The LaTeX Tagged PDF project — A status and progress report

Frank Mittelbach, Ulrike Fischer

Abstract

The LaTeX Tagged PDF project was started in spring 2020 and announced to the TeX community by the LaTeX Team at the (online) 2020 TUG conference. This short report describes the progress and status of this multi-year project.

Contents

1 Project overview

A tagged PDF is a PDF with additional semantic structure, which improves accessibility and reuse. To enable LaTeX to create such tagged PDFs, the LaTeX Tagged PDF project was initiated in Q4 of 2019 with a feasibility study produced for Adobe [18]. This led to a commitment by Adobe to financially support the project as proposed in that study. Unfortunately, due to the COVID-19 pandemic that flared up at that time, the execution of this commitment was delayed until Q3 of 2020. Despite this delay, the LaTeX Project Team started the effort in late spring 2020 (with at that time limited resources) and the project was announced to the TeX community at the (online) TUG 2020 conference, where the team also presented the first results from Phase I of the project [19].

The LaTeX Tagged PDF project is divided into six phases, each producing immediately usable applications. This ensures both early user benefits and early feedback to the project team. The phases are roughly aligned with the bi-yearly LaTeX release cycle, with each phase being expected to take one to three LaTeX releases.

In the feasibility study, all identified project tasks are given a unique number, in order to easily cross-reference them, describe their dependencies, and arrange them in the project schedule outlined in the study. In addition to referring to tasks by name, the current report also lists these task numbers to assist readers in finding more detailed information about a particular task by looking it up in that study [18].

1.1 Phase I — Prepare the ground

The purpose of the first phase was the implementation of the core functionality inside LaTeX that forms the basis for all the later work of creating a well-tagged PDF. This phase was completed in 2021 and contained three important milestones:

- “The Hook Management System” (task 2.2.5)
- “PDF Object Support” (task 2.2.6)
- “The Automated Testing Environment” (task 2.1.2)

As part of the work on Phase I we also identified two new tasks not covered in the feasibility study:

- “Add Generic Command Hooks” (task 2.2.5(b))
- “Provide a General Configuration Point Management” (task 2.2.5(c))

1.1.1 The hook management system

The “Hook Management System” was made available to the general public in the 2020 fall release of LaTeX and enhanced LaTeX with a generic hook interface and a variety of document, shipout, file and environment hooks [11, 13, 15].

Half a year of general use of the hook management system (by the team and by many third-party developers) showed that it needed some extensions and adjustments. This resulted in the add-on task 2.2.5(b) to augment the hook management system with a generic method to automatically add hooks to third-party commands when necessary.

These generic command hooks will allow us to patch third-party code from the outside (i.e., without taking over the maintenance of abandoned but otherwise functional packages) and this way simplifying the adoption of tagged PDF. The generic command hooks were implemented and made available in the 2021 spring release of LaTeX [12].

We also identified the need for “General Configuration Point Management”, similar to hook management but for configurations where only a single package or class can be in charge. This is needed to avoid packages patching into internal LaTeX commands and overwriting each other (partially) or overwriting tagging support code. For example, several packages currently attempt to alter the same internal commands of the output routine to insert some special code for footnote handling.

Conceptual work for this new task 2.2.5(c) has already been undertaken; package writer interfaces, similar to those of the hook management, will be provided during 2023.

doi.org/10.47397/tb/43-3/tb135mitt-tagged

1.1.2 PDF object support

“PDF Object Support”, code to support various PDF related tasks, is in part already included in the kernel through the `l3pdf` module of the L3 programming layer [5]. Further functionality is provided through the external bundle `pdfmanagement-testphase` [16], which was released in early 2021. This will be integrated into the kernel at a later stage.

As part of this task it was necessary to work with TeX engine developers to develop some engine patches for LuaTeX and pdfTeX, in order to enable these engines to fully support PDF 2.0 Structure Destinations (XeTeX was already capable out of the box). Such structure destination provides the same view mechanism as a destination, but references a structure element instead of a page and so creates a direct connection from a link to some content. With TeX Live 2022 and current MiKTeX, structure destinations are now created automatically if the `\DocumentMetadata` command, or the `tagpdf` package, are used to create a PDF 2.0 document.

