THE PracTeX Journal

Issue Table of Contents 2006, Number 1

[Published 2006-02-18]

Notices

Articles

Columns

THE PracTEX Journal

From the Editor: In this issue: Fonts

Lance Carnes

- Comment on this paper
- Send submission idea to editor

Issue 2006-1 is the first issue of the PracTeX Journal's second year. For this issue we tried announcing a "fonts" theme and were pleasantly surprised at the number of high-quality articles received. Many who use TeX have probably never used anything beyond the default fonts, while some may have tried `\usepackage{times}` but may not have tried `\usepackage{palatino}`. Hopefully, after perusing the articles in this issue, you will become more comfortable using new fonts and techniques.

The lead off article is Typographic Opportunities by Tamye Riggs of SOTA (Society of Typographic Aficionados). She gives an overview of the world of digital type, offers some tips for beginning and experienced users, and lists numerous resources. We contacted Tamye to provide an article and a font quiz, and she and her colleagues at SOTA provided everything we asked for and more — for the font quiz they even donated a valuable prize. If you would like to know more about type and type design, consider joining SOTA or attending the TypeCon conference, to be held this summer in Boston.

We are fortunate to have an article on LaTeX font usage by Walter Schmidt, who has been a major contributor to the TeX community in the area of fonts. In this article he answers some frequently-asked font questions, such as how to use the basic font selection commands and how to change the default fonts for a document.

Gerben C. Th. Wierda, Thomas A. Schmitz and Adam T. Lindsay describe font usage in Mac OS X Fonts in pdfTeX. This article decribes automated font installation with a Mac-based TeX system, but their techniques apply to other host systems as well. (Also see previous font-related articles by Schmitz and Lindsay.)

In Michael Spivak's The MathTimeProfessional Fonts we learn about the events and requirements that led this mathematician and math publisher to design the popular *MathTime Professional* fonts. Those who have read his beautifully typeset books or used his fonts for their own publications will be curious to know how and why he became a font designer. He concludes his article wistfully hoping this is the end of the font design era in his life, but I suspect he will be back with new designs in coming years.

Will Robertson writes about the relatively new set of Latin Modern fonts, a large collection of fonts for typesetting documents in many languages. The creators of these fonts, Bogus•aw Jackowski and Janusz M. Nowacki, intend this font collection to be the successor to Donald Knuth's Computer Modern fonts.

For those who need to use Asian fonts in their documents Helmer Aslaksen describes Using the CJK LaTeX Package, including how to install it, how to enter text, and other nuances of this package. The CJK (Chinese, Japanese, Korean) package, developed by Werner Lemberg, is widely used and this article provides a useful introduction to it, with several sample documents.

Stephen Hartke's Survey of Free Math Fonts shows numerous math font examples, showing how various math and text fonts combine in the same document. He shows where you can obtain the fonts and how to use them.

David Walden's popular Travels in TeX Land column describes his experiences buying, installing, and beginning to use the Lucida fonts. A good read for those who are "font shy", as David admits he was before writing this piece. Ask Nelly answers some commonly-asked questions about TeX font usage. And finally, for those who like puzzles and challenges, Distractions offers a few font quizzes with **valuable prizes** including fonts, books, and CDs. You can also read about the winners of last issue's Sudoku contests and see their solutions, including two TeX-based sudoku solvers!

PracTeX Journal readers provided insightful feedback. As you read the articles and columns please use the response links to send comments. If you use a technique from an article or column, be sure to contact the author and report how it worked for you. *The PracTeX Journal* is still evolving and your feedback will help us as we strive to improve it.

Next Issue

The 2006-2 issue will not have a theme and we will be accepting articles on any topic. If you would like to contribute an article or technical note on any aspect of LaTeX, TeX, or ConTeXt, please send an article outline to the editors.

Although many of the articles published by The PracTeX Journal are formatted with LaTeX, TeX, or ConTeXt, it would be nice to see more submissions written in straight text or in html (as your editor does). Brief technical notes, techniques that might be useful to others, or an account of how you tried to use something and (hopefully) eventually succeeded, are welcome.

Thanks

The Editorial Board and I want to thank the authors, columnists, and Ask Nelly answerers for their excellent pieces which make this journal possible. We also want to thank those who worked behind the scenes:

Reviewers and copy editors: Karl Berry, Jin-Hwan Cho, David L. Elliott, Baden Hughes, Werner Lemberg, John O'Rourke, Will Robertson, Thomas Schmitz, William Slough, Juan Luis Varona, David Walden, and Candy Yiu.

Production editors: Will Robertson, William Slough, and Paul Thompson.

See also other key people who make this publication possible.

And **special thanks** to the companies who donated prizes for the font quizzes in the Distractions column: Publish or Perish, Inc., Bigelow & Holmes Inc., SOTA (Society of Typographic Aficionados), and FontShop.

And finally, be sure to attend Practical TeX 2006, a workshop and conference being held this summer at Rutgers University in New Jersey. You'll meet other PracTeX Journal readers there, and learn first-hand about using LaTeX, TeX, and ConTeXt.

Lance Carnes
Editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

THE PracTEX Journal

Feedback

From Readers

- Comment on this paper
- Send submission idea to editor

I've just finished reading [Klaus Hoeppner's Strategies for including graphics in LaTeX documents] and I have to say it is the best document I've read on the subject. Clear, to the point, and extremely useful. A must-read!

Rui Maciel

TPJ

I received the PracTeX postcard yesterday and I really like it — the little mouse is superb!

Gianluca Pignalberi

[Gianluca was a *Distractions* contest winner. Be sure to enter one of the contests in this issue and possibly win a postcard or one of the first prizes. -Ed.]

TPJ

[Jim Hefferon's Minutes in Less Than Hours: Using LaTeX Resources] I, too, had to write my own minutes class — there are some classes out there, but the ones I looked at were oriented towards European-style minutes, which are a bit different from the minutes I was producing.

I did some things the way Jim did, but there are a few differences that might be of interest ... My minutes class may not be usable as is since it has a number of nonstandard dependencies, but there may be some ideas in there worth cribbing. If I ever get around to making my own class more generic, I will probably steal some ideas from Jim's class, as well.

Claire Connelly
Harvey Mudd College

[Andrew Mertz and William Slough Beamer by Example] This is a very useful paper. The authors are correct that the beamer user's guide, while admirably complete, is daunting at first glance. My sense is that one of the barriers to wider use of LaTeX is a lack of someone to hold the beginner's hand during those first steps. This article, and the PracTeX journal generally, are a great deal of help.

John Burt
Brandeis University

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

THE PracTEX Journal

Invitation to PracTeX'06

Robin Laakso

- Visit the Practical TeX 2006 home page
- Publicity flyer for the conference (please post)
- Email questions about conference
- Email submission idea for conference



Photos of Rutgers

On behalf of the TeX Users Group and Rutgers, The State University of New Jersey, I invite you to attend the Practical TeX 2006 workshop and/or conference. Both the workshop and conference will be held at Rutgers. Building on the success of PracTeX 2004 in San Francisco and PracTeX 2005 in Chapel Hill (NC), this year we're offering a full four-

day LaTeX workshop, followed by three days of talks and presentations. All aspects of the workshop and conference will focus on practical techniques for document production using LaTeX, TeX, ConTeXt, MetaPost, and friends.

Hope to see you there!

Robin Laakso
TeX Users Group

Practical TeX 2006

Workshops and Presentations on LaTeX, TeX, ConTeXt, and more

LaTeX Workshop: July 25-28, 2006
Practical TeX Conference: July 30-August 1, 2006

Bush Campus, Rutgers, the State University
Piscataway, New Jersey, USA

http://tug.org/practicaltex2006
conferences@tug.org

Keynote address: Barbara Beeton, American Mathematical Society and TeX Users Group

Conference web pages:

- **Register!**
- **Call for papers:** abstracts due April 1, 2006.
- Conference program & participants.
- LaTeX workshop.
- Conference sponsorship opportunities.
- Rutgers/New Jersey local information.
- Publicity flyer (please post).

This four-day conference focuses on practical techniques for document production using LaTeX, TeX, ConTeXt, MetaPost, and friends. It includes one day of classes and tutorials, followed by three days of presentations.

Hope to see you there!

Further information

Conference attendees will enjoy an opening night reception and an (optional) banquet one evening. Coffee and lunch will be served each day of the meeting. Located on the Busch Campus of Rutgers University in Piscataway, New Jersey, an easy train ride from New York City.

Conference fee and hotel information is available on the registration page.

Conference flyer and publicity

Mentioning the conference to colleagues and in any other contexts would be very much appreciated.

We'd also be grateful for any posting of this one-page PDF flyer (TeX source).

Sponsorship

If you or your organization would like to help sponsor the conference, numerous options are available, from a straight cash donation (always welcome!) to logos on the conference memorabilia. Please see this separate sponsorship page for details, or email us.

We are very grateful to many contributions.

Contact

Email: conferences@tug.org
Phone, fax, postal mail: see TUG office contact information.

Sponsored by the TeX Users Group. This conference follows 2004's Practical TeX conference in San Francisco and 2005's Practical TeX conference in Chapel Hill, NC.

Page generated February 28, 2006 ;TUG home page; search; contact webmaster.

THE PracTEX Journal

Typographic Opportunities

Tamye Riggs

Abstract

This introductory article describes the current market in digital type, offers some insights on the use of different typefaces, and offers some tips for everyone interested in typefaces, both novice and experienced users. Several on-line typeface resources are given, including type foundries, font development tools, and typograpic organizations, conferences, and discussion groups.

Tamye Riggs is a writer, typographer, and designer who's edited several magazines and books on typography and motion graphics. She also serves as the Executive Director of SOTA (The Society of Typographic Aficionados), a non-profit organization which publishes a magazine, "Interrobang," and produces the annual TypeCon conference, to be held this summer in Boston.

(Be sure to try the font quizzes devised by Tamye Riggs and Yves Peters.)

- Comment on this paper
- Send submission idea to editor

I was chatting recently with an executive from one of the world's largest font distributors. He casually informed me that, this year, his company would have more than 80,000 fonts available for download from their website. That's a staggering figure, but it's not entirely surprising.

Twenty years ago, the introduction of personal computers opened the world of publishing to the masses. The past decade has seen the rapid evolution of digital font production tools for individual use and the explosive growth of internet marketing and communications. Type designers worldwide now have the ability to produce fonts and make them instantly available over the internet. Some font producers market and sell from their own websites, while others prefer to distribute through large font houses. Either way, new type is available 24/7 for customers with internet access and a valid credit card.

With 80,000 fonts a few mouse clicks away, why do Times Roman and Arial still dominate the typographic landscape? Granted, these two fonts (or close variants)

come standard with most computer operating systems, and are also displayed in most web browsers. But just because they're easily accessible doesn't make them the right tools for every job.

System fonts, while appropriate for many tasks, are overused and often misused. Comic Sans is perfect for comic strip speech bubbles, kid's birthday party invitations, or a letter to a younger sibling. It's just not the best choice for professionally printed direct marketing pieces or business correspondence. The same goes for Times, Arial, Courier, and the rest — they're seen so much that readers may get bored or even completely ignore text when those fonts are used. I liken it to hearing a song on the radio too many times — what once sounded fresh and creative soon becomes annoying. Eventually, we tune it out.

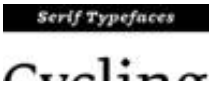
I'm not picking on system fonts, per se — they were developed to fit specific needs, and to give computer users fonts to work with right out of the box. For some purposes, it's easier and more practical to stick with commonly available fonts. When distributing an editable Word doc to a dozen people in different locations with different types of computers, it only makes sense to use cross-platform standards like Arial or Times.

Compatibility and commonality are not the only issues at hand. Some users prefer to stick with system fonts because they're already paid for. While saving money is always a consideration, anyone working with professional documents should consider unique typefaces to be an investment, a cost of doing business. Quality fonts from many commercial foundries and independent designers are affordably priced. Single fonts are often less than $20 each for standard licensing, with deeply discounted pricing available for multi-font families or large collections.

Many font retailers offer type viewers and advanced search engines on their websites. These tools make it possible to take a close look at typefaces before any money changes hands, enabling users to more easily find fonts that will meet their needs. There are also a number of type distributors that offer one or more fonts free of charge, as a marketing technique and a way of thanking their customers.

Take advantage of the opportunity to expand your typographic horizons. When you don't have to use a system font, don't. Seek out newer designs, or inspired classics that haven't been overexposed through the years. Typography is exciting — it's a kick to find a typeface that fits your style and delivers your message with finesse. And with so many fonts available, the possibilities are endless. You might even find a typeface that only a few other people in the world know about — it will seem almost as though the designer created it especially for you.

Alternative Types to Explore



Serif Typefaces

Cycling — Fedra Serif B, Typotheque

Quizzical — Delicato, Fountain

Amplify — FF Absara, FontFont

Readily — Bohemia, Linotype

Sincerely — Eidetic Neo, Emigre

Chapeau — Rongel, Feliciano Type Foundry

Worldly — Nicholas, ShinnType

Sans Serif Typefaces

Sans Serif Typefaces

Washing — Adesso, Présence Typo

Belgium — Freight, GarageFonts

Baysider — Helvetica Neue, Linotype

Charged — John Sans, Storm Type Foundry

Vegan — Chianti, Bitstream

Metropol — Mercury, Fountain

Grange — Maple, Process Type Foundry

Script Typefaces

Script Typefaces

Destroy — Ayres Royal, Wiescher Design

Gracefully — Sloop, Font Bureau

Specialty — P22 Hopper Josephine, P22

Evangeline — Cocktail Shaker, Font Diner

Welcoming — Pendulum, Canada Type

ITC Smack, International Typeface Corporation

Bello, Underware

Headline Typefaces

Avebury, Parkinson Type Design

Celestia Antiqua, MVB Design

Bfrika, Holland Fonts

Samba, Linotype

TX Tiny Tim, Typebox

Coquette, Mark Simonson Studio

Glyphic Neue, Typeco

Typographic Resources

There are thousands of type foundries, distributors, and independent designers making and selling fonts. Below is a representative sampling of companies, as well as other resources about typography.

Major foundries and distributors

Adobe
Bitstream
Font Haus
FontShop
Linotype
Monotype Imaging (formerly Agfa Monotype; includes ITC and Letraset labels)
MyFonts
Paratype
Phil's Fonts
URW
Veer

Large independent foundries and distributors

Elsner + Flake
Emigre

Font Bureau
FontFont
GarageFonts
House Industries
P22 (includes IHOF, Lanston, and other labels)
T-26

Small independents

3ip
Ascender Corporation
Altered Ego Fonts
Comicraft
Carter & Cone
Joshua Darden
Device
Feliciano Type Foundry
FlashFonts
Font Diner
Fountain
Hoefler & Frere-Jones
Holland Fonts
Letterror
LucasFonts
Orange Italic
Parkinson Type Design
Porchez Typofonderie
Présence Typo
Process Type Foundry
Psy/Ops
ShinnType
Mark Simonson Studio
Storm Type Foundry
Test Pilot Collective
Thirstype
Typebox
Typeco
Typotheque
Underware
Village
Virus

Large lists of foundries

Microsoft Typography Foundry List
Typophile Wiki

Font development tools

DTL FontMaster
FontLab (many font dev tools, including Fontographer, FontLab, and TypeTool)

Typographic organizations

Association Typographique Internationale (ATypI)
The Society of Typographic Aficionados (SOTA)
The Type Directors Club (TDC)

Type conferences

ATypI (Fall/mostly in Europe)
Kitabat Arabic Calligraphy and Typography Conference (Spring/Dubai)
St Bride (Fall/UK)
International Conference on Typography and Visual Communication (Summer/Greece)
TypeCon (Summer/North America)
Typo Berlin (Spring/Berlin)

Online typography discussion and news

Creative Pro
Daidala
Fontzone
Microsoft Typography News
Speak Up
Type Radio
Typographer
Typographica
Typophile

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

THE PracTEX Journal

Font selection in LaTeX
The most frequently asked questions

Walter Schmidt

Abstract

This article tries to answer the three most popular questions regarding font selection in LaTeX. Rather than providing ready-made answers, it tries primarily to provide some guidance through the existing documentation.

Walter Schmidt is well-known in the TeX community as the maintainer of PSNFSS (PostScript New Font Selection Scheme), a key element in the LaTeX system. He is also proficient in designing font macro packages for LaTeX, and is author of the mtpro package for MathTimeProfessional fonts and the lucimatx package for Lucida fonts. He maintains a math fonts web page (in German) containing samples and comparisons of several math fonts that can be used with LaTeX. Walter lives in Erlangen, Germany.

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

The PracTEX Journal
TPJ 2006 No 01, 2006-02-01
Article revision 2006/02/15

Font selection in LaTeX: The most frequently asked questions

Walter Schmidt

Abstract This article tries to answer the three most popular questions regarding font selection in LaTeX. Rather than providing ready-made answers, it tries primarily to provide some guidance through the existing documentation.

1 Basic commands

Having read any LaTeX introduction of your choice (for instance, [1]), you should be familiar with the basic commands for font selection. To start with, let's summarize them once again.

The following declarations let you choose between three pre-defined font families:

`\rmfamily`	selects a roman (i.e., serifed) font family
`\sffamily`	selects a sans serif font family
`\ttfamily`	selects a monospaced ("typewriter") font family

Within each font family, the following declarations select the "series" (i.e., darkness or stroke width),

`\mdseries`	regular
`\bfseries`	**bold**

and the "shape" (i.e., the form of the letters):

`\upshape`	upright
`\slshape`	·····
`\itshape`	*italic*
`\scshape`	Caps and Small Caps

These commands are "declarations", i.e., they remain in effect until the end of the current group or environment. For each declaration there exists a text-generating command as a counterpart; it typesets only its argument in the desired style, e.g., `\textsf` corresponds to `\sffamily`. See, e.g., [1], chapter 3.1 and appendix C.15.1.

Family, series and shape can be combined, e.g., `\bfseries\itshape` results in ***bold italic*** type. Notice, however, that not every possible combination is required to exist; for instance, many font families are lacking small caps.

This scheme is called NFSS (New Font Selection Scheme), and its official documentation [2] is available in every LaTeX system as a DVI or PDF document `fntguide.dvi` or `.pdf`.

2 How can I change the default fonts for the whole document?

Most likely, you know already that the three default font families used by LaTeX are

roman:	Computer Modern Roman
sans serif:	Computer Modern Sans Serif
monospaced:	`Computer Modern Typewriter`

Outside the world of TeX, these font families are far from popular, so the question asked in the title of the present section is perhaps the "top of the FAQs".

The families selected by `\rmfamily`, `\sffamily` and `\ttfamily` are determined by the values of the related macros `\rmdefault`, `\sfdefault` and `\ttdefault`. They can be altered using the well-known `\renewcommand`—provided that you know the name of the desired font family. You can easily verify this by adding the command

```
\renewcommand{\rmdefault}{ptm}
```

to the preamble of a document. `ptm` is the name under which the font family "Times" is installed in your LaTeX system, so all (roman) text in your example document should change from CM Roman to Times. (Most likely, you will now ask the question how to learn the name of a font family: please, be patient, it will be answered in the next section.)

If, however, there is any piece of maths in your example, you will notice that changing `\rmdefault` does *not* affect formulas. In the above case, they will still be typeset using the CM math fonts, which do not blend well with Times.

