Properties[page]) do
1289       table.insert (t,name)
1290     end
1291     table.sort (t)
1292     for _,name in ipairs(t) do
1293       token = token .. "/"..name.." ".. Properties[page][name]
1294     end
1295     token = "/Properties <<"..token..">>"
1296   end
1297   for i,name in ipairs(ResourceList) do
1298     if ltx.__pdf.Page.Resources[name] then
1299       token = token .. "/"..name.." "..ltx.pdf.object_ref("Page/Resources/"..name)
1300     end
1301   end
1302   return token
1303 end
1304
1305 -- the function is public, as I probably need it in tagpdf too ...
1306 function ltx.pdf.Page_Resources_Properties_gput (page,name,value) -- tex.count["g_shipout_re
1307   Properties[page] = Properties[page] or {}
1308   Properties[page][name]=value

```

```

1309 pdf.setpageresources(__pdf_backend_PageResources_gpush (page))
1310 end
1311
1312 function ltx.pdf.Page_Resources_gpush(page)
1313 pdf.setpageresources(__pdf_backend_PageResources_gpush (page))
1314 end
1315
1316 function ltx.pdf.object_ref (objname)
1317 if ltx.__pdf.object[objname] then
1318 local ref= ltx.__pdf.object[objname]
1319 return ref
1320 else
1321 return "false"
1322 end
1323 end
1324 </lua>

```

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.

B			
bool commands:		<code>\cs_new_protected:Npn</code>
<code>\bool_if:NTF</code> 594, 627, 651, 685, 707, 746, 794	 6, 87, 96, 112, 118, 124, 131, 138, 147, 174, 197, 207, 221, 233, 250, 261, 268, 275, 284, 293, 300, 307, 314, 323, 332, 340, 343, 349, 354, 357, 393, 405, 412, 443, 447, 458, 462, 463, 467, 471, 472, 476, 510, 526, 605, 695, 813, 837, 844, 851, 861, 865, 889, 902, 918, 922, 989, 1004, 1069, 1092, 1173, 1190, 1191
<code>\bool_new:N</code> 481	<code>\cs_new_protected:Npx</code> 106
<code>\bool_set_true:N</code> 930, 1012, 1105	<code>\cs_set_protected:Npn</code>	.. 490, 494, 498, 502, 506, 516, 518, 520, 522, 524, 537, 556, 575, 581, 587, 592, 599, 622, 646, 670, 674, 679, 683, 690, 700, 739, 779, 783, 788, 792, 799
box commands:			
<code>\box_dp:N</code> 943, 1024, 1116		
<code>\box_ht:N</code> 940, 1021, 1113		
<code>\box_new:N</code> 28, 1102		
<code>\box_set_dp:Nn</code> 1117, 1184		
<code>\box_set_ht:Nn</code> 1118, 1183		
<code>\box_set_wd:Nn</code> 1119, 1182		
<code>\box_use_drop:N</code> 1130, 1185		
<code>\box_wd:N</code> 937, 1018, 1110		
C		D	
clist commands:		dim commands:	
<code>\clist_const:Nn</code> 366	<code>\c_zero_dim</code> 1117, 1118, 1119
<code>\clist_map_function:NN</code> 828	<code>\directlua</code> 36
<code>\clist_map_inline:Nn</code>	377, 414, 438, 607		
cs commands:		E	
<code>\cs_generate_variant:Nn</code>	.. 4, 5, 16, 17	exp commands:	
<code>\cs_gset_eq:NN</code> 595, 596, 686, 687, 795, 796	<code>\exp_args:NNx</code> 815
<code>\cs_if_exist:NTF</code> 380	<code>\exp_args:Nnx</code> 629, 653
<code>\cs_new:Npn</code> 12, 192, 804, 870, 996, 1078, 1168, 1192	<code>\exp_args:Nnxx</code> 709, 733, 748, 772
		<code>\exp_args:Nx</code> 184, 295, 334, 625, 635, 649, 659, 703, 742, 1161

H

hbox commands:

- \hbox_gset:Nn 1103
- \hbox_set:Nn 928, 1010, 1175

hook commands:

- \hook_gput_code:nnn 39, 46, 54, 75, 79, 441
- \hook_gput_next_code:nn 1120
- \hook_gset_rule:nnnn 51, 59, 435, 436
- \hook_use:n .. 42, 49, 57, 67, 71, 77, 81

I

int commands:

- \int_compare:nNnTF 874, 876, 878, 951, 1032
- \int_const:Nn 984, 1064, 1099
- \int_gincr:N ... 150, 539, 558, 624, 648, 702, 706, 741, 745, 892, 904, 1098
- \int_if_exist:NTF 1198
