The **l3pdftools** module
temporary collection of pdf related commands
LaTeX PDF management testphase bundle

The LaTeX Project*

Version 0.96d, released 2023-12-09

1 **l3pdftools** documentation

This module collects a number of candidate commands for the l3pdf module

---

`\pdf_name_from_unicode_e:n` ⋆

`\pdf_name_from_unicode_e:n  {⟨content⟩}`

This converts ⟨*content*⟩ to a format suitable for a PDF Name. The output depends on the backend: For almost all backends it will first expand the content with `\text_expand:n` and then escape it in the way needed in a PDF Name with `\str_convert_pdfname:e`, and at last prepend a slash before. Typically such names use only ascii, but non-ascii is supported, but should be utf8 encoded. For example
`\pdf_name_from_unicode_e:n␣{A~B\c_percent_str␣C\c_hash_str␣D€}}`
will output `/A#20B#25C#23D#E2#82#AC`.

   With dvips it will expand the content with `\text_expand:n` and then wrap it in a `cvn` operation ("convert to name"). So the example above will output `(A␣B%C#D€)␣cvn` to the postscript. The content should not contain unbalanced parentheses with dvips.

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

\pdf_string_from_unicode:nnN   \pdf_string_from_unicode:nnN {⟨*format*⟩} {⟨*content*⟩} {⟨*tlvar*⟩}

This converts ⟨*content*⟩ following the rules defined by ⟨*format*⟩ and stores the result in ⟨*tlvar*⟩. The assignment is done locally. Non-ascii input should be utf8 encoded. Currently the following formats exist:

utf8/string-raw this converts with \str_set_convert:Nnnn into utf8/string.

utf8/string this converts into utf8/string and adds parentheses around the result.

utf8/URI-raw this converts with \str_set_convert:Nnnn into utf8/url and then replaces reserved and digits back from the percent encoding. Parentheses are escaped.

utf8/URI this converts into utf8/URI and adds parentheses around the result.

utf16/string-raw this converts with \str_set_convert:Nnnn into utf16/string.

utf16/string this converts into utf16/string and adds parentheses around the result.

utf16/hex-raw this converts into utf16/hex

utf16/hex this converts into utf16/hex and adds bracket around the result.

1.1   BDC operator / Properties resource

Entries to the /Properties dictionary in the page resources can be added with dvips only through side-effects: if a BDC-mark is created dvips/ghostscript will automatically create the necessary objects and names. To get a sensible abstraction the code does the same for the other backends if the core management code has been activated. This means that the behaviour of the command is different then. The \pdf_bdcobject:.. should only be used if the management is active.

\pdf_bdc:nn
\pdf_bdc:ee

\pdf_bdc:nn {⟨*tag*⟩} {⟨*dictionary content*⟩}

This command adds a BDC marked content operator to the current page stream. ⟨*tag*⟩ is the tag of this operator (without the leading slash), ⟨*dictionary content*⟩ is the content of the second argument. If the PDF resource management is active an dictionary object with the content is created and referenced with a name in the BDC operator. Without the resource management the content is used directly. It then depends on the backend how it is handled: with dvips a name is used while the pdfmode engines and dvipdfmx write the content into the stream.

\pdf_bdc_shipout:ee

New: 2023-08-18

`\pdf_bdc_shipout:ee {⟨tag⟩} {⟨dictionary content⟩}`

This command adds a BDC marked content operator to the current page stream. ⟨*tag*⟩ is the tag of this operator (without the leading slash), ⟨*dictionary content*⟩ is the content of the second argument.

Differently to `\pdf_bdc:ee` the arguments are not expanded when the command is *used*, but only at *shipout*. This requires new engines which allow to use the keyword `shipout` with the primitive `\special` and `\pdfliteral`. The content of ⟨*dictionary content*⟩ is added in the stream.

The command requires current engines and is not defined if an to old engine is detected!

\pdf_bdcobject:nn

New: 2020-07-03

`\pdf_bdcobject:nn {⟨tag⟩} {⟨object name⟩}`

This command adds a BDC marked content operator to the current page stream. ⟨*tag*⟩ is the tag of this operator (without the leading slash), ⟨*object name*⟩ is a the name of an dictionary object reserved with `\pdf_object_new:n` and filled with `\pdf_object_-write:nnn` with the properties of the BDC. Reusing a predefined object can save space but the command works correctly only if the resources management has been activated and should be used only if this can be ensured.

\pdf_bdcobject:n

Updated: 2020-07-03

`\pdf_bdcobject:n {⟨tag⟩}`

This command adds a BDC marked content operator to the current page stream. ⟨*tag*⟩ is the tag of this operator (without the leading slash). As object this commands uses the last anonymous dictionary object created with `\pdf_object_unnamed_write:nn`. It lies in the responsibility of the user that the last object is the wanted one. Like with `\pdf_bdcobject:nn` the command works correctly only if the resources management has been activated and should be used only if this can be ensured.