Changing the math fonts requires more effort than simply redefining a few macros. That's why alternative math fonts are usually accompanied by a *macro package*: loading the package effects all changes needed to replace the default (CM) math fonts; in many cases these packages take care of redefining `\rmdefault` appropriately, too. For instance, to change both text and formulas to Times, you would in fact have to add the following line to your document preamble, rather than the one given above:

```
\usepackage{mathptmx}
```

There are also macro packages that change only one of the text fonts, but provide additional features such as scaling.

This raises a few questions: You need to know *which* alternative fonts besides Computer Modern are available in your LaTeX system at all, you need to know the "LaTeX names" of the font families you want to use, and you need to know if there are any related macro packages available. These issues lead us to the next section:

3 Which font families are available in my LaTeX system?

There is a minimum set of alternative fonts that must *always* be available beside Computer Modern; the related collection of macro packages is often referred to as the "PSNFSS collection". In particular, it supports the use of the popular typefaces Times, Helvetica, Palatino and Charter (and a few others), and it supports math fonts that suit Times and Palatino. The related documentation [3] is available in every LaTeX distribution as a PDF file named `psnfss2e.pdf`, usually in the directory `doc/latex/psnfss`. Reading this document is *strongly* recommended. It tells the 'family names' of the supported fonts (such as `ptm` above), and it explains the usage of the related macro packages. (See [3], tables 1 and 3.)

Everything that goes beyond the PSNFSS collection is, strictly speaking, optional; i.e., only the documentation of your particular TeX distribution can tell

you which fonts are shipped with the system, and where the related documentation is installed.

Most contemporary TEX distributions come with (almost) all free text and math fonts that are available from CTAN; a good overview of the most popular ones is given in chapter 7 of the "LATEX Companion" [4].

Further font families that have been made available for use with LATEX are summarized in [5].

4 How can I change the fonts to be used for certain parts of the document?

A popular request is to customize the style of certain elements of the document, in particular the font used in the section headings and/or in captions. The style of these elements, including the font choice, is determined by the document class you are using. Unfortunately, the standard classes (article, report, book) do by default not provide any means for this kind of customization.

One solution is to use extra packages that add the required functionality; the most popular ones are the packages `titlesec` [6] and `sectsty` [7] to change the style of the section headings, and `caption` [8] to control the style of the captions of figures and tables. Use of these packages is described in detail in the related documentation.

Newer, alternative document classes often go a different way. The KOMA-script classes [9] as well as the Memoir class [10] provide various means for customization. As an example, let's take a look at the interface of the KOMA classes to control the style of section headings:

By default, the KOMA classes use the bold series of the sans-serif font family to typeset headings. To change this, the command

```
\setkomafont{sectioning}{...}
```

is provided. Its second argument is to contain all declarations to be applied when the section headings are typeset. A frequent requirement is to use the bold series of the roman font family instead (as in the standard classes), and additionally to apply `\boldmath`, so that mathematical elements in section headings are emboldened, too. Doing so is straightforward with the KOMA classes:

```
\setkomafont{sectioning}{\rmfamily\bfseries\boldmath}
```

Correspondingly, the style of captions can be controlled via the command `\setkomafont{caption}{...}`.

In general, with these extra packages and classes it becomes a matter of a single line of LaTeX commands to change the formatting of many parts of the document.

5 Conclusion

At first sight, font selection in LaTeX looks like a relatively complex issue, because it differs fundamentally from font handling in "classical" DTP programs. Yet, it isn't hard when you look in the documentation, and in this article we have tried to show some concise pointers.

References

[1] Leslie Lamport: LaTeX: A Document Preparation System.
2nd edition. Addison Wesley, 1994.

[2] LaTeX3 Project Team (Ed.): LaTeX2$_\varepsilon$ font selection.
Part of the LaTeX online documentation, file `fntguide.pdf`

[3] Walter Schmidt: Using common Postscript fonts with LaTeX.
Part of the LaTeX online documentation, file `psnfss2e.pdf`

[4] Frank Mittelbach et al.: The LaTeX Companion.
2nd edition. Addison Wesley, 2004.

[5] TeX User Group: Fonts and TeX.
http://tug.org/fonts

[6] Javier Bezos: The titlesec and titletoc packages.
CTAN:macros/latex/contrib/titlesec/

[7] Rowland McDonnel: The sectsty package.
CTAN:macros/latex/contrib/sectsty/

[8] Axel Sommerfeldt: The caption package.
CTAN:macros/latex/contrib/caption/

[9] Markus Kohm: The KOMA-script bundle.
CTAN:macros/latex/contrib/koma-script/

[10] Peter Wilson: The Memoir class.
CTAN:macros/latex/contrib/memoir/

THE PracTEX Journal

Mac OS X Fonts in pdfTeX

Gerben C. Th. Wierda, Thomas A. Schmitz and Adam T. Lindsay

Abstract

Installing a new font with your TeX installation can be a challenging task. This article documents an attempt to provide an automated solution for users running TeX on Apple's OS X. Using the fonts described here will only be possible for those who run this operating system; the way of making these fonts work with TeX should be of interest for all users. The article's level can be described as intermediate to advanced; it assumes some previous knowledge of TeX and fonts.

Gerben Wierda has been working on a book since 1995 but has been sidetracked by typesetting (where have we heard this before?), work, a child, an MBA, another child, more work, renovations, life in general and still more work. Having used mainly LaTeX since 1986, he recently switched his book project to ConTeXt. Gerben can be reached at `Gerben.Wierda@rna.nl`

Thomas A. Schmitz is Professor of Classics at Bonn University, Germany. After his last book project, he became so dissatisfied with the infamous word processor from a Redmond-based software company that he took the plunge and learned to use ConTeXt, a very nice TeX macro package. And he hasn't looked back ever since. Thomas can be reached at `thomas.schmitz@uni-bonn.de`

Adam Lindsay is a Research Fellow in Computing at Lancaster University, UK. He is an internationally recognized expert in MPEG-7, and finally approached the TeX world in 2002 after being required to edit large ISO standards using Microsoft Word. Being a Mac user, he is naturally passionate about fonts. Adam may be contacted at `atl@alum.mit.edu`

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

The PracTEX Journal
TPJ 2006 No 1
Rev. 2006–02–01

Mac OS X Fonts in pdfTEX

Gerben C. Th. Wierda · Thomas A. Schmitz · Adam T. Lindsay

Installing a new font with your TEX installation can be a challenging task. This article documents an attempt to provide an automated solution for users running Gerben Wierda's distribution of the TEX system on Apple's OS X. Using the fonts described here will only be possible for those who run this operating system; the way of making these fonts work with TEX should be of interest for all users. The article's level can be described as intermediate to advanced; it assumes some previous knowledge of TEX and fonts.

1 Introduction: It all began with a rant...

If you look into a TEX mailing-list or one of the newsgroups like `comp.text.tex`, you know the theme is a classic: using fonts in TEX. This time, it cropped up on the mailing-list "TeX on Mac OS X." In November 2005, some members complained they found it too hard to make the fonts that come with their system work with TEX.

At that point, Gerben Wierda privately contacted Thomas Schmitz. Gerben organizes the most popular (re-)distribution of the TEX-system for OS X. It is built on top of teTEX and TEXLive, but it also adds a couple of features that users have asked Gerben to add (such as additional fonts or the `prosper` class for LATEX). Gerben suggested that support for some of the fonts that come with the operating system should be made part of gwTEX. Soon, Adam Lindsay, who had already thought about a similar project, joined forces, and we spent a couple of days of frantic activity and hundreds of e-mail messages with assorted attachments until everything was in place and a first release of gtamacfonts could be made. This article will document what we did, how we did it, and how it was integrated into the distribution. We will give an outline of the philosophy behind it

all and of the general concept without going into the gory details here. Most of this technical stuff is amply documented elsewhere, e.g. in **Adam's article**[1] on OpenType fonts and **Thomas's article**[2] on TrueType fonts.

2 The General Concept

Support for the fonts involved two steps:

1. TEX needs a number of files in order to work with these fonts:

   - The metric files (`.tfm` and `.vf`) containing information about the boxes of the font's characters, the kerning, and the ligatures. All of this is needed by TEX to calculate where characters are to be placed;

   - a map file (`.map`) that declares the relation between these metric files and the font files that contain the actual outlines ("glyphs") of the characters (`.ttf`);

   - encoding files (`.enc`) that will allow us to extract a subset of 256 glyphs (shapes) from the fonts (which contain many more characters) per encoding (see below for an explanation);

   - for LATEX: font definitions (`.fd`) for different encodings and (optionally) packages (`.sty`) to facilitate choosing the fonts in your documents;

   - for ConTEXt: a typescript (`type-***.tex`) that organizes the fonts into families and allows using the normal font commands such as `{\em }` or `{\ss }`.

2. The fonts themselves have to be converted into a format that TEX can use. Macintosh systems typically put all the files for a font (regular, italic, bold, bold italic) into one big container ("font suitcase"). Mac OS X updated this concept with yet another type of container, the `.dfont` file. Ordinary TEX cannot work with these containers,[3] so

[1] `http://www.tug.org/pracjourn/2005-2/lindsay/lindsay.pdf`
[2] `http://www.tug.org/pracjourn/2005-2/schmitz/schmitz.pdf`
[3] But SIL International's XƎTEX can, and it also provides an interface to special aspects of the fonts.

the information needs to be extracted and put into regular font files. All the fonts shipped on newer Macintosh systems that are contained within `.dfont` files are TrueType fonts (`.ttf`) which can be used by pdfTEX, but not by `dvips`.

As we will see, most of the fonts demanded some special treatment, so we could not rely on a completely automated tool (such as `texfont` or `fontinst`) to do all the steps. We had to write quite a few of the files by hand, and we had to tweak some of the files that were produced by automatic tools.

3 Encodings

The first step was thinking about encoding vectors. An encoding vector defines a relation between TEX's internal representation of characters and glyphs (shapes of the 'characters' in a font). TEX is 8-bit-oriented and thus limited to encoding vectors with 256 positions. However, modern fonts may contain thousands of glyphs. A TEX encoding is therefore often a compromise on the availability of glyphs. Support for all the shapes in a font in TEX often means that it has to be broken up into many fonts, each with 256 possible glyphs. We decided we wanted support for the two most popular encodings:

1. EC, a.k.a. Cork encoding or T1 in the LATEX world. EC-encoded fonts offer numerous accented letters from many European languages. In LATEX, they are usually accompanied by a symbol set in the so-called TS1-encoding.

2. TEX 'n' ANSI, a.k.a. LY1 in the LATEX world. Texnansi has been more popular with ConTEXt users; it offers a reasonable number of accented letters and a selection of interesting symbols.

For most of the fonts, we could use the encoding files that come with teTEX. However, some of the fonts supply oldstyle numerals and/or small caps, so we had to define six additional encoding vectors:

☞ An EC- and texnansi-encoding that would produce oldstyle figures *and* small caps,

Compared to XƎTEX's interface, the method described here is less rich, but does not require a different TEX dialect, so it's more portable.

called `ec-sc.enc` and `texnansi-sc.enc`;

- an encoding that would produce oldstyle figures but regular lowercase letters (`ec-os.enc` and `texnansi-os.enc`);

- and, finally one that would produce lining figures and small caps, called `ec-sclf.enc` and `texnansi-sclf.enc`.

Unfortunately, this uniform approach did not work for the beautiful HoeflerText font (which has been used to typeset this document). Not only does the font have non-standard glyph names, but these names are not consistent across the regular, italic, and bold ("black") variants! We had to produce an entire set of EC- and texnansi-encodings for the variants of Hoefler.

4 Producing the Metric Files and the Fontmap

When we began producing the metric files from the fonts, we had to take one detail into account: TrueType fonts come in various forms. Macintosh TrueTypes have kerning- and ligature-information in parts of the files where the automated tools cannot find them.[4] That meant we could not use a tool such as `ttf2tfm` that would directly produce metric files from a TrueType file. Instead, we had to take a two-step approach:

1. Produce Adobe Font Metrics (`.afm`) from the fonts; these contain the kerning and ligature-information;

2. convert these `afm`s into TEX metrics, using one of the encoding files described above.

For step 1, we used the tool `fontforge`, written by **George Williams**[5]. It can produce a full `afm` from a Macintosh font, containing all the glyphs and the information about kerning and ligatures. Essentially, we used `fontforge` to convert the font from

[4] Before you leap to the conclusion that it's the case of the Macintosh platform being non-standard once again, consider for a moment that Apple invented the TrueType format.

[5] `http://fontforge.sourceforge.net/`

TrueType to PostScript Type 1. In the process, it produced an afm-file as well; we used this file and discarded the resulting .pfb file.

We could then take the resulting afms and produce TEX Font Metrics (.tfm) and Virtual Fonts (.vf) with the tool afm2tfm, which is part of a regular TEX installation. You can feed this tool an encoding and an afm, and it will convert it to a pair of vf and tfms. Since Adam and Thomas were both producing metrics, we had to be careful to be absolutely consistent with naming so that we could work in parallel. A run might look like this:

```
afm2tfm Didot.afm -T ec-sc.enc -v ec-sc-Didot ec-sc-raw-Didot
```

Where possible, afm2pl was substituted when generating texnansi-encoded files, as the tool avoided creating unnecessary virtual fonts.

There was one problem for which we could not find an automated solution: the generated HoeflerText metrics, for reasons we could not understand, never included ligatures for "fi" and "ffi." In the end, we edited the virtual fonts by hand and inserted two lines that define these ligatures.

After producing the metrics, we had to add a line to our mapfile (gtamacfonts.map) for every new tfm, in this form:

```
ec-sc-raw-Didot Didot 4 <Didot.ttf ec-sc.enc
```

After this was done, we now had a couple of dozen of new "fonts" (for TEX, every tfm constitutes a new font, even if it is just a variation of an existing font with a different encoding). In theory, you could now use them for TEXing with the primitive TEX \font command, but in order to use them with your favorite macro package, a few more files are necessary for everyday convenience.

5 Support for LATEX

In LATEX, support for a font usually consists of two types of files:

1. Font definitions (.fd) tell LATEX which tfms constitute the same font family: which one is the roman serif, italic, bold, slanted, small caps font? Their basic structure is easy to understand, but LATEX needs one fd for every encoding. For,

say, the Baskerville font, we have thus three files: `t1gtamacbaskerville.fd` for EC-encoding, `ts1gtamacbaskerville.fd` for the accompanying symbol font, and `ly1gtamacbaskerville.fd` for the texnansi-encoding.

2. A LaTeX-package (`.sty`) usually contains just a few lines of code containing, e.g., a redefinition of the main font in the form `\renewcommand{\rmdefault}{gtamacbaskerville}`, which will make the font family defined in `t1gtamacbaskerville.fd` the roman default for the document. The packages distributed with our implementation contain a few extra features as will be seen below.

The font definitions had to be hand-written. But it's a task that can be done very fast: these files define every imaginable combination and weight, but our fonts typically have only four of them (regular, italic, bold, bold-italic, maybe small caps). So you need to fill in the names of these four `tfm`s; all the rest of the lines just tells LaTeX to make appropriate substitutions. As an example, here's the font definition for Didot in texnansi-encoding (we have not included all of the lines just defining substitutions):

```
\ProvidesFile{ly1gtamacdidot.fd}

\DeclareFontFamily{LY1}{gtamacdidot}{}

\DeclareFontShape{LY1}{gtamacdidot}{m}{n} {<-> texnansi-Didot}{}
\DeclareFontShape{LY1}{gtamacdidot}{m}{sc}{<-> texnansi-sc-Didot}{}
\DeclareFontShape{LY1}{gtamacdidot}{m}{sl}{<-> ssub * gtamacdidot/m/it}{}
\DeclareFontShape{LY1}{gtamacdidot}{m}{it}{<-> texnansi-DidotItalic}{}

\DeclareFontShape{LY1}{gtamacdidot}{cb}{n} {<-> texnansi-DidotBold}{}
\DeclareFontShape{LY1}{gtamacdidot}{cb}{sc}{<-> ssub * gtamacdidot/cb/n}{}
\DeclareFontShape{LY1}{gtamacdidot}{cb}{sl}{<-> ssub * gtamacdidot/cb/it}{}
\DeclareFontShape{LY1}{gtamacdidot}{cb}{it}{<-> ssub * gtamacdidot/m/it}{}

\endinput
```

As you can see, only four lines contain actual definitions: for regular (`m/n`), small caps

(m/sc), italic ("slanted," m/sl), and bold (cb/n). All the other lines define "silent substitutions" (whence the ssub *).[6] Didot doesn't have bold italic, so gtamacdidot/cb/it is substituted by regular italic, gtamacdidot/m/it.

Once you have such a template, preparing fds for other encodings and other fonts is just a matter of find/replace.

However, we wanted to give users a bit more help, so we added some more features to our packages:

- Many of the fonts we make accessible are sans-serif fonts. They will almost always be used to go with a regular serif-font. Since design-sizes of fonts differ a lot, we wanted to add the possibility for users to scale these fonts in order to obtain a homogeneous look in their documents. We copied the code for doing this from other packages, such as helvet.sty. Users can now call the package with a "scale" option:

  ```
  \usepackage[scale=0.92]{gtamacgillsans}
  ```

  Because of the way LaTeX sets up fonts, the code for using this feature must be both in the fd and in the package itself, which is why we have not implemented scaling for fonts that are already supported and have their own font definitions. This was the case for the Latin Modern font (lmtt) which we defined as the default mono font.[7]

- The ConTEXt typescripts define collections of fonts consisting of serif, sans, and mono typefaces. We did the same for LaTeX, so the main serif packages (baskerville, didot, georgia, and hoefler) redefine rmdefault, sfdefault and ttdefault. We also added a default scale to make the sans-fonts look good with the main serif font. For mono, we use Latin Modern.

- HoeflerText and Didot offer oldstyle figures. Many users prefer to have these figures made the default in their documents (such as in this one). For them, we have added an option to these two packages, for example:

  ```
  \usepackage[osf]{gtamacdidot}
  ```

[6] If you want to know more about the structure and syntax of fd files, have a look at section 7.10.3 of the LaTeX-Companion.

[7] In ConTEXt, scaling is implemented on the level of the individual typeface-definition; in LaTeX, it has to be present in the fd file as well. For the moment, we have decided to stick with the fd-files distributed with tetex, so we haven't implemented scaling for Latin Modern, but we may change our minds for the next release...

When this package option is used, oldstyle figures are default and lining figures are made available in the small caps font, accessible with the `\textsc` command.

There are now four LaTeX packages that define a serif, sans, and typewriter font:

```
gtamacbaskerville.sty
gtamacdidot.sty
gtamacgeorgia.sty
gtamachoefler.sty
```

Seven packages define a sans font only:

```
gtamacfutura.sty
gtamacfuturacondensed.sty
gtamacgillsans.sty
gtamachelveticaneue.sty
gtamaclucidagrande.sty
gtamaclucidagrande.sty
gtamacverdana.sty
```

And, finally, one package defines a typewriter font:

```
gtamacamericantypewriter.sty
```

6 Support for ConTEXt

ConTEXt support was a bit more flexible than that with LaTeX because there is generally less pressure to present all features with a unified interface. Every font's set of features is different, so why not give users those features as needed?

