# LaTeX News

Issue 24, February 2016

## *Contents*

**LuaTeX support** 1

**Unicode data** 1

**More support for east European accents** 2

**Changes in Graphics** 2

**Changes in Tools** 2

**Improving support for Unicode engines** 2

## *LuaTeX support*

This release refines the LuaTeX support introduced in the 2015/10/01 release. A number of patches have been added to improve the behavior of ltluatex (thanks largely to code review by Philipp Gesang). The kernel code has been adjusted to allow for changes in LuaTeX v0.85–v0.88. Most notably, newer LuaTeX releases allow more than 16 write streams and these are now enabled for use by `\newwrite`, but also the experimental `newtoken` Lua library has been renamed back to `token` which required small adjustments in the LuaTeX setup.

The biggest change in LuaTeX v0.85–v0.87 compared to previous versions is that all the primitives (originally defined in pdfTeX) dealing with the PDF "back end" are no longer defined, being replaced by a much smaller set of new primitives. This does not directly affect the core LaTeX files in this release but has required major changes to the `.ini` files used by TeX Live and similar distributions to set up the format files. These changes in the LuaTeX engine will affect any packages using these back end commands (packages such as graphics, color, hyperref, etc.). Until all contributed packages are updated to the new syntax users may need to add aliases for the old pdfTeX commands. A new luapdftexalias package has been contributed to CTAN (not part of the core LaTeX release) that may be used for this purpose.

See also the sections below for related changes in the tools and graphics bundles.

## *Unicode data*

As noted in LaTeX News 22, the 2015/01/01 release of LaTeX introduced built-in support for extended TeX systems. In particular, the kernel now loads appropriate data from the Unicode Consortium to set `\lccode`, `\uccode`, `\catcode` and `\sfcode` values in an automated fashion for the entire Unicode range.

The initial approach taken by the team was to incorporate the existing model used by (plain) XeTeX and to pre-process the "raw" Unicode data into a ready-to-use form as `unicode-letters.def`. However, the relationship between the Unicode Consortium files and TeX data structures is non-trivial and still being explored. As such, it is preferable to directly parse the original (`.txt`) files at point of use. The team has therefore "spun-out" both the data and the loading to a new generic package, unicode-data. This package makes the original Unicode Consortium data files available in the `texmf` tree (in `tex/generic/unicode-data`) and provides generic loaders suitable for reading this data into the plain, LaTeX $2_\varepsilon$, and other, formats.

At present, the following data files are included in this new package:

- `CaseFolding.txt`
- `EastAsianWidth.txt`
- `LineBreak.txt`
- `MathClass.txt`
- `SpecialCasing.txt`
- `UnicodeData.txt`

These files are used either by LaTeX $2_\varepsilon$ or by expl3 (i.e. they represent the set currently required by the team). The Unicode Consortium provides various other data files and we would be happy to add these to the generic package, as it is intended to provide a single place to collect this material in the `texmf` tree. Such requests can be mailed to the team as usual or logged at the package home page: `https://github.com/latex3/unicode-data`.

The new approach extends use of Unicode data in setting TeX information in two ways. First, the `\sfcode` of all end-of-quotation/closing punctuation is now set to 0 (transparent to TeX). Second, `\Umathcode` values are now set using `MathClass.txt` rather than setting up only letters (which was done using an arbitrary plane 0/plane 1 separation). There are also minor refinements to the existing code setting, particularly splitting the concepts of case and letter/non-letter category codes.

For XeTeX, users should note that `\xtxHanGlue` and `\xtxHanSpace` are *no longer defined*, that no assignments are made to `\XeTeXinterchartoks` and that no `\XeTeXintercharclass` data is loaded into the format. The values which were previously inherited from the plain XeTeX setup files are *not* suitable for properly typesetting East Asian text. There are third-party packages addressing this area well, notably those in the CTeX bundle. Third-party packages may need adjustment to load the data themselves; see the unicode-data package for one possible loader.

