

News from the L^AT_EX Tagged PDF project: 2026

Ulrike Fischer

Abstract

The L^AT_EX Tagged PDF project was started in spring 2020 and announced to the T_EX community by the L^AT_EX Team at the (online) 2020 TUG conference. This short report describes the latest news of this multi-year project.

Contents

1	Introduction	169
2	Accessible math in PDF — finally!	169
3	Ending the testphase	169
4	Extending <code>\DocumentMetadata</code>	170
4.1	New code loaded	170
4.2	Command patches	170
4.3	New implementations	170
4.4	New interfaces	170
4.5	Effects on external packages	171
4.6	Incompatibilities	171
4.7	Which differences are relevant for users?171	
5	A moving target: How to keep track?	171

1 Introduction

Over the past years we have written a number of papers describing the motivation, goals and progress of the Tagged PDF project, e.g., [10, 3, 9, 2, 16] and we refer the reader to these papers for the background. Here we report on the latest updates.

More material can be found at the project github repository: github.com/latex3/tagging-project

2 Accessible math in PDF — finally!

Last year we announced that since June 2025 a concrete tool-chain exists to *demonstrate* that, and how, math tagged with MathML can be read [4]. In March 2026 the PDF Association finally announced officially that a workflow for accessible STEM content exists [15]; see figure 1.

Such a workflow (see figure 2) requires a *PDF producer* capable of tagging math with MathML. Such producers are currently LuaL^AT_EX and, with some restrictions, Word (it doesn't create MathML structure elements but the newest version is able to attach MathML as an associated file).

It further requires a *PDF viewer* that can pass on the MathML structure elements or the embedded associated MathML file to the assistive technology (AT). Currently this is, e.g., the case with Adobe reader [1], Foxit [5], and — as the first browser-based PDF viewer — Firefox [11]. Finally, it needs an AT

Accessible math in PDF – finally!



Math in PDF files is finally fully accessible, and can be navigated and read by capable assistive technology. This development, aligning creation software, modern PDF standards, compatible viewers and assistive technology, finally closes the accessibility gap between the web and downloadable STEM documents.

Article March 4, 2026

About PDF Association
staff

Figure 1: Screenshot of the PDF Association news

that can handle MathML in the PDF context, so a screen reader with the MathCat library [17], like the free NVDA [12] or (only in combination with Firefox) JAWS [18].

The workflow is still mostly Windows-specific, but with Firefox there is now a cross platform reader available. Orca [13], a GNU/Linux screen reader, is working on the MathML support.

In April 2026 the L^AT_EX Project Liaison Working Group (L^AT_EX LWG) at the PDF Association published a Best Practice Guide [6] that provides recommendations for creating STEM documents conforming to PDF 2.0 and PDF/UA-2 for this new workflow (figure 3). The Guide itself was compiled with LuaL^AT_EX and contains properly tagged math.

In June 2026, Boris Doubrov, the chair of the L^AT_EX LWG, Frank Mittelbach, David Carlisle and Neil Soiffer (the author of MathCat) introduced in a webinar [14] the Best Practice Guide and took questions from attendees.

3 Ending the testphase

For a long time the tagging support code, organized in modules, had to be loaded manually with the help of the “testphase” key. In view of the growing number of real world documents produced with the tagging code this was no longer adequate. As already announced at the 2025 TUG conference with the L^AT_EX release in November 2025 we made the “testphase” key obsolete and we *always* load all modules directly if `\DocumentMetadata` is used. Tagging is now activated with the “tagging” key.

```
\DocumentMetadata{tagging=on,pdfstandard=UA-2}
```

We also renamed the package from `pdfmanagement-testphase` to `pdfmanagement`.

For documents that want to load only the PDF management we provide a separate package, named `pdfmanagement`. Such documents should replace

```
\DocumentMetadata{pdfstandard=A-2b}
```

by

```
\RequirePackage{pdfmanagement}
\SetKeys{document/metadata}{pdfstandard=A-2b}
```

