LaTeX News

Issue 30, October 2019 (LaTeX release 2019-10-01)

Contents

LaTeX-dev formats now available 1
- Our hopes . . . 1
- Details please … . . . 2
- Setting up menu items . . . 2

Improving Unicode handling in pdfTeX 2

Improving file name handling in pdfTeX 2

Improving the `filecontents` environment 2

Making more user commands robust 2

Other changes to the LaTeX kernel 3
- Guard against `\unskip` in tabular cells . . . 3
- Fix Unicode table data . . . 3
- Improve `\InputIfFileExists`'s handling of file names . . . 3
- Improve interface for cross-references . . . 3
- Improve wording of a warning message . . . 3
- Avoid bad side-effects of `\DeclareErrorFont` . 3
- nfssfont: Make font table generation the default action . . . 3
- trace: Add package support in the kernel . . . 3

Changes to packages in the tools category 4
- array: Warn if primitive column specifiers are overwritten . . . 4
- multicol: Introduce `minrows` counter for balancing 4
- varioref: Better support for cleveref . . . 4
- xr: Support citations to bibliographies in external documents . . . 4

Changes to packages in the amsmath category 4
- amsmath: Introduce `\overunderset` command 4

Documentation updates 4
- Highlighting the standard NFSS codes for series 4
- LaTeX base and doc distribution reunited . . . 4

LaTeX-dev formats now available

We know that many of you, especially developers and maintainers of important packages, have a strong interest in a stable LaTeX environment.

In order to keep LaTeX very stable for users whilst allowing for further development to continue, we now have a development branch of LaTeX on GitHub containing development code for the upcoming release. When this code is ready for wider consumption and testing, we generate a pre-release of LaTeX from this development branch and make it available on CTAN.

For users of the TeX Live and MiKTeX distributions it is therefore now straightforward to test their documents and code against the upcoming LaTeX release with ease, simply by selecting a different program name (when using the command line) or by selecting a menu entry (after setting it up; see below).

If you do this then the latest version of the LaTeX development format will be used to process your document, allowing you to test the upcoming release with your own documents and packages. For example, if you run

```
pdflatex-dev myfile
```

then you will be greeted on the screen with something like `LaTeX2e <2019-10-01> pre-release-2` (identifying the pre-release format) instead of the normal `LaTeX2e <2018-12-01>`. In this pre-release you will find the latest new features that we have developed.

Our hopes

We don't expect everybody to start using the development formats to participate in testing, but we hope that people with a strong interest in a stable LaTeX environment (especially developers and maintainers of important packages) will use the new facilities and help us to ensure that future public releases of LaTeX do not (as has happened in the past) require some immediate patches because of issues that were not identified by our internal regression test suite or by other testing we do.

Any issue identified when using the development format should preferably be logged as an issue on GitHub, following the procedure outlined on our website at `https://www.latex-project.org/bugs/` including the use of the latexbug package as described.

Our bug reporting process normally states that issues involving third-party software are out of scope as we can't correct external packages; see [1]. However, in the particular case of the development format showing an incompatibility with a third-party package, it is fine to open an issue with us (in addition, please, to informing the maintainer of that package) so that we know about the problem and can jointly work on resolving it.

LaTeX News, and the LaTeX software, are brought to you by the LaTeX Project Team; Copyright 2019, license LPPL.

Details please …

More details and some background information about the concepts and the process are available in an upcoming *TUGboat* article: "The LaTeX release workflow and the LaTeX dev formats" [2].

Setting up menu items

While the command line call works out of the box if you have a recent TeX Live or MiKTeX installation, its use within an integrated editing environment doesn't at this point in time (maybe the developers of these editors will include it in the future). However, it is normally fairly simple to enable it as most (or even all?) of them provide simple ways to call your own setup. How this works in detail depends very much on the environment you use, so we can't give much help here.

But as an example: to provide an additional menu entry for `XeLaTeX-dev` on a MacBook all that is necessary is to copy the file `XeLaTeX.engine` to `XeLaTeX-dev.engine` and change the call from `xelatex` to `xelatex-dev` inside.

Improving Unicode handling in pdfTeX

Perhaps the most important improvement in this release is even better support for UTF-8 characters when using pdfTeX.[1]

When using a "Unicode engine", any Unicode character (that is not acting as a command, i.e., is not "active") can be used as part of the `\label`/`\ref` mechanism or can be displayed in a message or written to a file. In 8-bit engines, however, this was severely restricted: essentially you had to limit yourself to using ASCII letters, digits and a few punctuation symbols. With the new release, most of these restrictions have been removed and you now can write labels such as

```
\label{eq:größer}
```

or use accented characters, etc., as part of a `\typeout` message. The only requirement remaining is that only those UTF-8 characters that are also available for typesetting can be used, i.e., only those characters for which adequate font support is loaded. Otherwise you will get an error message stating that the particular Unicode character is not set up for use with LaTeX.