ConTEXt font support is generally achieved with typescript files (`type-*.tex`). These files include typescript definitions, which (for our purposes) fall into one of the following four categories:

1. Class–to–symbolic names that link a generic name (such as 'Serif' or 'SansItalic') with a name specific to the font family (such as 'HoeflerText-Regular' or 'GillSans-

LightItalic').

2. Symbolic–to–font names that link the specific font name above with an encoding–specific name that corresponds with an actual `.tfm` file on the user's system.

3. Map typescripts that trigger the loading of font map files. Map files create a relationship between the `.tfm` file that TEX knows about and a specific TrueType or Type 1 file. They are important as they tell the PDF building step where to find the font file that contains the glyphs.

4. Typeface typescripts group typescripts into family definitions. I would personally discourage using such preset definitions, as they have their limitations, but users find their convenience very compelling.

In the ConTEXt distribution, these groups of definitions are stored in different files (`type-syn`, `-enc`, `-map`, and `-exa`, respectively. In the gtamacfonts distribution, they are contained in one file (`type-gtamacfonts.tex`), for convenience.

The typescripts were written by hand, as each font family provided just enough variation to keep things interesting (and non–automatable). The real trick in typescripts is in the naming, keeping the typescript namespaces distinct, but overlapping enough to utilize the heavy redundancy across scripts. As an example, here are the typescripts for Gill Sans:

```
\starttypescript [sans] [gillsans] [name]
 \setups            [font:fallback:sans]
 \definefontsynonym [Sans]           [GillSans]
 \definefontsynonym [SansItalic]     [GillSans-Italic]
 \definefontsynonym [SansBold]       [GillSans-Bold]
 \definefontsynonym [SansBoldItalic] [GillSans-BoldItalic]
\stoptypescript
```

The above is a class–to–symbolic typescript: it points from the generic 'Sans' names to symbolic names specific to the Gill Sans font family. The `\setups` line is a compact, pre-defined way of making sure all of the alternatives within a font family (Slanted, BoldItalic, SmallCaps) point to default fallbacks if they are not over-ridden with an explicit definition we provide. For example, even though there is no slanted alternative listed, an `\sl` command would result in the SansItalic alternative, which, in the case

above, would resolve to whatever the GillSans-Italic font points to.

```
\starttypescript [sans] [gillsans-light] [name]
 \setups            [font:fallback:sans]
 \definefontsynonym [Sans]           [GillSans-Light]
 \definefontsynonym [SansItalic]     [GillSans-LightItalic]
 \definefontsynonym [SansBold]       [GillSans]
 \definefontsynonym [SansBoldItalic] [GillSans-Italic]
\stoptypescript
```

The above is a similar typescript that is triggered with a slightly different command. Instead of the normal alternative pointing to Gill Sans at a regular weight, it points to the light weight. The bold alternatives are similarly lightened to the regular variant.

```
\starttypescript [sans] [gillsans,gillsans-light] [texnansi,ec]
 \definefontsynonym [GillSans-Light]       [\typescriptthree-GillSansLight]
                                            [encoding=\typescriptthree]
 \definefontsynonym [GillSans-LightItalic] [\typescriptthree-GillSansLightItalic
                                            [encoding=\typescriptthree]
 \definefontsynonym [GillSans]             [\typescriptthree-GillSans]
                                            [encoding=\typescriptthree]
 \definefontsynonym [GillSans-Italic]      [\typescriptthree-GillSansItalic]
                                            [encoding=\typescriptthree]
 \definefontsynonym [GillSans-Bold]        [\typescriptthree-GillSansBold]
                                            [encoding=\typescriptthree]
 \definefontsynonym [GillSans-BoldItalic]  [\typescriptthree-GillSansBoldItalic]
                                            [encoding=\typescriptthree]
\stoptypescript
```

The above maps the symbolic names to the low-level font file names. Because of the multiple values in the `\starttypescript` lines, it actually serves the role of four typescripts at once: two weight variations multiplied by two encoding vectors. If selected with the EC encoding, the first line would resolve to:

```
 \definefontsynonym [GillSans-Light] [ec-GillSansLight] [encoding=ec]
```

Since we followed a strict convention in naming the font files, we're sure that the

system will locate an `ec-GillSansLight.tfm` font file.

```
\starttypescript [map] [all] [all]
 \loadmapfile [gtamacfonts.map]
\stoptypescript
```

The use of the map typescript in the `type-gtamacfonts` package is very straightforward. If you load the typescript file, you end up triggering a catch-all condition, and the `gtamacfonts.map` file is automatically loaded.

```
\starttypescript [HoeflerOldStyle] [texnansi,ec]
 \definetypeface [HoeflerOldStyle] [rm] [serif] [hoefleroldstyle]
                                [default] [encoding=\typescripttwo]
 \definetypeface [HoeflerOldStyle] [ss] [sans]  [gillsans]
                                [default] [encoding=\typescripttwo,rscale=0.96]
 \definetypeface [HoeflerOldStyle] [mm] [math]  [palatino]
                                [default] [rscale=0.90]
 \definetypeface [HoeflerOldStyle] [tt] [mono]  [modern]
                                [default] [encoding=\typescripttwo]
\stoptypescript
```

The final typescript that uses the Gill Sans font is one of the example typescripts. Again, we would rather discourage their use, because the font combination was rather restricted (by those fonts we could be reasonably sure are present on a modern gwTEX/ConTEXt system) and idiosyncratic to our own tastes. The `\definetypeface` commands are not difficult to cut-and-paste and adapt to your own tastes.

You can see how Gill Sans is named as the sans member of the family, scaled down slightly to match Hoefler Text's x-height. Again, the encoding is abstracted from the typescript.

All of the font families are defined in similar ways to the above, but as the underlying fonts have different family members, there are more than a few exceptions and caveats in the mix:

- There are both `hoefler` and `hoefleroldstyle` typescripts defined. The regular one defines an oldstyle figures variant, and the oldstyle typescript defines a lining figures variant. Both define a small-caps variant. Variants are a way of switching font features beyond the usual regular/bold/italic alternatives.

An example of variant usage is to switch to SMALL CAPS *within a stretch of italic text with a grouped* `{\Var[sc] small caps}` *command.*

- For Didot, we have defined an additional oldstyle figures flavor, but no special `\Var[]` variants are made available because the oldstyle figures are only available from the regular face, not italic or bold.

Didot is of the same family as *Bodoni*, and shares some characteristics with *Computer Modern*. The old-style **numerals**, however, evince a very different design: 123 456 789.

- American Typewriter, although it has no italic, is available in Regular, Light, Condensed, and Light-Condensed varieties.

- Helvetica Neue is also presented as two families: regular and light. The light family uses the fashionable HelveticaNeue-UltraLight weight. It's not very suitable for text usage, but might make for interesting headlines.

American Typewriter, although a slab-serif, typewriter–style face, is classed more as a **serif** for ConTEXt use because it's not terribly well adapted for code printouts. Helvetica Neue is an artful re-drawing of a classic face that *nearly* outstayed its welcome. It has led to a resurgence and even *more* ubiquity, not only in **office** applications, but in **graphic design**.

- There are only four fonts within the Futura family shipped by default on the Macintosh: Medium, Medium-Italic, Condensed-Medium, and Condensed-ExtraBold. These don't make for a satisfying family, but we provided two (in despair): Futura and Futura-Condensed.

- Gill Sans, as already discussed, is available in light- and regular-weight families.

- Lucida Grande, because of its exceptional glyph coverage, has the `qx` (Central Europe) and `t5` (Vietnamese) encodings enabled within ConTEXt, simply because it was easy to do so.

> I may not know much **tiếng Việt**, but I know my **glyphs**!

7 Symbols as a bonus feature in ConTEXt

As a surprise feature on the day before release, one of the authors dug up some old code and re-used it in order to enable use of the Hoefler Text Ornaments font. There are some attractive fleurons and borders within the font, and the authors thought it would be nice to have some basic support for users to experiment with. Only a couple commands are necessary within the `symb-gtahoefler` file, which takes the following form:

```
\loadmapfile[gtamacfonts]

\def\HoefO#1{\getglyph{HoeflerTextOrnaments}{#1}}

\startsymbolset[Hoefler Ornaments]

   \definesymbol[Hourglass]              [\HoefO{4}]
   \definesymbol[LeftHand]               [\HoefO{6}]
   \definesymbol[RightHand]              [\HoefO{7}]
   % ...

\stopsymbolset
```

With these definitions in place, the end user accesses the symbols in the Hoefler Text Ornaments font by loading the file (`\usesymbols[gtahoefler]`), loading the symbol set (`\setupsymbols[Hoefler Ornaments]`), and then calling the symbolic names with

commands such as `\symbol[Hourglass]`, and `\symbol[RightHand]`.

We know the font will be present if this symbol file is on a user's system, so all we need to do is ensure the `.map` file is loaded, define a convenience command, and name the symbols based on the names given directly in the encoding file. By combining these symbols, fairly pleasant effects (such as the acorn in the bulleted lists) can be created.

8 Packaging and Distribution

The results are organised according to the TEX Directory Structure (TDS). In our case, that means that there is a directory structure that can be grafted on top of an existing `texmf` tree. The various font-related files for this distribution are stored in `gtamacfont/` subdirectories for easy management:

```
./fonts/enc/dvips/gtamacfonts/
./fonts/map/pdftex/gtamacfonts/
./fonts/tfm/gtamacfonts/didot/ (etc)
./fonts/vf/gtamacfonts/didot/ (etc)
./doc/fonts/gtamacfonts/
./tex/latex/gtamacfonts/
./tex/context/gtamacfonts/
```

The gtamacfonts distribution has been made a part of the TEX **i-Installer**[8] i-Package. After installation and during configuration, the i-Package checks if in the `texmf.gwtex` tree, e.g. the directory `./fonts/truetype/gtamacfonts/didot` does not exist; if it doesn't, it creates it and populates it with an unpacked Mac Didot font using the `fondu` tool:

```
Didot.ttf
DidotBold.ttf
DidotItalic.ttf
```

[8] http://ii2.sourceforge.net/

To get this automatic conversion during the TEX i-Package configuration phase, it is necessary that the `fondu` tool has already been installed either manually or via the Fondu i-Package.[9]

9 Testing and Documentation

After installation with i-Installer, you can go to the

`/usr/local/teTeX/share/texmf.gwtex/doc/fonts/gtamacfonts/`

directory where you'll find several example files for LATEX and ConTEXt showing you how to use the fonts and which glyphs are available. The manual, `gtamacfonts.pdf`, is also available there.

10 Limitations and ToDo's

We hope that many users will find our support for Macintosh system-fonts useful, but we're also aware of a number of limitations, and there are a few features that we'd like to implement, but couldn't implement now for lack of time and skills:

1. The most important point is clear, yet it is well worth repeating: we are talking about TrueType fonts, so they will *only* work with pdfTEX, *not* with vanilla TEX and `dvips`. They will also be incompatible with packages and features that rely on PostScript features (such as `pstricks` or the font expansion parts of `microtype`). [*Character protrusion, however, is possible, and is activated in this document, as it is one of ConTEXt's standard features.*]

2. All fonts have some shortcomings. EC and texnansi define many characters; none of the fonts has them all. Some of the fonts are missing several weights and/or variants. Very few have additional features such as oldstyle figures and/or small caps.

[9] If `fontforge` is installed, the i-Package will also produce and store the AFM files.

3. Some fonts offer several weights such as light, ultralight, regular. In LaTeX, these could be made variants to the regular font, and one could even think of implementing switching to these alternative weights via options for the packages. This has not yet been done.

4. Both ConTEXt and LaTeX have a peculiar and not quite satisfying way of implementing support for oldstyle figures. In ConTEXt, it is either intrinsic to the font or enabled with a font variant call (such as `\Var[os]`). The old, and somewhat deprecated method of using the `\os` command is generally hard-wired into the MathItalic font and is a relic of some peculiarities of Computer Modern's old encoding conventions.

   In LaTeX, the command `\oldstylenums{}` produces oldstyle figures from the math font. Since we haven't redefined the math font, this would be Computer Modern, which doesn't look very good with these fonts. If the document uses the `textcomp` package, the command `\oldstylenums{}` takes the oldstyle figures from the accompanying `TS1` font if they are defined there. This is messy and will confuse inexperienced users. As a workaround, we recommend using `\textsc{}` in order to produce oldstyle figures (1234567890) where they exist and they are not default.

5. There is much more that could be done with the symbol support, both enhancing it within ConTEXt – for example, to take good advantage of the ornate border elements, there could be some enhanced pattern-generating macros – and to give basic support for LaTeX – we simply lacked the expertise to make it into a LaTeX package. We welcome any contributions in that area.

11 What the Future Holds...

We're quite happy that we could provide support for these fonts. We're looking forward to future developments. Now that the basic support is in place, it is fairly easy to add more fonts and to refine the support for the existing fonts. We also hope that our implementation will help reduce the wide-spread prejudice against TrueType fonts in TeX. And we hope that other users will feel welcome to join our efforts and contribute improvements.

Made with ConTEXt — Thanks, Hans!

THE PracTEX Journal

The MathTimeProfessional Fonts
Or, How I Wasted the Last Twenty Years of my Life

Mike Spivak

Abstract

I am a computer innocent who, through a series of historical accidents, ended up writing the amstex macro package, and a font design innocent who, through desperation for fonts that I was willing to use for my Calculus and Differential Geometry books, ended up designing the MathTime Professional fonts. It is a sobering reflection that these activities seem to have occupied a significant amount of my time during the last 20-25 years.

I often find it interesting to answer challenging questions about using TeX for typesetting. However, a severe allergy to LaTeX dictates that my solutions be couched in plain TeX, though presumably any well-versed LaTeXophile will be able to supply a translation. You may contact me at the MathTime Pro forum, or at Publish or Perish, Inc.

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

The *MathTime Professional* Fonts

Or, How I Wasted the Last Twenty Years of my Life

by Michael Spivak

The PracTEX Journal, February 2006

People who have been wearily following my dalliance in the world of fonts may be wondering how I came to have an interest in type design. The answer to that is very simple: I've never had any interest in type design. If, 20-some years ago, when I was contemplating the next edition of my *Calculus* book, some one had offered me a reasonable set of math fonts (ones that could be used with Times, or with the Baskerville font in which *Calculus* is typeset) for the modest price of $1,000, say, then I would have jumped at the chance to obtain it.

But no one did; on one side were the freebie people, who were producing lots of unsuitable fonts with MetaFont, on the other side real type design establishments, which couldn't be expected to foolishly invest time and money in the creation of fonts of interest only to the freebie freaks. So I had to foolishly supply the time and money myself.

Of course, I knew absolutely nothing about fonts. Professional type designers would probably say that I still know nothing about fonts, but I certainly know more than I did then, and a lot more than I ever wanted to know. What's more, back in those ancient times, MetaFont was the only tool I had even heard of, and, confounded as I was by elementary matters like the difference between `10pt` and `10pt#`, I despaired of ever learning enough MetaFont to do anything with it. So I relied on a MetaFont expert to produce some fonts, and had the pleasure of creating humongous pixel files, mailing back desired changes drawn on large printouts, reiterating this process *ad infinitum*, etc.

By the time it was possible to envision the eventual completion of this project, the prices of PostScript printers had descended from the stratosphere, and it became clear that I really needed PostScript fonts. So another artisan was hired to turn out PostScript versions. Now large sample characters were trivial to produce, and desired changes both easier to indicate and to realize, so that the project proceeded apace. So did the costs. Realizing that ignorance is not always financial bliss, I borrowed a Mac and learned the rudiments of Fontographer (at that time available only on the Mac, though I naturally switched to the PC version as soon as possible, thereby avoiding the insane issue of the same font being different files on two different machines). Thus I could not only produce PostScript fonts, but I could "draw" characters in the way that seemed natural to me, as outlines, rather than using "pens".

And thus were born the original *MathTime* fonts.

By the time I was ready to produce a typeset version of my *Differential Geometry* books, however, I lusted for more. I had to admit to myself that Knuth was right in insisting on separate designs for smaller type sizes, as in the days of cold-metal type; using scaled down versions of the 10pt fonts for superscripts just didn't look all that good (good enough for an elementary text like *Calculus*, perhaps, but not for an author's special pride and joy). But I didn't know how to go about making designs for smaller sizes, and I didn't find Knuth's smaller sizes for Computer Modern to be particularly compelling examples to follow.

Luckily, among the numerous type catalogs that had been accumulating on my shelves, in a small font catalog issued by Monotype (long before it had merged with Agfa) I found not only the presumably "true" Times New Roman font, but also something called Times Seven, and something else called Times Small Text. Since I had read in Walter Tracy's book *Letters of Credit* that Times New Roman was created in 9, 7 and $5\frac{1}{2}$ point sizes, I immediately surmised that Times Seven was the 7 point design and Times Small Text the $5\frac{1}{2}$ size (the people at Monotype were only able to tell me that they used Times Small Text for printing in small type sizes). That means that their Times, used at 10 point, is actually a magnification of a font meant to be used at 9 points, but, I rationalized, that isn't so bad, even if one were using the fonts together with Times text, since it's fine for the math fonts to be a bit more prominent than the text fonts (as they are, by design, in Knuth's Computer Modern).

I bought these fonts with great expectations, and sure enough, Times Seven looked ungainly when used at 10 points, but provided quite a nice 7 point font, and Times Small Text looked quite good at $5\frac{1}{2}$ points. I had struck gold. Opening the fonts Times Seven and Times Small Text in Fontographer, and comparing with Times, I was not only able to get a good idea of the sort of modifications required to produce smaller point sizes, but I was

even able to specify a sequence of various Fontographer-provided modifications to the characters in Times that produced a very creditable approximation to the shape of the characters in the other fonts. So now I knew how to get smaller sizes for new characters that I had produced. (There were still a lot of adjustments that had to be done by hand, making me wish that there were a programmable Fontographer, a sort of MetaFontographer.)

$$\left\{\begin{array}{rcl} \textit{MathTimeProfessional}\text{ equation: } \alpha\beta\gamma^{\alpha\beta\gamma^{\alpha\beta\gamma}} & & \\ \textit{MathTimeProfessional}\text{ letters} & \alpha\beta\gamma & \text{at text size } \alpha\beta\gamma \\ \text{for superscripts} & {\scriptstyle\alpha\beta\gamma} & \text{at text size } \alpha\beta\gamma \\ \text{for second-order superscripts} & {\scriptstyle\alpha\beta\gamma} & \text{at text size } \alpha\beta\gamma \end{array}\right\}$$

Along the way, I took advantage of the fact that I was working for myself, and not an overweening company or organization, to enhance the fonts with all sorts of other improvements of a more radical nature. I had always hated "extensible" parentheses, so I decided to have individually designed parentheses of large sizes. This turned out to be a somewhat complicated matter, especially when I later decided to produce other individually designed large delimiters, as well as individually designed square root signs:

$$\underbrace{\sqrt{\begin{pmatrix} \mathbb{A}_{11} & \dots & \mathbb{A}_{1n} \\ \mathcal{A}_{21} & \dots & \mathcal{A}_{2n} \\ \vdots & \ddots & \vdots \\ \mathfrak{A}_{n1} & \dots & \mathfrak{A}_{nn} \end{pmatrix}} \quad \text{instead of} \quad \sqrt{\begin{pmatrix} \mathbb{A}_{11} & \dots & \mathbb{A}_{1n} \\ \mathcal{A}_{21} & \dots & \mathcal{A}_{2n} \\ \vdots & \ddots & \vdots \\ \mathfrak{A}_{n1} & \dots & \mathfrak{A}_{nn} \end{pmatrix}}}$$

In the first place, even allowing 256 characters on the extension font, one would run out of room for all the additional characters needed. Even worse, the large ones simply couldn't be put on the font, because of PostScript restrictions on the size of characters in a font. Reducing the size of all the characters, and then using the font at a magnified size wouldn't work either, because the smaller size characters would end up so small that there wouldn't be enough space for all the points needed to specify them accurately. I had to place larger size delimiters on different fonts, a total of 3 different ones, which are then used at magnifications of 2, 4, and 8 times. This also required a different syntax than the admirable `\left...\right` [what one really wants is not *one* extension font, but an allowed *series* of extension fonts, where the `NEXTLARGEST` element on one font can be in another font].

