

# L<sup>A</sup>T<sub>E</sub>X News

Issue 26, January 2017

## Contents

$\varepsilon$ -T <sub>E</sub> X	1
Default encodings in X <sub>Y</sub> L <sup>A</sup> T <sub>E</sub> X and LuaL <sup>A</sup> T <sub>E</sub> X	1
<code>\showhyphens</code> in X <sub>Y</sub> L <sup>A</sup> T <sub>E</sub> X	2
The <code>fixltx2e</code> package	2
The <code>latexbug</code> package	2
Updates to <code>amsmath</code>	2
Updates to <code>tools</code>	2
An addendum to the release changes in 2015: page breaks and vertical spacing	2

## $\varepsilon$ -T<sub>E</sub>X

In L<sup>A</sup>T<sub>E</sub>X News 16 (December 2003) the team announced

We expect that within the next two years, releases of L<sup>A</sup>T<sub>E</sub>X will change modestly in order to run best under an extended T<sub>E</sub>X engine that contains the  $\varepsilon$ -T<sub>E</sub>X primitives, e.g.,  $\varepsilon$ -T<sub>E</sub>X or pdfT<sub>E</sub>X.

and also said

Although the current release does not *require*  $\varepsilon$ -T<sub>E</sub>X features, we certainly recommend using an extended T<sub>E</sub>X, especially if you need to debug macros.

For many years the team have worked on the basis that users will have  $\varepsilon$ -T<sub>E</sub>X available but had not revisited the above statements formally. As of the January 2017 release of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>,  $\varepsilon$ -T<sub>E</sub>X is *required* to build the format, and attempting to build a format without the extensions will fail.

Practically, modern T<sub>E</sub>X distributions provide the extensions in all engines other than the “pure” Knuth `tex`, and indeed parts of the format-building process already require  $\varepsilon$ -T<sub>E</sub>X, most notably some of the UTF-8 hyphenation patterns. As such, there should be no noticeable effect on users of this change.

The team expect to make wider use of  $\varepsilon$ -T<sub>E</sub>X within the kernel in future; details will be announced where they impact on end users in a visible way.

## Default encodings in X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X

The default encoding in L<sup>A</sup>T<sub>E</sub>X has always been the original 128-character encoding OT1. For Unicode based T<sub>E</sub>X engines, this is not really suitable, and is especially problematic with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X as in the major distributions this is built with Unicode based hyphenation patterns in the format. In practice this has not been a major problem as documents use the contributed `fontspec` package in order to switch to a Unicode encoded font.

In this release we are adding TU as a new supported encoding in addition to the previously supported encodings such as OT1 and T1. This denotes a Unicode based font encoding. It is essentially the same as the TU encoding that has been on trial with the experimental `tuenc` option to `fontspec` for the past year.

The X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X formats will now default to TU encoding and `lmr` (Latin Modern) family. In the case of LuaL<sup>A</sup>T<sub>E</sub>X the contributed `luaotfload` Lua module will be loaded at the start of each run to enable the loading of OpenType fonts.

The `fontspec` package is being adjusted in a companion release to recognise the new encoding default arrangements.

Note that in practice no font supports the full Unicode range, and so TU encoded fonts, unlike fonts specified for T1, may be expected to be incomplete in various ways. In the current release the file `tuenc.def` that implements the TU encoding-specific commands has made some basic assumptions for (for example) default handling of accent commands, and the set of command names is derived from the command names used for the UTF-8 support in the `inputenc` package, restricted roughly to the character ranges classically provided by the T1 and TS1 encodings, but is part of a longer term plan seen over recent releases to increase support for Unicode based T<sub>E</sub>X engines into the core L<sup>A</sup>T<sub>E</sub>X support.

If for any reason you need to process a document with the previous default OT1 encoding, you may switch encoding in the usual ways, for example

```
\usepackage[OT1]{fontenc}
```

or you may roll back all the changes for this release by starting the document with

```
\RequirePackage[2016/12/31]{latexrelease}
```

## *`\showhyphens` in $\XeTeX$*

Due to the way  $\XeTeX$  interfaces to font libraries, the standard definition of `\showhyphens` does not work. A variant definition has been available in the contributed `xltxtra` package, however a (slightly different) definition for `\showhyphens` is now included in  $\XeTeX$  by default. As usual this change will be undone if an earlier release is specified using the `latexrelease` package.

