The `documentmetadata-support` code[*]

Frank Mittelbach, Ulrike Fischer, LaTeX Project

October 31, 2024

1 Introduction

The kernel command `\DocumentMetadata`, which can be used as the very first declaration in a document (i.e., before `\documentclass`), defines metadata and other configuration data that applies to the document as a whole (using a key/value syntax).

While the underlying functionality is still under development (e.g., further keys will be added over time and keys marked temporary may vanish again) the code for `\DocumentMetadata` is placed in a separate bundle, so that it is easier to update it without the need to build a full LaTeX release. Over time the functionality will move fully into the kernel.

`\DocumentMetadata` also loads and activates the new PDF management code from pdfmanagement-testphase. As this forces the loading of the l3backend files, a backend that can't be detected automatically like `dvipdfmx`, must be set in the first `\DocumentMetadata`.

From a process perspective `\DocumentMetadata` loads the pdfmanagement-testphase code the first time it is called and then redefines itself to only manage key/value pairs in case it is called more than once. In particular, this means that a document without a `\DocumentMetadata` declaration has no code available for extended management of PDF output as needed for various features developed as part of the multi-year "Tagged PDF" project [1].

2 The `\DocumentMetadata` command

`\DocumentMetadata` `\DocumentMetadata{⟨key-value list⟩}`

The command should be used as the first command in a document, before `\documentclass`. It takes a key-value argument.

3 Currently supported key/values

Currently the following keys are implemented for `\DocumentMetadata`:

backend Passes the backend name to expl3. This is needed only if the needed backend can't be automatically determined or if the workflow used requires a special backend.

---

[*]This file has version 1.0j dated 2024-09-13, © LaTeX Project.

pdfversion Sets the PDF version explicitly, e.g., `pdfversion=1.7`

uncompress (no value) Forces an uncompressed pdf — mainly for debugging purposes.

lang Explicitly sets the Lang entry in the Catalog, e.g., `lang=de-DE`. If not given the default value used is `en-US`.

pdfstandard Choice key to set the pdf standard. Currently `A-1b`, `A-2a`, `A-2b`, `A-2u`, `A-3a`, `A-3b`, `A-3u`, `A-4`, `A-4E` and `A-4F` are accepted as values. The casing is irrelevant, `a-1b` works too. Note that using these key doesn't mean that the document actually follows the standard. LaTeX can neither ensure nor check all requirements of a standard, and not everything it can do theoretically has already been implemented. When setting an A-standard a color profile is included and the `/OutputIntent` is set and javascript action in hyperref are suppressed. The `u` variants do not enforce unicode, but they will pass the information to hyperref. The `a` variants do *not* enforce (or even test) a tagged pdf yet.

Starting with version 0.95s of `pdfmanagement-testphase` it is also possible to use the values `X-4`, `X-4p`, `X-5g`, `X-5n`, `X-5pg`, `X-6`, `X-6n`, `X-6p`, `UA-1` for a PDF/X and PDF/UA standard. These keys currently set *only* the relevant XMP-metadata. In version 0.95z support for `UA-2` has been added but note that `UA-2` hasn't been released yet. It should be used only together with pdf version 2.0.

`pdfstandard` can be used more than once to set overlapping standards, e.g:
`pdfstandard=A-2b,pdfstandard=X-4,pdfstandard=UA-1`

If XMP-metadata are added (see the following key `xmp`) the needed conformance marker for the standards are set.

More information can be found in the documentation of l3pdfmeta.

xmp A boolean, if set to false no XMP metadata are added to the PDF. The initial value is true. Details are described in the documentation of l3pdfmeta.

colorprofiles This allows to load icc-colorprofiles. Details are described in the documentation of l3pdfmeta.

testphase This key is used to load testphase code. The `testphase` key can only be used in the first `\DocumentMetadata`. The values it accepts and their effect will change over time, when testphase packages are added or removed or when the code is moved into the kernel. The key accepts a list of values and it can be used more than once.

