

The ltdocinit module

The L^AT_EX Project*

Version 0.951, released 2022-02-16

1 ltdocinit documentation

This small module defines `\DocumentMetadata` and the older alias `\DeclareDocumentMetadata` and the related keys. It also defines commands to store document properties in a global container.

When the kernel will provide `\DocumentMetadata` directly this module will slowly disappear.

1.1 `\DocumentMetadata`/`\DeclareDocumentMetadata`

<code>\DocumentMetadata</code>	<code>\DocumentMetadata{<key-value list>}</code>
<code>\DeclareDocumentMetadata</code>	<code>\DeclareDocumentMetadata{<key-value list>}</code> (deprecated)

Currently there is no dedicated location to declare settings concerning a document as a whole. Settings are placed somewhere in the preamble or with the class options or even with some package options. For some settings this can be too late, for example the pdf version can no longer be changed if a package has used code which already opened the PDF.

`\DocumentMetadata` as a new command should unify such settings in one place. It should be used before `\documentclass`. `\DocumentMetadata` can be used more than once.

With a L^AT_EX 2022-06-01 or newer (currently provided as `latex-dev`) the command is provided by the kernel and will load the pdfmanagement, with older L^AT_EX the pdfmanagement must be loaded first manually with `\RequirePackage{pdfmanagement-testphase}`.

The keys defined for `\DocumentMetadata` currently allows to set the PDF version, to set the PDF `/Lang`, to uncompress a pdf, to set the language and to declare a few PDF standards and to load some colorprofiles.

`\DocumentMetadata` is also used to activate the new PDF management code and it loads a number of required files for the PDF management code. As this forces the loading of the backend files, a backend which can't be detected automatically like `dvipdfmx`, must be set in the first `\DocumentMetadata`.

Currently the following keys are implemented

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.

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

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` and `A-3u` are accepted as values. The casing is irrelevant, `a-1b` works too. The underlying code to ensure the requirements (as far as they can be ensured) is still incomplete, but a color profile is included and the `/OutputIntent` is set. The `u` variants for example do not force unicode, but they will pass the information to `hyperref` and `hyperxmp`. The `a` variants do *not* enforce (or even test) a tagged pdf yet. More information can be found 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 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.

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,interwordspace}`. This phase is frozen.

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

The **testphase** key can only be used in the first `\DocumentMetadata`.

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`.

1.2 Container for document properties

The module provides a container where classes, packages and users can store properties of the document which are perhaps of interest or use for other packages or the author.

The properties are stored with a key `label/property`. The values can be retrieved expandably.

<hr/> <code>\AddToDocumentProperties</code> <hr/>	<code>\AddToDocumentProperties[\langle label \rangle]{\langle property \rangle}{\langle value \rangle}</code> <p>This stores $\langle value \rangle$ under the key $\langle label \rangle / \langle property \rangle$. By default $\langle label \rangle$ is the current package name <code>\@currname</code>. If another label is chosen, it should be one which avoids clashes with other packages using the container. The label <code>document</code> is reserved.</p>
<hr/> <code>\GetDocumentProperties</code> <hr/>	<code>\GetDocumentProperties{\langle label / property \rangle}</code> <p>Expands to the $\langle value \rangle$ corresponding to $\langle label / property \rangle$ in the container. If $\langle label / property \rangle$ is missing, this has an empty expansion. The result is returned within <code>\exp_not:n</code>, which means that the $\langle value \rangle$ does not expand further when appearing in an x-type argument expansion.</p>
<hr/> <code>\ShowDocumentProperties</code> <hr/>	<code>\ShowDocumentProperties</code> <p>This show the current content of the container.</p>

References

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

2 ltdocinit implementation

```

1 <@@=pdfmanagement>
2 <*header>
3 \ProvidesExplPackage{ltdocinit}{2022-02-16}{0.95l}
4   {Initialize document metadata}
5 </header>

```

`\DocumentMetadata` will be defined by the kernel in short time. So we define it and the keys here only if it is not already defined.

2.1 The keys for `\DocumentMetadata`

We define the keys first so that we can test if `\DocumentMetadata` exist (testing for the format date would be nice but would fail for the current latex-dev).