- \int_new:N 31, 32, 33, 869
- \int_use:N . 151, 152, 153, 157, 160, 161, 542, 550, 561, 569, 626, 631, 640, 650, 655, 664, 704, 711, 716, 718, 719, 723, 726, 727, 735, 743, 750, 755, 757, 758, 762, 765, 766, 774, 880, 992, 998, 1071, 1079, 1170

K

kernel internal commands:

- __kernel_backend_literal:n . 23, 540, 544, 559, 563, 577, 589, 601, 611
- __kernel_backend_literal_page:n 625, 649, 672, 681, 692, 703, 742, 781, 790, 801

L

lualua commands:

- \lualua: 144, 230, 281, 320

M

mode commands:

- \mode_leave_vertical: 1122

P

pdf commands:

- \pdf_object_last: 840, 847, 855
- \pdf_object_now:nn 583, 676, 785, 839, 846, 853
- \pdf_object_ref:n 809, 965, 970, 975, 980, 1045, 1050, 1055, 1060, 1136, 1143, 1150, 1158
- \pdf_object_write 451
- \pdf_xform_dp:n 1128, 1184
- \pdf_xform_ht:n 1127, 1183
- \pdf_xform_if_exist:n 1202
- \pdf_xform_wd:n 1126, 1182

pdf internal commands:

- __pdf_backend:n 114, 445, 452, 855, 1123, 1131, 1132, 1139, 1146, 1153, 1162, 1177
- __pdf_backend_bdc:nn 11, 478, 490, 516, 592, 595, 596, 597, 683, 686, 687, 688, 792, 795, 796, 797
- __pdf_backend_bdc_contobj:nn ... 581, 595, 674, 686, 783, 795
- __pdf_backend_bdc_contstream:nn 587, 596, 679, 687, 788, 796
- __pdf_backend_bdc_obj:n 11, 478, 498, 520, 556, 584, 646, 677, 739, 786
- __pdf_backend_bdc_obj:nn 11, 478, 494, 518, 537, 622, 700
- __pdf_backend_bmc:n 11, 478, 506, 524, 575, 670, 779
- __pdf_backend_catalog_gput:nn .. 18
- \g_pdf_backend_EmbeddedFiles_-int 868, 869, 874, 876, 878, 880, 892, 904
- __pdf_backend_EmbeddedFiles_-name: .. 868, 870, 895, 897, 907, 910
- \g_pdf_backend_EmbeddedFiles_-named_prop 884, 886, 893, 905
- \g_pdf_backend_EmbeddedFiles_-seq 884, 885, 896
- __pdf_backend_emc: 11, 478, 502, 522, 599, 690, 799
- __pdf_backend_luastring:n 100, 192, 201, 213, 214, 225, 240, 241
- \g_pdf_backend_name_int 30, 539, 542, 550, 558, 561, 569, 624, 626, 631, 640, 648, 650, 655, 664, 702, 704, 741, 743
- __pdf_backend_NamesEmbeddedFiles_-add:n 888, 889, 902, 918
- __pdf_backend_NamesEmbeddedFiles_-gpush:n 837, 844, 851, 861, 865
- \g_pdf_backend_object_int 1098, 1101
- __pdf_backend_object_last: 500, 570, 656, 665, 751, 775
- __pdf_backend_object_new:nn 379, 440
- __pdf_backend_object_ref:n 386, 454, 496, 551, 614, 632, 641, 712, 736
- __pdf_backend_object_write:nn .. 419, 441
- __pdf_backend_Page_gput:nn 4, 121, 131, 207, 268, 307, 343
- __pdf_backend_Page_gremove:n ... 4, 121, 138, 221, 275, 314, 349
- \g_pdf_backend_page_int 30

__pdf_backend_Page_primitive:n .	pdfdict commands:
..... 4, 121, 124, 184,	\pdfdict_gput:nnn
197, 261, 286, 295, 300, 325, 334, 340	. 133, 168, 270, 309, 345, 395, 407,
__pdf_backend_PageResources:n ..	465, 474, 629, 653, 709, 733, 748, 772
..... 443, 462, 471	\pdfdict_gremove:nn 140, 277, 316, 351
\c__pdf_backend_PageResources_-	\pdfdict_if_exist:nTF . 163, 729, 768
clist .. 366, 377, 414, 438, 607, 829	\pdfdict_item:nn 186, 809, 824
__pdf_backend_PageResources_-	\pdfdict_name:n
gpush:n 176, 177, 179, 417, 450, 609, 807,
..... 11, 478, 510, 526, 605, 695, 813	818, 823, 931, 952, 963, 968, 973,
__pdf_backend_PageResources_-	978, 1013, 1033, 1043, 1048, 1053, 1058
gpush_aux:n 804, 830	\pdfdict_new:n 165, 731, 770
__pdf_backend_PageResources_-	\pdfdict_show:n 776
gput:nnn 360, 393, 405, 447, 463, 472	\pdfdict_use:n 296, 335, 421, 958, 1039
__pdf_backend_PageResources_-	pdfmanagement internal commands:
obj_gpush: . 360, 412, 458, 467, 476	\g__pdfmanagement_active_bool ...