The `phase` key bundle testphase modules. They also all activate tagging.

phase-I This value loads code implementing the first phase of the project [1], i.e., it will load the tagpdf package. It will also activate tagging by issuing `\tagpdfsetup{activate,activate/spaces}`. This phase is frozen.

phase-II It differs from `phase-I` only in one point: It will additionally activate tagging of paragraphs with `\tagpdfsetup{para/tagging}`. In the upcoming months it will also enable automatic tagging of other basic document elements.

phase-III This is the current development phase. It differs from `phase-II` *a lot*: It will load new code for the tagging of lists, sectioning commands, table of contents and similar lists, graphics, minipages and floats. As it redefines many internals it is currently restricted to the use of standard classes (article, report, and book) and it supports only a limited number of add-on packages.

The various testphase modules can also be loaded individually (at least in theory, there can be hidden dependencies). If loaded like this, the tagpdf package is not loaded and tagging is not activated! The list of modules will change over time.

new-or-1 This patches a few commands related to the output routine. The patches are needed for the tagging of paragraphs, for the tagging of header and footer and to allow the PDF management to insert code which avoids that links happening at page breaks spills into the header and footer. This code is automatically loaded if the `testphase` values `phase-I`, `phase-II` or `new-or` are used.

new-or This loads more changes to the output routine required for the tagging. It is not compatible with every class! The code is also loaded by the `phase-II` value.

sec This adapts commands related to sectioning to make them tagging aware. The `sec` module is loaded by `phase-III`.

toc This adapts commands related to the table of contents and similar list to make them tagging aware. The `toc` module is loaded by `phase-III`.

graphic This enables tagging support for the `\includegraphics` command and the `picture` environment. This code is also loaded by the `phase-III` key.

block This reimplements lists and blocks environments and add tagging support. This code is also loaded by the `phase-III` key.

minipage This adds tagging support to `minipage` and `\parbox`. This code is also loaded by the `phase-III` key.

float This adds tagging support to floats. This code is also loaded by the `phase-III` key.

bib This adds tagging support to citations and bibliographies. This code is also loaded by the `phase-III` key.

text This module adds tagging support to the LaTeX logo and to the `\emph` command. This code is also loaded by the `phase-III` key.

marginpar This module adds tagging support to the `\marginpar` command. This code is also loaded by the `phase-III` key.

title This module add tagging support to the `\maketitle` command if a standard class is used. It also enhances the `\title` and `\author` commands to fill the XMP-metadata and set the window title. It is not compatible with packages and classes which redefine these commands too. The module is currently not loaded by any `phase` key.

math This adapts math for tagging. This is only a prototype. The module is currently not loaded by any `phase` key.

table This provides basic tagging for `tabular`, `longtable` and similar table environments. The module is currently not loaded by any `phase` key. Its use and restrictions is documented in `latex-lab-table.pdf`.

firstaid This contains small adjustments to external packages. The module is currently not loaded by any `phase` key.

debug This key activates some debug options. It takes a list of key-values as value. Currently the following keys are known:

para with the default and only value `show`. It will activate the `paratagging-show` option of `tagpdf`,

log with the values as described in the documentation `tagpdf`,

uncompress which does the same as `uncompress` as main key

pdfmanagement a boolean which allows to deactivate the pdfmanagement.

firstaidoff This accepts a comma lists of keywords and disables the patches related to them. More information can be found in the documentation of pdfmanagement-firstaid.

xmp-export This will export the XMP-metadata to a file `\jobname.xmpi`. with `debug={xmp-export=filename}` the file name can be changed. More information can be found in the documentation of l3pdfmeta of the pdfmanagement-testphase bundle.

tagpdf This loads the package tagpdf-debug which enhances various commands from tagpdf with additional debugging options. This can slow down the compilation!

