

# Fonts and Acrobat Making Electronic Paper Work

Martin Hosken  
SIL Non-Roman Script Initiative  
Copyright © 2001 SIL International



This presentation requires the following fonts:

SIL NRTC Presentation

Courier New

## Introduction

### PDF is great for:

- High Quality print images
- Highly interactive e-documents
- Small(ish) screen quality documents for Web
- Working across platforms
- Post production processing
- Interactive viewing
- Searching
- Public Specification

Need to make the right type of document



Things get easier because we don't have to use the keyboard to do the rendering by substitution, that has now gone into the font.

Don't things now become trivial with keyboards becoming, basically, a 1:1 mapping between a key and a Unicode value. There is no need for contexts now right?

## Making PDFs: Postscript

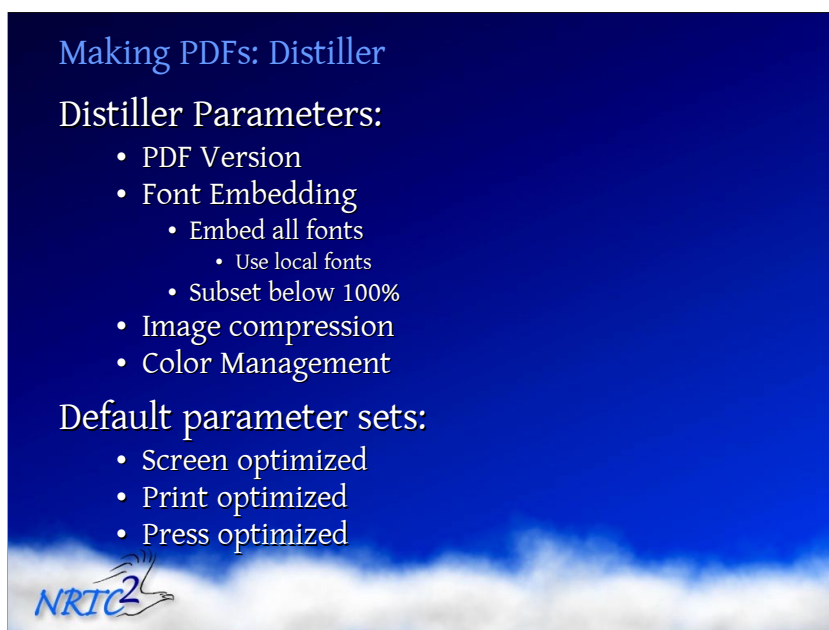
### Normal route: Postscript -> PDF

- Need a Postscript printer driver
- Choose carefully
  - Default **acrobat.ppd** with Adobe Printer Driver
  - Want colour kept
  - Want fonts embedded well

### PDF-Writer

- Rarely works (v4)
  - Font embedding seems buggy
  - Tries to be too simple





#### **PDF Version:**

1.2 is for Acrobat 3 and above and is fine for documents with simple TTF and no CID fonts. 1.3 is for Acrobat 4 and above and does good graphics and CID fonts, etc. 1.4 is for Acrobat 5 and adds little: transparency

Use the lowest version that meets your needs to maximise the viewers that can display your file

#### **Font Embedding**

For the most part, you want fonts embedded – after all, that is why you are making a PDF right? You don't want the poor reader to have to install all the fonts you used in making the document in the first place. This means you have to turn on Font Embedding. It is also wise to subset the fonts. This will mean that only the glyphs you use will end up in the font.

Sometimes fonts fail to embed because they don't have embedding permission (as set in the OS/2 table). This can cause a distilling process to fail. This is the most common reason I have encountered for distilling processes to fail.

#### **Image compression**

Use whichever image compression level you want for the target audience for the PDF. If you are sending things by email and it may get printed, I have found that 96dpi is a good compromise between size and 'image quality'. You may want to go lower, to say 72dpi to save space, or increase it to 300dpi for better print quality (or even 1200dpi for monochrome images (line drawings))

#### **Color Management**

This is only an issue if you are talking to a printer (as in a person!) and are dealing with colour. Likewise for the advanced options.

