

# The documentmetadata-support code\*

Frank Mittelbach, Ulrike Fischer, L<sup>A</sup>T<sub>E</sub>X Project

August 15, 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 L<sup>A</sup>T<sub>E</sub>X 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 1.0h dated 2024-03-26, © L<sup>A</sup>T<sub>E</sub>X 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. L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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: *L<sup>A</sup>T<sub>E</sub>X 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 \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.

```

10 \cs_set_protected:Npn \DocumentMetadata #1
11   {
12     \cs_if_eq:NNTF \documentclass \@twoclasseserror
13     { \msg_error:nn { meta } { after-class } }
14     {

```

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.

```

15     \bool_gset_true:N \g__pdfmanagement_active_bool
16     \keys_set_groups:nnn { document / metadata } {init}{ #1 }
17     %if no backend has been loaded force it now:
18     \str_if_exist:NF \c_sys_backend_str
19     {
20       \sys_load_backend:n {}
21     }

```

Now we load the extra backend code:

```

22     \ExplSyntaxOn\makeatletter
23     \file_input:n {l3backend-testphase-\c_sys_backend_str.def}
24     \ExplSyntaxOff\makeatother

```

Process the init keys and setup the generic driver.

```

25     \keys_set_filter:nnn { document / metadata } { init } { #1 }
26     \bool_if:NT \g__pdfmanagement_active_bool
27     {
28       \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 it exists) or the fall back language. Note: if `babel` is loaded without a language this gives the language `und`.

```

29     \g@addto@macro\@kernel@after@begindocument
30     {
31       \pdfdict_get:nnN {g__pdf_Core/Catalog}{Lang}\l__pdfmanagement_tmpa_tl
32       \quark_if_no_value:NT\l__pdfmanagement_tmpa_tl
33       {
34         \tl_if_empty:eTF { \BCPdata{main.language} }
35         { \tl_set:Ne \l__pdfmanagement_tmpb_tl { \BCPdata{language} } }
36         { \tl_set:Ne \l__pdfmanagement_tmpb_tl { \BCPdata{main.language} } }
37         \msg_warning:nne { meta } { lang-missing }{ \l__pdfmanagement_tmpb_tl }
38         \exp_last_unbraced:Ne
39         \AddToDocumentProperties{[document]}{lang}{\l__pdfmanagement_tmpb_tl}
40         \pdfmanagement_add:nne {Catalog} {Lang}{\l__pdfmanagement_tmpb_tl}
41       }
42     }
43   }

```

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

```

44     \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.

```

45     \cs_set_protected:Npn \DocumentMetadata ##1
46     {
47         \keys_set_filter:nnn { document / metadata } { init } { ##1 }
48         \str_remove_all:cn {opt@hyperref.sty}{customdriver=hgeneric-testphase}
49         \bool_if:NT \g_pdfmanagement_active_bool
50         {
51             \PassOptionsToPackage{customdriver=hgeneric-testphase}{hyperref}
52         }
53     }

```

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

```

54     \g_pdfmanagement_testphase_tl
55     \RequirePackage{pdfmanagement-firstaid}
56 }
57 }

```

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

### 4.3 `\DocumentMetadata` keys

```

58 \keys_define:nn { document / metadata }
59 {
60     backend .choices:nn =
61         { dvipdfmx , dvips , dvisvgm , luatex , pdftex , pdfmode , xdviPDFmx , xetex }
62         {
63             \sys_load_backend:n {#1}
64         },
65     backend .groups:n = { init } ,
66 }
67
68 \keys_define:nn { document / metadata }
69 {
70     ,pdfversion .code:n =
71     {
72         \pdf_version_gset:n { #1 }
73         \AddToDocumentProperties[document]{pdfversion}{#1}
74     }
75     ,uncompress .code:n =
76     {
77         \pdf_uncompress:
78     }
79     ,uncompress .value_forbidden:n = true
80     ,lang .code:n =
81     {
82         \pdfmanagement_add:nnn {Catalog} {Lang}{(#1)}
83         \AddToDocumentProperties[document]{lang}{#1}
84     }
85     ,xmpmeta .bool_gset:N = \g_pdfmeta_xmp_bool %see pdfmeta unused and undefined for now!
86     % this uses internal command from pdfmeta, it should probably move there ...
87     ,pdfstandard .code:n =
88     {