Sadly there is currently no easy way to check locally if a link points to a structure and to which one. The tag-view of Adobe Pro doesn’t show them and its html export ignores the structure. To test the new feature one has to check the internal PDF structure or use an online service like ngPDF [1], a demo site for new technology to derive HTML from tagged PDF in a predictable manner, developed by the PDF Association. When using ngPDF the HTML export of a tagged PDF with structure destinations contains links to the id of a structure:

```
<h1 id="6d4ff5-1">1 abc</h1>
<a href="#6d4ff5-1">1</a>
```

Without the structure destination the link would point only to a page related target:

```
<a href="#page-0">1</a>
```

1.1.3 The automated testing environment

The “Automated Testing Environment” is integral to achieving a successful completion to the project. It is essential to build up a large test suite on which all tests can be run and verified automatically. This is because we need to modify substantially the core LaTeX code without adversely affecting the millions of existing users who expect to be able to continue to reprocess their documents without finding any unexpected visual changes. Thus, even though this code is handled directly only by the LaTeX team, its existence and stability is of utmost importance to the success of the project.

1.2 Phase II — Provide tagging of simple documents

The main goal of Phase II is to provide automatic tagging of simple documents, excluding more complicated structures such as mathematics, tables, etc. This will be achieved by setting up the necessary core code that provides general mechanisms to deal with the issues around the automatic detection of paragraph text and its correct tagging, together with enabling a subset of the standard LaTeX document elements to produce the required tags.

Work on these two essential foundations for Phase II was started earlier, in parallel to finishing Phase I:

- “Core Tagging Support” (task 2.3.1); and
- Support for “Automated Paragraph Tagging” (task 2.3.2).

The main objective of these tasks is to provide the necessary infrastructure for the automatic tagging of relatively non-complex documents. The current focus is thus on the remaining major task for Phase II:

- “Implement tagging for the basic document elements of LaTeX” (task 2.3.3).

Work on Phase II took slightly longer than initially estimated due to additionally identified tasks that are either prerequisites for a successful completion of Phase II, or necessary for later phases and, for one reason or another, were best undertaken now, in parallel. We expect to close out Phase II be the end of 2022 with an out-of-sequence release of the `latex-lab` bundle, which was added in June 2022 to enable safe experimentation with new project code without disrupting workflows using production LaTeX [9].

1.2.1 Core tagging support

Tagging a PDF requires writing and managing various objects and literals in the PDF. The needed “Core Tagging Support” code is currently available as an add-on package (the package `tagpdf` [2]), since this allows for safe experimentation by those who wish to have tagged PDF output now, but without disrupting any user workflows. Once it is thoroughly tested, the code will be integrated into the kernel (this will form its own task in a later phase). How to use the code to create simple tagged PDF’s was described by Ulrike Fischer at the (online) 2021 TeX Users Group conference [3].

1.2.2 Automated paragraph tagging

Large parts of standard documents consist of simple paragraphs. For the success of the project it is of utmost importance that such paragraphs are tagged automatically and that paragraphs split over pages are handled correctly. The kernel extensions needed for such "Automated Paragraph Tagging" were finished in time for the 2021 spring release of LaTeX and announced at the (online) 2021 TUG conference by Frank Mittelbach [17]. They use marks and new hooks at the beginning and end of paragraphs [14].

1.2.3 Tagging for basic document elements

The goal of this task is to make standard LaTeX document elements tagging-aware, so that all the structural information they encapsulate is automatically transferred into an appropriate tag structure (including attributes) in the resulting PDF document. In the project schedule this task is split between phases II and III, starting in Phase II by concentrating on the high-level structural elements such as links, headers and footers, headings, lists, footnotes and tables of contents. The `tagpdf` package and the PDF management code already implement the automatic tagging of hyperlinks and the tagging of headers and footers (as artifacts). Automatic footnote tagging (with links) including special cases, such as footnotes broken across columns or pages, tagging of lists and tagging of tables of contents will be deployed with the release of `latex-lab` at the end of 2022. Later in Phase III (in 2023) the remaining basic document structures will be added, leaving more complex structures, such as tables, to later phases.

In this context, "automatic tagging" means that "identified document elements will be mapped using a default mapping to PDF tags without manual adjustments or fine-grained flexibility." Such flexibility will of course eventually be necessary in order to produce the highest quality of tagged PDF; therefore, this flexibility had to be catered for already in the underlying support code, which led to a new task:

- "Design and implement a general key/value interface" (task 2.3.3b).

This was identified as an additional prerequisite for successfully completing tasks 2.3.3 and 2.3.5. It must be made possible to extend the optional argument of standard commands and environments, for example, for sectioning and captions to accept key/value arguments to specify alternative text.

The task also includes the design and implementation of a general template mechanism for commands and environments using the key/value method for configuration. However, exposing these concepts on the user and package developer levels will require the design of interfaces for their configuration and manual overwriting, both of which are parts of later phases.

