

# LUAT<sub>E</sub>X 0.79

## an update

Bacho<sub>T</sub>E<sub>X</sub> 2014

**speed**

**nodes**

**expansion**

**parbuilder**

**backend**

**backlinks**

**properties**

**LUA calls**

**artefacts**

**callbacks**

**furthermore**

# speed

## issues

- there is this  $\text{T}_\text{E}\text{X}$  niche: automated workflows, complex and/or huge documents
- we need to find a compromise between convenience and efficiency
- we deal with font loading and processing, list manipulations
- we need to keep (slower) portable devices in mind

## recently done

- integrate  $\text{LUAJIT}\text{T}_\text{E}\text{X}$  in the source tree
- optimized aspects of the  $\text{T}_\text{E}\text{X}$ -LUA interface
- improved some of the logging (also more callbacks)

## todo

- locate bottlenecks in the engine
- cleanup some of the IO
- integrate (and improve) the (backend) error handling

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# nodes

## issues

- crossing the T<sub>E</sub>X-LUA boundary comes at a price
- with millions of calls (macro package features) there is a speed penalty
- some complex scripts are demanding (like devanagari)

## recently done

- an additional access model for stable and access-heavy node usage

## todo

- not that much conceptually (the current model is quite okay and probably as efficient as it can be)
- cleanup of merged engine code (redundant tests, nodes, etc)

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# expansion

## issues

- playing with the LUA parbuilder showed inefficiencies
- multiple auto-generated fonts were used for the stretched glyphs
- many calculations happened several times

## recently done

- (already done before) make the backend code more efficient
- introduce a stretch field in glyph nodes
- use that to pass information to the backend (no extra fonts)

## todo

- look into protrusion
- cleanup the frontend code a bit (delayed work)
- remove (or make optional) kern stretching
- specify stretch relative to glyph width and not emwidth

(t:/manuals/hybrid/parbuildertest/2013/test.tex)

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# parbuilder

## issues

- the parbuilder code is a merge of engine code
- some code is no longer needed or can be done better

## recently done

- fix some inconsistencies in the output of the parbuilder (math)

## todo

- create consistent output from the parbuilder
- make dir whatsits into proper core nodes and improve them a bit

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# backend

## issues

- front and backend are not clearly separated (inheritance of PDF<sub>TEX</sub>)
- some code has an experimental character

## recently done

- make the PDF properties more consistent (accessors)
- improve the LUA-<sub>TEX</sub> interface

## todo

- promote some generic features to core features (like images and transformations)
- implement some backend register setters as macros on top of LUA (cleaner code)
- separate the backend code even more
- remove some (never used) experimental code (or features that are done in LUA anyway)
- cleanup the code (no longer needed pdf prefixes etc.)

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# backlinks

## issues

- the internal lists were never really meant to be accessed
- some backlinks (prev nodes) were not okay (esp. in math lists)

## recently done

- make all lists properly back-linked
- fix the head nodes (that themselves can be next nodes of temp)

## todo

- check and double check the fixes

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# properties

## issues

- sometimes you want to carry information with nodes
- we cannot keep extending nodes (never ending story)

## recently done

- provide a (global) properties table that can have a table per node
- cleanup and copying is handled automatically

## todo

- (in `CONTEXT`;) move some test code in the core
- finish the experimental interface for associating data

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore



# LUA calls

## issues

- user nodes have a fixed set of values
- `\latelua` nodes are strings but functions can be handy too (they carry states)

## recently done

- add a LUA specific field (type)
- check such LUA values for string (compile and run) or functions (run)

## todo

- nothing

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# artefacts

## issues

- the code clearly shows that it's a 30<sup>+</sup> year mix
- the (impressive translation to c) code was a first step
- not all code (macros) reflects what it is meant to do (CWEB side effects)
- some documentation is (and was already) not in sync

## recently done

- fix bits and pieces

## todo

- make all more consistent
- add and fix the documentation

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# callbacks

## issues

- there are still missing callbacks

## recently done

- add the missing page flushing callbacks
- add callbacks for opening and closing files reporting
- provide bit more control over error messages

## todo

- add proper reporting for library messages

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore

# furthermore

## recently done

- the usual bug fixes
- the usual library updates
- some pending tracker issues

## todo (short term)

- all of this
- look into LUA 5.3 (especially new hybrid number concept)
- finish and integrate swiglib (end of year)

speed

nodes

expansion

parbuilder

backend

backlinks

properties

LUA calls

artefacts

callbacks

furthermore