#### **Default Parameter sets**

This is for Acrobat 4. Your mileage may vary

##### *Screen Optimized*

72dpi images (except monochrome which are 300dpi). No font embedding fonts demanded, with the base 14 fonts specifically not embedded. The aim of this is to make the smallest PDF that is sensible. PDF version is 1.2

##### *Print Optimized*

A good compromise between size and quality. This is aimed at printing from a normal computer printer. Version is set to 1.3, which I normally change. Images are subsampled to 300dpi. Fonts are embedded and subset

##### *Press Optimized*

Everything you might need for a full printing press work process. Includes thumbnails, size is no object. Subsampling as per print optimized, all fonts are embedded and subset. Colour management is left to the document – assuming that the generating program knows what it is doing with colour better than Acrobat. Every type of job ticket is included and every feature enabled. Clearly uses the highest version possible – 1.3 in this case.

# Making PDFs: Ghostscript

## Freeware

- v6.01 – last version that works!
  - Postscript level 1 or 2 only
- v7 – current version
  - Aiming for Distiller 4 equivalence
    - Level 3 CID fonts supported, etc.
    - Also v1.4 features
  - Type42 bug means fonts don't embed
- Randomises code values – can't search

## GSView

- No longer freeware ☹
- Not needed to create PDFs – batch file



Ghostscript is great. In many respects, it distills better than the Acrobat distiller. If it gets confused about fonts (which is not hard) it generates bitmaps for the glyphs. This looks very ugly on the screen, but if the resolution is high enough, the results print fine. The resulting files from v6.01, therefore are bigger than those from Acrobat distiller.

After v6.01, Ghostscript got saddled with a bug which occurs when converting from Type42 fonts. Thus the mechanism used by Windows Postscript printer drivers to embed TrueType fonts in Postscript. So Windows users tend to get stuffed by this bug. But since Ghostscript is Linux driven, the bug still hasn't been fixed ☹.

GSView is a GUI to Ghostscript to allow viewing of Postscript files. It is quite nice and certainly easier to use than generating tricky .bat files. But it is no longer free ☹. Ah well. Life moves on! At least we can still generate the PDF for free.

## Making PDFs: Others

### PDFTeX

- Generates PDF directly from TeX – no DVI
- Can interact with TTFs, in its own way
- No Unicode support

### Dvipdf

- From DVI to PDF
- No TTF support

### High end tools generate PDF directly

- Public format, so anyone can generate it



PDFTeX has the normal TeX limitation of 8-bit fonts. But apart from that it is really nice, if you are into TeX.

DVIPDF doesn't work with TrueType as far as I know

Since PDF is a public format, a number of software vendors are supporting it directly from their products. Of course, producing good PDF is hard, so they may have a few bugs, particularly in the area of fonts, which is so poorly documented in the PDF specification.

## Making PDFs: Yet more ways

### Write your own

- Perl Text::PDF module
  - FRET
  - Pdfbkt
  - Pdfstamp
- PHP, PDFLib, Acrobat SDK, etc.

### HTML converter (Acrobat v4)

- Open Web page
- Limited formatting
- No Unicode support



Unicode support may well be there in v5  
V5 loses Adobe capture.

## Fonts in PDF files

### Warning! Technical Information

## How fonts work in Acrobat 4

### 3 basic encodings

- 8-bit
- Unicode based
- Glyph based

### Glyph descriptions

- TTF
- Type 1 – postscript outlines
- Type 3 – PDF description outlines



There are 3 basic encodings, regardless of what the documentation might say  
Actually, the second two are achieved in pretty much the same way, but we will  
separate them due to motivation.



## TrueType: 8-bit

### Uses **cmap** table

- Assumes ANSI encoding – cp1252
- Or can use Symbol encoding
- Encoding vector theoretically
  - Relates 8-bit codes with **post** table entries
- Encoding vector and **post** table ignored
  - Maybe different on Mac
- Acrobat Reader just installs the font
  - Masks problems on source machine

### ToUnicode cmaps

- Maps from 8-bit code to Unicode
- For searching purposes



There is a lot of discussion about postscript names in the PDF spec and you would expect, from reading that, that if you have a post table in your font you can pretty well do what you want. Well, the implementation isn't like that, at least on the PC. It is all done via the cmap and it is all very klunky. It is noticeable that Distiller 4 gets around problems by subsetting fonts as symbol fonts, because it seems to be about the only way that works consistently.

Because Acrobat Reader used to try to install the font, and that would get masked (you have a font of the same name installed already, so the font install didn't work) people would subset the font to ensure that the name changed and so the install would work. This is still the case, sort of. But now Acrobat Reader does its own rendering, so this is less of an issue and works cross-platform. But Adobe felt the need to carry the tail and so we are lumbered with some old history. Perhaps one day we can get away from 8-bit fonts entirely.