```

```

89     \exp_args:Nne
90     \keys_set:nn {document / metadata} {\pdfstandard=\str_uppercase:n{#1}}
91   }
92 ,_pdfstandard .choices:nn =
93   {A-1B,A-2A,A-2B,A-2U,A-3A,A-3B,A-3U,A-4}
94   {
95     \prop_if_exist:cT { g__pdfmeta_standard_pdf/#1_prop }
96     {
97       \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/#1 _prop }
98     }
99     \AddToDocumentProperties [document]{pdfstandard}{#1}
100   }
101 ,_pdfstandard / A-4F .code:n =
102   {
103     \prop_if_exist:cTF { g__pdfmeta_standard_pdf/A-4F_prop }
104     {
105       \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4F_prop }
106     }
107     {
108       \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4_prop }
109       \prop_gput:Nnn \g__pdfmeta_standard_prop{conformance}{F}
110     }
111     \AddToDocumentProperties [document]{pdfstandard}{A-4F}
112   }
113 ,_pdfstandard / A-4E .code:n =
114   {
115     \prop_if_exist:cTF { g__pdfmeta_standard_pdf/A-4E_prop }
116     {
117       \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4E_prop }
118     }
119     {
120       \prop_gset_eq:Nc \g__pdfmeta_standard_prop { g__pdfmeta_standard_pdf/A-4_prop }
121       \prop_gput:Nnn \g__pdfmeta_standard_prop{conformance}{E}
122     }
123     \AddToDocumentProperties [document]{pdfstandard}{A-4E}
124   }
125 ,_pdfstandard / unknown .code:n =
126   {
127     \msg_warning:nnn{pdf}{unknown-standard}{#1}
128   }
129 ,testphase .multichoice:
130 ,testphase / tagpdf .code:n =
131   {
132     \tl_gput_right:Nn\g__pdfmanagement_testphase_tl
133     {
134       \file_if_exist_input:nF {tagpdf-latex-lab-testphase.ltx}
135       {
136         \RequirePackage{tagpdf}
137         \AddToDocumentProperties [document]{testphase/tagpdf}{loaded}
138         \tagpdfsetup{activate,para/tagging,activate/spaces}
139         \AddToDocumentProperties [document]{tagging}{active}
140         \AddToDocumentProperties [document]{tagging/para}{active}
141         \AddToDocumentProperties [document]{tagging/interwordspace}{active}
142       }

```

```

143     }
144   }
145 ,testphase / unknown .code:n =
146   {
147     \tl_gput_right:Nn\g__pdfmanagement_testphase_tl
148     {
149       \AddToDocumentProperties [document]{testphase/#1}{loaded}
150       \file_if_exist_input:nF {#1-latex-lab-testphase.ltx}
151       {
152         \msg_warning:nnn{meta}{latex-lab-pkg-missing}{#1}
153         \AddToDocumentProperties [document]{testphase/#1}{missing}
154       }
155     }
156   }
157 ,activate .multichoice:
158 ,activate / tagging .code:n =
159   {
160     \PackageWarning{pdfmanagement-testphase}
161     {The-activate-key-is-deprecated.\MessageBreak
162     Tagging-is-activated-with-'testphase=tagpdf'-directly}{ }
163   }
164 ,debug .code:n =
165   {
166     \keys_set:nn { document / metadata / debug } {#1}
167   }
168 ,debug / para .code:n =
169   {
170     \AddToHook
171     {
172       package/tagpdf/after
173     }
174     {
175       \tagpdfsetup{debug/show=para}
176     }
177   }
178 ,debug / log .code:n =
179   {
180     \AddToHook
181     {
182       package/tagpdf/after
183     }
184     {
185       \tagpdfsetup{debug/log=#1}
186     }
187   }
188 ,debug / tagpdf .code:n =
189   {
190     \AddToHook
191     {
192       package/tagpdf/after
193     }
194     {
195       \RequirePackage{tagpdf-debug}
196     }