All these additions were originally hacked together rather quickly for my own use, and then more carefully constructed when the *MathTimeProfessional* fonts were offered for sale. Examples of all these special features are not given here, but they may be found on the website `www.pctex.com`, where the fonts are sold. (Other interesting hacks, not made part of *MathTimeProfessional*, are mentioned in the colophon to the first volume of the *Differential Geometry* books.)

Of course, of the making of special features there is no end, and a few more (like the individually designed vertical and horizontal curly braces illustrated above) have been added to *MathTimeProfessional II*, the latest version, often in response to suggestions made on the PCTEX font forum. But I've provided so many fonts, and so many special features, that I have reason to hope that this dissolute period of my life may finally have come to completion.

THE PracTEX Journal

An exploration of the Latin Modern fonts

Will Robertson

Abstract

The Latin Modern fonts are a newly-created set of fonts with the principle aim of providing glyphs for as many languages as possible. There are a multitude of little-known font shapes in the package, however, and these will be explored in this paper.

Will Robertson is a PhD student in Mechanical/Mechatronic Engineering in the University of Adelaide, South Australia. His LaTeX experience came about from being distracted with it while writing various reports and papers on his "real" research, and now he's hooked. Will may be contacted at will@guerilla.net.au

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

The PracTEX Journal
TPJ 2006 No 01, 2006-02-01
Article revision 2005/02/20

An exploration of the Latin Modern fonts

Will Robertson

Email wspr81@gmail.com
Address School of Mechanical Engineering
University of Adelaide, SA
Australia, 5005

Abstract The Latin Modern fonts are a newly-created set of fonts with the principal aim of providing glyphs for as many languages as possible. There is a multitude of little-known font shapes in the package, however, and these will be explored here.

1 Introduction

The Latin Modern family[1] is a recent collection of fonts authored by Bogusław Jackowski and Janusz M. Nowacki [1]. They are intended as the successors to Donald Knuth's Computer Modern fonts for the Unicode age, to provide the means for typesetting as many languages as possible that use the Latin-based alphabet. The collection is vast: it contains sixty-nine fonts, each containing almost seven hundred glyphs, at time of writing, with more probable in the future. That's almost 69000 glyphs in total! A very small number of the glyphs are shown in figure 1, chosen mostly at random for their interesting shapes.

The Latin Modern fonts have been created with the MetaType1 system [2, 3], whose programmatic nature makes the idea of dealing with such a huge number of glyphs more palatable. The number of fonts in the collection is greater than the BlueSky Computer Modern Type 1 fonts [4] now used by default by all current LaTeX distributions, but fewer than in the enormous CM-Super collection (which also provides many glyphs for multilingual typesetting), whose fonts have been auto-traced from bitmaps and hence are of slightly inferior quality [5]. With the most recent releases, OpenType versions of the fonts have been made available for more general use. In this article, we shall look at the fonts the Latin Modern family provides and how they may be accessed in LaTeX.

1. This document describes version 0.99.3 of the Latin Modern fonts, which I should note are able to be both freely used and distributed in their provided form.

Copyright © Will Robertson, 2006

Υ ð Ê̊ Þ Ę Ŋ å̊ ǎ Ą̣ Ǽ

Figure 1: Ten of the 69000-odd glyphs in the Latin Modern collection.

2 NFSS refresher

To provide context, some brief details of LaTeX's font selection scheme are expounded here. Refer to the documentation [6] for further information. Three main families are defined for a document: the default roman, sans serif, and typewriter fonts. These are selected with the `\rmfamily`, `\sffamily`, and `\ttfamily` commands, respectively. Arbitrary font families are requested with the command `\fontfamily{...}`; all such `\font...` commands (more to be seen) must be appended by `\selectfont`, if nothing else, to perform the actual font selection.

Variations along two other font axes (other than family) are possible: series and shape. The former is used to express weight and width, such as bold or condensed, and combinations thereof. We will be using the `\fontseries{...}` command later to look at various weights of the Latin Modern fonts. The shape axis is used to express italics and small caps, among other more esoteric options. We shall be content in the shape axis to use the commands `\itshape`, `\slshape`, and `\scshape` to choose between the italic, oblique, and small caps shapes.

How do we discover all the codes used to express the families, series and shapes for each font? These are all defined within font definition (`.fd`) files, which are supplied one per font encoding. The most common encoding is `T1`, which provides glyphs for many, but not all, European languages. To discover the font shapes available in the Latin Modern collection, then, these files must be located within the TeX distribution. They are found in the `texmf/tex/latex/lm` directory (where this is located will be system dependent), and investigation here will yield all of Latin Modern's secrets.

The encodings currently supported by the Latin Modern fonts in LaTeX are: `T1`, for most European languages; `QX`, a variant of `T1` that is more suitable for Slavonic languages (including the fk ligature, cf. fk); `LY1`, which supports a mixture of common symbols and accented letters; `T5`, for Vietnamese; and `TS1`, a large collection of miscellaneous symbols to accompany `T1`.

3 The same-old

Everyone is familiar with the default TEX fonts. The Latin Modern fonts are selected with, in the preamble,[2]

```
\usepackage{lmodern}
\usepackage[T1]{fontenc}
```

which should make barely any visible changes to already existing documents; these fonts are an *extension* of Computer Modern, not a new design.

To begin, the three default families are shown, using common LATEX font selecting commands. In the examples shown in this article, indented entries indicate that the previous outdented command(s) are still active.

Roman Perhaps simply because he could, Knuth included a large amount of variation in the fonts he designed for TEX. Certainly, no one since has really matched his efforts. The descendants of his fonts still bear this curious hallmark: the Latin Modern Roman family contains both slanted *and* italic shapes.

`\rmdefault`	Latin Modern Roman
`\itshape`	*Latin Modern Roman Italic*
`\slshape`	*Latin Modern Roman Oblique*
`\scshape`	Latin Modern Roman Small Caps
`\bfseries`	**Latin Modern Roman Bold Extended**
`\itshape`	***Latin Modern Roman Bold Italic Extended***
`\slshape`	***Latin Modern Roman Bold Oblique Extended***

Sans serif Variations here must wait until later; here are the 'standard four'. Note that the sans serif family does not have a true italic, nor small caps.

`\sffamily`	Latin Modern Sans
`\slshape`	*Latin Modern Sans Oblique*
`\bfseries`	**Latin Modern Sans Bold**
`\slshape`	***Latin Modern Sans Bold Oblique***

2. Change `T1` to another option (`LY1`, `QX`, `T5`), or combination thereof, depending on which glyphs you require/which language(s) you are typesetting.

Typewriter The italic shape here is perhaps a little unpleasant, and the fact that it has small caps is quite unusual considering that the sans serif family does not.

```
\ttfamily      Latin Modern Typewriter
   \itshape    Latin Modern Typewriter Italic
   \slshape    Latin Modern Typewriter Oblique
   \scshape    LATIN MODERN TYPEWRITER SMALL CAPS
   \bfseries   Latin Modern Typewriter Dark
      \slshape Latin Modern Typewriter Dark Oblique
```

The majority of the shapes demonstrated above are available in the vector Computer Modern fonts (that is, the current LaTeX defaults). See Appendix A for a complete comparison of the fonts available in the Computer Modern and Latin Modern collections (most of which we have yet to see). The bold typewriter fonts above, however, are completely new to Latin Modern. While the original METAFONT fonts were completely parameterised such that changes like this were easily possible, its bitmap output format is very outdated and rarely used these days.

4 Interlude — optical sizes

In the old days of printing, fonts were made of metal and were literally one to a size. The characters in a font for the body text of a book would look noticeably different to that same font at a larger size for titling. Nowadays, computer-based fonts can be scaled linearly to any size imaginable, but well designed fonts are still made available with variations based on the intended size of the output. In brief, the smaller a font is, the less fine its intricacies must be in order to survive the transfer from (possibly imperfect) printed page or low-resolution screen to eye. Conversely, a font designed to be large can be more delicately rendered.

For the original Computer Modern fonts, designed in METAFONT, the optical size could be chosen exactly for any size. Due to disk space constraints, specific sizes were chosen as canonical, which were then inherited when they were converted to the PostScript Type 1 format. The Latin Modern fonts, in turn, also preserve these canonical sizes for all of the 'major' shapes, although such a profusion of optical sizes is almost certainly unnecessary, since there needn't be such a great range of font sizes in a single document.

Latin Modern Roman, design size 5 pt
Latin Modern Roman, design size 6 pt
Latin Modern Roman, design size 7 pt
Latin Modern Roman, design size 8 pt
Latin Modern Roman, design size 9 pt
Latin Modern Roman, design size 10 pt
Latin Modern Roman, design size 12 pt
Latin Modern Roman, design size 17 pt

Figure 2: The optical size range of Latin Modern Roman, each font at 12 pt.

The set of optical sizes for Latin Modern Roman is shown in figure 2, the largest number for any of the Latin Modern families. The non-linear nature of the scaling is immediately apparent, and it is quite clear how the characteristics change from robust to delicate, most significantly in the widths and stroke thicknesses of the characters, as the design size increases.

The Latin Modern fonts with a range of optical sizes are: roman upright, italic, oblique, and bold extended; sans upright and oblique; and typewriter upright. These optical size variations constitute 32 of 69 fonts in the collection.

5 Non-default weights

As previously mentioned, the Latin Modern collection shares with the Computer Modern fonts some shapes that are not often used in practice, probably due to the fact that they can't be accessed with the 'normal' NFSS commands such as `\emph` and `\textbf`.

5.1 Other bold shapes

A non-extended version of the roman bold exists. Unfortunately, it is available in but a single design size (unlike its extended counterpart), and lacks true italics.

`\bfseries`	**Latin Modern Roman Bold Extended**
`\fontseries{b}\selectfont`	**Latin Modern Roman Bold**
`\fontseries{b}\slshape`	***Latin Modern Roman Bold Oblique***

5

The sans serif family has a similar 'secret' bold shape:

`\sffamily`	
`\bfseries`	**Latin Modern Sans Bold**
`\fontseries{sbc}\selectfont`	Latin Modern Sans Demi Condensed
`\fontseries{sbc}\slshape`	***Latin Modern Sans Demi Condensed Oblique***

5.2 Italic small caps

The slantsc package allows NFSS declarations `\slshape` and `\scshape` to be combined in order to select oblique small caps. (Or `\itshape` for truly italic small caps if they exist.) With `\usepackage{slantsc}`, it is possible to select

`\scshape\slshape`	*Latin Modern Roman Oblique Small Caps*
`\ttfamily\scshape\slshape`	*Latin Modern Typewriter Oblique Small Caps*

Oblique or italic small caps are scarce in traditional typesetting, but their use is becoming more popular in modern times.

5.3 The new typewriter shapes

Quite recently in the lifetime of the Latin Modern collection, the typewriter fonts have been supplemented with extra shapes, including the 'Typewriter Dark' fonts previously seen. Also present are light and condensed light shapes, the latter being a $\frac{2}{3}$ reduction in width; that is, 120 characters in condensed light will fit in the space for 80 regular typewriter letters. Note that every character in every weight and shape of the typewriter fonts has the same width so that the letter grid remains constant when switching between styles.

`\ttfamily`	
`\fontseries{dk}\selectfont`	**Latin Modern Typewriter Dark**
`\fontseries{dk}\slshape`	***Latin Modern Typewriter Dark Oblique***
`\fontseries{lt}\selectfont`	Latin Modern Typewriter Light
`\fontseries{lt}\slshape`	*Latin Modern Typewriter Light Oblique*
`\fontseries{lc}\selectfont`	Latin Modern Typewriter Light Condensed
`\fontseries{lc}\slshape`	*Latin Modern Typewriter Light Condensed Oblique*

One may wonder why the light weights were produced. As the medium typewriter face is relatively heavy, it does not have much contrast with the new dark weight; compare the example on page 3 with the one on the previous page. So, in situations in which the bold face is to be used, the light face should be selected as the 'normal' typewriter weight. This can be performed with the following declaration in the preamble:

```
\DeclareFontFamily{T1}{lmtt}{}
\DeclareFontShape{T1}{lmtt}{m}{n}{<-> ec-lmtl10}{}
\DeclareFontShape{T1}{lmtt}{m}{\itdefault}{<-> ec-lmtlo10}{}
\DeclareFontShape{T1}{lmtt}{\bfdefault}{n}{<-> ec-lmtk10}{}
\DeclareFontShape{T1}{lmtt}{\bfdefault}{\itdefault}{<-> ec-lmtko10}{}
```

(For the `T1` encoding; adapt as required for the other encodings by looking in the `...lmtt.fd` files, as discussed in section 2.) The use of `\bfdefault` and `\itdefault` permit font selection with the commands `\textbf` and `\emph`.

6 Other families

As well as the secret weights mentioned above, there are entire *families* in the Latin Modern collection of which many people may be unaware.

6.1 Sans extended

The family 'Latin Modern Sans Extended' (sometimes referred to as 'Sans Quotation' due to Knuth's original use for it) is an extended version of the default sans serif family, intended for use at small font sizes (its nominal design size is 8 pt).

```
\renewcommand\sfdefault{lmssq}
```

`\sffamily`	Latin Modern Sans Extended
`\slshape`	*Latin Modern Sans Extended*
`\bfseries`	**Latin Modern Sans Extended**
`\slshape`	***Latin Modern Sans Extended***

The variation in sans bold is interesting with regard to the condensed sans shown in section 5.1, but the shapes aren't entirely suitable for combination since they have different x-heights arising from their different design sizes:

Condensed Bold Extended

6.2 Typewriter proportional

As the era of teletext computers draws ever more distant, perhaps the idea of a fixed width font can be thought to be archaic. The Latin Modern Typewriter family has an accompanying variable width design, for those who wish to use it:

`\renewcommand\ttdefault{lmvtt}`	
`\ttfamily`	Latin Modern Typewriter Proportional
`\slshape`	*Latin Modern Typewriter Proportional Oblique*
`\fontseries{lt}\selectfont`	Latin Modern Typewriter Proportional Light
`\fontseries{lt}\slshape`	*Latin Modern Typewriter Proportional Light Oblique*
`\fontseries{dk}\selectfont`	**Latin Modern Typewriter Proportional Dark**
`\fontseries{dk}\slshape`	***Latin Modern Typewriter Proportional Dark Oblique***

It can be seen that here, as in the fixed-width typewriter fonts, every alphabet has the same horizontal width. Again, if the bold face is to be used for contrast, better results will be achieved by selecting the light face as default. This can be effected in a similar manner as before: (section 5.3, refer in this case to `t1lmvtt.fd`)

```
\DeclareFontFamily{T1}{lmvtt}{}
\DeclareFontShape{T1}{lmvtt}{m}{n}{<-> ec-lmvtl10}{}
\DeclareFontShape{T1}{lmvtt}{m}{\itdefault}{<-> ec-lmvtlo10}{}
\DeclareFontShape{T1}{lmvtt}{\bfdefault}{n}{<-> ec-lmvtk10}{}
\DeclareFontShape{T1}{lmvtt}{\bfdefault}{\itdefault}{<-> ec-lmvtko10}{}
```

7 Future additions

The Latin Modern fonts are approaching their first final release, but there are still some fonts to incorporate from Computer Modern. In order to provide a complete replacement, the mathematical glyphs need to be adapted, but this will mostly be a job of 'taking' them — no new glyphs, at this stage, need to be created.

Two more font shapes will be included: (also look out for a slanted Dunhill)

`\fontfamily{cmr}\fontshape{ui}\selectfont`	Computer Modern Unslanted *italic*
`\fontfamily{cmdh}\selectfont`	Computer Modern Dunhill

8

These fonts exist primarily to demonstrate the 'meta-ness' of the Computer Modern fonts, in that obliqueness of the italics and the stem height of the roman, to name but two parameters in the design, may be varied orthogonally. Their use is not particularly widespread.

The Latin Modern team will no doubt continue to supplement their fonts with additional glyphs and variations. I hope they create more ligatures ('fb', 'fj', 'fh', ...) and language-specific features, which will become more useable in the TEX world when OpenType starts becoming more the *de facto* font standard (see XƎTEX for a Unicode-capable TEX variant that supports OpenType fonts and their advanced features [7]).

As an example of a language-specific feature, French people (or at least, French *typesetters*) like more space before their punctuation than English-speakers. It is possible to provide for this within the OpenType fonts themselves, to be activated at the discretion of the user, without having to deal with messy 'Babel-like' active character techniques that perform this service currently.

8 Conclusions

This concludes our tour of the different shapes of the Latin Modern font collection, which are the more multilingual replacements of the vector Computer Modern fonts. They have been exhibited in the belief that they are not as well known as they deserve, for much time and effort has been spent to supplement each of the fonts with hundreds of extra glyphs.

These fonts are already the default in ConTEXt, and in time are planned to become the default fonts in LATEX, although exactly when remains to be seen. We have seen some shortfalls and awkwardness with LATEX's font selection scheme in being able to select, in a straightforward manner, the large variety of shapes and weights that the collection offers. Brief examples detailing how to overcome these problems have been given, but more work is required. In the future, we look forward to the creation of a better user interface for this purpose, either specifically for these fonts, or in general with a 'newer' font selection scheme.

References

[1] Bogusław Jackowski and Janusz M. Nowacki. Latin Modern: Enhancing Computer Modern with accents, accents, accents. *TUGboat*, 24(1):64–74, 2003. URL http://www.tug.org/TUGboat/Articles/tb24-1/jackowski.pdf.

[2] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. MetaType1: a METAPOST-based engine for generating Type 1 fonts. *MAPS*, 26:111–119, 2001. URL http://www.ntg.nl/maps/pdf/26_15.pdf.

[3] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. Programming PS Type 1 fonts using MetaType1: Auditing, enhancing, creating. *TUGboat*, 24(3):575–581, 2003. URL http://www.tug.org/TUGboat/Articles/tb24-3/jackowski.pdf.

[4] Computer Modern and AMSFonts in Type 1 (PostScript) form. Website. URL http://www.ams.org/tex/type1-fonts.html.

[5] Vladimir Volovich. CM-Super: Automatic creation of efficient Type 1 fonts from METAFONT fonts. *TUGBoat*, 24(1):75–78, 2003. URL http://www.tug.org/TUGboat/Articles/tb24-1/volovich.pdf.

