The documentmetadata-support code*

Frank Mittelbach, Ulrike Fischer, LaTeX Project

July 11, 2025

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:

*This file has version 1.0s dated 2025-07-07, © LaTeX Project.

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.

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 A-standard. values. The casing is irrelevant, `a-1b` works too. Note that using this 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 any 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.

Beside the A-standards it is also possible to use the values `X-4`, `X-4p`, `X-5g`, `X-5n`, `X-5pg`, `X-6`, `X-6n`, `X-6p` for a PDF/X and `UA-1` and `UA-2` for PDF/UA standard. `UA-2` should only be used together with PDF 2.0. Currently these keys set *only* the relevant XMP-metadata. They do not validate or enforce special requirements (e.g., the `UA` standards do not automatically activate tagging).

`pdfstandard` can be used more than once to set overlapping standards, e.g: `pdfstandard=A-2b,pdfstandard=X-4,pdfstandard=UA-1`. It is also possible to pass a list like `pdfstandard={UA-2,A-4F}`.

If XMP-metadata are added (see the following key `xmp`) the necessary conformance markers 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.

tagging This key allows to activate, deactivate or partially deactive the tagging support. It accepts the three values `on`, `off` and `draft`. When used, the key loads the tagpdf package and all modules that are loaded with `testphase=latest`. `tagging=off` deactivates then in the `class/before` hook the tagging commands. `tagging=draft` leaves the tagging commands active but it deactivates the writing of the structure tree at the end of the compilation. This can save time when drafting a longer document but preserves, e.g., MC-content marker in the PDF stream and warnings and errors from tagpdf if the structure is faulty.

tagging-setup This key allows to configure the tagging. It accepts all keys that can also be used in `\tagpdfsetup`; see the tagpdf documentation. Additionally, it accepts two keys to change the list of testphase modules loaded:

modules With this key it is possible to overwrite the list of modules loaded with by the `tagging` key. It will always load the `phase-II` module, so `modules={block,title}` does the same as `testphase={phase-II,block,title}`. Using this key is generally not needed, but if used the `tagging` key should only be used before the `tagging-setup` key to avoid that it resets the list of modules to the `latest set` again.

extra-modules This key allows to load modules that haven't been added to the `latest` set yet.

testphase This key is used to load testphase code. With the introduction of the `tagging` and `tagging-setup` keys it is no longer necessary for a user to specify the `testphase` key; it is only retained for internal testing and to support documents written during the early days of the tagging project.

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 value `latest` loads all modules that we recommend so that it is not necessary to specify individual modules.

The `phase` keys 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!

BBox This helps to debug BBox values of graphics, see the documentation of latex-lab-graphic. It does nothing if the graphic testphase module is not loaded.

References

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

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.