1.3 Preparatory work for tasks in Phases III and IV

For a number of technical and practical reasons we have diverged from the original schedule layout and have already started work on tasks planned for later phases. These include:

- "Design and implement an extended cross-reference mechanism for LaTeX" (task 2.2.2)
- "Provide an interface for specifying all types of document metadata" (task 2.3.4)
- "Design and implement hyperlinking and move it into the LaTeX kernel" (task 2.2.3)
- "Standards compliance" (task 2.3.9)

1.3.1 Interface to document metadata

The status and outlook on the implementation of document metadata (task 2.3.4) is described in a separate article in the current issue [4].

1.3.2 Hyperlinking improvements

User interfaces (and backend code) for hyperlinking facilities are currently provided mainly by the `hyperref` package. With the new PDF Object Support described above, large parts of the backend code have already been moved into the LaTeX kernel or into the `pdfmanagement-testphase` package. At the time when structures such as footnotes, headings, and tables of contents are made tagging-aware, native hyperlinking support will be added as well, and the no-longer-needed patches in `hyperref` suppressed.

1.3.3 Standards compliance

Support for various PDF/A standards is provided by the `pdfmanagement-testphase` package (`l3pdfmeta` module). The code will add the typically-needed color profiles and PDF objects, and suppress forbidden actions such as JavaScript code.

It should be noted that LaTeX cannot check all requirements of a standard and that an external validator such as veraPDF should be used for this. Support for PDF/X standards is currently only provided in the form of XMP metadata entries.

1.4 Cross-phase tasks

A number of tasks require attention and action across all phases. Up to now these are:

- "Define a change strategy to safely extend LaTeX without causing serious issues for the worldwide user base" (task 2.1.1)
- "Developer acceptance testing for finished tasks" (task 2.4.1)
- "Coordinate updates to external packages" (task 2.4.3)

Altering and enhancing LaTeX without disrupting existing documents and workflows is an important goal of the project. For this a number of tools have been implemented:

- The `\DocumentMetadata` interface allows tagged documents and non-tagged documents to be processed by the same LaTeX format by changing only one line.
- Experimental code is kept first in external packages such as `pdfmanagement-testphase` and `tagpdf`, or the `latex-lab` bundle.
- The `firstaid` package allows us to temporarily patch external packages if it turns out that they are incompatible with a change.
- The `latex-dev` releases give package authors time and opportunity to test changes and report problems, and the LaTeX team time to contact package authors and coordinate updates.
- The package `tagpdf-base` provides dummy versions of the core tagging commands, thus supporting the writing of commands and environments which are properly tagged if the user activates tagging, but which also work without tagging.
- The `\IfDocumentMetadataTF` kernel command allows testing if the new interface has been used in a document.
- The `\MakeLinkTarget` kernel command provides a dummy version of the command used by `hyperref` that creates anchors, and so allows writing commands and environments with built-in hyperlinking features which are activated if the user loads `hyperref`.

1.5 Interface to project code for users

As part of the metadata task (2.3.4), we provided a `\DocumentMetadata` command in the LaTeX kernel to be used as the very first declaration in a document (i.e., before `\documentclass`). This allows us to load the PDF management code and enable tagging and other project-related code. In short, by using this declaration the user indicates that this is a document to which tagging should be applied. That is, it serves a similar role as the switch from `\documentstyle` (old LaTeX 2.09 pre-1994) to `\documentclass` (modern LaTeX). This eases the transition and allows old and new code to coexist.

In this way, we also avoid users having to load special packages to test new features, instead everything boils down to giving a simple line

```
\DocumentMetadata{testphase=phase-II,...}
```

at the start of the document which then loads everything necessary to make use of currently-available results from a particular phase.

This interface has been deployed in LaTeX 2022 June release and from that point onwards it is available to every LaTeX user.

1.6 Summary of current status

Phase I of the project was completed previously and the results are deployed and in general use today.

Phase II is near completion, with most tasks implemented and deployed, and an expected closeout with the release of the `latex-lab` bundle at the end of 2022. This will then enable automatic tagging of LaTeX documents with a (still fairly) simple element structure by adding the aforementioned

```
\DocumentMetadata{testphase=phase-II,...}
```

at the top of a document.[1] Thus the major aim of Phase II is to enable many existing LaTeX documents to be reprocessed to produce tagged, and hence accessible, PDF output with no change to the source file other than adding a line like the above.

The goal of the later phases is then to expand this scope with more and more document elements being recognized, and to support adjustments to the tagging, thereby increasing the quality of the tagged PDF output. Eventually, the temporary interface within `\DocumentMetadata`, responsible for loading the `tagpdf` support package, will also no longer be necessary.

Overall, we can confidently state that the project progress is in good shape and within the main plan boundaries and will continue to be so.