[6] LaTeX3 Project Team. *LaTeX2ε font selection*. URL http://www.latex-project.org/guides/fntguide.pdf.

[7] Jonathan Kew. XeTeX, the Multilingual Lion: TeX meets Unicode and smart font technologies. *TUGboat*, 26(2):115–124, 2005. URL http://www.tug.org/TUGboat/Articles/tb26-2/kew.pdf. (Not available to non-TUG members at time of publication).

A Summary of Computer vs. Latin Modern

The following table summarises the different font shapes and weights available in the Computer Modern (C.M.) and Latin Modern (L.M.) collections. A black bullet (•) indicates the presence of that font(s), and the grey bullet (•) indicates future plans for it.

It can be seen clearly from this table that the extra fonts in the Latin Modern collection are slanted variants that can be created mechanically: no *new* designs have been created for this project. This indicates where the effort of the project has gone: thousands of extra glyphs and hundreds of thousands of kerning pairs.

FAMILY	SERIES	SHAPE	C.M.	L.M.
Roman	Medium	Upright, Slanted, Italic, Small Caps	•	•
		Slanted Small Caps		•
		Upright italic	•	•
	Bold	Upright	•	•
		Slanted		•
	Bold Extended	Upright, Slanted, Italic	•	•
Fibonacci	Medium	Upright	•	
Dunhill	Medium	Upright	•	•
		Slanted		•
Sans Serif	Medium	Upright, Slanted	•	•
	Semibold Condensed	Upright	•	•
		Slanted		•
	Bold Extended	Upright	•	•
		Slanted		•
Sans Quotation	Medium	Upright, Slanted	•	•
	Bold Extended	Upright	•	•
		Slanted		•
Typewriter	Medium	Upright, Italic, Slanted, Small Caps	•	•
		Slanted Small Caps		•
	Light, Dark	Upright, Slanted		•
	Light Condensed	Upright, Slanted		•
Typewriter Proportional	Medium	Upright, Slanted	•	•
	Light, Dark	Upright, Slanted		•

THE PracTEX Journal

Chinese TeX: Using the CJK LaTeX Package, Unicode TrueType Fonts and PDFTeX under Windows

Helmer Aslaksen

Abstract

The goal of this article is to help users with no past experience with CJK (Chinese, Japanese, Korean) typesetting to include some pieces of CJK text in a TeX document using the CJK LaTeX package, TrueType fonts and pdfTeX under Windows.

Helmer Aslaksen is Associate Professor of Mathematics at the National University of Singapore. He has used TeX since 1988, but is still learning more and more about its power. You can reach Helmer at aslaksen@math.nus.edu.sg

Note: The following is a snapshot of the author's web page on this subject as of the publication date of this issue of The PracTeX Journal. Look at his web page for the latest version.

- Comment on this paper
- Send submission idea to editor

I studied English, French and German in school in Norway, and Chinese for two years at UC Berkeley. (Unfortunately, my pronunciation is horrible in all languages!) Because of my web page on the Chinese calendar, I need to put Chinese characters and pinyin (phonetic representation of Chinese using the roman alphabet) on the web and in my TeX files.

The goal of this article is very modest. If you are an expert on CJK (Chinese, Japanese, Korean) typesetting, you will probably not learn anything new. But if you need to include some pieces of CJK text in your TeX document, then I hope that this article will prove useful.

There are two issues involved in using CJK in TeX. First you need to input the CJK text in your text editor, and then you need to compile it under TeX.

The CJK LaTeX package by Werner Lemberg is a wonderful tool, but it can be a bit hard to install, especially if you're not using Linux. Fortunately, it is now fairly easy to install under Windows using either TeXLive or MiKTeX. Using Unicode TrueType fonts and PDFTeX involves some additional steps, so I wanted to share my solution with other people.

In this article you will learn about

Inputting Chinese Characters

There are many ways to do this, but if you just need to write a few words, a simple solution is to use MS Word and the MS IME. Here are some links about the MS IME.

- Microsoft Global Input Method Editors (IMEs)
- What is an IME (Input Method Editor) and how do I use it?
- IME Tutorial

Save as plain text and choose the appropriate encoding, either UTF-8, GB2312 or Big5. Make sure that no characters appear in red! Then open the text files in your TeX editor and copy over to your TeX document. I use WinEdt, and I cannot see the Chinese characters there. If I open the text files in Notepad, the Chinese appear in the UTF-8 case, but not in the other. However, if I try to copy the UTF from Notepad to WinEdt, it does not come out right, I must open the text file in WinEdt.

Here are three sample files, `test-UTF8.tex`, `test-GB2312.tex` and `test-Big5.tex`.

If you're looking for a good Windows Unicode editor, you may want to check out EmEditor. You can use the MS IME with it! They give out academic licenses for free! I have never managed to

get Emacs to work under Windows, but if you manage, I would love to hear from you!

Some Unicode editors may add the "BOM" (Byte Order Mark) at the very beginning of the file, even if the output file encoding is set to UTF-8. The BOM under UTF-8 is the byte sequence `0xEF 0xBB 0xBF`. If the output file starts with those three bytes, they should be removed, or you may get strange warnings, log entries, or even errors while processing with LaTeX.

Installing CJK under Windows

This is no problem with either MiKTeX or TeXLive. Just select the packages the usual way.

Compiling the Example Files

Under MiKTeX, the GB and Big5 example files work fine, but the UTF8 example does not work. There are several ways of resolving this. One method is to convert TrueType fonts to PostScript Type 1 format by creating tfm, pfb and map files. The MiKTeX version of the CJK package has done this for the gbsnlp and bsmilp fonts used in the GB and Big5 examples, but not for the cyberbit font used in the UTF8 example. However, PDFTeX can use TrueType fonts directly, by creating tfm, enc and map files. I will describe how to use this method for the cyberbit font.

- Download `Cyberbit.ZIP` from ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/Cyberbit.ZIP". Rename `Cyberbit.ttf` to `cyberbit.ttf` and put it in `localtexmf\fonts\truetype\bitstream`.
- Run

  ```
  ttf2tfm cyberbit.ttf -w cyberb@Unicode.sfd@ > cyberbit.log
  ```

  This creates 165 tfm and 165 enc files. The -w option is important! That is how you get the enc files.
- Put all the tfm files in `localtexmf\fonts\tfm\bitstream\cyberb`.
- Put all the enc files in `localtexmf\pdftex\cyberb`.
- Check if `texmf\ttf2tfm\base\ttfonts.map` contains the line

  ```
  cyberb@Unicode@ cyberbit.ttf
  ```

  If not, create the file `localtexmf\ttf2tfm\base\ttfonts.map` with that line in it.
- Download delloye.free.fr/cyberbit.map, rename it `cyberb.map`, replace `cyberbit` with `cyberb`, except for `cyberbit.ttf`, throughout the file, and place it in `localtexmf\pdftex\cyberb`.
- Create the file `localtexmf\web2c\updmap.cfg` with the line

  ```
  Map cyberb.map #localtexmf\pdftex\cyberb\cyberb.map
  ```

  in it.
- Run `mkfntmap` (or `updmap` or `initexmf --mkmaps`).
- Refresh the file name database.

The cyberb name is a remnant of old file name restrictions. If you want to write `cyberbit` instead of `cyberb`, you'll have to do the following.

- Run

  ```
  ttf2tfm cyberbit.ttf -w cyberbit@Unicode.sfd@ > cyberbit.log
  ```

- Put `cyberbit@Unicode@ cyberbit.ttf` in `localtexmf\ttf2tfm\base\ttfonts.map`.
- Use delloye.free.fr/cyberbit.map, as it is.
- Put the enc and map files in `localtexmf\pdftex\cyberbit`.
- Add the line

  ```
  Map cyberbit.map #localtexmf\pdftex\cyberbit\cyberbit.map
  ```

  to localtexmf\web2c\updmap.cfg.
- Download delloye.free.fr/c70cyberbit.fd into `localtexmf\tex\latex\cyberbit`, or copy `texmf\tex\latex\CJK\UTF8\c70song.fd` into that directory, rename it `c70cyberbit.fd` and replace cyberb with cyberbit.

This should work on most modern TeX systems. If your system insists on making pk files, you need to fix your `updmap`.

With TeXLive, it is enough to create the tfm files (no need to use the `-w` option) and to edit `ttfonts.map`. However, this will only generate pk fonts.

If you want to run the `CJKbabel.tex` file, you can download the `t5.sty` file from CTAN, put it somewhere in localtexmf and refresh the file name database. Another alternative is to use the vntex package.

Some useful links.

- Using Truetype fonts and Unicode in Pdflatex by Otfried Cheong.
- How to make LaTeX (teTeX) handle unicode and CJK in MacOSX by Pai H. Chou.

Thanks to Werner Lemberg (the author of the CJK package), Danai Sae-Han, and Harald Hanche-Olsen for patient e-mails, to Olivier Delloye for his useful posting, and to Pai H. Chou for his helpful web page!

Combining Simplified and Traditional Characters

A simple solution is of course to use Unicode, but it can also easily be achieved by using the `\CJKencoding` command. You may want to look at sample-utf8.tex and sample-gb-big5.tex.

Character Encoding Conversion

Sometimes your Chinese text doesn't come out right, because it uses the wrong character encoding. This used to be a big problem, but now you can do character encoding conversion with Chinese Encoding Converter at Erik E. Peterson's On-line Chinese Tools.

Finding Unicode Codes

I often need to know the Unicode code for Chinese characters, either for TeX or HTML. You can input the characters in MS Words and copy them into Chinese Character Dictionary - Unicode Version at Erik E. Peterson's On-line Chinese Tools. You have to select the box for showing Unicode Value in the results and select UTF-8, and not Unicode, for the input. The other version, Chinese Character Dictionary, will not work, since it does not have the UTF-8 option. To convert to octal, you can use Conversion Table - Decimal, Hexadecimal, Octal, Binary.

You can also use Convert characters to Unicode at pinyin.info.

Creating CJK Bookmarks with Hyperref and Beamer

To get anything unusual, like for instance pinyin tone marks, into the bookmarks when using hyperref, you have to use the `\texorpdfstring{}{}` command, for example `\section {\texorpdfstring{\Jie2 \Qi4}{Ji\'e Q\`i}}`. This example was easy, because the marks for tones 2 and 4 can be obtained by standard accented characters. If you want to get the first tone, you have to first get the Unicode code from either Reading and Writing Chinese Characters and Pinyin on the Web Using Unicode, Code Charts (PDF Version) at Unicode Home Page or from the file puenc.def in the hyperref package. You must then use the unicode option for the hyperref or beamer package, and write for example `\section{\texorpdfstring {\Tang1 \Ruo4 \Wang4}{T{\001\001}ng Ru\`o W\`ang}}`. If you want to get the Chinese characters, you can use Chinese Character Dictionary - Unicode Version, and write for example `\section{\texorpdfstring{···}{\156\157 \202\345 \147\033}}`.

For more info on pinyin tone marks you may look at my page on Reading and Writing Chinese Characters and Pinyin on the Web Using Unicode.

Helmer Aslaksen
Department of Mathematics
National University of Singapore

Page generated February 28, 2006 ;TUG home page; search; contact webmaster.

THE PracTEX Journal

A Survey of Free Math Fonts for TeX and LaTeX

Stephen Hartke

Abstract

We survey free math fonts for TeX and LaTeX, with examples, instructions for using LaTeX packages for changing fonts, and links to sources for the fonts and packages.

Stephen G. Hartke is a VIGRE Research Assistant Professor of Mathematics at the University of Illinois at Urbana-Champaign. He can be reached at hartke@gmail.com

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

The PracTEX Journal
TPJ 2006 No 01, 2006-02-15
Article revision 2006-02-03

A Survey of Free Math Fonts for TEX and LATEX

Stephen G. Hartke

Email lastname @ gmail .dot. com
Address Department of Mathematics
University of Illinois at Urbana-Champaign
Urbana, IL, 61801 USA

Abstract We survey free math fonts for TEX and LATEX, with examples, instructions for using LATEX packages for changing fonts, and links to sources for the fonts and packages.

Contents

Copyright © 2006 Stephen G. Hartke
Permission is granted to distribute verbatim or modified copies of this document provided this notice remains intact.

1 Introduction

One of the biggest challenges in selecting a font for TEX or LATEX is that there are not very many math fonts that match the plethora of available text fonts. It's reasonably easy to use an arbitrary Postscript Type 1 font in TEX for text (see Philipp Lehman's Font Installation Guide [1]), but obtaining and configuring a matching math font from scratch is a demanding task. Thus, there are few math fonts for TEX, and in particular very few free ones. However, in the past few years, several very nice free fonts have been released. The goal of this article is to list all of the free math fonts and to provide examples.

"Free" here means fonts that are free to use (both commercially and non-commercially) and free to distribute, but not necessarily free to modify. I also am biased towards listing fonts that have outline versions in PostScript Type 1 format suitable for embedding in Postscript PS or Adobe Acrobat PDF files. Donald E. Knuth originally designed the METAFONT system for producing fonts for TEX in bitmap format. PS or PDF files that have embedded bitmap fonts do not display well in Adobe Acrobat Reader,[1] to the point of being almost unreadable on the screen, and are also noticeable when printing at extremely high resolutions (on photo-setters, for instance). Since outline fonts contain mathematical descriptions of the curves used in each glyph, they can be scaled to any resolution while retaining image quality.

The fonts listed here are categorized according to their origin: whether originally designed for TEX, related to the standard Postscript fonts, or other free fonts. A font's origin does not particularly bear on its quality or suitability for typesetting mathematics. No recommendations or evaluations of the fonts are given here, as people's tastes in fonts vary greatly. The goal of this survey is simply to make authors aware of all their options.

Most of the fonts can be selected by including a single package in the preamble of the user's LATEX file (the *preamble* is the section after "`\documentclass{}`" and before "`\begin{document}`"). The line or lines to include for each font are listed in the caption of the sample figure. For example "`\usepackage{fourier}`" uses Utopia and Fourier-GUTenberg, as shown in the sample LATEX file in Section 6.

Walter A. Schmidt also has a survey in German of math fonts [3] that con-

1. Starting with version 6, Adobe Acrobat Reader displays bitmap fonts fine. The free PDF viewers Ghostview and xpdf have always displayed bitmap fonts accurately.

centrates more on commercial fonts. Schmidt's survey has several examples that show different pairings between text fonts and math fonts.

2 Fonts Originally Designed for TEX

These fonts were originally designed for use with TEX, using either METAFONT or MetaType1 [2].

Computer Modern: Knuth created Computer Modern [5] as the default font for TEX. The font set includes serif, sans serif, and monospaced text faces, and corresponding math fonts. The math symbol set is very complete. Computer Modern is *the* font for TEX, which leads some to claim that the font is overused. The characters are fairly thin and light, and so are not as readable on screen in small sizes or from high-resolution laser printers.[2] In a comparison by Raph Levien, the printing in Knuth's *Digital Typography* [7] is heavier than the digital version or from a laser printer.

Type 1 versions of Computer Modern from Blue Sky Research and Y&Y, Inc. have been made freely available by the American Mathematical Society (AMS) and a collection of publishers and other technical companies [8,4]. Basil K. Malyshev has also released a free Type 1 version of Computer Modern [9], originally for use with his TEX system BaKoMa TEX.

Computer Modern has been extended to include more characters, particularly for non-English European languages. These fonts include European Computer Modern by Jörg Knappen and Norbert Schwarz (METAFONT only) [10]; Tt2001 by Szabó Péter (converted into Type 1 format from METAFONT sources using `textrace`; Tt2001 has been superseded by CM-Super, which Péter recommends) [12,11]; CM-Super by Vladimir Volovich (also converted using `textrace`) [14,13]; and Latin Modern by Bogusław Jackowski and Janusz M. Nowacki (extended from the Blue Sky AMS fonts using MetaType1) [16,15].

2. When on screen, the fonts are usually anti-aliased, often into a gray blur because the stems are not thick enough to fill a pixel. When printed with a high-resolution laser printer, the fonts are shown accurately, but I think are too thin. With a medium-resolution printer like an inkjet, there's enough resolution to show the form of the letters (unlike on screen), but the low-resolution "bulks up" the letters compared to a high-resolution laser printer, with the letters thus appearing darker.

Figure 1: Computer Modern (using the Blue Sky and Y&Y Type 1 fonts; no package necessary).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath i\jmath j k\kappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\mu\nu v\upsilon w\omega\varpi x\chi y\psi z$ $\infty \propto \emptyset\varnothing$ đð ɘ

The SliTEX font (`lcmss`) is a sans serif text face that has wide letters and high x height. Its high readability makes it extremely suitable for slide presentations. However, there is no matching math font. SliTEX sans serif can be set as the primary text font using TEXPower's `tpslifonts.sty` [17].

Computer Modern Bright: This a sans serif font with corresponding math font derived from Computer Modern by Walter A. Schmidt [18]. CM-Super contains Type 1 versions of the text fonts in T1 encoding, and Harald Harders created Type 1 versions of the text and math fonts called `hfbright` [19] using `mftrace`.

Concrete and Euler or Concrete Math: The Concrete font was created by Knuth for his book *Concrete Mathematics* [20]. Hermann Zapf was commissioned by the AMS to create the math font Euler for use in *Concrete Mathematics*. Type 1 versions of Concrete in T1 encoding are available in the CM-Super collection [13], and Type 1 versions of Euler are available in the Blue Sky collection from the AMS [8] and in the BaKoMa collection [9]. The `eulervm` package by Walter Schmidt [23, 24] implements virtual fonts for Euler that are more efficient to use with LATEX. Ulrik Vieth created the Concrete Math fonts [21] to match the Concrete text fonts;

Figure 2: CM Bright (`\usepackage{cmbright}`; output uses the `hfbright` fonts).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath i\jmath j k\kappa\varkappa l\ell\lambda m n\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho q r s t\tau\pi u\mu\nu\upsilon w\omega\varpi x\chi y\psi z\ \infty\ \propto\ \emptyset\varnothing$ d$\eth$ $\ni$

the only free versions are implemented in METAFONT. The `ccfonts` package by Walter Schmidt [22] changes the text font to Concrete and changes the math font to the Concrete Math fonts if `eulervm` is not loaded.

Iwona and Kurier: The fonts Iwona and Kurier were created by J. M. Nowacki [25, 26] using the MetaType1 system based on typefaces by the Polish typographer Małgorzata Budyta. The two fonts are very similar, except that Kurier avoids "ink traps" with gaps in its strokes. The packages have complete math support in both TEX and LATEX.

Antykwa Półtawskiego: J. M. Nowacki created the font Antykwa Półtawskiego [27] using the MetaType1 system based on a typeface by Polish typographer Adam Półtawski. The package `antpolt` has no math support at this time, and requires the encoding to be set to QX or OT4.

Antykwa Toruńska: The font Antykwa Toruńska was created by J. M. Nowacki [29, 28] using the MetaType1 system based on a typeface by the Polish typographer Zygfryd Gardzielewski. The package `anttor` has complete math support in both TEX and LATEX.

Figure 3: Concrete text with Euler math (`\usepackage{ccfonts,eulervm}` `\usepackage[T1]{fontenc}`). Note that Concrete does not have a bold font, so Computer Modern is used instead. Non-bold text output uses the CM-Super Concrete fonts.

