

Typesetting Nightmares

Hans Hagen, PRAGMA ADE

It does not take much for users (and customers) to realize that \TeX is a programming language. This often results in the perception that you can do anything you want, and make people believe that you can do better than other, less open applications. Combine this with the fact that developers seldom admit that something cannot be done, and the ingredients of a typographic programming nightmare are there. The complication arises from the fact that:

- opposite to desktop publishing applications, \TeX sees a document as a sequence of content
- where \TeX based macro-packages tend to organize fonts and measures, designers follow a more random path
- where \TeX loves structure, authors want to put any thought on paper, being structured or not, which results in not only interfering data, but also in the wish to escape from \TeX 's machinery
- one reason for choosing \TeX is its ability to typeset math, and typesetting that often conflicts with pure text typesetting
- \TeX tries to do its best to typeset beautiful paragraphs, but frequently the (not producible by \TeX) counterpart is considered more beautiful or adequate

This means that in order to fulfil the needs of authors and designers, one sometimes has to bend \TeX 's rules and cook up rather complicated macros. In this presentation I will discuss a couple of last year's (typo)graphical programming nightmares.