

L^AT_EX News

Issue 37, June 2023 — DRAFT version for upcoming release (2023-06-01)

Contents

Introduction	1
New functionality offered as part of the “L^AT_EX Tagged PDF” project	1
Documentation improvements	1
Displaying the exact release dates for L ^A T _E X . . .	1
New or improved commands	2
Providing copy and show functions for environments	2
Code improvements	2
doc: Index _ correctly	2
Default definition for \do	2
doc: Support the upquote package	2
New key for filecontents	2
A further hook for shipping out pages	2
Displaying release information in the .log . . .	3
Bug fixes	3
Incompatibility between doc and unicode-math	3
A fix for \hspace	3
Improve spacing at top of minipages	3
Changes to packages in the amsmath category	3
Changes to packages in the graphics category	3
Changes to packages in the tools category	3
Fix handling of nested multicols	3

Introduction

New functionality offered as part of the “L^AT_EX Tagged PDF” project

We have now enabled new automatic tagging functionality for additional L^AT_EX elements, among them most display environments, standard sectioning commands and content, figure and table listings. This can be activated through

```
\DocumentMetadata{testphase=phase-III}
```

At this point in time tagging support is only available for a restricted set of documents, i.e., those that use one of the basic document classes (`article`, `report`,

and `book`) and only use commands and environments described in Lamport’s L^AT_EX manual.

Using other document classes or adding additional packages in the preamble may work (or may partially work) but at this stage it is not very likely, at least not for packages or classes that excessively alter internals of L^AT_EX.

Also note that there are still several environments and commands described in the L^AT_EX manual that do not have tagging support yet, notably `tabulars`, `floats`, `tabbing` and the various `math` environment. They will get this support as part of `phase-III` between now and the June release.

A prototype for math tagging (including support for the `amsmath` environments) is already available, but it is mainly intended for experimentation and feedback and the resulting tagging is by no means the way we envision it to be eventually. If you like to try it out use the following line:

```
\DocumentMetadata{testphase={phase-III,math}}
```

Note that the math tagging code at this point in time will clash with packages that redefine the `$` (which then may lead to strange errors) and that packages that use math mode for non-mathematical constructs may result in surprising output as far as tagging is concerned. Feedback on which packages fail with the code in one or the other way would be appreciated.

The `latex-lab` bundle contains various (still untagged) documentation files about the new code that can be accessed with `texdoc -l latex-lab`.

Feedback is welcome! Please use <https://github.com/latex3/latex2e/discussions/1010>.

Documentation improvements

Displaying the exact release dates for L^AT_EX

In some situations it is necessary to find out the exact release dates for older version of the L^AT_EX format, for example, when you need to use different code in a package depending on the availability of a certain feature and you therefore want to use `\IfFormatAtLeastTF{<date>}` or the rather horrible construction `\@ifl@t@r\fmtversion{<date>}`, if you want to cater for formats that are older than 2020.

Or you know that your package is definitely not going to work with a format before a certain `<date>`, in which case you could use `\NeedsTeXFormat{LaTeX2e}[<date>]`

to ensure that any user is alerted if their format is too old.

The big problem is to know the exact $\langle date \rangle$ to put into such commands and in the past that was not that easy to find. You could have looked in the file `changes.txt`, but that is hidden somewhere in your installation and if you try `texdoc -l changes.txt` you get more than thirty results and the right file is by no means the first.

Yukai Chou (@muzimuzhi) kindly provided a patch for this, so that we now have the exact dates for each L^AT_EX format listed in an easy to remember place: in `ltnews.pdf` and that file conveniently also contains all major features and changes to L^AT_EX over the years—one of which is most likely the reason you need the $\langle date \rangle$ date in the first place.

It is given in parentheses in the newsletter title, thus this newsletter tells you that on 2023-06-01 the command `\NewEnvironmentCopy`, a new `shipout` hook, etc. was made available. And looking into `ltnews.pdf` you can now easily find out that the L^AT_EX3 programming layer was added on 2020-02-02 (because the date was so nice) and not on the first of the month. *(github issue 982)*

New or improved commands

Providing copy and show functions for environments

To copy a command definition we have introduced `\NewCommandCopy` in 2022. This even allows you to copy commands that consist of several internal components, such as robust commands or those with a complex signature. To do the same with environments, e.g., to define the environment `myitemize` to be equivalent to `itemize`, you can now write

```
\NewEnvironmentCopy{myitemize}{itemize}
```

There are also `\Renew...` and `\Declare...`, which may be useful depending on the circumstances.