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

A∧∆∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
αabβcϑdδeϵεfζξgγhℏℎιijjkκϰlℓλmnηθϑoσσφφϕpppqrstτπuμνϖvwωϖxχyψz ∞ ∝ ∅∅dð ϑ

3 Core Postscript Fonts

When Adobe introduced Postscript in 1984, they defined 35 core fonts (in 10 typefaces) that must be present in all Postscript interpreters. In 1996, URW++ released a replacement set for the core fonts under the GNU General Public License. The URW++ fonts were primarily released for use with Ghostscript, a free Postscript interpreter. Table 1 lists the original Postscript fonts, along with the URW++/Ghostscript equivalents. Each font can be used as the default text font by selecting the indicated LaTeX package from the PSNFSS distribution [30].

Avant Garde and Kerkis Sans: The font Kerkis Sans was created by Antonis Tsolomitis [31,32] by extending Avant Garde to include Greek and additional Latin characters. The resulting fonts are stand-alone and can be used by applications outside of TeX. The package `kerkis` sets the sans serif font to Kerkis Sans; there is no package option to set Kerkis Sans to be the primary text font.

Figure 4: Concrete text with Concrete math (`\usepackage{ccfonts}` `\usepackage[T1]{fontenc}`). Note that Concrete does not have a bold font, so Computer Modern is used instead. Non-bold text output uses the CM-Super Concrete fonts.

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏℎiιjjkκϰlℓλmnηθϑoσς*ϕφϕpρϱqrstτπuμνυvwωϖxχyψz ∞ ∝ ∅∅dð ∋

Figure 5: Iwona text and math (`\usepackage[math]{iwona}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in 6G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYYΨZ 1234567890
*aαbβc6dδeϵεfζξgγhℏℎiιjkκϰlℓλmnηθϑoσς*ϕφϕpρϱqrstτπuμνυvwωϖxχyψz ∞ ∝ ∅∅dð ∋

Figure 6: Kurier text and math (`\usepackage[math]{kurier}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

A/ΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYΥΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath ijk\kappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\mu\nu\nu\upsilon w\omega\varpi x\chi y\psi z$ $\infty \propto \emptyset\varnothing$ d$\eth$ $\Im$

Figure 7: Antykwa Półtawskiego text (`\usepackage{antpolt}` and `\usepackage[QX]{fontenc}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYΥΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath ij\jmath k\kappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\mu\nu\nu\upsilon w\omega\varpi x\chi y\psi z$ $\infty \propto \emptyset\varnothing$ d$\eth$ $\Im$

Figure 8: Antykwa Toruńska text and math (`\usepackage[math]{anttor}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

ΑΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYYΨZ 1234567890
$\alpha\alpha b\beta c\eth d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath ijk\kappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\mu\nu\nu\nu w\omega\varpi x\chi y\psi z \infty \propto \emptyset\varnothing \mathrm{d}\eth \ni$

Adobe Postscript	URW++/Ghostscript	# of fonts	package
Avant Garde	URW Gothic L	4	`avant`
Bookman	URW Bookman L	4	`bookman`
Courier	Nimbus Mono L	4	`courier`
Helvetica	Nimbus Sans L	8	`helvet`
New Century Schoolbook	Century Schoolbook L	4	`newcent`
Palatino	URW Palladio L	4	`palatino`
Symbol	Standard Symbols L	1	—
Times	Nimbus Roman No. 9 L	4	`times`
Zapf Chancery	URW Chancery L	1	`chancery`
Zapf Dingbats	Dingbats	1	—

Table 1: Core Postscript fonts and URW++/Ghostscript equivalents.

Figure 9: Kerkis text and math (`\usepackage{kmath,kerkis}`; the order of the packages matters, since `kmath` loads the `txfonts` package which changes the default text font).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYϒΨZ 1234567890
*aαbβcϑdδeεεfζξgγhℏħiιjjkκϰlℓſmnηϑoσς*φϕ℘pρϱqrstτπuμνυωωϖxχyψz ∞ ∝ ∅∅dð ϵ

Bookman and Kerkis: The font Kerkis was created by Antonis Tsolomitis [31, 32] by extending URW Bookman L to include Greek and additional Latin characters. The resulting fonts are stand-alone and can be used by applications outside of TEX. A font of math symbols is included, but not used by the LATEX package. The package `kmath` uses txfonts for math symbols and uppercase Greek letters.

New Century Schoolbook and Millennial or fouriernc: The Millennial math font of the current author contains Greek letters and other letter-like mathematical symbols. A set of virtual fonts is provided that uses New Century Schoolbook for Latin letters in math, Millennial for Greek and other letter-like symbols, and txfonts and Computer Modern for all other symbols, including binary operators, relations, and large symbols. This font is still in development, but will hopefully be released in 2006. The `fouriernc` package of Michael Zedler [33] uses New Century Schoolbook for text and Latin letters in mathematics, and the Greek and symbol fonts from the Fourier-GUTenberg package for the remaining mathematical symbols.

Figure 10: New Century Schoolbook with Millennial math
(`\usepackage{millennial}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏℏıiıjjkκ≺lℓλmnηθϑoσς*ϕφ℘pρϱqrstτπuμνυvwωϖxχyψz ∞ ∝ ∅d∂ ϑ

Figure 11: New Century Schoolbook with Fourier math
(`\usepackage{fouriernc}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏℏıiıjjkκκlℓλmnηθϑoσς*ϕφφpρϱqrstτπuμνυvwωϖxχyψz ∞∝ ∅∅d∂ ϑ

Figure 12: Palatino text with pxfonts math (`\usepackage{pxfonts}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔVBCDΣEFΓGHIJKLMNOΘΩƱPΦΠΞQRSTUVWXYϒΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏħιiijjkϰlℓλmnηθϑoσς*ϕφφ*p*ρϱ*q*rsτπuμνυvwωϖxχyψz* ∞ ∝ ∅∅dð ∋

Palatino and pxfonts, Pazo, or mathpple: Young Ryu created the pxfonts collection [34], which contains Greek and other letter-like symbols, as well as a complete set of geometric symbols, including the AMS symbols. Diego Puga created the Pazo math fonts, which include the Greek letters and other letter-like symbols in a style that matches Palatino. The LaTeX package `mathpazo` (now part of PSNFSS [30]) uses Palatino for Latin letters, Pazo for Greek and other letter-like symbols, and Computer Modern for geometric symbols. The LaTeX package `mathpple` (also part of PSNFSS [30]) uses Palatino for Latin letters and slanted Euler for Greek and other symbols. Since Hermann Zapf designed both Palatino and Euler, the designs mesh well. An alternate use of Euler is using the `eulervm` package. Ralf Stubner added small caps and old-style figures to URW Palladio L in the FPL package [36], and Walter Schmidt extended these fonts in the FPL Neu package [37].

Times and txfonts, Belleek, mathptmx, or mbtimes: Young Ryu created the txfonts collection [38], which contains Greek and other letter-like symbols, as well as a complete set of geometric symbols, including the AMS symbols. The `txfonts` package also includes a very nice typewriter font, `txtt`. Belleek was created by Richard Kinch [39, 40] and is a drop-in replacement for the commercial fonts required by the `mathtime` package (now part of PSNFSS [30]). The LaTeX package

Figure 13: Palatino text with Pazo math (`\usepackage{mathpazo}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYYΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏℎiɪjjkϰℓlℓλmnηθϑoσς*ϕφ℘pρϱqrstτπuμνϋvwωϖxχyψz ∞ ∝ ∅∅d∂ ϑ ϑ

Figure 14: Palatino text with Euler math (`\usepackage{mathpple}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYΥΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏℎiɪjjkϰℓlℓλmnηθϑoσσ*ϕφ℘pρϱqrstτπuμνϋvwωϖxχyψz ∞ ∝ ∅∅d∂ ϑ ϑ

Figure 15: Times text with txfonts math (`\usepackage[varg]{txfonts}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYϒΨZ 1234567890
aαbβcϑdδeϵεfζξgγhℏℎiιjȷkϰlℓλmnηθϑoσς$\phi\varphi\wp$*pρϱqrstτπuμνυvωϖxχyψz* ∞ ∝ ∅∅dð ∋

`mathptmx` (also part of PSNFSS [30]) uses Times for Latin letters and Symbol for Greek and other symbols. Michel Bovani created the `mbtimes` package by using Omega Serif for text and Latin and Greek letters in mathematics. `mbtimes` also includes symbol fonts and a set of calligraphic letters. Omega Serif is the primary font for Omega, a 16-bit extension of TEX by John Plaice and Yannis Haralambous [43].

The STIX fonts project [41] is a collaboration of several academic publishers to create a set of Times-compatible fonts containing every possible glyph needed for mathematical and technical publishing. These fonts are still in development, with a scheduled release in the middle of 2006.

Note that Adobe Reader 7.0 replaces Times with Adobe Serif MM if Times or the Ghostscript equivalent Nimbus Roman No. 9 L is not embedded in the PDF file. Adobe Serif MM only has an oblique version, not a real italics, and thus, the primary text and Latin letters in mathematics will not match letters taken from additional fonts. This problem can be avoided by embedding Times or the Ghostscript equivalent Nimbus Roman No. 9 L into the PDF file. Also, I have heard (but not personally verified) that the Windows version of Adobe Reader displays Times New Roman when Times is not embedded. The upright versions of the two typefaces are very similar, but the italics are noticeably different (consider the z, for instance).

Figure 16: Times text with Belleek math (`\usepackage{mathtime}`; output uses the Belleek fonts).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma\,; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

A$\Lambda\Delta\nabla$BCDΣEFΓGHIJKLMNO$\Theta\Omega\mho$P$\Phi\Pi\Xi$QRSTUVWXY$\Upsilon\Psi$Z 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar i\imath j\jmath k\kappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\mu\nu\upsilon w\omega\varpi x\chi y\psi z\infty\propto\emptyset\varnothing\eth\eth\ni$

Figure 17: Times text with Symbol math (`\usepackage{mathptmx}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

A$\Lambda\Delta\nabla$BCDΣEFΓGHIJKLMNO$\Theta\Omega\mho$P$\Phi\Pi\Xi$QRSTUVWXY$\Upsilon\Psi$Z 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar i\imath j\jmath k\kappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\ \nu\upsilon w\omega\varpi x\chi y\psi z\infty\propto\emptyset\varnothing\eth\eth\ni$

Figure 18: Omega Serif text with Omega math (`\usepackage{mbtimes}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
$a\alpha b\beta c\eth d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar i\imath j\jmath k\varkappa\varkappa l\ell\lambda mn\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho qrst\tau\pi u\mu\nu\nu vw\omega\varpi x\chi y\psi z\ \infty\ \propto\ \emptyset\varnothing d\eth\ \ni$

Helvetica, Courier, and Zapf Chancery do not have matching math fonts. Courier and Zapf Chancery are inappropriate for mathematics anyway, but Helvetica is sometimes used for presentations and posters. The free fonts MgOpenModerna [44] and FreeSans [45] would be natural choices for the Greek letters in a Helvetica mathematics font.

4 Other Free Fonts

Several other fonts have been released for use with free open-source software. LaTeX packages have been created for most of these fonts.

Bitstream Vera Sans and Arev Sans: Bitstream Vera was released by Bitstream in cooperation with the Gnome Foundation [46] as a high quality scalable free font for use with free open-source software. Bitstream Vera serif, sans serif, and sans mono are available in text using the `bera` package by Malte Rosenau and Walter A. Schmidt [47]. Tavmjong Bah created Arev Sans [49] by extending Bitstream Vera Sans to include Greek, Cyrillic, and many mathematical symbols. The current author created the LaTeX package `arev` [48] using Arev Sans for text and math letters and bold Math Design fonts for Bitstream Charter for symbols.

Figure 19: Arev Sans text with Arev math (`\usepackage{arev}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϑPΦΠΞQRSTUVWXYΥΨZ 1234567890
*aαbβcϑdδeϵεfζξgγhℏℎiιjjkκϰlℓλmnηθϑoσς*ϕφ℘pρϱqrstτπuμνvυwωϖxχyψz
∞ ∝ ∅∅dð ɜ

Bitstream Charter and Math Design: Bitstream Charter [50] was donated by Bitstream for use with X Windows. The Math Design fonts for Bitstream Charter created by Paul Pichaureau [51] are very complete, including Greek letters, symbols from Computer Modern, and the AMS symbols. Charis SIL [52] might be an alternate source for Greek letters that match Bitstream Charter more closely. Another possibility for a math font is to use the Euler fonts with the `charter` and `eulervm` packages.

URW Garamond and Math Design: URW Garamond No. 8 [53] is available under the Aladdin Free Public License as part of the GhostPCL project. The Math Design fonts for URW Garamond created by Paul Pichaureau [51] are very complete, including Greek letters, symbols from Computer Modern, and the AMS symbols.

Utopia and Fourier or Math Design: Utopia [54] was donated by Adobe for use with X Windows. Michel Bovani created Fourier-GUTenberg [55] as an accompaniment to Utopia and is very complete, containing both Greek letters and standard and AMS symbols. The Math Design fonts for Utopia of Paul Pichaureau [51] are also very complete, including Greek letters and AMS symbols.

Figure 20: Bitstream Charter text with Math Design math (`\usepackage[charter]{mathdesign}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath\imath j\jmath k\kappa\varkappa l\ell\lambda m n\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho q r s t\tau\pi u\mu\nu\nu\upsilon w\omega\varpi x\chi y\psi z$ $\infty \propto \emptyset\varnothing$ đð ɜ

Figure 21: URW Garamond text with Math Design math (`\usepackage[garamond]{mathdesign}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩ℧PΦΠΞQRSTUVWXYϒΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath\imath j\jmath k\kappa\varkappa l\ell\lambda m n\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho q r s t\tau\pi u\mu\nu\nu\upsilon w\omega\varpi x\chi y\psi z$
$\infty \propto \emptyset\varnothing$ đð ɜ

Figure 22: Utopia text with Fourier-GUTenberg math (`\usepackage{fourier}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYΥΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath\imath j\jmath k\kappa\kappa l\ell\lambda m n\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho q r s t\tau\pi u\mu\nu\nu\nu w\omega\varpi x\chi y\psi z\infty\propto$ ∅∅dð ϶

Figure 23: Utopia text with Math Design math (`\usepackage[utopia]{mathdesign}`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\mathrm{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

AΛΔ∇BCDΣEFΓGHIJKLMNOΘΩϒPΦΠΞQRSTUVWXYΥΨZ 1234567890
$a\alpha b\beta c\partial d\delta e\epsilon\varepsilon f\zeta\xi g\gamma h\hbar\hbar\imath\imath j\jmath k\kappa\varkappa l\ell\lambda m n\eta\theta\vartheta o\sigma\varsigma\phi\varphi\wp p\rho\varrho q r s t\tau\pi u\mu\nu\nu\nu w\omega\varpi x\chi y\psi z$
$\infty\propto$ ∅∅dð ϶

Using METAFONT, Achim Blumensath created the package `MnSymbol` [56], which contains geometric symbols (no Greek or other letter-like symbols) in varying optical sizes that match the commercial font Adobe MinionPro. The `MnSymbol` package also contains traced Type 1 versions. `MnSymbol` is free; however the package `MinionPro` of Achim Blumensath, Andreas Bühmann, and Michael Zedler [57] which uses `MnSymbol` requires a license from Adobe for the font MininonPro.

5 Comparison of Features

Table 2 shows a comparison of the different features in each package. The only packages that have optical sizes are Computer Modern, CM Bright, Concrete, Euler, and MnSymbol. Except for when the `eulervm` package is used, Latin math letters are taken from the italic text font. An asterisk after a font name indicates that the package has a version of that style in its own font files.

The only sans serif fonts with matching math fonts are CM Bright and Arev Sans. Both work well for presentations. Computer Modern sans serif, CM Bright, Arev Sans, Bera Sans, Kerkis Sans, Helvetica, and Avant Garde all work well as sans serif fonts that accompany a primary roman font. Computer Modern typewriter, `txtt` (from txfonts), Luxi Mono [59], and Bera Mono all work well as typewriters fonts.

There are several other free fonts easily used in LaTeX, notably the Bera fonts, Luxi Mono, and efont-serif [60]. Malte Rosenau converted the Bitstream Vera fonts into Type 1 format, renaming the fonts to Bera [47]. Bera includes serif, sans, and mono. Bera Serif does not have a matching italic font, but the DejaVu fonts [58] are an extension of Bitstream Vera that include a true serif italic, as well as Greek and Cyrillic for all three styles. Except for Bera Sans and Arev Sans, none of the previous fonts have matching math fonts.

6 Creation of this Survey

It might be technically feasible to create a font survey such as this article as a single TeX document. This document, however, was not created in that fashion for two reasons. First, it would be an inordinate amount of work to switch between fonts within the same document. The authors of the LaTeX packages put in a

Package	Text	Greek	CM sym	AMS sym	Calligr	Blkbd	boldmath
computer modern	cm	cm	cm	ams	cm	ams	yes
cmbright	cmbright	cmbright	cm*	cm*	cm*	ams	no
ccfonts,eulervm	concrete	euler	euler	ams	euler	ams	yes
concmath	concrete	concrete	concmath	concmath	concmath	concmath	no
iwona	iwona	iwona	iwona	iwona	cm*	ams	yes
kurier	kurier	kurier	kurier	kurier	cm*	ams	yes
anttor	anttor	anttor	anttor	anttor	anttor	ams	yes
kmath,kerkis	kerkis	kerkis	txfonts	txfonts	txfonts	txfonts	yes
millennial	nc schlbk	millennial	txfonts	txfonts	txfonts	ams	no
fouriernc	nc schlbk	fourier	fourier	fourier	fourier	fourier	yes
pxfonts	palatino	pxfonts	txfonts*	txfonts*	txfonts*	pxfonts	yes
mathpazo	palatino	pazo	cm	ams	cm	pazo	yes
mathpple	palatino	euler	euler	ams	cm	ams	yes
txfonts	times	txfonts	txfonts	txfonts	txfonts	txfonts	yes
mathtime (Belleek)	times	belleek	belleek	ams	cm	ams	no
mathptmx	times	symbol	cm	ams	rsfs	ams	no
mbtimes	omega	omega	mbtimes	ams	rsfs*	esstix	yes
mathdesign (Charter)	charter	md charter	md charter	md charter	rsfs*	ams	yes
arev	arev	arev	md charter	md charter	cm	fourier	yes
mathdesign (Garamond)	garamond	md garamond	md garamond	md garamond	rsfs*	ams*	yes
fourier	utopia	fourier	fourier	fourier	fourier	fourier	yes
mathdesign (Utopia)	utopia	md garamond	md utopia	md utopia	rsfs*	ams*	yes

Table 2: Comparison of the features of different packages.

```
\documentclass{article}
\include{sampleformat}
  \usepackage{fourier}
\begin{document}
  \include{textfragment}
\end{document}
```

Figure 24: Sample LaTeX file for `fourier`. The file `sampleformat.tex` contains page layout commands, such as setting the margins and removing the page numbers. The file `textfragment.tex` contains the text and mathematics fragment to be displayed. Both included files are used by every sample LaTeX file. The line "`\usepackage{fourier}`" was changed for each sample to the package listed in the sample's caption.

considerable amount of effort to set up the fonts for a document, and it would be silly to duplicate their work. Second, we want to show to a reader exactly what he or she will get by using that package.

