

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

MetaPost 0.99...99

an experiment

BachoT_EX 2014

LUA in T_EX

- it all started out with `\directlua`
- and after that we got access to dimensions and counters
- the first magic came from the ability to print to T_EX
- eventually we got access to internals and the machinery
- that became a natural way to extend T_EX in arbitrary ways

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

LUA in METAPOST

- the idea is to have a mechanism similar to `\directlua`
- as MetaPost is a rather complete language, so overloading behaviour is not needed

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

the implementation

- this time it started out with scantokens
- with read-only access to strings, numerics, booleans
- contrary to $\text{T}_{\text{E}}\text{X}$'s `\directlua` we don't mess with the input stack
- instead we can hook in a function that can return a string
- on top of that one can build an interface
- in principle this mechanism is not bound to Lua

LUA in $\text{T}_{\text{E}}\text{X}$

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

the basics

metapost

```
numeric n ; n := scantokens("123.456") ;
```

lua

```
local m = mplib.new {  
  ...  
  script_runner = function(s) return loadstring(s)() end,  
  script_error = function(s) print(s) end,  
  ...,  
}
```

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

the helpers

<code>m:get_numeric(name)</code>	returns a numeric (double)
<code>m:get_boolean(name)</code>	returns a boolean (true or false)
<code>m:get_string (name)</code>	returns a string

<code>mplib.get_numeric(m,name)</code>	returns a numeric (double)
<code>mplib.get_boolean(m,name)</code>	returns a boolean (true or false)
<code>mplib.get_string (m,name)</code>	returns a string

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

METAFUN

<code>print(...)</code>	returns one or more values
<code>pair(x,y)</code> <code>mp.pair(t)</code>	returns a proper pair
<code>triplet(x,y,z)</code> <code>triplet(t)</code>	returns an rgb color
<code>quadruple(w,x,y,z)</code> <code>quadruple(t)</code>	returns an cmyk color
<code>format(fmt,...)</code>	returns a formatted string
<code>quoted(fmt,...)</code> <code>quoted(s)</code>	returns a (formatted) quoted string
<code>numeric(name)</code>	gets a numeric from MetaPost
<code>boolean(name)</code>	gets a boolean from MetaPost
<code>string(name)</code>	gets a string from MetaPost

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do

to do

- read-only access to pairs, triplets, quadruplets and maybe paths
- update the MetaFun manual (has to be done anyway)

LUA in T_EX

LUA in
METAPOST

the imple-
mentation

the basics

the helpers

METAFUN

to do