__pdf_backend_Pages_primitive:n 594, 685, 794
..... 86, 87, 96, 106, 112, 118	\pdfnames 18
__pdf_backend_pdfmark:n	\pdfpageref 2
..... 492, 496, 500, 504, 508, 908	prg commands:
__pdf_backend_ref_label:nn	\prg_new_conditional:Npnn 1196
..... 6, 16, 153, 719, 758	\prg_new_eq_conditional:NNn .. 1202
__pdf_backend_ref_value:nn	\prg_return_false: 1200
..... 12, 17, 161, 727, 766	\prg_return_true: 1199
\g__pdf_backend_resourceid_int ..	prop commands:
..... 30, 150, 151, 152,	\prop_count:N 952, 1033
153, 157, 160, 161, 706, 711, 716,	\prop_gclear:N 931, 1013
718, 719, 723, 726, 727, 735, 745,	\prop_gput:Nnn 181, 450, 893, 905
750, 755, 757, 758, 762, 765, 766, 774	\prop_gset_eq:NN 176
__pdf_backend_ThisPage_gpush:n .	\prop_if_empty:NTF
..... 4, 121, 174, 250, 293, 332, 357 416, 609, 806, 962,
__pdf_backend_ThisPage_gput:nn .	967, 972, 977, 1042, 1047, 1052, 1057
..... 4, 121, 147, 233, 284, 323, 354	\prop_if_exist:NTF 177, 817
\g__pdf_backend_thispage_-	\prop_map_function:NN 186, 822
shipout_tl 4	\prop_map_inline:Nn 179
\l__pdf_backend_tmpa_box	\prop_new:N 26, 886
..... 25, 928, 937, 940,	
943, 983, 1010, 1018, 1021, 1024,	
1063, 1175, 1182, 1183, 1184, 1185	
\l__pdf_backend_xform_bool . 481,	
627, 651, 707, 746, 930, 1012, 1105	
__pdf_backend_xform_if_exist:n .	
..... 1196, 1202	
__pdf_backend_xform_new:nnnn ...	
..... 921, 922, 1004, 1092, 1190	
__pdf_backend_xform_ref:n	
921, 996, 1078, 1125, 1168, 1179, 1192	
__pdf_backend_xform_use:n	
..... 921, 989, 1069, 1173, 1191	
\g__pdf_tmpa_prop ... 25, 176, 181, 186	
\l__pdf_tmpa_tl	
..... 25, 154, 163, 165, 168, 720,	
729, 731, 734, 759, 768, 770, 773, 776	

	R
	ref commands:
	\ref_label:nn 4, 9, 152, 718, 757
	\ref_value:nn 5, 14, 160, 726, 765
	\relax 64
	\RequirePackage 3
	S
	scan commands:
	\scan_stop: 993, 1075
	seq commands:
	\seq_gput_right:Nn 896
	\seq_new:N 885
	\special 427, 429
	sys commands:
	\sys_if_engine luatex:TF 94

T

\TeX and \LaTeX 2 ϵ commands:

$\backslash\text{bsphack}$	8
$\backslash\text{esphack}$	10
$\backslash\text{@kernel@after@shipout@background}$	62, 65
$\backslash\text{@kernel@after@shipout@lastpage}$..	69
$\backslash\text{@kernel@before@shipout@background}$	64
$\backslash\text{g@addto@macro}$	64, 65, 69
$\backslash\text{zref@extractdefault}$..	156, 722, 761
$\backslash\text{zref@labelbylist}$	151, 715, 754

tex commands:

$\backslash\text{tex_directlua:D}$	98, 209, 223, 380, 382
$\backslash\text{tex_global:D}$	89, 126, 815
$\backslash\text{tex_immediate:D}$	945, 1026
$\backslash\text{tex_latelua:D}$	
.....	199, 235, 252, 397, 398, 635, 659
$\backslash\text{tex_luaescapestring:D}$	194
$\backslash\text{tex_pdfextension:D}$	847
$\backslash\text{tex_pdflastxform:D}$	986, 1066
$\backslash\text{tex_pdfnames:D}$	840

$\backslash\text{tex_pdfpageattr:D}$	126
$\backslash\text{tex_pdfpageresources:D}$	815
$\backslash\text{tex_pdfpagesattr:D}$	89
$\backslash\text{tex_pdfrefxform:D}$	991, 1071
$\backslash\text{tex_pdfxform:D}$	945, 1026
$\backslash\text{tex_special:D}$	108, 263, 302
$\backslash\text{tex_the:D}$	937, 940,
.....	943, 1018, 1021, 1024, 1110, 1113, 1116
$\backslash\text{tex_unexpanded:D}$	194

tl commands:

$\backslash\text{c_space_tl}$	542, 550, 561, 569, 626,
.....	650, 704, 743, 897, 1126, 1127, 1128
$\backslash\text{tl_const:Nn}$	935, 938,
.....	941, 1016, 1019, 1022, 1108, 1111, 1114
$\backslash\text{tl_if_exist:NTF}$	62
$\backslash\text{tl_new:N}$	27
$\backslash\text{tl_set:Nn}$	154, 720, 759
$\backslash\text{tl_to_str:n}$	
.....	936, 939, 942, 985, 992, 998,
.....	1017, 1020, 1023, 1065, 1073, 1079,
.....	1100, 1109, 1112, 1115, 1170, 1198