In order to accomplish these goals, a small LaTeX file (see Figure 24 for an example) was made for each font that loaded the appropriate packages and then loaded a common text fragment for display. Each file was LaTeXed and then converted to an EPS file using `dvips` with the -E option. The -E option creates a tight bounding box around the text. The main file `survey.tex` then included each of these graphics, and was compiled with `pdflatex`. For some reason, `dvips` created an unusable one-page PS file when including `mbtimes.eps`. HeVeA was used to convert `survey.tex` directly to HTML.

Acknowledgements

Thanks to Michael Zedler, Ulrik Vieth, Karl Berry, William Slough, and the anonymous referees for helpful comments.

References

[1] Philipp Lehman, The Font Installation Guide on CTAN:/info/Type1fonts/fontinstallationguide.

[2] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk, MetaType1 on CTAN:/fonts/utilities/metatype1

[3] Walter A. Schmidt, Mathematikschriften für LaTeX, http://home.vr-web.de/was/mathfonts.html.

[4] American Mathematical Society (AMS) webpage for Computer Modern Type 1 fonts, http://www.ams.org/tex/type1-fonts.html.

[5] Donald E. Knuth, *Computer Modern Typefaces*, Addison-Wesley Pub. Co., 1986.

[6] Raph Levien, Effect of gain on appearance of Computer Modern, http://levien.com/type/cmr/gain.html.

[7] Donald E. Knuth, *Digital Typography*, Stanford, California: Center for the Study of Language and Information, 1999.

[8] Blue Sky Research and Y&Y, Inc., Computer Modern Type 1 fonts on CTAN:/fonts/cm/ps-type1/bluesky.

[9] Basil K. Malyshev, BaKoMa Computer Modern Type 1 and TrueType fonts on CTAN:/fonts/cm/ps-type1/bakoma.

[10] Jörg Knappen and Norbert Schwarz, European Computer Modern fonts on CTAN:/fonts/ec.

[11] Szabó Péter, Tt2001 fonts on CTAN:/fonts/ps-type1/tt2001.

[12] Szabó Péter, webpage for textrace and Tt2001 fonts, http://www.inf.bme.hu/~pts/textrace.

[13] Vladimir Volovich, CM-Super on CTAN:/fonts/ps-type1/cm-super.

[14] Vladimir Volovich, CM-Super: Automatic creation of efficient Type 1 fonts from METAFONT fonts, *TUGboat*, 24 (1) 2003, 75–78.

[15] Bogusław Jackowski and Janusz M. Nowacki, Latin Modern on CTAN:/fonts/ps-type1/lm.

[16] Bogusław Jackowski and Janusz M. Nowacki, Latin Modern: Enhancing Computer Modern with accents, accents, accents, *TUGboat*, 24 (1) 2003, 64–74.

[17] TEXPower LATEX style files by Stephan Lehmke, http://texpower.sourceforge.net.

[18] Walter A. Schmidt, CM Bright on CTAN:/fonts/cmbright.

[19] Harald Harders, hfbright on CTAN:/fonts/ps-type1/hfbright.

[20] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete Mathematics*, Addison-Wesley, 1989.

[21] Ulrik Vieth, Concrete Math fonts on CTAN:/fonts/concmath.

[22] Walter Schmidt, ccfonts on CTAN:/macros/latex/contrib/ccfonts.

[23] Walter Schmidt, eulervm on CTAN:/fonts/eulervm.

[24] Walter Schmidt, Euler-VM: Generic math fonts for use with LATEX, *TUGboat*, 23 (3/4) 2002, 301–303.

[25] Janusz M. Nowacki, Iwona on CTAN:/fonts/iwona.

[26] Janusz M. Nowacki, Kurier on CTAN:/fonts/kurier.

[27] Janusz M. Nowacki, Antykwa Półtawskiego on CTAN:/fonts/psfonts/polish/antp.

[28] Janusz M. Nowacki, Antykwa Toruńska on CTAN:/fonts/antt.

[29] Janusz M. Nowacki, Antykwa Toruńska: an electronic replica of a Polish traditional type, *TUGboat*, 19 (3) 1998, 242–243.

[30] Sebastian Rahtz and Walter A. Schmidt, PSNFSS on CTAN:/macros/latex/required/psnfss.

[31] Antonis Tsolomitis, The Kerkis font family, *TUGboat*, 23 (3/4) 2002, 296–301.

[32] Antonis Tsolomitis, Kerkis on CTAN:/`fonts/greek/kerkis`.

[33] Michael Zedler, fouriernc on CTAN:/`fonts/fouriernc`.

[34] Young Ryu, pxfonts on CTAN:/`fonts/pxfonts`.

[35] Diego Puga, Pazo Math fonts on CTAN:/`fonts/mathpazo`.

[36] Ralf Stubner, FPL font on CTAN:/`fonts/fpl`.

[37] Walter Schmidt, FPL Neu package, `http://home.vr-web.de/was/x/FPL/`.

[38] Young Ryu, txfonts on CTAN:/`fonts/txfonts`.

[39] Richard Kinch, Belleek fonts on CTAN:/`fonts/belleek`.

[40] Richard J. Kinch, Belleek: A call for METAFONT revival, *TUGboat*, 19 (3) 1998, 244–249.

[41] STIX Fonts project, `http://www.stixfonts.org`.

[42] Michel Bovani, mbtimes at `ftp://ftp.gutenberg.eu.org/pub/gut/distribs/mbtimes/`.

[43] John Plaice and Yannis Haralambous, Omega at `http://omega.enstb.org`.

[44] MgOpenModerna, one of the MgOpen fonts, `http://www.ellak.gr/fonts/mgopen`.

[45] FreeSans, one of the Free UCS Outline Fonts, `http://savannah.nongnu.org/projects/freefont`.

[46] Bitstream Vera, released by Bitstream in cooperation with the Gnome Foundation, `http://www.gnome.org/fonts`.

[47] Malte Rosenau, Bera Postscript Type 1 fonts (converted from Bitstream Vera fonts, which necessitated the name change) and LaTeX support files by Walter A. Schmidt, on CTAN:/`fonts/bera`.

[48] Tavmjong Bah and Stephen Hartke, Arev Sans on CTAN:/`fonts/arev`.

[49] Tavmjong Bah, Arev Sans, http://tavmjong.free.fr/FONTS.

[50] Bitstream Charter on CTAN:/fonts/charter.

[51] Paul Pichaureau, Math Design fonts on CTAN:/fonts/mathdesign.

[52] Charis SIL, http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=CharisSILfont.

[53] URW Garamond on CTAN:/fonts/urw/garamond.

[54] Adobe Utopia on CTAN:/fonts/utopia.

[55] Michel Bovani, Fourier-GUTenberg on CTAN:/fonts/fourier-GUT.

[56] Achim Blumensath, MnSymbol on CTAN:/fonts/mnsymbol.

[57] Achim Blumensath, Andreas Bühmann, and Michael Zedler, MinionPro on CTAN:/fonts/minionpro.

[58] DejaVu fonts, http://dejavu.sourceforge.net.

[59] Luxi Mono on CTAN:/fonts/LuxiMono.

[60] efont-serif at http://openlab.jp/efont/serif/.

THE PracTEX Journal

Travels in TeX Land: Using the Lucida fonts

David Walden

Abstract

This paper describes buying, installing, and beginning to use the Lucida fonts. Then it describes some more exploration in the world of fonts in the context of the Lucida fonts

David Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. You can contact him at
dave@walden-family.com

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

The PracTEX Journal
TPJ 2006 No 01, 2006-02-18
Article revision 2006-01-31

Travels in TEX Land: Using the Lucida fonts

David Walden

Abstract This paper describes buying, installing, and beginning to use the Lucida fonts. Then it describes some more exploration in the world of fonts in the context of the Lucida fonts.

1 Introduction

In this column in each issue I muse on my wanderings around the world of TEX. Since this issue is about fonts, I thought it would be appropriate for me to do some wandering in the domain of fonts. Thus, I decided to try to use the Lucida fonts that the TEX Users Group (TUG) recently made arrangements to sell.

The Lucida fonts were designed by Charles Bigelow and Kris Holmes. They comprise a comprehensive set of fonts (serif, sans serif, bold, typewriter, etc.) aimed at looking good on low-resolution output devices. According to Charles Bigelow,[1] one of the differences between Times and Lucida Bright is that Lucida Bright has a bigger x-height so it looks bigger than Times at the same point size which offers some advantages at small sizes on screen and in laser printing where resolutions are low. However, as PDF has become popular, which allows zooming of documents, a big x-height is not as crucial as it was 18 years ago when Bigelow and Holmes first designed Lucida Bright or 20 years ago when they first brought out Lucida Sans. Also, Lucida Bright is a bit more tightly fitted so it can work well in small sizes in short lines and two- or three column pages in journals and magazines. Lucida Bright has the harmonized Lucida Sans, Lucida Sans Typewriter, Greek, various math symbols, and so on, for a more harmonious look, which many people like, when a seriffed typeface is used with sans-serif, fixed-pitch, and so on.

1. These descriptions are excerpted and slightly paraphrased but substantially quoted from emails from Bigelow to Karl Berry and me. Bigelow also has a more comprehensive description at http://tug.org/store/lucida/designnotes.html.

Copyright © 2006 David C. Walden

Bigelow makes that further point that even though some of the original motivations for the design of the Lucida fonts are no longer as important as they were (e.g., x-height in today's world where a zoom capability is ubiquitous), the font remains popular because now people just like the look of it. It's one of those form-follows-function deals that happen fairly often in the world of design where a practical or technical solution of one era becomes simply a stylistic look of a later era — like the millions of people who today wear blue jeans for style rather than hard manual work.

The Lucida fonts were originally make available to the TEX community on a proprietary basis by Y&Y, Inc., of which Berthold and Blenda Horn were the principals. Y&Y created the hints for the Type 1 Lucida fonts and created the necessary TEX font metrics (`.tfm`) for all of the Lucida fonts, including the fussy details TEX needs for math typesetting. Y&Y also developed the first set of macros to facilitate use of the Lucida fonts with TEX. Unfortunately, in 2004 Y&Y went out of business (http://www.tug.org/yandy/). Fortunately, TUG was able to make an arrangement with Bigelow and Holmes to continue to make the Lucida fonts available to the TEX community; for more details, see http://tug.org/lucida/.

The macros originally written by Y&Y were later largely superseded for use with standard LATEX by macros developed by David Carlisle and distributed as the `lucidabr` package.[2] Also, Walter Schmidt did significant reorganization, clean-up, and improvement of the TEX related aspects of Lucida fonts: he updated and rearranged the font metrics, virtual fonts, fd files and the related documentation. Lance Carnes, through his company PCTEX, provided sponsorship for Schmidt's work.

PCTEX, Inc. (www.pctex.com) has also arranged with Bigelow and Holmes to distribute the Lucida fonts to the TEX community. The PCTEX distribution includes the `lucimatx` package, a later package also developed by Walter Schmidt aimed at being an improvement on the `lucidabr` package (the use of which I assume in my discussion in the rest of this column). The `lucimatx` package is available from PCTEX independently of buying the Lucida fonts from PCTEX, for $15.

Walter Schmidt has explained to me that the PCTeX and TUG distributions

2. The source files says indicates that Sebastian Rahtz also had a hand in improving the Y&Y macros before David Carlisle, and I've heard of some involvement by Karl Berry as well.

contain identical fonts and metric files and thus, there is complete compatibility between the distributions and my discussion in sections 3 to 5 of this column applies to both distributions of the fonts.

2 Buying the Lucida fonts

Let me begin by saying that I have always avoided doing anything significant with fonts in LaTeX. Installing or using a new font has seemed too complicated to bother with, especially since I don't have a very discerning eye for fonts and those built into LaTeX and its basic classes seemed good enough to me. So, trying to acquire and use the Lucida fonts is my first significant fonts effort.

The Lucida fonts are available from TUG at http://tug.org/store/lucida/. I clicked on this URL and then clicked where it says, "general order form for the Lucida fonts" which took me to http://tug.org/store/lucida/order.html. There I put a number 1 in the box for "Lucida Complete - Individual Member" (with a price of $90) and then clicked the "Order Lucida fonts" button. That resulted in a message page saying to "Please go back to fix errors: You must indicate that you accept the TUG end user license terms to order Lucida fonts." So I went back and checked the box next to "Please read the TUG Lucida end-user license and then check here if you accept the terms" and then again clicked the "Order Lucida fonts" button.[3]

"Order Lucida fonts" took me to a page where I could give my credit card number, name, address, etc. I filled in what appeared to be needed and clicked on "Order Complete"; however, it told me I had failed to check the type of my credit card. (I've often wondered why on-line sales require the user to say VISA, Master Card, or whatever. I would have assumed that information was somehow implicit in the credit card number. Is it not?)

I checked the box for my type of credit card and clicked on "Order Complete" again. The TUG server displayed a nice receipt for me to print out that said, "Thank you very much for your order. The TUG office will notify you when it has

3. Notice that I said nothing about reading the http://tug.org/store/lucida/ page or reading the end user license terms. I suspect that I am like many users who download computer software. I want or need to use it and so I figure I'm stuck with whatever terms the seller requires. Hopefully I'm not agreeing to anything too crazy.

been processed."[4] That made it sound like nothing would happen immediately, so I went back and read the first two paragraphs of the "Order Lucida fonts" page which told me (second paragraph), "We regret that, at present, access to the Lucida fonts is not instant. A human must review your order and process your payment. We make every effort to do this within 24 hours or less, and send you the access information for the fonts. Please feel free to contact the TUG office if there is a delay." This was sort of a relief. It was easy enough to order the Lucida fonts, but since it was Sunday, I probably wouldn't have to do any frustrating work like trying to actually install the Lucida fonts until probably at least Monday. Whew!

3 Downloading and installing the Lucida fonts

Once my payment to TUG for the Lucida fonts was processed, I was given a URL to a location in the TUG server store. I clicked on that and accepted the option of downloading to my computer a zip archive called `lucida-complete.zip`. The instructions that came with the downloaded zip archive told me to unzip it into my local `texmf` directory, if I had one. I do, so I did.

Unfortunately, my first attempt at unzipping caused a problem. Allume's Stuffit Delux program which I use on my MS Windows XP system created a directory `localtexmf/lucida-complete` and unzipped the Lucida distribution into that directory. However, the correct installation of the Lucida file structure is for it to be distributed across the `doc`, `tex`, `fonts`, etc., directories that reside immediately below `localtexmf`, exactly as in the zip file without any intervening directories. Thus, I had to delete the `localtexmf/lucida-complete` directory and to try another unzip program. The 7-zip File Manager program did the correct unzipping of `lucida-complete.zip` to the destination `localtexmf`.

The instructions for installing the Lucida fonts next told me to refresh the file name database. I use MikTEX as it comes with TUG's 2004 ProTEXt distribution. With MikTEX and Windows XP, one refreshes the file name database by clicking the Start button, clicking Programs on the resulting menu, clicking MikTEX on the next resulting menu, clicking MikTEX Options, and clicking the Refresh Now button on under the General tab of the MikTEX Options dialog window.

4. The TUG server also sent me an email that contained the same information as the receipt.

4 First try with the Lucida fonts

Having installed the Lucida fonts, it was time to use them. My first use was this column. Click on the link on the HTML page for this paper for the file `pre-lucida.pdf` to see how this column worked before using the Lucida fonts. To convert this column to use of the Lucida fonts, I followed the installation instruction where it told me to look at the file `doc/fonts/lucida/lucida-sample.pdf` in my local `texmf` directory. I skipped to the top of the second page of that file and followed the instruction telling me to put the following code at the beginning of my LaTeX file:

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{lucidabr}
```

except I specified the pracjourn class of this journal rather than the article class. Then I simply recompiled the file for this column with PDFLaTeX, and everything seemed basically to work. See the file `walden.pdf` to see the resulting output.

5 Some more exploration

According to the `doc/fonts/lucida/lucida-sample.pdf` file, there are lots of other aspects of the Lucida fonts that my use of them with this column thus far doesn't illustrate. Therefore, my next effort was a little artificial exercise aimed solely at finding what else I see about the extent of the Lucida fonts and getting a grasp on how to effect each.[5] The right side of each line shows the LaTeX commands to causes the font changes shown on the left side of each line. Read `doc/fonts/lucida/lucida-sample.pdf` to get a little understanding for the need for the `\fontfamily` and `\selectfont` stuff below. In section 6, I'll say more about the LaTeX shape, series, and family font commands.

5. This exercise is adapted (almost copied, really) from the `doc/fonts/lucida/lucida-sample.tex` file that comes with the Lucida distribution. That document and its `.pdf` file noted above contain additional information that I don't include here.

LucidaBright	
LUCIDABRIGHT-SMALL CAPS	\scshape
LucidaBright-Italic	\itshape
LucidaBright-Italic	also works with \em
LucidaBright-Oblique	\slshape
LucidaBright-Demi	\bfseries
LUCIDABRIGHT-DEMI-SMALL CAPS	\bfseries\scshape
LucidaBright-DemiItalic	\bfseries\itshape
LucidaSans-Typewriter	\ttfamily
LucidaSans-Typewriter Oblique	\ttfamily\slshape
LucidaSans-Typewriter Bold	\ttfamily\bfseries
LucidaSans-Typewriter BoldOblique	\ttfamily\bfseries\slshape
LucidaSans	\sffamily
LucidaSans-Italic	\sffamily\itshape
LucidaSans-Demi	\sffamily\bfseries
LucidaSans-DemiItalic	\sffamily\bfseries\itshape
LucidaSans-Bold	\sffamily\fontseries{ub}\selectfont
LucidaSans-BoldItalic	\sffamily\itshape\fontseries{ub}\selectfont
Lucida Typewriter	\fontfamily{hlct}\selectfont
Lucida Typewriter Oblique	\fontfamily{hlct}\selectfont\slshape
Lucida Typewriter Bold	\fontfamily{hlct}\selectfont\bfseries
Lucida Typewriter BoldOblique	\fontfamily{hlct}\selectfont\slshape\bfseries
LucidaFax	\fontfamily{hlx}\selectfont
LucidaFax-Italic	\fontfamily{hlx}\selectfont\itshape
LucidaFax-Bold	\fontfamily{hlx}\selectfont\bfseries
LucidaFax-BoldItalic	\fontfamily{hlx}\selectfont\itshape\bfseries
LucidaCasual	\fontfamily{hlcn}\selectfont
LucidaCasual-Italic	\fontfamily{hlcn}\selectfont\itshape

LucidaCalligraphy-Italic	`\fontfamily{hlce}\selectfont`
LucidaHandwriting-Italic	`\fontfamily{hlcw}\selectfont`
Lucida Blackletter	`\fontfamily{hlcf}\selectfont`

6 Lucida and math

Personally, I don't use math in my LaTeX work to any significant extent. Thus, I have not tried the math fonts here, with one trivial exception. For extensive examples of use of the Lucida fonts for math, see the file `lucida-amsmath.pdf` which accompanies this column.

There is one issue slightly related to math that I will mention. I have been in the habit of using `$\{$`, `$\}$`, and `$\backslash$` to produce the left brace, right brace, and backslash characters in some situations, for example, in verbatim text in footnotes. Because my eye has not been very sensitive about fonts, I never paid much attention to the result.

However, with the Lucida `typewriter` font, the difference between fonts in `\this{}` and `\this{}` is apparent, even to me. Thus, I was reminded that in LaTeX one can get left and right braces simply by typing `\{` and `\}` which have the advantages of not switching to a math font and being fewer characters to type.

The problem with using `$\backslash$` to produce a backslash became particularly apparent when I tried to produce the characters `\/` while discussing italic correction in the following section. I never knew before that LaTeX has the built-in command `\textbackslash` for just this purpose and which can be used to produce a better version of `\/`. (I confess, however, that I actually switched to mostly using `\char'\\` to produce backslashes in typewriter mode. It is stranger looking but fewer characters to type than `\textbackslash`, although I slightly worry it is some sort of violation of good LaTeX programming practice.)

7

7 Lucida and the NFSS

Although I have used LaTeX for a number of years, I have never really learned more about fonts than is described in section 4.1 of Kopka and Daly's *Guide to LaTeX, Fourth Edition*, and I skipped reading some of what was there. I had previously looked at several more comprehensive descriptions of the New Font Selection Scheme (NFSS), but my eyes always glazed over after a sentence of two. I'm just not interested in reading documentation in the abstract. However, now that I was actually loading and trying to use the Lucida fonts (and had the set of font files in front of me to look at), it seemed like a good time to try again to understand more about NFSS. So, I carefully reread section 4.1 and then skimmed Appendix A of Kopka and Daly's book.

The point of the NFSS is apparently to make the font "attributes" be as independent of each other to the greatest extent possible, that is, to the extent there are fonts available to support the different combinations of attributes. There are (more or less) five sorts of attributes:

- Family (typewriter, sans serif, and Roman, i.e., serif)
- Shape (upright, italic, slanted, and small caps)
- Series (medium weight or boldface)
- Size (5pt, 7pt, 8pt, 9pt, 12pt, etc.)
- Encoding (OT1, T1, TS1, etc.)

So, in some sense we want available to us all of the elements in the cross-product of these attributes:

$$\textit{family} \times \textit{shape} \times \textit{series} \times \textit{size} \times \textit{encoding}$$

7.1 Family, shape, and series

In the first three groups of examples of section 5, we saw independent use of declarations of font families, shapes, and series.

There we also saw an example of a font declaration that doesn't quite fit in a family, shape, or series: `\em`, for "emphasis." The declaration `\em` is the same as `\itshape` if the mode is already non-italic; but, if the mode is already italic, `\em` switches to a non-italic mode. However, it may be better to use the

\emph command form (noted below) of this declaration because it also puts italic correction at both ends of the italic text so the user does not have to do it manually with \/.

There is one other font declaration that doesn't quite fit in a family, shape, or series: \normalfont, which switches things back to a defined default. The \normalfont declaration was used for the first example in section 5 (LucidaBright alone). The default font is defined in the lucidabr style file that was loaded with the \usepackage{lucidabr} command.[6]

There is a font command parallel to each font declaration. The whole set is:

Family

