Taco Hoekwater

July 20, TUG 2007
INGREDIENTS
INGREDIENTS

lua 5.1
lua libraries
lua 5.1
lua libraries
bits of fontforge
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
\TeX
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
\TeX
PDF\TeX
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
\TeX
\pdf\TeX
ALEPH
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
\TeX
\pdfTeX
ALEPH
C code
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
\TeX
PDFT\TeX
ALEPH
C code
lua 5.1
lua libraries
bits of fontforge
bits of dvipdfmx
\TeX
\pdftex
\aleph
C code
bugs
Timeline

• Project started — TUG 2005
• CSU & TUG funding approved — May 2006
• Binary snapshots since — August 2006
• Mailing list and website started — September 2006
• First public beta — TUG 2007
• Release candidate — TUG 2008
Features

• TEX extensions
• Unicode support
• Extended font subsystem
• Lua interface libraries
• Direct Lua execution
• Callbacks
TEX EXTENSIONS

• configurable startup
• \catcode
• tables
• attributes
• dynamic memory
• integrated pool file
• compressed formats
Unicode support

• UTF-8 input and output
• Unicode hyphenation patterns
• Commands like \char
• Fonts can be Unicode encoded
Extended font subsystem

- totally configurable through lua code
- TEX (TFM) and OMEGA (OFM) fonts
- OpenType fonts
- TrueType subsetting
- artificial fonts
- dynamic memory usage
Lua interface libraries

- tex — general TEX access
- pdf — routines related to pdf output
- texio — writing to the log and terminal
- font — accessing font internals
- status — \textsc{luatex} status information
- kpse — file searching
- callback — setting up callback hooks
- token — handling \textsc{tex} tokens
- node — handling typeset nodes
Direct lua execution

• expandable `directlua` command
• postponed `latelua` command
• multiple lua interpreters
• extra lua libraries:
  - `lpeg` — pattern matching
  - `lfs` — filesystem extensions
  - `md5` — checksumming
  - `zip` — reading zip archives
  - `gzip & zlib` — reading and writing compressed files
  - `unicode` — utilities for unicode
  - `fontforge` — font loading
• `texlua` script interpreter
Callbacks

• callbacks hook into the internal processing
• they make \texttt{LUATEX} run a lua function you define
• some extend, some replace core functionality
• no connection to the \texttt{TEX} input language
• equivalent to compiled executable code
Callback example

```directlua
function read
  tfm = ...
  archive = zip.open('texmf-fonts.zip')
  if archive then
    tfmfile = archive:open(name .. '.tfm')
    if tfmfile then
      data = tfmfile:read('*all')
      return true, data, #data
    end
  end
  return false, nil, 0
end

callback.register('read_font_file', read)
```
Types of callbacks

- finding files
- reading and preprocessing input
- font definitions
- token creation
- node list handling
- information display
Project Crew

- Hans Hagen --- general overview and website
- Hartmut Henkel --- pdf backend
- Taco Hoekwater --- coding and manual
- Arthur Reutenauer --- binaries and testing
- Martin Schröder --- release support
Contact

• Website: http://www.luatex.org
• Repository and bug tracker: http://scm.foundry.supelec.fr/projects/luatex
• Mailing list: dev-luatex@ntg.nl http://www.ntg.nl/mailman/listinfo/dev-luatex