References

[1] Frank Mittelbach and Chris Rowley: *LATEX Tagged PDF — A blueprint for a large project.* https://latex-project.org/publications/indexbyyear/2020/

4 The Implementation

```
1 ⟨@@=pdfmanagement⟩
2 ⟨*code⟩

3
4 \RequirePackage{pdfmanagement-testphase}
5 \ExplSyntaxOn\makeatletter
```

4.1 Variables

These variable definitions are currently also done in ltdocinit. They can be removed from there once latex-lab has been updated to provide them too.

\g__pdfmanagement_firstaidoff_clist A list to store the firstaid code which should be disabled

```
6 \clist_if_exist:NF \g__pdfmanagement_firstaidoff_clist
7   { \clist_new:N \g__pdfmanagement_firstaidoff_clist }
```

(*End of definition for* `\g__pdfmanagement_firstaidoff_clist`.)

\g__pdfmanagement_testphase_tl a tl to store the testphase loading code so that we can load them at the end of the command.

```
8 \tl_if_exist:NF \g__pdfmanagement_testphase_tl
9   { \tl_new:N \g__pdfmanagement_testphase_tl }
```

(*End of definition for* `\g__pdfmanagement_testphase_tl`.)

4.2 Kernel changes with \DocumentMetadata

```
10 \cs_new_protected:Npn \@kernel@before@DocumentMetadata
11   {
12     \ifx\Umathcode\@undefined
13     \fontencoding{T1}
14     \renewcommand\encodingdefault{T1}
15     \fi
16   }
```

(*End of definition for* \@kernel@before@DocumentMetadata. *This function is documented on page* **??**.)

4.3 \DocumentMetadata

\DocumentMetadata \DocumentMetadata should not be used after \documentclass so we error in this case. It can be used more than once but follow-up calls should not do the initialization code.

```
17 \cs_set_protected:Npn \DocumentMetadata #1
18   {
19     \cs_if_eq:NNTF \documentclass \@twoclasseserror
20       { \msg_error:nn { meta } { after-class } }
21       {
```

Load general format changes

```
22         \@kernel@before@DocumentMetadata
```

The wanted backend must be detected first, we read the init key and then force the loading of the backend. The backend can contain management commands, so the boolean should be set to true first.

```
23         \bool_gset_true:N \g__pdfmanagement_active_bool
24         \keys_set_groups:nnn { document / metadata} {init}{ #1 }
25         %if no backend has been loaded force it now:
26         \str_if_exist:NF \c_sys_backend_str
27           {
28             \sys_load_backend:n {}
29           }
```

Now we load the extra backend code:

```
30         \ExplSyntaxOn\makeatletter
31           \file_input:n {l3backend-testphase-\c_sys_backend_str.def}
32         \ExplSyntaxOff\makeatother
```

Process the init keys and setup the generic driver.

```
33         \keys_set_filter:nnn  { document / metadata } { init } { #1 }
34         \bool_if:NT \g__pdfmanagement_active_bool
35           {
36             \PassOptionsToPackage{customdriver=hgeneric-testphase}{hyperref}
```

Finally we setup the language default. This is done after the begindocument hook so that it can pick up settings from babel. If the Catalog dictionary already contains a lang value we do nothing, otherwise we use the value stored in \BCPdata, either the main language (if its exists) or the fall back language. Note: if babel is loaded without a language this gives the language **und**.

```
37             \g@addto@macro\@kernel@after@begindocument
38               {
```

```
39              \pdfdict_get:nnN {g__pdf_Core/Catalog}{Lang}\l__pdfmanagement_tmpa_tl
40              \quark_if_no_value:NT\l__pdfmanagement_tmpa_tl
41               {
42                 \tl_if_empty:eTF { \BCPdata{main.language} }
43                  { \tl_set:Ne \l__pdfmanagement_tmpb_tl { \BCPdata{language} } }
44                  { \tl_set:Ne \l__pdfmanagement_tmpb_tl { \BCPdata{main.language} } }
45                 \msg_warning:nne { meta } { lang-missing }{ \l__pdfmanagement_tmpb_tl }
46                 \exp_last_unbraced:Ne
47                  \AddToDocumentProperties{[document]{lang}{\l__pdfmanagement_tmpb_tl}}
48                 \pdfmanagement_add:nne {Catalog} {Lang}{(\l__pdfmanagement_tmpb_tl)}
49               }
50            }
51         }
```

