

# LuaTeX

closing in on 0.50

TUG 2009

- ◆ Since we reported on the state of LuaTeX at KTUG, Dante and Bachotek no fundamental extensions have been made.
- ◆ The focus has been on a 0.40 freeze (for TeXlive), debugging, cleanup and converting the Pascal code into C as a prelude to 0.50 (euroTeX).
- ◆ We promised a production version. The 0.40 version that will be on TeXlive is quite stable and usable. Currently we're moving to 0.50.
- ◆ The current version of LuaTeX is quite compatible with pdfTeX. Of course documents can come out differently because of the following reasons:
  - ◆ We have Unicode hyphenation patterns independent of the font encoding.
  - ◆ There is more granularity in metrics when not using traditional fonts.
  - ◆ Hyphenation, ligaturing and kerning are separated stages.
- ◆ But such differences occur anyway over time (as one updates resources) unless a frozen distribution is used.

- ◆ The paragraph builder will become more accessible (easier to do now).
- ◆ The output routine will be opened up.
- ◆ Directional issues will be cleaned up (oriental T<sub>E</sub>X project).
- ◆ The backend will be restructured (in progress).
- ◆ We will explore and implement some pending wishes.
- ◆ Of course we will do whatever comes to our mind.
- ◆ It's up to you (and your package writers) to use it or not.

- ◆ One can ignore the opening up and just use Lua for manipulations, and generation and piping of data.
- ◆ In ConT<sub>E</sub>Xt we make extensive use of the opening up. The development of LuaT<sub>E</sub>X and ConT<sub>E</sub>Xt MkIV goes in sync.
- ◆ We're quite lucky that ConT<sub>E</sub>Xt users are testing new releases as soon as they show up and the `contextgarden` has binaries for many platforms.
- ◆ At the upcoming ConT<sub>E</sub>Xt meeting we hope will discuss the impact of what has been done so far and will be done after 0.50.
- ◆ Areas that have been touched are:
  - ◆ all input and output
  - ◆ most of fonts, especially OpenType
  - ◆ many manipulations like character casing
  - ◆ much of math (and more to come)
  - ◆ everything structure and referencing
  - ◆ color and other attributes
  - ◆ the complete pdf backend
  - ◆ a tree based xml fontend
  - ◆ the usual tools (scripts)
- ◆ Some of the changes are real large (state).

- ◆ No matter what nice things we do, the big question is: can LuaTeX be used at all, given all the changes in the code base.
- ◆ We ship some basic plain TeX support as part of the ConTeXt distribution as proof that you can use LuaTeX without ConTeXt as well, but of course we don't touch plain itself.
- ◆ I do use an older version (0.30) in a few small and non-critical workflows, but especially the rewritten structure related code has quite some impact.
- ◆ All rewrites go through several stages because we adapt LuaTeX when we feel it makes sense (beta features).
- ◆ We measure usability in several ways:
  - ◆ How fast is a simple run in an editor (startup, pdf building)?
  - ◆ How well does it integrate in workflows (toolkits, overhead)?
  - ◆ How fast is a run (or multiple runs when needed)?
- ◆ Some aspects are out of our control, like for instance the impact of the console on runtime (per char, refresh delay, fonts, buffer).

Let's take a closer look at some tests:

- ◆ We constantly check changes with a couple of test documents: `mk`, `metafun`, `luatexref-t`. We only release when these documents process well.
- ◆ I made a rather dumb baseline (mostly `otr` and `backend`) performance test (just the word `test` on many pages).

```
input load time          - 0.109 seconds
stored bytecode data    - 184 modules, 45 tables, 229 chunks
node list callback tasks - 4 unique tasks, 4 created, 20980 calls
cleaned up reserved nodes - 29 nodes, 10 lists of 1427
node memory usage       - 19 glue_spec, 2 dir
h-node processing time  - 0.312 seconds including kernel
attribute processing time - 1.154 seconds
used backend            - pdf (backend for directly generating pdf output)
loaded patterns         - en:us:pat:exc:2
jobdata time           - 0.078 seconds saving, 0.047 seconds loading
callbacks               - direct: 86692, indirect: 13364, total: 100056
interactive elements    - 178 references, 356 destinations
v-node processing time  - 0.062 seconds
loaded fonts            - 43 files: ....
fonts load time        - 1.030 seconds
metapost processing time - 0.281 seconds, loading: 0.016 seconds, execution: 0.156 seconds,
                        n: 161
result saved in file    - luatexref-t.pdf
luatex banner           - this is luatex, version beta-0.42.0
control sequences       - 31880 of 147189
current memory usage    - 106 MB (ctx: 108 MB)
runtime                 - 12.433 seconds, 164 processed pages, 164 shipped pages,
                        13.191 pages/second
```

```
input load time          - 0.125 seconds
stored bytecode data    - 184 modules, 45 tables, 229 chunks
node list callback tasks - 4 unique tasks, 4 created, 24295 calls
cleaned up reserved nodes - 116 nodes, 29 lists of 1411
node memory usage       - 21 attribute, 23 glue_spec, 7 attribute_list, 7 local_par, 2 dir
h-node processing time  - 1.763 seconds including kernel
attribute processing time - 2.231 seconds
used backend            - pdf (backend for directly generating pdf output)
loaded patterns         - en:us:pat:exc:2 en-gb:gb:pat:exc:3 nl:nl:pat:exc:4
language load time     - 0.094 seconds, n=4
jobdata time           - 0.062 seconds saving, 0.031 seconds loading
callbacks               - direct: 98199, indirect: 20257, total: 118456
xml load time          - 0.000 seconds, lpath calls: 46, cached calls: 31
v-node processing time  - 0.234 seconds
loaded fonts           - 69 files: ....
fonts load time        - 28.205 seconds
metapost processing time - 0.421 seconds, loading: 0.016 seconds, execution: 0.203 seconds,
                        n: 65
graphics processing time - 0.125 seconds including tex, n=7
result saved in file    - mk.pdf
metapost font generation - 0 glyphs, 0.000 seconds runtime
metapost font loading   - 0.187 seconds, 40 instances, 213.904 instances/second
luatex banner          - this is luatex, version beta-0.42.0
control sequences      - 34449 of 147189
current memory usage    - 454 MB (ctx: 465 MB)
runtime                 - 50.326 seconds, 316 processed pages, 316 shipped pages,
                        6.279 pages/second
```

**MK, the history of LuaT<sub>E</sub>X upto 0.50**



```
input load time          - 0.109 seconds
stored bytecode data    - 184 modules, 45 tables, 229 chunks
node list callback tasks - 4 unique tasks, 4 created, 33510 calls
cleaned up reserved nodes - 39 nodes, 93 lists of 1432
node memory usage       - 249 attribute, 19 glue_spec, 82 attribute_list, 85 local_par, 2 dir
h-node processing time  - 0.562 seconds including kernel
attribute processing time - 2.512 seconds
used backend            - pdf (backend for directly generating pdf output)
loaded patterns         - en:us:pat:exc:2
jobdata time            - 0.094 seconds saving, 0.031 seconds loading
callbacks               - direct: 143950, indirect: 28492, total: 172442
interactive elements    - 214 references, 371 destinations
v-node processing time  - 0.250 seconds
loaded fonts            - 45 files: l.....
fonts load time         - 1.794 seconds
metapost processing time - 5.585 seconds, loading: 0.047 seconds, execution: 2.371 seconds,
                        n: 1772, external: 15.475 seconds (7 calls)
mps conversion time     - 0.000 seconds, 1 conversions
graphics processing time - 0.499 seconds including tex, n=74
result saved in file    - metafun.pdf
luatex banner           - this is luatex, version beta-0.42.0
control sequences       - 32587 of 147189
current memory usage    - 113 MB (ctx: 115 MB)
runtime                 - 43.368 seconds, 362 processed pages, 362 shipped pages,
                        8.347 pages/second
```

```
\dorecurse {n} {test \page}
```

runtime in seconds (pages per second)

### Dell M90 Laptop, Vista Ultimate / 32

engine	30	300	2000	10000
<i>xetex</i>	1.81 (16)	2.45 (122)	6.97 (286)	29.20 (342)
<i>pdftex</i>	1.28 (23)	2.07 (144)	6.96 (287)	30.94 (323)
<i>luatex</i>	1.48 (20)	2.36 (127)	7.85 (254)	34.34 (291)

### Dell 2950 Server, SuSe Linux 11 / 64

engine	30	300	2000	10000
<i>xetex</i>	0.92 (32)	1.89 (158)	8.74 (228)	42.19 (237)
<i>pdftex</i>	0.49 (61)	1.14 (262)	5.23 (382)	24.66 (405)
<i>luatex</i>	1.07 (27)	1.99 (150)	8.32 (240)	38.22 (261)

**Baseline performance**