<code>\g__pdfmanagement_firstaidoff_clist</code>	<p>A list to store the firstaid code which should be disabled</p> <pre> 6 <*package> 7 \clist_new:N \g__pdfmanagement_firstaidoff_clist </pre> <p>(End definition for <code>\g__pdfmanagement_firstaidoff_clist</code>.)</p>
<code>\g__pdfmanagement_testphase_tl</code>	<p>a tl to store the testphase loading code so that we can load them at the end of the command.</p> <pre> 8 \tl_new:N \g__pdfmanagement_testphase_tl </pre> <p>(End definition for <code>\g__pdfmanagement_testphase_tl</code>.)</p> <pre> 9 \cs_if_free:NT \DocumentMetadata 10 { 11 \keys_define:nn { document / metadata } 12 { 13 backend .choices:nn = </pre>


```

14         { dvipdfmx , dvips , dvisvgm , luatex , pdftex , pdfmode , xdvipdfmx , xetex }
15     {
16         \sys_load_backend:n {#1}
17     },
18     backend .groups:n = { init } ,
19 }
20
21 \keys_define:nn { document / metadata }
22 {
23     ,pdfversion .code:n =
24     {
25         \pdf_version_gset:n { #1 }
26         \AddToDocumentProperties[document]{pdfversion}{#1}
27     }
28     ,uncompress .code:n =
29     {
30         \pdf_uncompress:
31     }
32     ,uncompress .value_forbidden:n = true
33     ,lang .code:n =
34     {
35         \pdfmanagement_add:nnn {Catalog} {Lang}{#{1}}
36         \AddToDocumentProperties[document]{lang}{#1}
37     }
38     %,xmpmeta .bool_gset:N = \g_pdfmeta_xmp_bool %see pdfmeta unused and undefined for now
39     % this uses internal command from pdfmeta, it should probably move there ...
40     ,pdfstandard .code:n =
41     {
42         \exp_args:Nnx
43         \keys_set:nn {document / metadata} {_pdfstandard=\str_uppercase:n{#1}}
44     }
45     ,_pdfstandard .choices:nn =
46     {A-1B,A-2A,A-2B,A-2U,A-3A,A-3B,A-3U}
47     {
48         \prop_if_exist:cT { g__pdfmeta_standard_pdf/#1_prop }
49         {
50             \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/#1_prop }
51         }
52         \AddToDocumentProperties [document]{pdfstandard}{#1}
53     }
54     ,_pdfstandard / unknown .code:n =
55     {
56         \msg_warning:nnn{pdf}{unknown-standard}{#1}
57     }
58     ,testphase .multichoice:
59     ,testphase / tagpdf .code:n =
60     {
61         \tl_gput_right:Nn\g__pdfmanagement_testphase_tl
62         {
63             \file_if_exist_input:nF {tagpdf-latex-lab-testphase.ltx}
64             {
65                 \RequirePackage{tagpdf}
66                 \AddToDocumentProperties [document]{testphase/tagpdf}{loaded}
67                 \tagpdfsetup{activate,paratagging,interwordspace}

```



```

68         \AddToDocumentProperties [document]{tagging}{active}
69         \AddToDocumentProperties [document]{tagging/para}{active}
70         \AddToDocumentProperties [document]{tagging/interwordspace}{active}
71     }
72 }
73 }
74 ,testphase / unknown .code:n =
75 {
76     \tl_gput_right:Nn\g__pdfmanagement_testphase_tl
77     {
78         \AddToDocumentProperties [document]{testphase/#1}{loaded}
79         \file_if_exist_input:nF {#1-latex-lab-testphase.ltx}
80         {
81             \msg_warning:nnn{document}{latex-lab-pkg-missing}{#1}
82             \AddToDocumentProperties [document]{testphase/#1}{missing}
83         }
84     }
85 }
86 ,activate .multichoice:
87 ,activate / tagging .code:n =
88 {
89     \PackageWarning{pdfmanagement-testphase}
90     {The~activate~key~is~deprecated.\MessageBreak
91     Tagging~is~activated~with~'testphase=tagpdf'~directly}{ }
92 }
93 ,debug .code:n =
94 {
95     \keys_set:nn { document / metadata / debug } {#1}
96 }
97 ,debug / para .code:n =
98 {
99     \AddToHook
100     {
101         package/tagpdf/after
102     }
103     {
104         \tagpdfsetup{paratagging-show}
105     }
106 }
107 ,debug / log .code:n =
108 {
109     \AddToHook
110     {
111         package/tagpdf/after
112     }
113     {
114         \tagpdfsetup{log=#1}
115     }
116 }
117 ,debug / uncompress .code:n =
118 {
119     \pdf_uncompress:
120 }
121 ,debug / pdfmanagement .bool_gset:N = \g__pdfmanagement_active_bool

