LaTeX News

Issue 22, January 2015

New LaTeX 2ε bug-fix policy

Introduction

For some years we have supplied bug fixes to the LaTeX 2ε kernel via the fixltx2e package. This kept the kernel stable, but at the expense of meaning that most users did not benefit from bug fixes, and that some compromises which were made to save space in the machines of the time are still affecting most users today.

In this release we have started a new update policy. All the fixes previously available via fixltx2e are now enabled *by default* in the format, as are some further extensions for extended TeX engines, ε-TeX, XeTeX and LuaTeX. Compatibility and stability are still important considerations, and while most users will not notice these improvements, or will want to benefit from them, a new latexrelease package is provided that will revert all the changes and re-instate the definitions from earlier releases. The package can also be used with older releases to effectively *update* the kernel to be equivalent to this 2015 release.

A new document, `latexchanges`, is distributed with the release that documents all the changes to documented commands since the 2014 LaTeX release, and will be updated in future releases if further changes have been made.

The latexrelease package

As noted above a new package is available to manage differences between LaTeX releases. If you wish to revert all changes back to the definitions as they were in previous releases you may start your document requesting the LaTeX release from May 2014:

```
\RequirePackage[2014/05/01]{latexrelease}
\documentclass{article}
```

Conversely if you start a large project now and want to protect yourself against possible future changes, you may start your document

```
\RequirePackage[2015/01/01]{latexrelease}
\documentclass{article}
```

Then the version of latexrelease distributed with any future LaTeX release will revert any changes made in that format, and revert to the definitions as they where at the beginning of 2015.

If you wish to share a document using the latest features with a user restricted to using an older format, you may use the form above and make the latexrelease package available on the older installation. The package will then update the format definitions as needed to enable the older format to work as if dated on the date specified in the package option.

The `\IncludeInRelease` command

The mechanism used in the latexrelease package is available for use in package code. If in your zzz package you have

```
\RequirePackage{latexrelease}
\IncludeInRelease{2015/06/01}
    {\zzz}{\zzz definition}
  \def\zzz......new code
\EndIncludeInRelease
\IncludeInRelease{0000/00/00}
  {\zzz}{\zzz definition}
  \def\zzz....original
\EndIncludeInRelease
```

then in a document using a format dated 2015/06/01 or later, the "new code" will be used, and for documents being processed with an older format, the "original" code will be used. Note the format date here may be the original format date as shown at the start of every LaTeX run, or a format date specified as a package option to the latexrelease package.

So if the document has

```
\RequirePackage[2014/05/01]{latexrelease}
\documentclass{article}
\usepackage{zzz}
```

then it will use the *original* definition of `\zzz` even if processed with the current format, as the format acts as if dated 2014/05/01.

Limitations of the approach

The new concept provides full backward and forward compatibility for the LaTeX format, i.e., with the help of a current latexrelease package the kernel can emulate all released formats (starting with 2014/06/01[1]).

However, this is not necessarily true for all packages. Only if a package makes use of the `\IncludeInRelease` functionality will it adjust to the requested LaTeX release date. Initially this will only be true for a few selected packages and in general it may not even be

[1] Patching an older format most likely works too, given that the changes in the past have been minimal, though this isn't guaranteed and hasn't been tested.

advisable for packages that have their own well-established release cycles and methods.

Thus, to regenerate a document with 100 % compatible behavior it will still be necessary to archive it together with all its inputs, for example, by archiving the base distribution trees (and any modifications made). However, the fact that a document requests a specific LaTeX release date should help identifying what release tree to use to achieve perfect accuracy.

Updates to the kernel

Updates incorporated from fixltx2e

The detailed list of changes incorporated from fixltx2e is available in the new `latexchanges` document that is distributed with this release. The main changes are that 2-column floats are kept in sequence with one column floats, corrections are made to the `\mark` system to ensure correct page headings in 2-column documents, several additional commands are made robust.

ε-TeX register allocation

LaTeX has traditionally used allocation routines inherited from plain TeX that allocated registers in the range 0–255. Almost all distributions have for some years used ε-TeX based formats (or XeTeX or LuaTeX) which have 2^{15} registers of each type (2^{16} in the case of LuaTeX). The etex package has been available to provided an allocation mechanism for these extended registers but now the format will by default allocate in a range suitable for the engine being used. The new allocation mechanism is different than the etex package mechanism, and supports LuaTeX's full range and an allocation mechanism for LaTeX floats as described below.

On ε-TeX based engines, an additional command, `\newmarks` is available (as with the etex package) that allocates extended ε-TeX marks, and similarly if XeTeX is detected a new command `\newXeTeXintercharclass` is available, this is similar to the command previously defined in the `xelatex.ini` file used to build the `xelatex` format.

Additional LaTeX float storage

LaTeX's float placement algorithm needs to store floats (figures and tables) until it finds a suitable page to output them. It allocates 18 registers for this storage, but this can often be insufficient. The contributed morefloats package has been available to extend this list; however, it also only allocates from the standard range 0–255 so cannot take advantage of the extended registers. The new allocation mechanism in this release incorporates a new command `\extrafloats`. If you get the error: `Too many unprocessed floats.` then you can add (say) `\extrafloats{500}` to the document preamble to make many more boxes available to hold floats.

Built-in support for Unicode engines

The kernel sources now detect the engine being used and adjust definitions accordingly, this reduces the need for the "`.ini`" files used to make the formats to patch definitions defined in `latex.ltx`.

As noted above the format now includes extended allocation routines.

The distribution includes a file `unicode-letters.def` derived from the Unicode Consortium's Unicode Character Data files that details the upper and lower case transformation data for the full Unicode range. This is used to set the `lccode` and `uccode` values if a Unicode engine is being used, rather than the values derived from the T1 font encoding which are used with 8-bit engines.

Finally `\typein` is modified if LuaTeX is detected such that it works with this engine.

l3build

This release has been tested and built using a new build system implemented in Lua, intended to be run on the texlua interpreter distributed with modern TeX distributions. It is already separately available from CTAN. This replaces earlier build systems (based at various times on make, cons, and Windows bat files). It allows the sources to be tested and packaged on a range of platforms (within the team, OS X, Windows, Linux and Cygwin platforms are used). It also allows the format to be tested on XeTeX and LuaTeX as well as the standard pdfTeX/ε-TeX engines.

Hyperlinked documentation and TDS zip files

As well as updating the build system, the team have looked again at exactly what gets released to CTAN. Taking inspiration from Heiko Oberdiek's latex-tds bundle, the PDF documentation provided now includes hyperlinks where appropriate. This has been done without modifying the sources such that users without hyperref available can still typeset the documentation using only the core distribution. At the same time, the release now includes ready-to-install TDS-style zip files. This will be of principal interest to TeX system maintainers, but end users with older machines who wish to manually update LaTeX will also benefit.