

L^AT_EX News

Issue 23, October 2015

Contents

Enhanced support for LuaT_EX	1
Names of LuaT _E X primitive commands	1
T _E X commands for allocation in LuaT _E X	2
Predefined Lua functions	2
Support for older releases and plain T _E X	2
Additional LuaT _E X support packages	2
More Floats and Inserts	2
Updated Unicode data	2
Support for Comma Accent	2
Extended inputenc	2
Pre-release Releases	2
Updates in tools	2

Enhanced support for LuaT_EX

As noted in L^AT_EX News 22, the 2015/01/01 release of L^AT_EX introduced built-in support for extended T_EX systems.

The range of allocated register numbers (for example, for count registers) is now set according to the underlying engine capabilities to 256, 32768 or 65536. Additional allocators were also added for the facilities added by ε -T_EX (`\newmark`) and X_YT_EX (`\newXeTeXintercharclass`). At that time, however, the work to incorporate additional allocators for LuaT_EX was not ready for distribution.

The main feature of this release is that by default it includes allocators for LuaT_EX-provided features, such as Lua functions, bytecode registers, catcode tables and Lua callbacks. Previously these features have been provided by the contributed `luatex` (Heiko Oberdiek) and `luatexbase` (Élie Roux, Manuel Pégourié-Gonnard and Philipp Gesang) packages. However, just as noted with the `etex` package in the previous release, it is better if allocation is handled by the format to avoid problems with conflicts between different allocation schemes, or definitions made before a package-defined allocation scheme is enabled.

The facilities incorporated into the format with this release, and described below, are closely modelled on the `luatexbase` package and we thank the authors, and

especially Élie Roux, for help in arranging this transition.

The implementation of these LuaT_EX features has been redesigned to match the allocation system introduced in the 2015/01/01 L^AT_EX release, and there are some other differences from the previous `luatexbase` package. However, as noted below, `luatexbase` is being updated in line with this L^AT_EX release to provide the previous interface as a wrapper around the new implementation, so we expect the majority of documents using `luatexbase` to work without change.

Names of LuaT_EX primitive commands

The 2015/01/01 L^AT_EX release for the first time initialised LuaT_EX in `latex.ltx` if LuaT_EX is being used. Following the convention used in the contributed `lualatex.ini` file used to set up the format for earlier releases, most LuaT_EX-specific primitives were defined with names prefixed by `luatex`. This was designed to minimize name clashes but had the disadvantage that names did not match the LuaT_EX manual, or the names used in other formats, and produced some awkward command names such as `\luatexluafunction`. From this release the names are enabled without the `luatex` prefix.

In practice this change should not affect many documents; relatively few packages access the primitive commands, and many of those are already set up to work with prefixed or unprefixed names, so that they work with multiple formats.

For package writers, if you want to ensure that your code works with this and earlier releases, use unprefixed names in the package and ensure that they are defined by using code such as:

```
\directlua{tex.enableprimitives("",
    tex.extraprimitives(
        "omega", "aleph", "luatex"))}
```

Conversely if your document uses a package relying on prefixed names then you can add:

```
\directlua{tex.enableprimitives("luatex",
    tex.extraprimitives(
        "omega", "aleph", "luatex"))}
```

to your document.

Note the compatibility layer offered by the `luatexbase` package described below makes several commands available under both names.

As always, this change can be reverted using:
`\RequirePackage[2015/01/01]{latexrelease}`
at the start of the document.

TEX commands for allocation in LuaTEX

For detailed descriptions of the new allocation commands see the documented sources in `lualatex.dtx` or chapter N of `source2e`; however, the following new allocation commands are defined by default in LuaTEX: `\newattribute`, `\newcatcodetable`, `\newluafunction` and `\newwhatsit`. In addition, the commands `\setattribute` and `\unsetattribute` are defined to set and unset Lua attributes (integer values similar to counters, but attached to nodes). Finally several catcode tables are predefined: `\catcodetable@initex`, `\catcodetable@string`, `\catcodetable@latex`, `\catcodetable@atletter`.

Predefined Lua functions

If used with LuaTEX, L^ATEX will initialise a Lua table, `luatexbase`, with functions supporting allocation and also the registering of Lua callback functions.

Support for older releases and plain TEX

The LuaTEX allocation functionality made available in this release is also available in plain TEX and older L^ATEX releases in the files `lualatex.tex` and `lualatex.lua` which may be used simply by including the TEX file: `\input{lualatex}`. An alternative for old L^ATEX releases is to use:

`\RequirePackage[2015/10/01]{latexrelease}`
which will update the kernel to the current release, including LuaTEX support.

Additional LuaTEX support packages

In addition to the base L^ATEX release two packages have been contributed to the `contrib` area on CTAN. The `ctablestack` package offers some commands to help package writers control the LuaTEX `catcodetable` functionality, and the `luatexbase` package replaces the previously available package of the same name, providing a compatible interface but implemented over the `lualatex` code.

More Floats and Inserts

If ϵ -TEX is available, the number of registers allocated in the format to hold floats such as figures is increased from 18 to 52.

The extended allocation system introduced in 2015/01/01 means that in most cases it is no longer necessary to load the `etex` package. Many classes and packages that previously loaded this package no longer do so. Unfortunately in some circumstances where a package or class previously used the `etex` `\reserveinserts` command, it is possible for a document that previously worked to generate an error

“no room for a new insert”. In practice this error can always be avoided by declaring inserts earlier, before the registers below 256 are all allocated. However, it is better not to require packages to be re-ordered and in some cases the re-ordering is complicated due to delayed allocations in `\AtBeginDocument`.

In this release, a new implementation of `\newinsert` is used which allocates inserts from the previously allocated float lists once the classical register allocation has run out. This allows an extra 52 (or in LuaTEX, 64 thousand) insert allocations which is more than enough for practical documents (by default, L^ATEX only uses two insert allocations).

Updated Unicode data

The file `unicode-letters.def` recording catcodes, upper and lower case mappings and other properties for Unicode characters has been regenerated using the data files from Unicode 8.0.0.

Support for Comma Accent

The command `\textcommabelow` has been added to the format. This is mainly used for the Romanian letters ȘșȚț. This was requested in latex/4414 in the L^ATEX bug tracker.

Extended inputenc

The `utf8` option for `inputenc` has been extended to support the letters s and t with comma accent, U+0218–U+021b. Similarly circumflex w and y U+0174–U+0177 are defined. Also U+00a0 and U+00ad are declared by default, and defined to be `\nobreakspace` and `\-` respectively.

The error message given on undefined UTF-8 input characters now displays the Unicode number in U+hex format in addition to showing the character.

Pre-release Releases

The patch level mechanism has been used previously to identify L^ATEX releases that have small patches applied to the main release, without changing the main format date.

The mechanism has now been extended to allow identification of pre-release versions of the software (which may or may not be released via CTAN) but can be identified with a banner such as
LaTeX2e <2015/10/01> pre-release-1
Internally this is identified as a patch release with a negative patch level.

Updates in tools

The `multicol` package has been updated to fix the interaction with “here” floats that land on the same page as the start or end of a `multicols` environment.