\pdf_bmc:n

New: 2019-10-17

`\pdf_bmc:n {⟨tag⟩}`

This command created a BMC marked content operator. The argument is the tag without the leading slash. It can be e.g. used for simple artifact markers.

\pdf_emc:

New: 2019-06-30

`\pdf_emc:`

This command closes the BDC marked content operator opened with `\pdf_bdc:nn`. It should be on the same page as the bdc-command.

```
\pdf_object_new:n      {module/objA}
\pdf_object_write:nnn {module/objA}{dict}{/Type/Artifact}
\pdf_bdc:nn {Span}{module/objA}
text
\pdf_emc:
```

2 l3pdftools implementation

```
1 ⟨∗header⟩
2 \ProvidesExplPackage{l3pdftools}{2023-12-09}{0.96d}
3   {candidate commands for l3pdf---LaTeX PDF management testphase bundle}
```

3

2.1 Conversions and export functions

`\pdf_name_from_unicode_e:n`
`\pdf_name_from_unicode_e:V`

```
7 \cs_generate_variant:Nn \str_convert_pdfname:n { e }
8
9 \cs_new:Npn \pdf_name_from_unicode_e:n #1
10   {
11     \__kernel_pdf_name_from_unicode_e:n { #1 }
12   }
13
14 \cs_generate_variant:Nn \pdf_name_from_unicode_e:n {V}
```

(*End of definition for* `\pdf_name_from_unicode_e:n`*. This function is documented on page 1.*)

The convert command must use a different value the source encoding depending on the engines. Until the PR in str-convert is active we add the alias here too

```
15 \bool_lazy_any:nTF
16   {
17     \sys_if_engine_luatex_p:
18     \sys_if_engine_xetex_p:
19   }
20   {
21     \prop_gput:Nnn \g__str_alias_prop { default } {  }
22   }
23   {
24     \prop_gput:Nnn \g__str_alias_prop { default } { utf8 }
25   }
```

`\pdf_string_from_unicode:nnN`

```
26 \msg_new:nnn {pdfmanagement}{ unknown-convert}
27   {
28     Unknown~string~conversion~method~'#1'!
29   }
30 \cs_new:Npn \pdf_string_from_unicode:nnN #1 #2 #3
31   {
32     \cs_if_exist_use:cF { __pdf_string_from_unicode_#1:nN }
33       {
34         \msg_error:nnn { pdf } { unknown-convert } {#1}
35         \use_none:nn
36       }
37     { #2 } #3
38   }
39
40 \cs_generate_variant:Nn \pdf_string_from_unicode:nnN {nVN}
```

(*End of definition for* `\pdf_string_from_unicode:nnN`*. This function is documented on page 2.*)

Most converter are simply wrapper around the str-convert commands and so use the same names, with the addition raw if no delimiters are added. The exception is the one for url's: it reverts most of the percent encodings and escapes the parentheses. That's why its name is URI instead of url. The current code is probably quite slow and will need a replacement.