Note, however, that the restrictions on what characters can be used in the names of commands have not changed.

What is not possible when using an 8-bit engine such as pdfTeX is to use characters other than ASCII letters as part of a command name. This is due to the fact that all other characters in such engines are not single character tokens, but in fact consist of a sequence of bytes and this is not supported in command names.

[1] The Japanese engines e-pTeX and e-upTeX can't use these features yet as they don't support the primitive `\ifincsname`. Work is under way to resolve this in the engines.

Improving file name handling in pdfTeX

A related change is that file names used as part of `\input`, `\includegraphics`, etc., commands can now contain any Unicode characters allowed by the file system in use, including spaces. In this case, even characters that can't be typeset (due to lack of font support) can be used.

Improving the `filecontents` environment

The `filecontents` environment now supports an optional argument in which you can specify that it is allowed to `overwrite` an already existing file; by default nothing is written if a file with the given name exists anywhere in the search tree. An alternative name for this option is `force`. Even then the environment will refuse to write to `\jobname.tex` to avoid clobbering its own input file. However, if you use a different extension on your input file you could still overwrite it (there is no way to test for that).

There is also an option `nosearch`, which specifies that only the current directory is examined for an existing file, not the whole TeX inputs tree. This is useful if you want to write a local copy of a standard system file. Finally, `noheader` prevents writing a preamble to the file (this is the same as using the star form of the environment).

Another change is that this environment is now allowed anywhere in the document, which means it provides everything (and more) of what the now obsolete filecontents package provided.

Making more user commands robust

In the early days of LaTeX many commands were fragile, i.e., they needed `\protect` in front of them when used in places such as section headings and other "moving arguments", etc. In LaTeX 2_ε many of these commands were made robust, but still a fairly large number remained unnecessarily fragile.

In this release of LaTeX we have now made a lot more commands robust. There is a very small collection of commands that must stay fragile because their expansion (maybe partially) at just the right time is critical. Yet others are unlikely to ever be needed in a "moving argument".

Doing this for `\begin` and `\end` was rather tricky as the standard mechanism with `\DeclareRobustCommand` doesn't work here, at least not for `\end` as that needs to expand during typesetting without generating a `\relax` (from the `\protect`). Such a token would start a new row in table environments, such as `tabular`, etc. Furthermore, some packages try to look into the definition of `\end` by expanding it several times. Thus expansion with `\expandafter` had to produce exactly the same result as before. But in the end we overcame

that hurdle too, so now environments are automatically robust if used in places like headings or `\typeout` and so forth.

What hasn't been tackled yet is the redefinitions in amsmath: this package redefines a number of basic math constructs that are now robust, so that they become fragile again once the package is loaded. This area will be addressed in a followup release. *(github issue 123)*

Other changes to the LaTeX kernel

Guard against `\unskip` in tabular cells

If a `tabular` or `array` cell started with a command that started with an `\unskip` then centering the column broke because the stretching glue on the left got removed. The fix for this was to add a minuscule, and hence unnoticeable, additional space after the stretching space: removing this extra space causes no problems.

This change was also applied in the array package. *(github issue 102)*

Fix Unicode table data

`U+012F` which is "i with ogonek" produced a "dotless i with ogonek" by mistake. This has been corrected. *(github issue 122)*

The Unicode slots `27E8` and `27E9` have been mapped to `\textlangle` and `\textrangle` which is the recommended mapping. In the past they raised a LaTeX error. *(github issue 110)*

When doing cut-and-paste from other documents or websites, f-ligatures and others ligatures might end up as single Unicode characters in your file. In the past those got rejected by LaTeX. We now define those Unicode slots and map them back to the sequence of individual characters constituting the ligature. If supported by the current font (which is normally the case) they are then reconstructed as ligatures and thus get typeset as desired. Otherwise they will come out as individual characters which is still better than an error message. *(github issue 154)*

Improve `\InputIfFileExists`'s handling of file names

In rare circumstances it was possible that `\InputIfFileExists` would work incorrectly, e.g., a construction such as

```
\InputIfFileExists{foo}{\input{bar}}{}
```

would not load the files `foo.tex` and `bar.tex` but would load `bar.tex` twice. This has been corrected. *(github issue 109)*

Improve interface for cross-references

The packages fncylab and varioref provided a slightly improved definition of `\refstepcounter` which allowed the internal `\p@..` commands to receive the counter value as an argument, instead of acting as a simple prefix. This supports more complex formatting of the value in the reference.

These packages also provided the command `\labelformat` to help in the specification of such formatting in an easy way. For example, `\labelformat{equation}{eq.~(#1)}` specifies that references to equations automatically come out as "eq. (5)" or similar. As such a `\labelformat` declaration means a `\ref` command can no longer be successfully used at the start of a sentence, the packages also provided `\Ref` for such scenarios.

Both of these commands, `\labelformat` and `\Ref`, are now removed from the packages and instead made available in the kernel so there is no need to load additional packages.