`\pdfmanagement_add:nnn` has collected values in this hook.

```
52         \hook_use_once:n {pdfmanagement/add}
```

Now we redefine `\DocumentMetadata` so that it only process the keys on any further calls.

We need to update the hyperref option if the active status changes.

```
53         \cs_set_protected:Npn \DocumentMetadata ##1
54           {
55             \keys_set_filter:nnn  { document / metadata } { init } { ##1 }
56             \str_remove_all:cn {opt@hyperref.sty}{customdriver=hgeneric-testphase}
57             \bool_if:NT \g__pdfmanagement_active_bool
58              {
59                \PassOptionsToPackage{customdriver=hgeneric-testphase}{hyperref}
60              }
61           }
```

Load more modules, the testphase code and the firstaid code. The code is only loaded in the first `\DocumentMetadata` call!

```
62         \g__pdfmanagement_testphase_tl
63         \RequirePackage{pdfmanagement-firstaid}
64       }
65   }
```

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

4.4   `\DocumentMetadata` keys

```
66 \keys_define:nn { document / metadata }
67   {
68     backend .choices:nn =
69       { dvipdfmx , dvips , dvisvgm , luatex , pdftex , pdfmode , xdvipdfmx , xetex }
70       {
71         \sys_load_backend:n {#1}
72       },
73     backend .groups:n = { init } ,
74   }
75
76 \keys_define:nn { document / metadata }
77   {
78     ,pdfversion .code:n =
79       {
80         \pdf_version_gset:n { #1 }
```

```
81        \AddToDocumentProperties[document]{pdfversion}{#1}
82      }
83    ,uncompress .code:n =
84      {
85        \pdf_uncompress:
86      }
87    ,uncompress .value_forbidden:n = true
88    ,lang .code:n =
89      {
90        \pdfmanagement_add:nnn {Catalog} {Lang}{(#1)}
91        \AddToDocumentProperties[document]{lang}{#1}
92      }
93  %,xmpmeta .bool_gset:N = \g_pdfmeta_xmp_bool %see pdfmeta unused and undefined for now!
94  % this uses internal command from pdfmeta, it should probably move there ...
95    ,pdfstandard .code:n =
96      {
97        \exp_args:Nne
98        \keys_set:nn {document / metadata} {_pdfstandard=\str_uppercase:n{#1}}
99      }
100   ,_pdfstandard .choices:nn =
101     {A-1B,A-2A,A-2B,A-2U,A-3A,A-3B,A-3U,A-4}
102     {
103       \prop_if_exist:cT { g__pdfmeta_standard_pdf/#1_prop }
104         {
105           \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/#1 _prop }
106         }
107       \AddToDocumentProperties [document]{pdfstandard}{#1}
108     }
109   ,_pdfstandard / A-4F .code:n =
110     {
111       \prop_if_exist:cTF { g__pdfmeta_standard_pdf/A-4F_prop }
112         {
113           \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4F_prop }
114         }
115         {
116           \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4_prop }
117           \prop_gput:Nnn   \g__pdfmeta_standard_prop{conformance}{F}
118         }
119       \AddToDocumentProperties [document]{pdfstandard}{A-4F}
120     }
121   ,_pdfstandard / A-4E .code:n =
122     {
123       \prop_if_exist:cTF { g__pdfmeta_standard_pdf/A-4E_prop }
124         {
125           \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4E_prop }
126         }
127         {
128           \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4_prop }
129           \prop_gput:Nnn   \g__pdfmeta_standard_prop{conformance}{E}
130         }
131       \AddToDocumentProperties [document]{pdfstandard}{A-4E}
132     }
133   ,_pdfstandard / unknown .code:n =
134     {
```

```
135        \msg_error:nnn{pdf}{unknown-standard}{#1}
