

Old wine in
new bottles

1 observations

- I Most ebooks are just books (or try to be).
- II Only a small portion has (or needs) design.
- III To what extent appreciation matters is hard to measure.
- IV Vendor locking is spoiling much.
- V 10 years of low res screens have made readers tolerant.
- VI Publishers already lost the edge.
- VII Eventually authors will publish themselves.

2

What is an ebook

- I Nicest is it being a pdf (some design).
- II Easiest is it being an xhtml file (with some css).
- III Pointless it is being a frozen app.
- IV We can already provide a pdf for paper and screen for quite a while.
- V We can consider providing an xhtml alongside as reflowable variant.
- VI Who knows what we can provide in the future.

3 The starting point

- I No output is better than the input.
- II Fixing bad coding is a waste of energy.
- III Not that many publishers want to invest in coding.
- IV Not that many tools enforce structure.
- V The real good devices still have to come but we can be ready for it.
- VI The intelligence has to be in the macro package, not in the engine.

4 Implementation

- I Some users have to produce tagged pdf (to satisfy validators).
- II It helps that many commands in ConT_EXt are built upon a relatively small generic set.
- III So, given some basic structure, supporting tags is easy.
- IV It integrates in the core.
- V And as a side effect an export was equally easy to support.
- VI The overhead is not that large (upto 10%).
- VII Instead of going for freaky solutions (no need for challenges) we assume proper structure.
- VIII It's not to hard to extend the current features.

5 Consequence for users

- I Use `\startchapter` cum suis, but that was already the MkIV fashion.
- II Preferable use `\startitem` instead of `\item`.
- III Playing safe means tagging paragraphs with `\startparagraph`.
- IV Use style environments instead of font switches.
- V Just use whatever structural markup that ConTEXt already provides for ages.

An 6 example

- I A normal input with T_EX commands (cld-mkiv.tex)
- II Regular (tagged) output (cld-mkiv.pdf)
- III Structured output (cld-mkiv.xml)
- IV Reflowable output (cld-mkiv-export.xhtml)
- V Dynamic styling (cld-mkiv-export.css)
- VI Basic template (cld-mkiv-export.template)