LaTeX News

Issue 42, November 2025 — DRAFT version for upcoming release (LaTeX release 2025-11-01)

Contents

Introduction

to write

News from the Tagged PDF project

Expanding the `\DocumentMetadata` command

In 2022 we introduced the `\DocumentMetadata` with a twofold purpose: to provide a dedicated place for document wide settings and metadata, and to act as a trigger command to identify documents that want to load new code. The latter allows the use of the new, extended interfaces essential for the Tagging Project but also useful also without tagging.

Initially, using `\DocumentMetadata` with an empty argument loaded only the PDF management code and a new hyperref driver was used. Since November 2024 `\DocumentMetadata` changes the default encoding from OT1 to T1; and since June 2025 it also changes the default PDF version from 1.7 to 2.0.

Additional code in latex-lab (needed, e.g., for the tagging project) had to be loaded explicitly by using the `testphase` or the new `tagging` key in the argument of `\DocumentMetadata`. Whilst this allowed for the selective loading and testing of the new code, it also produced problems for classes and packages adapting their code for the tagging project since it was difficult to test which parts of the latex-lab code were active.

In this release we therefore extend `\DocumentMetadata` even further: it will now load directly all the code that one get also when using the `tagging=off` or the `testphase=latest` key.

The values `phase-I`, `phase-II`, `phase=III` of the `testphase` key will no longer load different code variants but only activate tagging. Extra modules not yet incorporated in the `latest` set of modules can still be loaded by using the `testphase` key.

For documents that want to load the PDF management but do not want the new tagging support code we provide a dedicated package. Such documents should replace

```
\DocumentMetadata{pdfversion=1.7,pdfstandard=a-3b}
```

by

```
\RequirePackage{pdfmanagement}
\SetKeys{document/metadata}{pdfversion=1.7,pdfstandard=a-3
```

Requiring or testing for the Tagging support code

Classes or packages that are written only for the new code loaded by `\DocumentMetadata` can use the new command `\NeedsDocumentMetadata` at the begin of the class or package file. It will produce a suitable error message if the tagging support code has not been loaded.

Classes and package that want to support both legacy documents and newer documents using `\DocumentMetadata` can now use `\IfDocumentMetadataTF` to test whether the new code has been loaded – eventually in combination with a test of the date of the format. To test whether the PDF management has been loaded, the test `\IfPDFManagementActiveTF` is provided.

Normalizing key names for block environments

to write

Contexts in typesetting

to write if latex-lab-context makes it to the next stage before 2025/11/01

New or improved commands

Support separate font families for script fonts

In TEX's math processing separate fonts can be selected for text, script and scriptscript sizes. LATEX's NFSS traditionally uses the same font family at different sizes, handling adjustment needed for making fonts appear better in a script location through the use of optical sizes. This works great for traditional TEX fonts, but for OpenType fonts this leads to issues. OpenType MATH assumes the font in a script location has separate features set and therefore received specific adjustments.

To support this without relying on heuristics based on the font size, a new command `\DeclareMathScriptfontMapping` has been added. It takes 3 pairs of encoding/family arguments to indicate that for the first pair when used as the math main font the second and the third should be used as the script and scriptscript font, respectively. *(github issue 1707)*

Code improvements

Avoid strange warnings about font substitutions

A font series value such as `sbc` contains both the weight (`sb`, i.e. "semibold") and the width (`c`, i.e. "condensed") of the font. If you want to reset only one of the two to "medium" and keep the other, you can use `\fontseries{m?}` or `\fontseries{?m}`: The former switches `sbc` to `c`, the latter switches `sbc` to `sb`. However, if the resulting series did not exist, you got strange warnings, e.g.:

```
LaTeX Font Warning:
Font shape 'OT1/cmss/c/n' undefined
using 'OT1/cmss/m?/n' instead on input line 7.
LaTeX Font Warning:
Font shape 'OT1/cmss/m?/n' undefined
using 'OT1/cmss/m/n' instead on input line 7.
```

This has now been corrected so that you get a single, more meaningful warning:

```
LaTeX Font Warning:
Font shape 'OT1/cmss/c/n' undefined
using 'OT1/cmss/m/n' instead on input line 7.
```

If the `m` series does not exist either, you will still get strange warnings, but this should only affect very few fonts. The source file was also tidied up a little on this occasion. *(github issue 1727)*

Allow multiple family names in `\ProcessKeyOptions`

The ability to process key–value options was introduced into the kernel in the June 2022 release [3], with the command `\ProcessKeyOptions` carrying out the option assignment. In the original version, this takes an optional argument which can select one key family (namespace) for options. We have now extended this to take a comma list of possible families. *(github issue 1756)*

Control of value expansion in keys

Normally, key–value input is treated "as is", with no expansion of either key names or values. However, there are occasions when the expansion of selected values is useful. We have now extended the key handling for templates (`\DeclareInstance`, etc.) and for keys created using the L3 programming layer to allow selective expansion. In both cases, the syntax uses a trailing colon and a single letter specifier: these letters are those used in `\ExpandArgs` or the L3 programming layer. For example, to use the values of the LATEX 2_ε variable `\@itemdepth`, one could have settings

```
key-a:c =  @itemdepth ,
key-b:v =  @itemdepth
```

This facility will *automatically* be available in any package setup macro using the L3 programming layer, for example siunitx. *(github issue 1801)*

Support word exclusion in case changing

Work on improving automatic case changing over previous releases has continued. We have now added the ability to 'register' words for exclusion from case changing, using `\DeclareLowercaseExclusions`, `\DeclareTitlecaseExclusions` and `\DeclareUppercaseExclusions`.

Bug fixes

Support active characters correctly with `\DeclareRobustCommand`

The mechanism used by `\DeclareRobustCommand` creates an internal command which has a space added to the name of the document one: so `\foo␣` for a command `\foo`. That fails if applied to an active character: unlike normal commands, these have to be exactly one character long. Due to the way the implementation works, to date this would result in redefining `\␣` every time `\DeclareRobustCommand` was used with an active character. This has now been corrected: robust active characters are now created using the engine `\protected` mechanism and do not use an internal auxiliary. They still work in file names and labels to give the character itself. *(github issue 345)*

Changes to packages in the tools category

Updating the status of some components

The tools bundle contains a range of packages with different usage profiles. Some of these were necessary in the transition from LATEX 2.09 to LATEX 2_ε, while others are very widely used in current documents (for example array). We have therefore marked a small number of packages in tools as *retained only for historical and stability reasons*, and where relevant pointed to more up-to-date alternatives; the list is:

- enumerate: use enumitem instead
- rawfonts: retained as part of LaTeX 2.09 support
- somedefs: retained as part of LaTeX 2.09 support
- theorem: use amsthm instead
- verbatim: use fancyvrb instead

References

[1] Leslie Lamport. *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 2nd edition, 1994. ISBN 0-201-52983-1. Reprinted with corrections in 1996.

[2] LaTeX Project Team. *LaTeX 2ε news 1–42*. November 2025. `https://latex-project.org/news/latex2e-news/ltnews.pdf`

[3] LaTeX Project Team. *LaTeX 2ε news 35*. June 2022. `https://latex-project.org/news/latex2e-news/ltnews35.pdf`

[4] LaTeX Project Team. *LaTeX 2ε news 41*. June 2025. `https://latex-project.org/news/latex2e-news/ltnews41.pdf`