136      }
137    ,testphase .multichoice:
138    ,testphase / tagpdf .code:n =
139      {
140        \tl_gput_right:Nn\g__pdfmanagement_testphase_tl
141          {
142            \file_if_exist_input:nF {tagpdf-latex-lab-testphase.ltx}
143              {
144                 \RequirePackage{tagpdf}
145                 \AddToDocumentProperties [document]{testphase/tagpdf}{loaded}
146                 \tagpdfsetup{activate,para/tagging,activate/spaces}
147                 \AddToDocumentProperties [document]{tagging}{active}
148                 \AddToDocumentProperties [document]{tagging/para}{active}
149                 \AddToDocumentProperties [document]{tagging/interwordspace}{active}
150              }
151          }
152      }
153    ,testphase / unknown .code:n =
154      {
155        \tl_gput_right:Nn\g__pdfmanagement_testphase_tl
156          {
157             \AddToDocumentProperties [document]{testphase/#1}{loaded}
158             \file_if_exist_input:nF {#1-latex-lab-testphase.ltx}
159              {
160                 \msg_warning:nnn{meta}{latex-lab-pkg-missing}{#1}
161                 \AddToDocumentProperties [document]{testphase/#1}{missing}
162              }
163          }
164      }
165    ,activate .multichoice:
166    ,activate / tagging .code:n =
167      {
168        \PackageWarning{pdfmanagement-testphase}
169         {The~activate~key~is~deprecated.\MessageBreak
170          Tagging~is~activated~with~'testphase=tagpdf'~directly}{}
171      }
172    ,debug .code:n =
173      {
174        \keys_set:nn { document / metadata / debug } {#1}
175      }
176    ,debug / para .code:n =
177      {
178        \AddToHook
179          {
180            package/tagpdf/after
181          }
182          {
183             \tagpdfsetup{debug/show=para}
184          }
185      }
186    ,debug / log .code:n =
187      {
188        \AddToHook
```

```
189              {
190                package/tagpdf/after
191              }
192              {
193                  \tagpdfsetup{debug/log=#1}
194              }
195          }
196        ,debug / tagpdf .code:n =
197          {
198            \AddToHook
199              {
200                package/tagpdf/after
201              }
202              {
203                  \RequirePackage{tagpdf-debug}
204              }
205          }
206        ,debug / uncompress .code:n =
207          {
208            \pdf_uncompress:
209          }
210        ,debug / pdfmanagement .bool_gset:N = \g__pdfmanagement_active_bool
211        ,debug / firstaidoff .clist_gset:N = \g__pdfmanagement_firstaidoff_clist
212    }
213
214
```

4.5   Messages

```
215  %UFi is meta the right module name here?
216  \prop_gput:Nnn \g_msg_module_type_prop { meta } { LaTeX }
217  \prop_gput:Nnn \g_msg_module_name_prop { meta } { DocumentMetadata }
218
219  \msg_new:nnn { meta } { after-class }
220            {
221                \token_to_str:N \DocumentMetadata \c_space_tl
222                should~be~used~only~before~\token_to_str:N\documentclass
223            }
224  \msg_new:nnn { meta } { latex-lab-pkg-missing }
225            {
226              LaTeX-lab~package~'#1'~not~found.
227             }
228  \msg_new:nnn { meta } { lang-missing }
229            {
230              The~language~has~not~been~set~in~\token_to_str:N
231              \DocumentMetadata.\\Setting~it~to~'#1'~as~fallback.
232            }
233  \ExplSyntaxOff\makeatother
234  ⟨/code⟩
```

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.