```



```

122     ,debug / firstaidoff .clist_gset:N = \g__pdfmanagement_firstaidoff_clist
123   }
124 }

```

2.2 \DocumentMetadata/\DeclareDocumentMetadata

At first we have to define the older alias `\DeclareDocumentMetadata` if `\DocumentMetadata` is already defined in a newer LaTeX version. Here we load the latex-lab support (if it hasn't been loaded before) and redefine then `\DeclareDocumentMetadata` to `\DocumentMetadata`

```

125 \cs_if_exist:NT \DocumentMetadata
126 {
127   \cs_new_protected:Npn \DeclareDocumentMetadata
128   {
129     \cs_if_free:NT \DocumentMetadata
130     {
131       \file_if_exist_input:nF {documentmetadata-support.ltx}%
132       {
133         \@latex@error
134         {No~support~files~for~\noexpand\DeclareDocumentMetadata~found}
135         {
136           Is~the~'LaTeX-lab'~bundle~installed?
137           \MessageBreak
138           Without~it,~the~declaration~is~ignored.
139         }
140         \let\DeclareDocumentMetadata\@gobble
141       }
142     }
143     \RenewDocumentCommand \DeclareDocumentMetadata {}
144     { \DocumentMetadata }
145     \DocumentMetadata
146   }
147 }

```

Now we provide a definition for `\DocumentMetadata` if it doesn't exist yet.

```

148 \cs_if_free:NT \DocumentMetadata
149 {

```

`\DocumentMetadata` should for now 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.

```

150   \prop_gput:Nnn \g_msg_module_type_prop { document } { LaTeX }
151   \prop_gput:Nnn \g_msg_module_name_prop { document } { DocumentMetadata }
152   \msg_new:nnn
153     { document } { setup-after-documentclass }
154     {
155       \token_to_str:N \DocumentMetadata \c_space_tl
156       should~be~used~only~before~\token_to_str:N\documentclass
157     }
158   \msg_new:nnn
159     { document } { latex-lab-pkg-missing }
160     {
161       LaTeX-lab~package~'#1'~not~found.
162     }

```


`\DocumentMetadata`

```
163 \NewDocumentCommand\DocumentMetadata { m }
164 {
165   \cs_if_eq:NNTF \documentclass \@twoclasseserror
166   { \msg_error:nn { document } { setup-after-documentclass } }
167   {
```

The wanted backend must be detected first, we read the init key and then force the loading by loading expl3. The backend can contains management commands, so the boolean should be set to true first.

```
168     \bool_gset_true:N \g__pdfmanagement_active_bool
169     \keys_set_groups:nnn { document / metadata } {init}{ #1 }
170     %if no backend has been loaded force it now:
171     \str_if_exist:NF \c_sys_backend_str
172     {
173       \sys_load_backend:n {}
174     }
```

Now we load the extra backend code

```
175     \ExplSyntaxOn\makeatletter
176     \file_input:n {l3backend-testphase-\c_sys_backend_str.def}
177     \ExplSyntaxOff\makeatother
```

set the default language, process the rest of the keys, and setup the generic driver

```
178     \keys_set_filter:nnn { document / metadata } { init } { lang=en-US, #1 }
179     \bool_if:NT \g__pdfmanagement_active_bool
180     {
181       \PassOptionsToPackage{customdriver=hgeneric-testphase}{hyperref}
182     }
```

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

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

Redefine `\DocumentMetadata` so that it only process the keys. We need to update the `hyperref` option if the active status changes.

```
184     \RenewDocumentCommand\DocumentMetadata { m }
185     {
186       \keys_set_filter:nnn { document / metadata } { init } { ##1 }
187       \bool_if:NTF \g__pdfmanagement_active_bool
188       {
189         \str_remove_all:cn
190         {opt@hyperref.sty}{customdriver=hgeneric-testphase}
191         \PassOptionsToPackage
192         {customdriver=hgeneric-testphase}{hyperref}
193       }
194       {
195         \str_remove_all:cn
196         {opt@hyperref.sty}{customdriver=hgeneric-testphase}
197       }
198     }
```

Load more modules, the testphase code and the firstaid code.

```
199     \g__pdfmanagement_testphase_tl
200     \RequirePackage{pdfmanagement-firstaid}
201   }
202 }
```


At last we need to provide the older alias here too

```
203 \NewDocumentCommand\DeclareDocumentMetadata{}{\DocumentMetadata}  
204 }
```

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

2.3 Container for document Properties

The container for the document properties is a prop

```
\g__pdfmanagement_documentproperties_prop  
205 \prop_new:N \g__pdfmanagement_documentproperties_prop %
```

(End definition for \g__pdfmanagement_documentproperties_prop.)