## *The `fixltx2e` package*

As described in  $\LaTeX$  News 22, the `fixltx2e` package has become obsolete with the new update policy. Since 2015 it has just made a warning and exited. In this release we have re-introduced all the code from the original fixes in the 2014  $\LaTeX$  but guarded by `\IncludeInRelease{2015/01/01}`. So for current releases `fixltx2e` still just displays a warning but for old releases, whether that is an old format, or a format with the version date reset via the `latexrelease` package, the fixes in the original `fixltx2e` will be applied.

This improves the ability to run old documents in a way that is compatible with contemporary formats. If you have a 2014 document that used `\usepackage{fixltx2e}` and you add `\RequirePackage[2014/01/01]{latexrelease}` and process it with the current format then `latexrelease` will undo most changes made since 2014, but now when the document includes `fixltx2e` it will act like a 2014 version of the package and apply the code fixes, not just give a warning that the package is obsolete.

## *The `latexbug` package*

As explained in more detail at the  $\LaTeX$  Project website<sup>1</sup> a new package, `latexbug`, has been produced to help produce test files to accompany bug reports on the core  $\LaTeX$  distribution. This is being published separately to CTAN at the same time as this release. By using the `latexbug` package you can easily check that the packages involved in the test are all part of the core release. The  $\LaTeX$  project cannot handle bug reports on contributed packages, which should be directed to the package maintainer as given in the package documentation.

## *Updates to `amsmath`*

The `amsmath` package has two updates at this release.

- The spacing to the left of the `aligned` and `gathered` environments has been fixed: the spurious thin space is no longer added by default. Package options control this to revert to the original behaviour where required; see the `amslatex` guide for further details.

- The large delimiters around generalised fractions (for example in the `\binom` construct) did not work in previous releases if using  $\LuaTeX$  or  $\XeTeX$  with OpenType math fonts. This is related to the lack of specific metrics for this use in the OpenType Math table. In principle  $\LuaTeX$  has two additional named metrics to control the delimiters but these are not initialised by default, and in  $\XeTeX$  it does not seem possible to make them work at all. So for Unicode  $\TeX$  systems, a new implementation of `\genfrac` is used at this release that uses `\left\right` internally but parameterised to give spacing as close to the original as possible. The implementation in (pdf) $\TeX$  is unaffected.

## *Updates to tools*

The `array` package has been updated to fix a longstanding but previously unreported issue with unwanted interactions between tables in the page head or foot and the body of the page, as reported in PR tools/4488. There is also an update to the  $\LuaTeX$  support in `bm`.

## *An addendum to the release changes in 2015: page breaks and vertical spacing*

In 2015 we announced the introduction of the roll-back/roll-forward concept to manage bug fixes and additions to core  $\LaTeX$  in a manageable way. We also announced at that time that we now incorporate all fixes from `fixltx2e` into the kernel (as the old mechanism produced problems instead of improving the situation). Refer to `ltnews22.pdf` for details.

One of the fixes from `fixltx2e` was for a glaring bug in `\addvspace` that was originally detected in the mid-nineties and back then added to the `fixltx2e` support package. In certain situations `\addvspace` would result in a page/column break below the baseline of the last line. As a result documents using `\flushbottom` would show a clear misalignment (even more prominent when typesetting in two-column mode).

Starting with release 2015/01/01 this is now finally corrected already in the kernel and not only in `fixltx2e`. In nearly all circumstances this will either make no difference to existing documents, or it will locally improve the visual appearance of that document without changing anything on other pages. However, by the nature of the change it is also possible that there are further non-local changes to the page breaks due to the different break positions introduced by the fix.

Thus, for documents that have been written before 2015 and that should be preserved unchanged at all costs you may have to add

```
\RequirePackage[2014/01/01]{latexrelease}
```

at the top of the document, to roll back the format to a date before the policy change.

<sup>1</sup><https://www.latex-project.org/bugs/>