2 Software releases

In the period between 2020–Q4 and 2022–Q4 the team has implemented and distributed five main LaTeX releases that have a direct bearing on the progress of the project. Important features of the releases are summarized below; the releases also contain other improvements not directly related to the Tagged PDF project.

LaTeX Release 2020-10, see [6]

- Provide the new hook management for LaTeX (task 2.2.5)
- Move the `xparse` interfaces to the kernel (needed for various later tasks to provide document-level interfaces)

[1] The data to place into the `\DocumentMetadata` argument is temporary at this point and will change over the course of the project, e.g., `testphase=phase-II` means apply the code for Phase II — something that will not be necessary once the code is finalized and integrated with the LaTeX kernel.

LaTeX Release 2021-06, see [7]

- Extending hook management to paragraphs (needed for task 2.3.2)
- Extending hook management to commands (needed for Phases II & III tasks)

LaTeX Release 2021-11, see [8]

- Consolidation release
- Corrections and improvements to the hook management system after extensive use by the project team and by third-party developers

LaTeX Release 2022-06, see [9]

- `\DocumentMetadata` interface (needed for task 2.3.4)
- Introduction of the `latex-lab` bundle (needed to allow safe user experimentation with new functionality from the project)
- First part of the new key/value handling in the kernel (needed for several tasks)

LaTeX Release 2022-11, see [10]

- Auto-detecting new key/value arguments, e.g., in `\section` or `\caption` (implements task 2.3.3b; needed for several other tasks)

References

[1] DualLab. Next generation PDF. ngpdf.com

[2] U. Fischer. *The tagpdf package.* ctan.org/pkg/tagpdf

[3] U. Fischer. On the road to Tagged PDF: About StructElem, marked content, PDF/A and squeezed Bärs. *TUGboat* 42(2):170–173, 2021. https://doi.org/10.47397/tb/42-2/tb131fischer-tagpdf

[4] U. Fischer, F. Mittelbach. Adding XMP metadata in LaTeX. *TUGboat* 43(3):263–267, 2022. https://doi.org/10.47397/tb/43-3/tb135fischer-xmp

[5] LaTeX Project Team. *The L3kernel package.* ctan.org/pkg/l3kernel

[6] LaTeX Project Team. LaTeX news, issue 32, 2020. www.latex-project.org/news/latex2e-news/ltnews32.pdf

[7] LaTeX Project Team. LaTeX news, issue 33, 2021. www.latex-project.org/news/latex2e-news/ltnews33.pdf

[8] LaTeX Project Team. LaTeX news, issue 34, 2021. www.latex-project.org/news/latex2e-news/ltnews34.pdf

[9] LaTeX Project Team. LaTeX news, issue 35, 2022. www.latex-project.org/news/latex2e-news/ltnews35.pdf

[10] LaTeX Project Team. LaTeX news, issue 36 — draft, 2022. www.latex-project.org/news/latex2e-news/ltnews36.pdf

[11] LaTeX Project Team. *LaTeX's hook management*, 2022. mirrors.ctan.org/macros/latex/base/lthooks-doc.pdf

[12] LaTeX Project Team. *The ltcmdhooks module*, 2022. mirrors.ctan.org/macros/latex/base/ltcmdhooks-doc.pdf

[13] LaTeX Project Team. *The ltfilehook documentation*, 2022. mirrors.ctan.org/macros/latex/base/ltfilehook-doc.pdf

[14] LaTeX Project Team. *The ltpara.dtx code*, 2022. mirrors.ctan.org/macros/latex/base/ltpara-doc.pdf

[15] LaTeX Project Team. *The ltshipout package*, 2022. mirrors.ctan.org/macros/latex/base/ltshipout-doc.pdf

[16] LaTeX Project Team. *The pdfmanagement-testphase package*, 2022. ctan.org/pkg/pdfmanagement-testphase

[17] F. Mittelbach. Taming the beast — Advances in paragraph tagging with pdfTeX and XeTeX (2021): Automatic paragraph tagging with the pdfTeX and XeTeX engine now possible. www.latex-project.org/news/2022/09/06/TUG-online-talks-21-22/

[18] F. Mittelbach, U. Fischer, C. Rowley. LaTeX Tagged PDF feasibility evaluation. latex-project.org/publications/2020-tagged-pdf-feasibility.pdf

[19] F. Mittelbach, C. Rowley. LaTeX Tagged PDF—a blueprint for a large project. *TUGboat* 41(3):292–298, Nov. 2020. latex-project.org/publications/2020-FMi-TUB-tb129mitt-tagpdf.pdf

◇ Frank Mittelbach
Mainz, Germany
https://www.latex-project.org

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