In addition, we offer a `\ShowEnvironment` command, which displays the `\begin` and `\end` code of the environment passed as an argument. E.g., `\ShowEnvironment{center}` results in the following output:

```
> \begin{center}=environment:
> ->\trivlist \centering \item \relax .
<recently read> }
1. ... \ShowEnvironment{center}
> \end{center}:
> ->\endtrivlist .
<recently read> }
1. ... \ShowEnvironment{center}
```

(github issue 963)

Code improvements

doc: Index _ correctly to write

(github issue 943)

Default definition for \do

The command `\do` with its nice public name is in reality an internal command inherited from plain T_EX for list processing. However, it only got a definition when `\begin{document}` was executed, with a result that a user definition in the preamble was unconditionally overwritten at this point. To properly alert the user that this command is not freely available we now already provide a definition in the format so that `\newcommand` and friends produce a proper error message instead of providing a definition that doesn't last. *(github issue 975)*

doc: Support the upquote package

The default quote and backquote characters in typewriter fonts are typographical quotes, e.g., the input

```
\verb*/'prog 'my input'/'
```

is rendered as `'prog my input'` and not as ``prog my input`` as preferred by many programmers.

This can be adjusted, for example, with the `upquote` package, which results in the second output. However, for historical reasons `doc` had its own definition of `\verb` and `verbatim` and as a consequence the two packages did not cooperate. This has now been fixed and loading `upquote` together with `doc` has the desired effect. *(github issue 953)*

New key for filecontents

The `filecontents` environment warns on the terminal if a file gets overwritten even if that is intentional, e.g., when you write a temporary file over and over again. To make the warning less noisy in this case we added a new `nowarn` key that redirects the overwriting warning to the transcript file. We think that some record of the action is still required to help with debugging, thus it is not completely silenced. The warning that nothing gets written, because the file already exists (and the `force` key was not used), is not altered and still shows up on the terminal. *(github issue 958)*

A further hook for shipping out pages

Since October 2020 the `shipout` process offers a number of hooks to adjust what is happening before, during, and after the `\shipout`. For example, with the `shipout/before` hook, packages can reset code they have altered (e.g., `\catcodes` during verbatim-like processing) and with `shipout/background` and `shipout/foreground` material can be added to the pages. Details are given in [1]. However, what was missing was a hook that allows a package writer to

manipulate the completed page (with foreground and background attached) just before the actual shipout happens.

For this we now provide the additional hook `shipout`. One use case (sometimes needed in print production) is to mirror the whole page via `\reflectbox` including all the extra data that may have been added into the foreground background. *(github issue 920)*

Displaying release information in the .log

\LaTeX displays its release information at the very beginning of the \LaTeX run on the terminal and also writes it to the transcript file if that is already opened at this point. While this is normally true, it is not the case if the \LaTeX run was started passing additional \TeX code on the command line, e.g.,

```
pdfflatex '\PassOptionsToClass{11pt}{article}
\input{myarticle}'
```

In this case the release information is displayed when `\PassOptionsToClass` is processed but the transcript file is only opened when the output file name is known, i.e., after `\input` has been seen, and as a result the release information is only shown on the terminal.

To account for this scenario, we now repeat the release information also at the very end of the transcript file where we can be sure that it is open and ready to receive material. *(github issue 944)*

Bug fixes

Incompatibility between doc and unicode-math

The `unicode-math` package alters the catcode of `|` but does not adjust its value for use in `doc`, with the result that “or” module, i.e., $\langle A|B \rangle$ are displayed in a strange way. This is now fixed with some firstaid code that will eventually be moved into `unicode-math`. *(github issue 820)*

A fix for \hspace

The change to `\hspace`, done in 2020 to make it `calc` aware, had the unfortunate side effect that starting a paragraph with `\hspace` would result in the execution of `\everypar` inside a group (i.e., any local changes would immediately be revoked, breaking, for example, `wrapfig` in that special situation). This got fixed with the 2022-11 PL1 hotfix, so was already corrected in the previous release, but only now documented in a newsletter. *(github issue 967)*

Improve spacing at top of minipages

A list and several other document elements add some vertical space in front of them. However this should not happen at the beginning of a box (such as a `minipage`) and normally it doesn't, because \TeX automatically drops such spaces at the start of a vertical list. However,

if there is some invisible material, such as a `\color` command, some `hyperref` anchor, a `\write` or something else, then the list is no longer empty and \TeX no longer drops spaces, hence the difference.

With the new paragraph handling introduced in 2021 it is now finally possible to detect and avoid this problem and apply appropriate counter measures so that from now on the spacing will be always correct. *(github issue 989)*

Changes to packages in the amsmath category

Changes to packages in the graphics category

Changes to packages in the tools category

Fix handling of nested multicols

If `multicols` environments have been nested into each other (the inner one boxed) it could fail if the boxed environment appeared near a page break. The problem was that the output routine was called while the `\hsize` was still altered to fit the column width of the inner `multicols` — thereby messing up the placement of columns of the page. This has now been fixed. *(github issue 1002)*

References

- [1] Frank Mittelbach, \LaTeX Project Team: *The ltshipout documentation*.
Run `texdoc ltshipout-doc` to view.