A New Kind of TEX

I The evolution of pdfTeX

I pdftex 1.50: frozen stable version of pdftex

II luatex: basic cleanup and lua support (developers test version)

III orientaltex: official deliverable of an funded project

IV metatex: formal experimental version with metapost

V pdftex 2.00: the first official stable version of metatex

II How we proceed

I start with existing engines (pdftex, partial aleph)

II move from 8 bit character handling to utf-8

III remain (mostly) downward compatible and provide an migration path

IV get rid of some interfering optimizations and clean up code base

V provide complete control at each stage using Lua (callbacks)

VI manipulate input an multiple places (e.g. ascii and utf regexp)

VII provide (node) list manipulation (e.g. dedicated regexp)

VIII in addition to specials, provide attribute states (push/pop)

III Reasons for using Lua

I it's lightweight compared to other scripting languages

II we want to permit multiple instances during a run

III the interpreter is efficient and fast enough for our purpose

IV it has proven to be stable and is widely accepted

V it has reached a mature state (version 5)

VI the language is conceptually clean and concise

VII it has some pleasant natural limitations

VIII you don't need huge bulky manuals

IV Talking and Tweaking

With regards to interfaces, think of:

I
- tex.dimen[<number>]
- tex.dimen["csname"]
- tex.dimen.csname

II
- tex.parindent

III
- language.number
- font.glyph[1234].width
- tex.languages[0]
- tex.fonts[789]

IV
- command.hbox
- command.vbox
- command.insert
- command.display
- command.alignment

V
- kpse.find_file
- kpse.expand_path

VI
- texio.input_line
- texio.write

VII
- pdf.type (ascii output)
- pdf.outchar

VIII
- . . .

V How can we benefit from Lua

We can improve existing methods:

I replace parts of macro packages

II optimize computational extensive tasks

III replace external methods by internal ones

But we can also be more drastic:

IV read from zip files (packaging)

V use sockets to talk with other processes

VI replace and/or extending kpse (separation, integration)

VII extend the typesetting engine with external methods (using serialization)