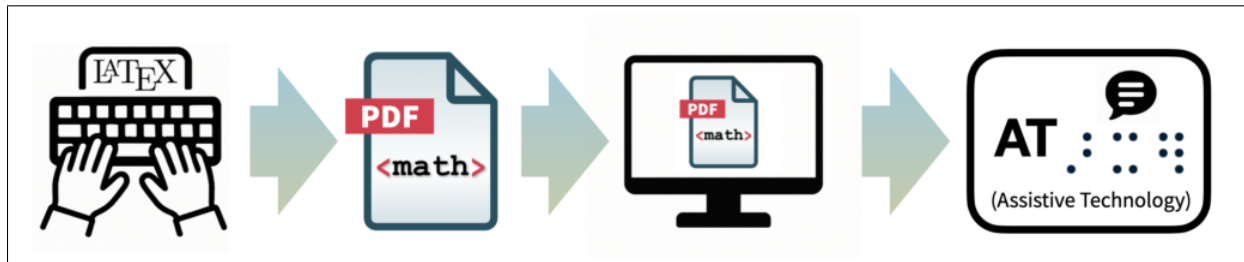
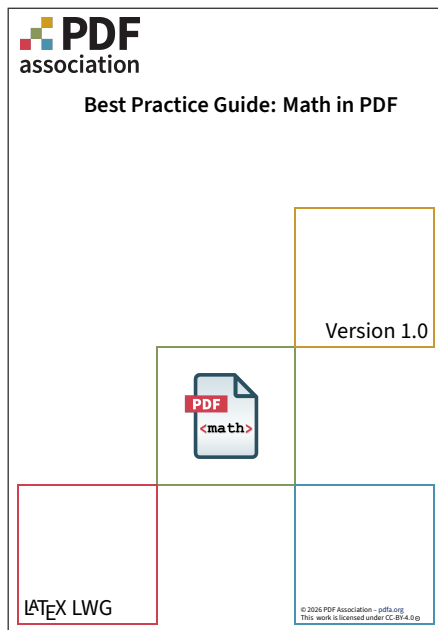


Figure 2: The workflow for accessible math

Figure 3: Cover of *Best Practice Guide: Math in PDF*

4 Extending `\DocumentMetadata`

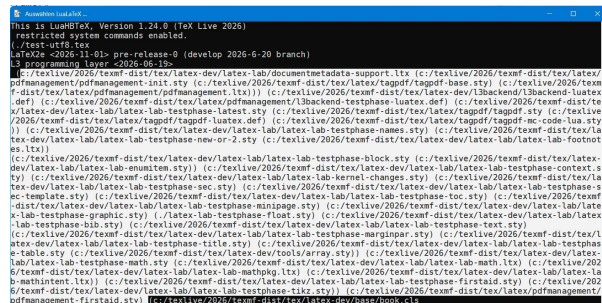
The fact that `\DocumentMetadata` now loads all modules has a number of consequences:

- A rather large amount of new code is loaded through the `latex-lab` modules.
- This code patches or extends various command definitions.
- Various commands have a new implementation.
- Some interfaces change.
- The code affects also external packages.
- Various packages and classes are incompatible with the new code.

4.1 New code loaded

The screenshot in figure 4 gives an impression of the amount of code involved: typically more than 30 additional files are loaded.¹

¹ Future L^AT_EX versions will combine files and move code into the kernel and so reduce the number of files loaded.

Figure 4: Screenshot of the additional files loaded by `\DocumentMetadata`

4.2 Command patches

The new code changes or extends the definitions of various internal and user command definitions to insert tagging sockets and/or hooks. Examples are `\contentsline`, `\mbox`, `\@savemarbox` (marginpar), `\makeph0nt` (used by `\phantom`), `\verb`, and more ...

4.3 New implementations

In other cases the new code completely reimplements document elements. Every inline and display math grabs its content as an argument before processing it. Footnotes use sockets. Blocks, lists, theorems and — since the latest L^AT_EX release — headings are now based on templates. As an example, here is the new definition of the `\section` command:

```
\DeclareDocumentCommand\section
  {s ={\shorttitle}o m}
  {\ParseLaTeXeHeading{section}{#1}{#2}{#3}}
```

4.4 New interfaces

As can be seen from the `={\shorttitle}o` in the above definition the interface of heading commands has changed: the optional argument accepts a key-value list. With this, it is possible to configure a heading on the fly and to adapt the texts used in the table of contents and the bookmarks separately:

```

\section
[
  placement      = top, %at top of page
  number-format= \fbox{\theheading},
  prefix         = Part~,
  title-decls    = \scshape,
  toc            = TOC,
  bookmark      = BOOKMARK
]
{The Bear}

```

The output would then look like this:

Part 1 THE BEAR

The interfaces of block and list environments have been extended and now have optional key-value arguments also.

4.5 Effects on external packages

The new code loaded by `\DocumentMetadata` not only changes standard $\text{\LaTeX}$ commands and environments, but also affects external packages. We suppress, emulate and patch packages to adapt them to the new code and to add at least partial tagging support. A number of packages and classes have also been adapted by their maintainers.

Examples of packages that are suppressed and emulated are `enumitem` (not yet complete), `amsthm`, `alltt` and `footmisc`. Commands from, e.g., `amsthm`, `fancyvrb`, `float` and `tikz` have been patched. Others contain tagging support directly, e.g.: `eqnlines`, `fcolumn`, `incgraph`, `linguistix`, `marginalia`, `marginnote`, `pdfcomment`, `tcolorbox`.

Finally, `ltx-talk`, a new class for accessible presentations, works only with `\DocumentMetadata`.

4.6 Incompatibilities

Various packages and classes are not yet compatible with the new code. The status list [8] mentions 659 incompatible and partially compatible packages and classes.²

4.7 Which differences are relevant for users?

In short we now have two different $\text{\LaTeX}$ versions: legacy $\text{\LaTeX}$ and the new “`\DocumentMetadata`” $\text{\LaTeX}$, which differ in various ways. But which differences are really relevant?

