LaTeX News

Issue 29, December 2018 (LaTeX release 2018-12-01)

Contents

Introduction 1

Bug reports for core LaTeX 2ε and packages 1

Changes to the LaTeX kernel 1
- UTF-8: updates to the default input encoding . 1
- Fixed \verb* and friends in XeTeX and LuaTeX 1
- Error message corrected 2
- Fixed fatal link error with hyperref 2
- Avoid page breaks caused by invisible commands 2
- Prevent spurious spaces when reading table of contents data 2
- Prevent protrusion in table of contents lines . . 2
- Start L-R mode for \thinspace and friends . . 3
- Guarding \pfill in doc 3

Changes to packages in the tools category 3
- Sometimes the trace package turned off too much 3
- Update to xr 3
- Column data for multicols* sometimes vanished 3
- Extension to \docolaction in multicol 3
- Prevent color leak in array 3
- Support fragile commands in array or tabular column templates 3

Changes to packages in the amsmath category 3

Website updates 3
- Publications area reorganized and extended . . 3
- Japanese translations of the user's guide 4

Introduction

The December 2018 release of LaTeX is a maintenance release in which we have fixed a few bugs in the software: some are old, some newer, and they are mostly rather obscure.

Bug reports for core LaTeX 2ε and packages maintained by the Project Team

In Spring 2018 we established a new issue tracking system (Github issues at https://github.com/latex3/latex2e/issues) for both the LaTeX core and the packages maintained by the LaTeX Project Team, with an updated procedure for how to report a bug or problem.

Initial experience with this system is good, with people who report problems following the guidelines and including helpful working examples to show the problem—thanks for doing this.

The detailed requirements and the workflow for reporting a bug in the core LaTeX software is documented at

https://www.latex-project.org/bugs/

with further details and discussion in [1].

Changes to the LaTeX kernel

UTF-8: updates to the default input encoding

In the April 2018 release of LaTeX we changed the default encoding from 7-bit ASCII to UTF-8 when using classic TeX or pdfTeX; see *LaTeX News 28* [2] for details.

Now, after half a year of experience with this new default, we have made a small number of adjustments to further improve the user experience. These include:

- Some improvements when displaying error messages about UTF-8 characters that have not been set up for use with LaTeX, or are invalid for some other reason; *(github issues 60, 62 and 63)*
- The addition of a number of previously missing declarations for characters that are in fact available with the default fonts, e.g., \j "ȷ" (0237), \SS "SS" (1E9E), \k{} "˛" (02DB) and \.{} "˙" (02D9);
- Correcting the names for \guillemetleft "«" and \guillemetright "»" in all encoding files. These correct names are in addition to the old (but wrong) Adobe names: Adobe mistakenly called them Guillemot, which is a sea bird. *(github issue 65)*
- Added \Hwithstroke ("Ħ") and \hwithstroke ("ħ") necessary for typesetting Maltese. *(https://tex.stackexchange.com/q/460110)*

Fixed \verb* and friends in XeTeX and LuaTeX

The original \verb* and verbatim* in LaTeX were coded under the assumption that the position of the space character (i.e., ASCII 32) in a typewriter font contains a visible space glyph "␣". This is correct for pdfTeX with the most used font encodings OT1 and T1. However, this unfortunately does not work for Unicode engines using the TU encoding since the space character slot (ASCII 32) then usually contains a real (normal) space, which

LaTeX News, and the LaTeX software, are brought to you by the LaTeX Project Team; Copyright 2018, license LPPL.

has the effect that `\verb*` produces the same results as `\verb`.

The `\verb*` code now always uses the newly introduced command `\verbvisiblespace` to produce the visible space character and this command will get appropriate definitions for use with the different engines. With pdfTeX it will simply use `\asciispace`, which is a posh name for "select character 32 in the current font", but with Unicode engines the default definition is

```
\DeclareRobustCommand\verbvisiblespace
   {\leavevmode
    {\usefont{OT1}{cmtt}{m}{n}\asciispace}}
```

which uses the visible space from the font Computer Modern Typewriter, regardless of the currently chosen typewriter font. Internally the code ensures that the character used has exactly the same width as the other characters in the current (monospaced) font; thus, for example, code displays line up properly.