```
{\rmfamily ...}   \textrm{...}    % serif font
{\ttfamily ...}   \texttt{...}    % typewriter font
{\sffamily ...}   \textsf{...}    % sans serif font
```

Shape

```
{\upshape ...}    \textup{...}    % upright font
{\itshape ...}    \textit{...}    % italic font
{\slshape ...}    \textsl{...}    % slanted font
{\scshape ...}    \textsc{...}    % small caps font
```

Series

```
{\mdseries ...}   \textmd{...}    % medium weight font
{\bfseries ...}   \textbf{...}    % boldface font
```

Other

```
{\normaltext ...} \textnormal{...} % default font
{\em ...}         \emph{...}       % emphasis font
```

The font declaration names can also be used for environments, for example:[7]

6. Lots of other things relating to use of the Lucida fonts are also defined in the style file. There is a parallel style file, **lucbmath**, for when a user only wants the Lucida math fonts.

7. Morten Høgholm says that he prefers creating a new environment, e.g,

```
\newenvironment*{fontchange}[1]{#1\ignorespaces}{}
\begin{fontchange}{\itshape\bfseries} Text.  \end{fontchange}
```

which he says, "gives you the option to combine declarations, and it is a proper environment."

```
\begin{itshape}
. . .
\end{itshape}
```

7.2 Size

LaTeX has commands such as the following for selecting the font size: `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, etc. Like many other font sets, Lucida set of fonts is scaled geometrically from one set of fonts and, thus, any size from 1 point to a very large number of points is available. Between them, the class file one is using plus the `lucidabr` package define the size of the fonts that are mapped into each of the font size commands.

7.3 Encoding

At the top of the `.tex` file for this column, between the document class command and the command to load the `lucidabr` style package, we find these two commands:[8]

```
\usepackage[T1]{fontenc}
\usepackage{textcomp}
```

These commands have to do with the coding attributes.

Each font has a certain number of characters in it. Historically, fonts had 128 characters, and this is the default that comes with the Computer Modern fonts that are the LaTeX default. Today, many fonts have 256 characters. Also, there must be a mapping between the individual characters and the character positions in the font (that is, between a particular character and a number between 0 and 255). The command `\usepackage[T1]{fontenc}` specifies the so-called Cork coding of 256 characters. *There is no support with the Lucida fonts for LaTeX's default OT1 encoding.* The `\usepackage{textcomp}` command makes available an auxiliary coding (TS1) of almost 100 special characters known as the Text Companion fonts (see Table 7.6 on page 363 of *The LaTeX Companion, Second Edition*).

8. See page 5 for the complete sequence.

7.4 The generalized font selection commands

As is implicit in the last five groups of examples in section 5, some of the fonts we want may be outside the defaults that comes with the `lucidabr` style file. Thus, to select these fonts, we must use the lower level commands. The section 5 examples used the following commands `\fontfamily`, `\fontseries`, and `\selectfont` in addition to the usual font declarations.

The NFSS uses six lower level commands to specify the font to be used, and the usual font declarations used earlier in this section translate into these six lower level commands:

```
\fontencoding{ }
\fontfamily{ }
\fontseries{ }
\fontshape{ }
\fontsize{ }{ }
\selectfont
```

Various of the first five commands above are given to select the desired parameters of the font, and then the `\selectfont` command is given to actually select the font.

Table 7.25 on page 414 of *The LaTeX Companion* shows nine "weight classs" and nine "width classes" that can be used with the `\fontseries` command; one of these weight classes, `ub`, is used in the examples in section 5 specify ultra bold. The most common series parameters are `m` for medium (normal) weight and `b` for bold.

Table 7.26 on page 415 of *The LaTeX Companion* shows six "shape classifications" that can be used with the `\fontshape` command. Four of them seem to be used with the Lucida fonts: `n` (normal), `it`, `sl`, and `sc`. There is no explicit use of this command in the examples of section 5.

Table 7.27 on page 416 of *The LaTeX Companion* shows over twenty possible "standard font encodings" for use with LaTeX. The Lucida fonts support the T1, TS1, and LY1 encodings.

The `\fontsize` command takes two parameters — a font size and a baseline skip size. Page 415 of *The LaTeX Companion* gives this example:

```
\fontsize{14.4}{17}
```

which means `14.4pt` or `\Large` characters with a `17pt` space from the baseline of one line of letters to the baseline of the next line of letters.

The parameter of the `\fontfamily` command appears to be part of the names of the actual font files that come with the distribution. For instance, Lucida Casual's `\fontfamily{hlcn}` command in section 5 appears to refer to files for three different codings (T1, TS1, and LY1) of a specific font — the files named `ly1hlcn.fd`, `ly1hlcn.fd`, and `ly1hlcn.fd`. In any case, following is a chart from a file named `lucida.txt` by Walter Schmidt that is part of the Lucida distribution. Notice that the codes with an asterisk in the first column are the codes that are shown in `\fontfamily` commands in the examples of section 5.

```
NFSS classification
===================

family   series      shape(s)       FontName(s)
-------------------------------------------------------------
hlh      m, b        n, it, sl, sc  LucidaBright,
                                    LucidaBright-Italic,
                                    LucidaBright-Demi,
                                    LucidaBright-DemiItalic,
                                    LucidaBrightSlanted,
                                    LucidaBrightSmallcaps,
                                    LucidaBrightSmallcaps-Demi
hlhj     m           n, it, sc      LucidaBright,
         b           n, sc          LucidaBright-Italic,
                                    LucidaCalligraphy-Italic,
                                    LucidaBright-Demi,
                                    LucidaBrightSmallcaps,
                                    LucidaBrightSmallcaps-Demi
hls      m, b, ub(*) n, it          LucidaSans, LucidaSans-Italic,
                                    LucidaSans-Demi,
                                    LucidaSans-DemiItalic,
                                    LucidaSans-Bold,
                                    LucidaSans-BoldItalic
hlct(*)  m, b        n, sl          LucidaTypewriter,
                                    LucidaTypewriterBold,
```

			LucidaTypewriterOblique, LucidaTypewriterBoldOblique
hlst	m, b	n, sl	LucidaSans-Typewriter, LucidaSans-TypewriterOblique, LucidaSans-TypewriterBold, LucidaSans-TypewriterBoldOblique
hlx(*)	m, b	n, it	LucidaFax,LucidaFax-Italic, LucidaFax-Demi, LucidaFax-Italic
hlce(*)	m	it	LucidaCalligraphy-Italic
hlcn(*)	m	n, it	LucidaCasual, LucidaCasual-Italic
hlcw(*)	m	it	LucidaHandwriting-Italic
hlcf(*)	m	n	LucidaBlackletter

Acknowledgements

Karl Berry suggested this project and helped me all along the way. Charles Bigelow reviewed section 1 for accuracy and allowed his words to be used and paraphrased. Walter Schmidt also reviewed section 1 for accuracy and provided words for me to paraphrase. Will Robertson provided better definitions of the TEX and LATEX logos to work with the Lucida fonts in the context of the pracjourn class. Morten Høgholm reviewed section 7 on NFSS for accuracy and suggested several important corrections and improvements. Two anonymous reviewers also provided suggestions for improvement, as did editor Lance Carnes.

Biographical note

David Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. More history is at www.walden-family.com/dave.

THE PracTEX Journal

Ask Nelly: Which fonts can be accessed from the TeX Live distribution just using usepackage? What is the difference between Fonts and Typefaces? Which are the best fonts for typesetting math?

The Editors

Abstract

Ask Nelly is a question and answer column. Nelly is the quiet person who sits at the back corner desk, who knows a lot, and when asked any question is always ready with a patient answer. If Nelly doesn't know the answer, Nelly will know an expert who has the answer. Feel free to Ask Nelly about any aspect of LaTeX, TeX, Context, etc.

Q: Which fonts can be accessed from the TeX Live distribution just using usepackage?

A: This question is answered in this issue's article by Walter Schmidt. See Section 3, and Reference [3]. Following are some additional notes by editorial board member Steve Peter:

Every LaTeX distribution provides support for almost all of the fonts in what is called the standard "PSNFSS collection". Some distributions provide for additional fonts, such as lucida and lucsans, which require the commercial Lucida fonts (available from TUG and PCTeX). The PSNFSS fonts are documented in `psnfss2e.pdf` which you can find in your TeX installation at `/texmf/doc/latex/psnfss`, or at CTAN.

In addition, there are a number of packages outside of PSNFSS that can be used to control fonts. The Latin Modern package (lmodern) switches to the PostScript version of Latin Modern, an enhanced version of Computer Modern.

In addition, the bera package sets the text fonts to varieties of the bera font (a modified version of Vera), luximono will set your typewriter font to Luxi Mono, kerkis switches to the Kerkis fonts, beton switches to the Concrete fonts, punk gives you Knuth's Punk font, but only

in a bitmapped version. As you can see, most of the packages are named the same as the fonts they provide support for, so poke around the `/texmf/tex/latex` directory for additions.

TpJ

Q: What is the difference between Fonts and Typefaces?

A: Fonts and typefaces are very different things, even though people tend to use the terms interchangeably. Typefaces are designs like Bembo, Gill Sans or Papyrus. Type designers create typefaces, using software programs to shape the individual letters. A few still draw the letters by hand and then scan the drawings into a type-design application.

Fonts are the things that enable the printing of typefaces. Type foundries produce fonts. Sometimes designers and foundries are one and the same, but creating a typeface and producing a font are two separate functions.

From Design to Font

The 16th-century French designer Claude Garamond created the typeface that now carries his name. Creating the design was a multistage process. First he cut a letter (backward) on the end of a steel rod. The completed letter was called a punch. Next he took the punch and hammered it into a flat piece of soft brass to make a mold of the letter. A combination of molten lead, zinc and antimony was poured into the mold, and the result was a piece of type whose face was an exact copy of the punch. After Garamond made punches for all the letters he would use and cast as many pieces of type as he thought he would need, he put the type into a typecase. The resulting collection of letters was a font of type.

Many Fonts — One Typeface

Over the years, there have been hand-set fonts, machine-set fonts, phototype fonts and now digital fonts of the Garamond typeface. Currently there are TrueType, PostScript Type1 and OpenType fonts of Garamond. There are Latin 1 fonts of Garamond, used to set most of the languages in Western Europe, and Greek and Cyrillic fonts, which enable the setting of these alphabets. All these fonts are of the Garamond typeface design.

This question was answered by typography author Allan Haley, Director of Words and Letters at Monotype Imaging and chairman of the SOTA board of directors.

TpJ

Q: Which are the best fonts for typesetting math?

A: If you ask this question of a dozen mathematicians, math book and journal publishers, and others who use math fonts, you will likely get a dozen different answers. Choosing the best TeX math font may be a matter of reviewing some of those currently available, and making

sure the fonts fit well with the text font you want to use.

There are at least two good math font surveys available. In this issue there is a good survey article of free math fonts by Stephen G. Hartke. Walter Schmidt maintains a web site which gives examples of various free and non-free math fonts combined with text fonts (his page is in German — use Google translation tools or similar if you like, and remember that "fette" (bold) in German may translate as "fat" or "grease" in English).

Also in this issue are two articles on non-free math fonts. David Walden writes about Lucida fonts, which contain both text and math fonts. Michael Spivak describes how the MathTime Professional fonts, which are a good match with Times and Baskerville text fonts, took over his life.

TPJ

- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.

THE PracTEX Journal

Distractions: Name-that-Font

The Editors

Abstract

This column offers some amusements as a distraction from writing serious LaTeX and TeX documents. In this issue we offer several font-related contests with valuable first prizes, and PracTeX Journal souvenir postcard prizes for second and third place.

- Name-That-Font for Type Novices — Prize: the book *Stop Stealing Sheep & Find Out How Type Works*, donated by FontShop.
  This contest was devised by Yves Peters with hints by Tamye Riggs. Both are active in SOTA (Society of Typographic Aficionados) and Typophile.
- Name-That-Font for Type Gurus — Prize: books and CDs, *Indie Fonts 1 & 2*, donated by SOTA.
  This contest was devised by Yves Peters with hints by Tamye Riggs. Both are active in SOTA and Typophile.
- Typeface quiz — Prize: **Lucida fonts** (PCTeX, TUG), donated by Bigelow & Holmes Inc.
  These questions about typefaces and designers are by Charles Bigelow of Bigelow & Holmes.
- Math font quiz — Prize: ***MathTime Professional fonts***, donated by Publish or Perish Inc.
  A challenging set of math font questions by Michael Spivak, Publish or Perish, Inc.
- Sudoku contest answers
  Winners from the previous Distractions contest.

Feel free to suggest or send ideas for other distractions.

Rules for all contests: Send your answers by using the email link next to each contest. Enter as many contests as you like, but only one entry per person per contest please. **Submissions must be received by March 31, 2006**. Winners of each contest will be chosen as follows: after the submission deadline entries will be picked at random, and the first three with the most correct answers will receive the first, second, and third prizes for that contest. All decisions are final. Good luck.

Name that Font!

by Yves Peters and Tamye Riggs of SOTA (Society of Typographic Aficionados).

Test your knowledge of serifs and swashes with this typographic treasure hunt. Below are text samples set in twelve typefaces currently available in digital format. Your assignment is to discover the name of each font shown. Each phrase is a clue that will help you identify that particular typeface. We've added an extra hint above each image for good measure.

Google, a thesaurus, and free association are your friends. Some of these glyphic gems are found in the far reaches of the internet, while others may be as close as your operating system. Compare letterforms with the type viewers available on most font foundry websites — these handy tools will offer the physical evidence you need to ensure you've found the right type.

The first part of this quiz is easy — six of the world's most common fonts are awaiting your examination. Be warned — the second set of challenging characters may cause alphabetic anxiety.

Hint: Refer to Tamye Riggs's article in this issue for links to online type samples. The words in the boxes and the hints above them do help. Some hints refer to the typeface name and some to the type designer.

These two contests were designed by **Yves Peters**, a (typo)graphic designer, rock drummer, and father of three who trained at FSI/MetaDesign Berlin and did a three-year stint as type expert/technical advisor at the Belgian FontShop. His talent for being able to identify most typefaces on sight, while utterly useless in daily life, garnered him the prestigious role of moderator of the Typophile Type Identification forum.

TypoNovice

Enter this contest.

by Yves Peters and Tamye Riggs. First prize is the book, *Stop Stealing Sheep & Find Out How Type Works,* second edition, by Erik Spiekermann and E.M. Ginger; published by Adobe Press. Retail value $30. Donated by FontShop.

1) Think inside the box.

Times
New Roman

2) Is it Willie or Ray?

as moonlight
through the pines

3) Lines and spines.

Literary
Personality

4) The “Brown” identity.

International Shipment

5) Go to Helvetica.

Deutsche Bundespost

6) Related to Jillette's partner?

FINANCIAL SERVICES

TypoGuru

Enter this contest.

by Yves Peters and Tamye Riggs. First prize is the two-book set with included CD-ROMs: *Indie Fonts* and *Indie Fonts 2*, both first editions, edited by James Grieshaber, Richard Kegler, and Tamye Riggs; published by P-Type Publications. Retail value $80. Donated by SOTA.

1) Concrete jungle?

2) Chuck's chum.

3) Not dry.

Chulalongkorn
Bangkok University

4) Related to Zsa Zsa?

Uncharted
Territories

5) Deep thoughts.

NICCOLO
MACHIAVELLI

6) Muscular, en español.

Typeface Name Quiz

Enter this contest.

by Charles Bigelow. First prize is **Lucida fonts** (PCTeX, TUG), donated by Bigelow & Holmes Inc., retail value $145.

1. What famous novel (and the movie or movies made from it) has at least three characters having the last names of two well known type designers?

2. What typeface has had two different authorized names, one meaning bizarre or monstrous and the other signifying a country?

3. What typeface is named after the author of the first book printed in that type, a philosophical dialogue between two men hiking up a mountain?

Math Font Quiz

Enter this contest.

by Michael Spivak. First prize is ***MathTime Professional fonts***, donated by Publish or Perish, Inc., retail value $179.

The quiz. It's challenging — answer as many questions as you can. (The quiz was typeset with *MathTime Professional* fonts, and the large symbols in questions 2 and 6 come from them.)

Runner-up prizes: a TeX Bug cartoon postcard

- Comment on this paper
- Send submission idea to editor

Page generated February 28, 2006 ; TUG home page; search; contact webmaster.