4

```
41  %% TODO Names need a review when it is clear which converters
42  %% are actually needed
43  %% string conversions and printing
44  %% we assume here that the text purify step has been done. The input is
45  %% a list of (utf8) chars.
46  %% str convert, not expandable.
47  %  filespec (attachment view)  tests:
48  %  utf8:  gr\303\274\303\237e.txt
49  %    %doesn't work, umlaut wrong,
50  %  utf8 with BOM \357\273\277gr\303\274\303\237e.txt
51  %    %doesn't work, umlaut wrong, bom visible
52  %  utf16 with BE: (FEFF)
53  %  \376\377\000g\000r\000\374\000\337\000e\000.\000t\000x\000t %works
54  %              xetex converts to <feff0067007200fc00df0065002e007400780074>
55  %  utf16 with BE / HEX: <FEFF0067007200FC00DF0065002E007400780074> works
56
57  % bookmarks: as pdfoutline uses () currently only utf16 with BE is usable.
58  % check if one can use HEX too when directly writing the object
59  % ==========
60  % uri: utf16BE/string seems not to work, hex neither
61  %      utf8/string works but not on macos,
62  %      so a specfic utf8/url variant is needed
63  % ==========
64  % "input" is utf8 for pdftex, empty (native) for unicode engine
65  % commands to output literal strings  (...)
66
67  \cs_new_protected:cpn { __pdf_string_from_unicode_utf8/string-raw:nN }  #1 #2
68    {
69      \str_set_convert:Nnnn #2
70        { #1 }
71        { default }
72        {utf8/string}
73    }
74
75  \cs_new_protected:cpn { __pdf_string_from_unicode_utf8/string:nN } #1 #2
76    {
77      \use:c { __pdf_string_from_unicode_utf8/string-raw:nN } { #1 } #2
78      \str_put_left:Nn #2  {(}
79      \str_put_right:Nn #2 {)}
80    }
81  % special url command:
82  \cs_new_protected:cpx { __pdf_string_from_unicode_utf8/URI-raw:nN }  #1 #2
83    {
84      \exp_not:N \str_set_convert:Nnnn #2
85        { #1 }
86        { default }
87        {utf8/url}
88      \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 3A} {:}
89      \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 2F} {/}
90      \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 23} {\c_hash_str}
91      \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 5B} {[}
92      \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 5D} {]}
93      \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 40} {\c_atsign_str}
```

```
94    \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 21} {!}
95    \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 24} {\c_dollar_str}
96    \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 26} {\c_ampersand_str}
97    \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 27} {'}
98    \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 2A} {*}
99    \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 2B} {+}
100   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 2C} {,}
101   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 3B} {;}
102   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 3D} {=}
103   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 3F} {?}
104   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 30} {0}
105   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 31} {1}
106   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 32} {2}
107   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 33} {3}
108   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 34} {4}
109   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 35} {5}
110   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 36} {6}
111   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 37} {7}
112   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 38} {8}
113   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 39} {9}
114   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 28} {\c_backslash_str(}
115   \exp_not:N \str_replace_all:Nnn #2 {\c_percent_str 29} {\c_backslash_str)}
116   }
117
118 \cs_new_protected:cpn { __pdf_string_from_unicode_utf8/URI:nN } #1 #2
119   {
120     \use:c { __pdf_string_from_unicode_utf8/URI-raw:nN } {#1} #2
121     \str_put_left:Nn #2  {(}
122     \str_put_right:Nn #2 {)}
123   }
124 % with utf16 with BE marker
125 \cs_new_protected:cpn { __pdf_string_from_unicode_utf16/string-raw:nN }  #1 #2
126   {
127     \str_set_convert:Nnnn #2
128       { #1 }
129       { default }
130       {utf16/string}
131   }
132
133 \cs_new_protected:cpn { __pdf_string_from_unicode_utf16/string:nN } #1 #2
134   {
135     \use:c { __pdf_string_from_unicode_utf16/string-raw:nN } {#1} #2
136     \str_put_left:Nn #2  {(}
137     \str_put_right:Nn #2 {)}
138   }
139
140 \cs_new_protected:cpn { __pdf_string_from_unicode_utf16/hex-raw:nN } #1 #2
141   {
142     \str_set_convert:Nnnn #2
143       { #1 }
144       { default }
145       {utf16/hex}
146   }
147
```

```
148 \cs_new_protected:cpn { __pdf_string_from_unicode_utf16/hex:nN } #1 #2
149   {
150     \use:c { __pdf_string_from_unicode_utf16/hex-raw:nN } {#1} #2
151     \str_put_left:Nn #2  {<}
152     \str_put_right:Nn #2 {>}
153   }
154
```

(*End of definition for* `__pdf_string_from_unicode_utf8/string-raw:nN` *and others.*)

2.1.1 BDC operator commands

```
155 \cs_new_protected:Npn  \pdf_bdc:nn #1 #2 { \__pdf_backend_bdc:nn { #1 }{ #2 } }
156 \cs_generate_variant:Nn \pdf_bdc:nn {ee}
157
158 \cs_new_protected:Npn  \pdf_bdc_shipout:ee #1 #2
159   {
160     \bool_if:NTF\l__pdfmanagement_delayed_shipout_bool
161       {
162         \__pdf_backend_bdc_shipout:ee { #1 }{ #2 }
163         \cs_gset_eq:NN \pdf_bdc_shipout:ee \__pdf_backend_bdc_shipout:ee
164       }
165       {
166         \msg_error:nn {pdfmanagement}{delayed-shipout}
167         \cs_gset_eq:NN \pdf_bdc_shipout:ee  \use_none:nn
168       }
169   }
170 \cs_new_protected:Npn  \pdf_bdcobject:nn #1 #2 { \__pdf_backend_bdcobject:nn { #1 }{ #2 } }
171 \cs_new_protected:Npn  \pdf_bdcobject:n #1    { \__pdf_backend_bdcobject:n  { #1 } }
172 \cs_new_protected:Npn  \pdf_bmc:n #1    { \__pdf_backend_bmc:n  { #1 } }
173 \cs_new_protected:Npn  \pdf_emc:          { \__pdf_backend_emc: }
```

(*End of definition for* `\pdf_bdc:nn` *and others. These functions are documented on page 2.*)

```
174 ⟨/package⟩
```

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.