It is possible to redefine this command to select your own character, for example

```
\DeclareRobustCommand\verbvisiblespace
   {\textvisiblespace}
```

will select the "official" visible space character of the current font. This may look like the natural default, but it wasn't chosen as our default because many fonts just don't have that Unicode character, or they have one with a strange shape. *(github issues 69 and 70)*

Error message corrected

Trying to redefine an undefined command could in a few cases generate an error message with a missing space, e.g., `\renewcommand\1{...}` gave

```
LaTeX Error: \1undefined.
```

This is now fixed. *(github issue 41)*

Fixed fatal link error with hyperref

If an `\href` link text gets broken across pages, pdfTeX and LuaTeX will generate a fatal error unless both parts of the link are internally at the same boxing level. In two-column mode that was not the case if one of the pages had spanning top floats. This has now been changed so that the error is avoided. *(github issue 94)*

Avoid page breaks caused by invisible commands

Commands like `\label` or `\index` could generate a potential page break in places where a page break was otherwise prohibited, e.g., when used between two consecutive headings. This has now been corrected. If for some reason you really want a break and you relied on this faulty behavior, you can always add one using `\pagebreak`, with or without an optional argument. *(github issue 81)*

Prevent spurious spaces when reading table of contents data

When table of contents data is read in from a `.toc` file, the new-line character at the end of each line is converted by TeX to a space. In normal processing this is harmless (as TeX is doing this input reading whilst in vertical mode and each line in the file represents a single line (paragraph) in the table of contents. If, however, this is done in horizontal mode, which is sometimes the case, then these spaces will appear in the output. If you then omit some of the input lines (e.g., because you do not display TOC data below a certain level), then these spaces accumulate in the typeset output and you get surprising, and unwanted, gaps inside the text.

The new code now adds a `%` sign at the end of problematic lines in the `.toc` file so that TeX will not generate such spaces that may survive to spoil the printed result. As some third party packages have augmented or changed the core LaTeX functionality in that area (for example, by adding additional arguments to the commands in TOC files) the code uses a conservative approach and the `%` signs are added only when certain conditions are met. Therefore some packages might require updates if they want to benefit from this correction, especially if they unconditionally overwrite LaTeX's `\addcontentsline` definition. *(github issue 73)*

Prevent protrusion in table of contents lines

In TeX's internal processing model, paragraph data is one of the major data structures. As a result, many things are internally modeled as paragraphs even if they are not conceptually "text paragraphs" in the traditional sense. In a few cases this has some surprising effects that are not always for the better. One example is standard TOC entries, where you have heading data followed by some dot leaders and a page number at the right, produced, for example, from this:

Error message corrected 2

The space reserved for the page number is of a fixed width, so that the dots always end in the same place. Well, they did end in the same place until the advent of protrusion support in the TeX engines. Now, with the microtype package loaded, it is possible that the page number will protrude slightly into the margin (even though it's typeset inside a box) and as a result this page number box gets shifted. With enough bad luck this can get you another dot in the line, sticking out like the proverbial sore thumb, as exhibited in the question on StackExchange that triggered the correction.

LaTeX now takes care that there will be no protrusion happening on such lines, even if it is generally enabled for the whole document. *(https://tex.stackexchange.com/q/172785)*

Start L-R mode for \thinspace and friends

In LaTeX, commands that are intended only for paragraph (L-R) mode are generally careful to start paragraph mode if necessary; thus they can be used at the start of a paragraph without surprising and unwanted consequences. This important requirement had been overlooked for a few horizontal spacing commands, such as `\thinspace` (a.k.a. "`\,`"), and for some other support commands such as `\smash` or `\phantom`. Thus they ended up adding vertical space when used at the beginning of a paragraph or, in the case of `\smash`, creating a paragraph of their own. This has now been corrected, and a corresponding update has been made to the amsmath package, in which these commands are also defined. *(github issues 49 and 50)*

Guarding \pfill in doc

For presenting index entries pointing to code fragments and the like, the doc package has a `\pfill` command that generates within the index a line of dots leading from the command name to the page or code line numbers. If necessary it would automatically split the entry over two lines. That worked well enough for a quarter century, but we discovered recently that it is broken inside the ltugboat class, where it sometimes produces bad spacing within continuation lines.