Each font may have a ToUnicode map which is designed to describe a relationship between the codes stored in the document and the Unicode values for those codes. For 8-bit fonts this is from the 8-bit code to Unicode. This helps searching, at least it should in v5

## TrueType: CID

### Type 0 fonts

- Relate CID (Character ID) to glyph
- Two layer model
  - Data is CID
  - Map to GID
- Glyph encoded data has a identity cmap
- Unicode encoded data
  - Turns font's **cmap** into PDF cmap
- Embedded TTF no longer needs **cmap** table
- Also ToUnicode cmap



Type 0 fonts are a way of separating the internals of the font from the stored data. You can even have multiple fonts pretending to be a single font (at least in theory).

## Conclusion

### Make good Postscript

- Or generate PDF directly

### Make good PDF

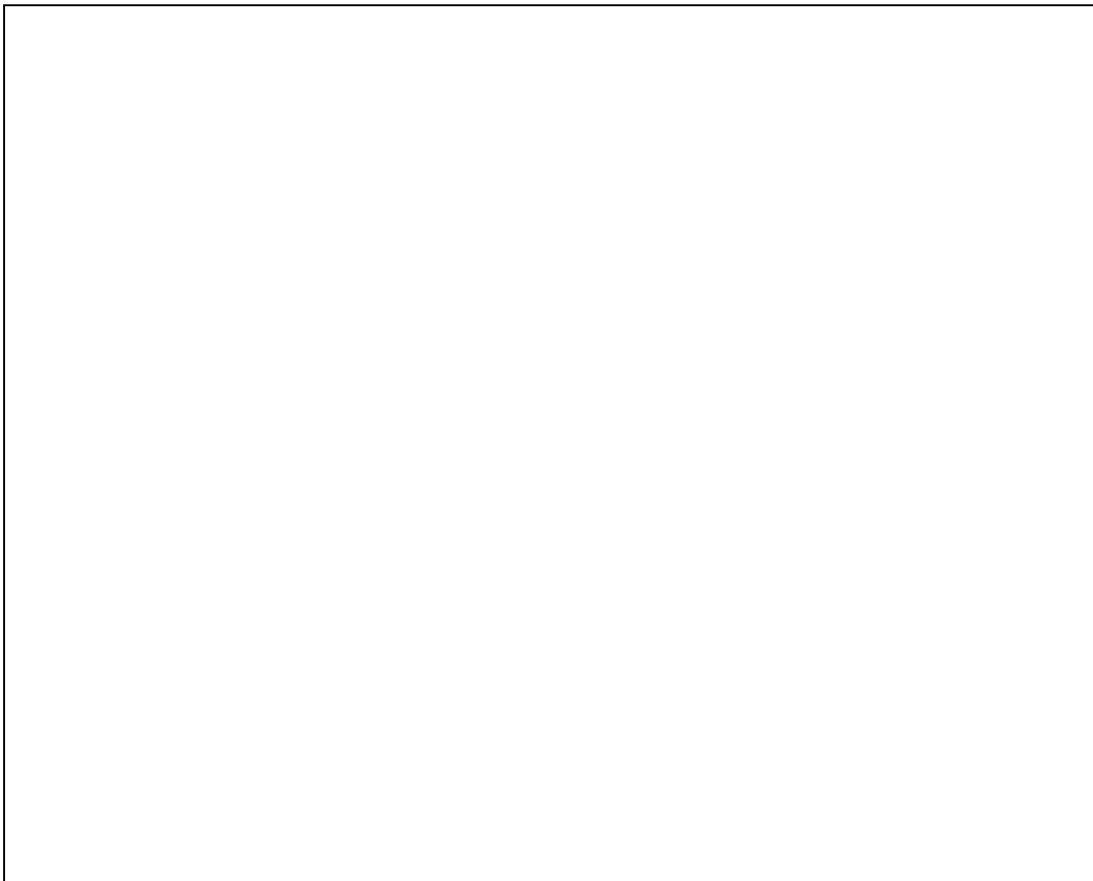
- Make sure all fonts are embedded
- Check on another machine



The answer to PDF issues is all about making good PDF and that can be a little tricky, but if you set things up right and use the latest software (i.e. v4 or later), then things go a whole lot easier.

Questions

Questions and comments?



## Contact Information

### For More Information Contact:

- Non-Roman Script Initiative  
SIL International  
7500 West Camp Wisdom Rd.  
Dallas, TX 75236  
(972) 708-7440  
[fonts@sil.org](mailto:fonts@sil.org)

This presentation is Copyright ©2001 SIL International,  
and may not be reproduced without permission