```





<b>D</b>	
debug (key) .....	1
\documentclass .....	1, 5, 12, 214
\DocumentMetadata .....	1, 2, 5, 6, 10, 213, 223
<b>E</b>	
\emph .....	3
exp commands:	
\exp_args:Nne .....	89
\exp_last_unbraced:Ne .....	38
\ExplSyntaxOff .....	24, 225
\ExplSyntaxOn .....	5, 22
<b>F</b>	
file commands:	
\file_if_exist_input:nTF ...	134, 150
\file_input:n .....	23
<b>H</b>	
hook commands:	
\hook_use_once:n .....	44
<b>I</b>	
\includegraphics .....	3
<b>K</b>	
keys commands:	
\keys_define:nn .....	58, 68
\keys_set:nn .....	90, 166
\keys_set_filter:nnn .....	25, 47
\keys_set_groups:nnn .....	16
<b>L</b>	
lang (key) .....	1
<b>M</b>	
\makeatletter .....	5, 22
\makeatother .....	24, 225
\maketitle .....	3
\marginpar .....	3
\MessageBreak .....	161
metadata keys:	
backend .....	1
colorprofiles .....	1
debug .....	1
lang .....	1
pdfstandard .....	1
pdfversion .....	1
testphase .....	1
uncompress .....	1
xmp .....	1
msg commands:	
\msg_error:nn .....	13
\g_msg_module_name_prop .....	209
\g_msg_module_type_prop .....	208
\msg_new:nnn .....	211, 216, 220
\msg_warning:nnn .....	37, 127, 152
<b>P</b>	
\PackageWarning .....	160
\parbox .....	3
\PassOptionsToPackage .....	28, 51
pdf commands:	
\pdf_uncompress: .....	77, 200
\pdf_version_gset:n .....	72
pdfdict commands:	
\pdfdict_get:nnN .....	31
pdfmanagement commands:	
\pdfmanagement_add:nnn .....	5, 40, 82
pdfmanagement internal commands:	
\g__pdfmanagement_active_bool ...	15, 26, 49, 202
\g__pdfmanagement_firstaidoff_-	
clist .....	6, 203
\g__pdfmanagement_testphase_tl ..	
.....	8, 54, 132, 147
\l__pdfmanagement_tmpa_tl ...	31, 32
\l__pdfmanagement_tmpb_tl .....	
.....	35, 36, 37, 39, 40
pdfmeta commands:	
\g_pdfmeta_xmp_bool .....	85
pdfmeta internal commands:	
\g__pdfmeta_standard_prop .....	
.....	97, 105, 108, 109, 117, 120, 121
pdfstandard (key) .....	1
pdfversion (key) .....	1
prop commands:	
\prop_gput:Nnn .....	109, 121, 208, 209
\prop_gset_eq:NN .....	97, 105, 108, 117, 120
\prop_if_exist:NTF .....	95, 103, 115
<b>Q</b>	
quark commands:	
\quark_if_no_value:NTF .....	32
<b>R</b>	
\RequirePackage .....	4, 55, 136, 195
<b>S</b>	
str commands:	
\str_if_exist:NTF .....	18
\str_remove_all:Nn .....	48
\str_uppercase:n .....	90
sys commands:	
\c_sys_backend_str .....	18, 23
\sys_load_backend:n .....	20, 63
<b>T</b>	
\tagpdfsetup .....	138, 175, 185
testphase (key) .....	1

T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:		
\@kernel@after@begindocument . . .	29	\tl_new:N . . . . . 9
\@twoclasseserror . . . . .	12	\tl_set:Nn . . . . . 35, 36
\g@addto@macro . . . . .	29	token commands:
\title . . . . .	3	\token_to_str:N . . . . . 213, 214, 222
tl commands:		
\c_space_tl . . . . .	213	<b>U</b>
\tl_gput_right:Nn . . . . .	132, 147	uncompress (key) . . . . . 1
\tl_if_empty:nTF . . . . .	34	
\tl_if_exist:NTF . . . . .	8	<b>X</b>
		xmp (key) . . . . . 1