Improve wording of a warning message

The kernel now says "Trying to load …" instead of "Try loading …" in one of its informal messages to match style of similar messages. *(github issue 107)*

Avoid bad side-effects of `\DeclareErrorFont`

As a side effect of setting up the error font for NFSS, this declaration also changed the current font size back to 10pt. In most circumstances that doesn't matter, because that declaration was meant to be used only during the format generation and not during a LaTeX run. However, it has turned out to be used by some developers in other places (incorrectly in fact: e.g., inside some `.fd` files) where resetting the size causes havoc seemingly at random. The command has now changed to not produce such side effects. *(gnats issue latex/4399)*

nfssfont: Make font table generation the default action

With the small file `nfssfont.tex` it is possible to produce font tables and other font tests in the style set up by Don Knuth. In nearly all cases a font table is wanted, so this action has been made the default. Now one can simply hit enter instead of having to write `\table\bye`.

trace: Add package support in the kernel

The trace package implements the commands `\traceon` and `\traceoff` that work like `\tracingall` but skip certain code blocks that produce a lot of tracing output. This is useful when debugging, to suppress uninteresting tracing from, for example, loading a font. Code blocks that should not be traced need to be surrounded by the commands `\conditionally@traceoff` and `\conditionally@traceon`.

The LaTeX kernel now provides dummy definitions for these two commands so that package writers can use them in their packages regardless of trace being loaded or not.

Changes to packages in the tools category

array: Warn if primitive column specifiers are overwritten

With `\newcolumntype` it is possible to define your own column specifiers for a `tabular` preamble; it is also possible to change existing ones. However, doing that for a primitive column specifier, such as `c`, is seldom a good idea, since then its functionality becomes unavailable. The package was therefore supposed to warn the user in this case, but due to a missing `\expandafter` in the code it never did—now it does. *(github issue 148)*

multicol: Introduce `minrows` counter for balancing

When there are only a few lines of text on a page at the end of a `multicols` environment, balancing the columns often looks rather odd: such as three columns each containing a single line. The balancing behavior can now be controlled through the counter `minrows` (default is 1) which specifies that, after balancing, there must be at least that many lines in the first column. Thus, if you set `minrows` to `2` then you would get a distribution of `2+1+0` lines and if set to three, the result would be `3+0+0` instead of the default `1+1+1`.

What is most appropriate really depends on the circumstances, but this now gives you the tools to make local or global adjustments.

varioref: Better support for cleveref

The varioref package has been internally updated to provide better interfaces for packages such as hyperref and cleveref.

It also has a new package option `nospace` that stops varioref from meddling with space in front of its commands. The original behavior was always somewhat problematical and it is suggested that all new documents use this option (which should really have been the default).

Support was also added for the Arabic language through the option `arabic`.

xr: Support citations to bibliographies in external documents

The xr package can be used to cross-reference an external LaTeX document. This means that even when a work is split over different documents (that need to be processed separately), `\ref` or `\pageref` can use labels from any document, creating links between them. This facility has now been extended so that `\cite` commands and their cousins can now also reference bibliographies in external documents; this feature was first provided in the package xcite by Enrico Gregorio.

Note that for technical reasons xr doesn't work with hyperref. Use xr-hyper instead if you need the latter package.

Changes to packages in the amsmath category

amsmath: Introduce `\overunderset` command

The amsmath package has always offered the commands `\overset` and `\underset` to produce binary operators with something set above or below. But sometimes one needs to put something above and something below: The newly added `\overunderset` makes this easily possible.

Documentation updates

There are a number of documentation updates in files on the documentation page of the project website [4].

Highlighting the standard NFSS codes for series

The *Font Selection Guide* [3] has been updated to strongly recommend that the standard codes should be used when providing font support. The reason for this recommendation is explained here.

The font selection scheme uses a number of standard codes for `\fontseries` and `\fontshape` to ensure that different fonts are comparable, e.g., that you get a "light" weight if you specify `l` and "extra bold" when you write `eb`, etc. Over the years people came up with a number of other creative short codes like `k`, `j`, `t` and others with the result that changing a font family required different codes and thus prevented users from easily mixing and matching different families. Some work has been undertaken to get back to a coherent scheme and all the font families supported through the program `autoinst` are now producing the standard codes again.

LaTeX base and doc distribution reunited

For a long time the LaTeX distribution available from CTAN was split into several parts to allow them to be uploaded or downloaded separately. As this is these days more confusing than helpful we have recombined the base part with the documentation part (as both are anyway always updated together). Thus the package latex-doc is no longer separately available from CTAN but contained in the latex-base distribution.

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] Frank Mittelbach: *The LaTeX release workflow and the LaTeX dev formats*. In: TUGboat, 40#2, 2019. https://latex-project.org/publications/

[3] LaTeX Project Team: *LaTeX 2ε font selection*. https://latex-project.org/documentation/

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