For “normal users”, the relevant differences are small interface changes — mostly additional optional key-value arguments —, small layout changes from bug corrections, e.g., the placement of some footnotes,

² We thank here mbertucci who greatly extended and improved the list and the test files in the past year.

and a rather large set of incompatible packages and classes.

For package, class and “template” authors these differences are relevant also. Additionally, they have to handle the new configuration options and need to learn how to make their commands and environments tagging aware.

5 A moving target: How to keep track?

There are already quite a lot of changes and more will come. The new code is a moving target: more commands will be made tagging-aware, more commands will be reimplemented with templates, more packages will adapt their code to support tagging.

How to keep track?

Using AI is not really a sensible way to learn how to make an accessible PDF with $\text{\LaTeX}$: While AIs quite eagerly offer long $\text{\LaTeX}$ examples with lots of comments, the quality ranges from nonsensical documents using incompatible packages, to passable with small errors if the prompts are too naive. AI has no good notion about our timelines and how the code is changing, and doesn’t know what we are planning and working on. Without good knowledge about accessible PDFs in general and the status of the $\text{\LaTeX}$ Tagged PDF project in particular, it is quite difficult to access and correct the answers or to write better prompts.

So, I recommend the more traditional learning methods:

- Use the new code to get used to it.
- Read the $\text{\LaTeX}$ News: `texdoc ltnews`.
- Read the documentation:


```
texdoc -l latex-lab
texdoc usrguide
texdoc clsguide
texdoc source2e
```
- Use the $\text{\LaTeX}$ github [7] to read and write issues and discussions and give feedback.
- Use the $\text{\LaTeX}$ communities:
 - Ask questions.
 - Answer questions.
 - Discuss with people.
 - Spread the knowledge.

It is work to learn how to handle new documents with `\DocumentMetadata`. But it is worth the effort: the new code not only adds tagging support but in the process also implements new, useful features for document authors and class designers. And as a bonus you get *Accessible Math* — *finally!*

References

- [1] Adobe. Adobe Reader. www.adobe.com/acrobat/pdf-reader.html
- [2] U. Fischer. On the road to Tagged PDF: About StructElem, marked content, PDF/A and squeezed BÄrs. *TUGboat* 42(2):170–173, 2021. doi.org/10.47397/tb/42-2/tb131fischer-tagpdf
- [3] U. Fischer, F. Mittelbach. Automated tagging of L^AT_EX documents — what is possible today, in 2023? *TUGboat* 44(2):262–266, 2023. doi.org/10.47397/tb/44-2/tb137fischer-tagging23
- [4] U. Fischer, F. Mittelbach. News from the L^AT_EX Tagged PDF project: 2025. *TUGboat* 46(2):244–247, 2025. doi.org/10.47397/tb/46-2/tb143fischer-tagging25
- [5] Foxit Software Incorporated. Foxit Reader. www.foxit.com
- [6] L^AT_EX LWG/PDF Association. Best practice guide: Math in PDF, 2026. pdfa.org/resource/best-practice-guide-math-in-pdf/
- [7] L^AT_EX Project. The L^AT_EX Tagged PDF repository. github.com/latex3/tagging-project
- [8] L^AT_EX Project. Tagging status of L^AT_EX packages and classes, 2026. latex3.github.io/tagging-project/tagging-status/
- [9] F. Mittelbach, U. Fischer. The L^AT_EX Tagged PDF project — a status and progress report. *TUGboat* 43(3):268–272, 2022. doi.org/10.47397/tb/43-3/tb135mitt-tagged
- [10] F. Mittelbach, U. Fischer. Enhancing L^AT_EX to automatically produce tagged and accessible PDF. *TUGboat* 45(1):52–59, 2024. doi.org/10.47397/tb/45-1/tb139mitt-deims24
- [11] Mozilla Corporation. Firefox. firefox.com
- [12] NV Access. NVDA. nvaccess.org
- [13] Orca Developers. Orca screen reader, 2026. orca.gnome.org
- [14] PDF Association. Webinar — Accessible Mathematical Content in PDF. pdfa.org/event/webinar-accessible-mathematical-content-in-pdf/
- [15] PDF Association. Accessible math in PDF — finally!, 2026. pdfa.org/accessible-math-in-pdf-finally/
- [16] C. Rowley, U. Fischer, F. Mittelbach. Accessibility in the L^AT_EX kernel — experiments in Tagged PDF. *TUGboat* 40(2):157–158, 2019. tug.org/TUGboat/tb40-2/tb125rowley-tagpdf.pdf
- [17] N. Soiffer. MathCAT: Math capable assistive technology. nsoiffer.github.io/MathCAT
- [18] Vispero. JAWS. vispero.com/jaws-screen-reader-software/

◇ Ulrike Fischer
 Bonn, Germany
<https://www.latex-project.org>