## *More support for east European accents*

As noted in LaTeX News 23, comma accent support was added for `s` and `t` in the 2015/10/01 release. In this release a matching `\textcommaabove` accent has been added for U+0123 (`\c{g}`, ģ) which is the lower case of U+0122 (`\c{G}`, Ģ). In the OT1 and T1 encodings the combinations are declared as composites with the `\c` command, which matches the Unicode names "latin (capital|small) letter g with cedilla" and also allows `\MakeUppercase{\c{g}}` to produce `\c{G}`, as required. In T1 encoding, the composite of `\c` with `k`, `l`, `n` and `r` are also declared to use the comma below accent rather than cedilla to match the conventional use of these letters.

The UTF-8 `inputenc` option `utf8` has been extended to support all latin combinations that can be reasonably constructed with a (single) accent command an a base character for the T1 encoding so ģ, ų and similar characters may be directly input using UTF-8 encoding.

## *Changes in Graphics*

The changes in LuaTeX v0.87 mean that the color and graphics packages no longer share the `pdftex.def` file between LuaTeX and pdfTeX. A separate file `luatex.def` (distributed separately) has been produced, and distributions are encouraged to modify `graphics.cfg` and `color.cfg` configuration files to default to the `luatex` option if LuaTeX v0.87 or later is being used. The team has contributed suitable `.cfg` files to CTAN to be used as models.

Normally it is best to let the local `graphics.cfg` automatically supply the right option depending on the TeX engine being used; however the color and graphics (and so graphicx) packages have been extended to have an explicit `luatex` option comparable to the existing `pdftex` and `xetex` options.

The trig package has been updated so that pre-computed values such as sin(90) now expand to digits (`1` rather than the internal token `\@one` in this case). This allows them to be used directly in PDF literal strings.

## *Changes in Tools*

LuaTeX from version v0.87 no longer supports the `\write18` syntax to access system commands. A new package shellesc has been added to tools that defines a new command `\ShellEscape` that may be used in all TeX variants to provide a consistent access to system commands. The package also defines `\write18` in LuaTeX so that it continues to access system commands as before; see the package documentation for details.

## *Improving support for Unicode engines*

Stability concerns are always paramount when considering any change to the LaTeX $2_\varepsilon$ kernel. At the same time, it is important that the format remains usable and gives reliable results for users. For the Unicode TeX engines XeTeX and LuaTeX there are important differences in behavior from classical (8-bit) TeX engines which mean that identical default behaviors are not appropriate. Over the past 18 months the team has addressed the most pressing of these considerations (as detailed above and in LaTeX News 22 and 23), primarily by integrating existing patches into the kernel. There are, though, important areas which still need consideration, and which *may* result in refinements to kernel support in this area in future releases.

The default font setup in LaTeX $2_\varepsilon$ at present is to use the `OT1` encoding. This assumes that hyphenation patterns have been read using appropriate codes: the `T1` encoding is assumed. The commonly-used hyphenation patterns today, hyph-utf8, are set up in this way for 8-bit engines (pdfTeX) but for Unicode engines use Unicode code points. This means that hyphenation will be incorrect with Unicode engines unless a Unicode font is loaded. This requires a concept of a Unicode font encoding, which is currently provided by the fontspec package in two versions, `EU1` and `EU2`. The team is working to fully understand what is meant by a "Unicode font encoding", as unlike a classical TeX encoding it is essentially impossible to know what glyphs will be provided (though each slot is always defined with the same meaning). There is also an overlap between this area and ideas of language and writing system, most obviously in documents featuring mixed scripts (for example Latin and Cyrillic).

As well as these font considerations, the team is also exploring to what extent it is possible to allow existing (8-bit) documents to compile directly with Unicode engines without requiring changes in the sources. Whether this is truly possible remains an open question.

It is important to stress that changes will only be made in this area where they do *not* affect documents processed with $\varepsilon$-TeX/pdfTeX (i.e. documents which are written for "classical" 8-bit TeX engines). Changes will also be made only where they clearly address deficiencies in the current setup for Unicode engines (i.e. where current behaviors are wrong).