\AddToDocumentProperties

```
206 \NewDocumentCommand\AddToDocumentProperties{0{\@currname}mm}  
207 {  
208 \exp_args:NNx  
209 \prop_gput:Nnn \g__pdfmanagement_documentproperties_prop  
210 {  
211 \tl_if_blank:eTF {#1}{top-level/}{#1/} #2  
212 }  
213 { #3}  
214 }
```

(End definition for \AddToDocumentProperties. This function is documented on page 3.)

\GetDocumentProperties

```
215 \NewExpandableDocumentCommand\GetDocumentProperties{m}  
216 {  
217 \prop_item:Nn \g__pdfmanagement_documentproperties_prop {#1}  
218 }
```

(End definition for \GetDocumentProperties. This function is documented on page 3.)

\ShowDocumentProperties

```
219 \msg_new:nnn { pdfmanagement } { show-properties }  
220 {  
221 The~following~document~properties~have~been~stored:  
222 #1  
223 }  
224 \NewDocumentCommand\ShowDocumentProperties {}  
225 {  
226 \msg_show:nnx {pdfmanagement}{show-properties}  
227 {  
228 \prop_map_function:NN \g__pdfmanagement_documentproperties_prop \msg_show_item:nn  
229 }  
230 }
```

(End definition for \ShowDocumentProperties. This function is documented on page 3.)

```
231 \endpackage
```


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.

A	
\AddToDocumentProperties	3, 26, 36, 52, 66, 68, 69, 70, 78, 82, <u>206</u>
\AddToHook	99, <u>109</u>
B	
bool commands:	
\bool_gset_true:N	168
\bool_if:NTF	179, 187
C	
clist commands:	
\clist_new:N	7
cs commands:	
\cs_if_eq:NNTF	165
\cs_if_exist:NTF	125
\cs_if_free:NTF	9, 129, 148
\cs_new_protected:Npn	127
D	
\DeclareDocumentMetadata	1, 6, 127, 134, 140, 143, 203
\documentclass	1, 6, 156, 165
\DocumentMetadata	1–3, 6, 7, 9, 125, 129, 144, 145, 148, 155, <u>163</u>
E	
exp commands:	
\exp_args:NNx	208
\exp_args:Nnx	42
\exp_not:n	3
\ExplSyntaxOff	177
\ExplSyntaxOn	175
F	
file commands:	
\file_if_exist_input:nTF ..	63, 79, 131
\file_input:n	176
G	
\GetDocumentProperties	3, <u>215</u>
H	
hook commands:	
\hook_use_once:n	183
K	
keys commands:	
\keys_define:nn	11, 21
\keys_set:nn	43, 95
\keys_set_filter:nnn	178, 186
\keys_set_groups:nnn	169
L	
\let	140
M	
\makeatletter	175
\makeatother	177
\MessageBreak	90, 137
msg commands:	
\msg_error:nn	166
\g_msg_module_name_prop	151
\g_msg_module_type_prop	150
\msg_new:nnn	152, 158, 219
\msg_show:nnn	226
\msg_show_item:nn	228
\msg_warning:nnn	56, 81
N	
\NewDocumentCommand ...	163, 203, 206, 224
\NewExpandableDocumentCommand ...	215
\noexpand	134
P	
\PackageWarning	89
\PassOptionsToPackage	181, 191
pdf commands:	
\pdf_uncompress:	30, 119
\pdf_version_gset:n	25
pdfmanagement commands:	
\pdfmanagement_add:nnn	7, 35
pdfmanagement internal commands:	
\g__pdfmanagement_active_bool ...	121, 168, 179, 187
\g__pdfmanagement_documentproperties_prop	205, 209, 217, 228
\g__pdfmanagement_firstaidoff_clist	6, 122
\g__pdfmanagement_testphase_tl ...	8, 61, 76, 199
pdfmeta commands:	
\g_pdfmeta_xmp_bool	38
pdfmeta internal commands:	
\g__pdfmeta_standard_prop	50
prop commands:	
\prop_gput:Nnn	150, 151, 209
\prop_gset_eq:NN	50