The reason turned out to be a redefinition of the LaTeX command `\nobreakspace` (`~`) inside the class ltugboat, which removed any preceding space (and thus unfortunately also removed the dots on the continuation line). While one can argue that this is a questionable redefinition (if only done by a single class and not generally), it has been in the class so long that changing it would certainly break older documents. So instead we now guard against that removal of space. *(github issues 25 and 75)*

Changes to packages in the tools category

Sometimes the trace package turned off too much

The trace package is a useful little tool for tracing macro execution: it hides certain lengthy and typically uninteresting expansions resulting from font changes and similar activities. However, it had the problem that it also reset other tracing settings such as `\showoutput` in such situations, so that you couldn't use `\showoutput` in the preamble to get symbolic output of all the pages in the document. This has now been corrected.

Update to xr

The xr package has been updated so that the code that reads the `.aux` file has been made more robust. It now correctly ignores conditionals (added by hyperref and other packages) rather than generating low level parsing errors. *(https://tex.stackexchange.com/a/452321)*

Column data for multicols sometimes vanished*

In certain situations involving `multicols*`, when there are more explicit `\columnbreak` requests than there are columns on the current page, data could vanish due to the removal of an internal penalty marking the end of the environment. This has been corrected by explicitly reinserting that penalty if necessary. *(github issue 53)*

Extension to \docolaction in multicol

The `\docolaction` command can be used to carry out actions depending on the column you are currently in, i.e., first, any inner one (if more than two) or last. However, if the action generates text then there is the question: is this text part of the current column or the one after? That is, on the next run, do we test before or after it, to determine in which column we are?

This is now resolved as follows: if you use `\docolaction*` any generated text by the chosen action is considered to be after the test point. But if you use the command without the star then all the material it generates will be placed before the test point to determine the current column, i.e., the text will become part of the current column and may affect the test result on the next run.

Prevent color leak in array

In some cases the color used inside a `tabular` cell could "leak out" into the surrounding text. This has been corrected. *(github issue 72)*

Support fragile commands in array or tabular column templates

The preamble specifiers `p`, `m` and `b` each receives a user supplied argument: the width of the paragraph column. Normally that is something harmless, like a length or a simple length expression. But in more complicated settings involving the calc package it could break with a low-level error message. This has now been corrected. *(https://tex.stackexchange.com/q/459285)*

Changes to packages in the amsmath category

The changes in the kernel made for `\thinspace`, `\smash`, etc. (see above) have been reflected in the amsmath package code, so that loading this package doesn't revert them. *(github issues 49 and 50)*

Website updates

Publications area reorganized and extended

To help readers to find relevant information in more convenient and easy ways, the area of the website covering publications by the LaTeX Project Team was reorganized and extended (many more abstracts added). We now provide the articles, talks and supplementary data structured both by year and also by major topics [4]. Feel free to take a look.

Japanese translations of the user's guide
Yukitoshi Fujimura has kindly translated into Japanese two documents that are distributed with standard LaTeX. These are:

- LaTeX 2ε for authors;
- User's Guide for the amsmath Package [5].

They can be found on the website documentation page [3]. You will now also find there a typeset version of the full LaTeX 2ε source code (with index etc.) and a number of other goodies.

References

[1] Frank Mittelbach: *New rules for reporting bugs in the LaTeX core software*. In: TUGboat, 39#1, 2018. https://latex-project.org/publications/

[2] *LaTeX News, Issue 28*. In: TUGboat, 39#1, 2018. https://latex-project.org/news/latex2e-news/

[3] *LaTeX documentation on the LaTeX Project Website*. https://latex-project.org/documentation/

[4] *LaTeX Project publications on the LaTeX Project Website*. https://latex-project.org/publications/

[5] American Mathematical Society and The LaTeX Project: *User's Guide for the amsmath Package* (Version 2.1). April 2018. Available from https://www.ctan.org and distributed as part of every LaTeX distribution.