luaalignstate — Access the alignment counter in LuaTEX

Ruiyuan Chen

2026/11/15 v1.0

In TEX, the alignment brace counter, also known as "master counter" (according to the TEXbook) or `align_state` (according to the source code), is an internal variable that is used to track the balance between all braces `{` and `}` seen so far, and plays a subtle role in the handling of `\halign`/`\valign` and derived macros such as LATEX's `tabular`. The variable is not exposed to the macro language or even to the Lua libraries in LuaTEX. This makes certain rare bugs difficult to track down,[1] and also prevents some useful alignment tricks that would otherwise be possible.[2]

This package provides a hack in LuaTEX to directly inspect and modify the alignment brace counter. There is an underlying format-independent Lua module, as well as a LATEX3 programming wrapper interface.

1 Lua interface

The Lua module is loaded with

```
local luaalignstate = require"luaalignstate"
```

which returns a table containing the following functions.

`luaalignstate.getset()`

```
new, old = luaalignstate.getset(function (old_alignstate)
    return new_alignstate
end)
```

The main interface. Determines the current alignment counter, and optionally updates it in one go. If the provided function is or returns `nil`, no update is made.

TEXhackers note: This function works by executing an empty `\halign`. As such, it *must not be called* while scanning the preamble of another alignment (via e.g., `\span\directlua`); doing so would lead to memory corruption.

Because of this, the function will throw a fatal error if it detects that it was likely called from an alignment preamble, namely if `align_state` < -500000. (This will also happen outside of alignment preambles, if you for some reason set `align_state` to a value less than -500000.)

`luaalignstate.get()`

```
alignstate = luaalignstate.get()
```

`luaalignstate.set()`

```
new, old = luaalignstate.set(new_alignstate)
```

`luaalignstate.add()`

```
new, old = luaalignstate.add(increment)
```

Simple wrappers around `getset()`.

[1] See e.g., https://github.com/latex3/latex2e/issues/1468.

[2] For example, it is difficult to create a fully general mechanism that allows a colspan to be specified in a table cell other than at the very beginning.

2 LaTeX3 interface

The LaTeX package is loaded with

```
\RequirePackage{luaalignstate}
```

This loads the Lua module globally, and defines the following L3 programming functions.

`\luaalignstate_value:` ⋆ Expands in one step to the current value of the alignment counter (in decimal).

TEXhackers note: As with the Lua functions, this is *not* safe to expand (via `\span`) while scanning the preamble of an alignment.

Note that `\luaalignstate_value:` can be somewhat tricky to expand correctly in the desired ambient brace level: for example,

```
\tl_set:No \l_my_tl {\luaalignstate_value:}
```

would likely set `\l_my_tl` to one or two more than the alignment counter value at the point that it is executed, depending on the implementation of `\tl_set:No`.

For this reason, the following getter functions are provided as an alternative.

`\luaalignstate_get:N` `\luaalignstate_get:N` ⟨*tl var*⟩

Sets the ⟨*tl var*⟩ (locally) to the current value of the alignment counter.

`\luaalignstate_use:n` ⋆ `\luaalignstate_use:n {`⟨*code*⟩`}`

Expands in two steps to ⟨*code*⟩ followed by a brace group containing the current value of the alignment counter.

`\luaalignstate_use_inline:n` ⋆ `\luaalignstate_use_inline:n {`⟨*inline function*⟩`}`

Expands in two steps to ⟨*inline function*⟩ applied to the alignment counter value.

`\luaalignstate_set:n` `\luaalignstate_set:n {`⟨*int expr*⟩`}`

Sets the alignment counter to the given ⟨*int expr*⟩.

TEXhackers note: The ⟨*int expr*⟩ may refer to the previous alignment counter value via `\luaalignstate_value:`, which within the ⟨*int expr*⟩ expands to the value at the point where `\luaalignstate_set:n` is called (i.e., outside any braces surrounding or part of ⟨*int expr*⟩).

On the other hand, the ⟨*int expr*⟩ should not refer directly to `\currentgrouplevel`, `\currentgrouptype`, or `\lastnodetype`, as their values will be wrong.

`\luaalignstate_add:n` `\luaalignstate_add:n {`⟨*int expr*⟩`}`

Adds ⟨*int expr*⟩ to the alignment counter.

TEXhackers note: This may be expressed in terms of `\luaalignstate_set:n`, but is ≈ 2 ops faster. Unlike above, here the ⟨*int expr*⟩ should *not* refer to `\luaalignstate_value:`.