

Abstracts

Dynamic presentations using `TeXpower` and `PSTricks`

David M. Allen, University of Kentucky

The nature of the full version of this presentation requires that it be viewed on a screen rather than paper, as dynamic features cannot be illustrated on paper. An electronic version of the full paper is available from the author's web site: <http://www.ms.uky.edu/~allen/>.

A typical presentation consists of displaying a sequence of slides (a metaphor for screens) in a predetermined order. This presentation is to demonstrate methods for preparing dynamic presentations in the following contexts:

1. Rather than showing a set of slides in predetermined order, one may select the slides and their order after the presentation starts. This would likely be in response to questions from the audience.
2. If the discussion gets deep, it may be useful to visit a web site.
3. A math professor might want to show a multiline derivation one line at a time to focus attention to the current point of discussion.
4. An engineer might want to show a graphic depicting the assembling of a device one part at a time.

The \LaTeX packages used in this endeavor and their URL's follow.

Items 1 and 2 are implemented using the *hyperref* package, <http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref/>. Extensive facilities for navigation within a document, between documents, and on the web are provided by *hyperref*.

Items 3 and 4 are implemented using *TeXpower*, <http://texpower.sourceforge.net/>. *TeXpower* is a \LaTeX package providing incremental display and special effects similar to those found in Microsoft PowerPoint.

Item 4 assumes there are graphics to be displayed, and my examples use graphics produced by the *PSTricks* package, <http://www.pstricks.de/>. *PSTricks* provides a user friendly front end to the PostScript language. It is a generic \TeX package providing extensive computational graphics capabilities.

Typesetting critical editions of poetry with `poemscol`

John Burt, Brandeis University

`poemscol` provides macros for \LaTeX for setting collections of poetry. It provides the structures required to produce a critical edition of the kind specified by the Modern Language Association's Committee on Scholarly Editions, providing line numbering, endnote sections for textual variants (both substantives and accidentals), emendations, explanatory notes, and an index of titles and first lines. It provides running headers of the form "Emendations to pp. xx-yy" for the endnotes sections. It provides structures for different kinds of poetic text. It automatically marks every occasion where a stanza break falls on a page break. Aids for preparing parallel-text (as for instance editions with facing-page translations) editions are under development.

(Full papers on `poemscol` were published in *TUGboat* 22(4) and *The PracTeX Journal* 2005-3. *Ed.*)

Indexing, `MakeIndex`, and `SAS`

Ronald Fehd, CDC

\LaTeX provides the `fancyvrb` package which can be very useful in preparing a document providing an overview of a collection of computer programs. This paper examines the theory of indexing and the \LaTeX `MakeIndex` package. The author provides two `SAS` programs which read all programs in a project directory and then write an index of intra- and inter-program references.

MathML via `TeX4ht` and other tools

Eitan Gurari

The support provided by graphical browsers for the HTML standard was a major ingredient in developing the Internet into a popular medium for archiving and distributing general content. Two recent advancements suggest a similar bright future for mathematical content expressed with the MathML standard. The Mozilla Firefox browser, released last November, now offers native support for MathML. Also, the MathPlayer version 2 plug-in for MS Internet Explorer, which is easily installed and was released a year ago, is now capable of serving general MathML files.

This presentation will provide insight into how `TeX4ht` produces MathML from \LaTeX sources, and will consider issues involved in creating MathML with `TeX4ht` and other tools.

L^AT_EX and PitStop: An unusual but powerful alliance

Mirko Janc, INFORMS

I will share some experiences in preparing art files for inclusion in L^AT_EX in the production cycle in our Institute. We publish 11 scholarly journals in Operations Research using L^AT_EX with a special font setup (presented at the TUG 2003 conference in Hawaii).

Powerful L^AT_EX math typesetting capabilities coupled with PitStop, a commercial Acrobat plugin, enable easy relabeling of figures with most complex math. Unlike other methods, exact positioning and scaling is a breeze. We also use this same method for updating colored covers where color issues are at stake, so the underlying PDF template can be properly preserved.

Some other related “tricks” to get clean art ready for proper inclusion in L^AT_EX will also be discussed.

An introduction to XeT_EX

Jonathan Kew

Professor Donald Knuth’s T_EX is a typesetting system with a wide user community, and a range of supporting packages and enhancements is available for many types of publishing work. However, it dates back to the 1980s and is tightly wedded to 8-bit character data and custom-encoded fonts, making it difficult to configure T_EX for many complex-script languages.

This paper will introduce XeT_EX, a system that extends T_EX with direct support for modern OpenType and AAT fonts and the Unicode character set. This makes it possible to typeset almost any script and language with the same power and flexibility as T_EX has traditionally offered in the 8-bit, simple-script world of European languages. XeT_EX (currently available on Mac OS X, but possibly on other platforms in the future) integrates the T_EX formatting engine with technologies from both the host operating system (Apple Type Services, CoreGraphics, QuickTime) and auxiliary libraries (ICU, TECKit), to provide a simple yet powerful system for multilingual and multiscript typesetting.

The most significant extensions which XeT_EX provides are its native support for the Unicode character set, replacing the myriad of 8-bit encodings traditionally used in T_EX with a single standard for both input text encoding and font access; and an extended `\font` command that provides direct access by name to all the fonts installed in the user’s com-

puter. It also provides a mechanism to access many of the advanced layout features of modern fonts.

Additional features that will also be discussed include built-in support for a wide variety of graphic file formats, and an extended line-breaking mechanism that supports Asian languages such as Chinese or Thai that are written without word spaces.

Finally, we look briefly at some user-contributed packages that help integrate the features of XeT_EX with the established L^AT_EX system. Will Robertson’s `fontspec.sty` provides a simple, consistent user interface in L^AT_EX for loading both AAT and OpenType fonts, and accessing virtually all of the advanced features these fonts offer; Ross Moore’s `xunicode.sty` is a package that allows legacy L^AT_EX documents to be typeset using native OS X fonts without converting the input text entirely to Unicode, by supporting traditional T_EX input conventions for accents and other ‘special’ (i.e., non-ASCII) characters.

(We expect to publish the full paper in the next issue of *TUGboat*. *Ed.*)

Producing beautiful slides with L^AT_EX: An introduction to the HA-prosper package

Tristan Miller

In this paper, we present HA-prosper, a L^AT_EX package for creating overhead slides. We describe the features of the package and give examples of their use. We also discuss what advantages there are to producing slides with L^AT_EX versus the presentation software typically bundled with today’s office suites.

(The full paper on HA-prosper was published in *The PracT_EX Journal* 2005-2. *Ed.*)

T_EX font installation and usage

Steve Peter

This talk is designed to be a near-comprehensive roadmap of installing and using fonts with T_EX (except for bitmapped fonts). We will start with the basics of T_EX font handling (TFMs, etc.), along with a discussion of the major font technologies (PostScript, TrueType, and OpenType) and T_EX’s virtual fonts. Then we move to NFSS and fontinst, followed by T_EXfont and ConT_EXt typescripts. Time permitting, we will configure an expert font, complete with fi, fl, ff, ffi, and ffl ligatures, suitable for professional typesetting.

(We expect to publish the full paper in a future issue of *TUGboat*. *Ed.*)