

Introduction to Fonts

Important terms

- **Glyph:** A 'glyph' is the shape of a character in a font
- **.notdef glyph:** A '.notdef' glyph is a particular character that is used when glyph selection fails
- **Empty glyph:** An 'empty glyph' is a glyph without contour (shape), e.g. a blank

Font basics

Each block of text in a PDF document consists of four sets of data.

- The **encoded characters** which are sequences of bytes that represent the individual character codes that make up the text in the page
- The **font data** which is a group of glyphs (character visualizations) accessed by a unique number called a **Glyph ID (present in the font)**
- A map that links the encoded character codes to Glyph IDs (This mapping can be called encoding or glyph selection or glyph mapping)
- An optional map that links the character codes to Unicode values. This map is not needed when displaying the PDF but is required to allow the user to extract text content from the document (Mapping to Unicode: explained below)

How a page description in PDF makes use of fonts

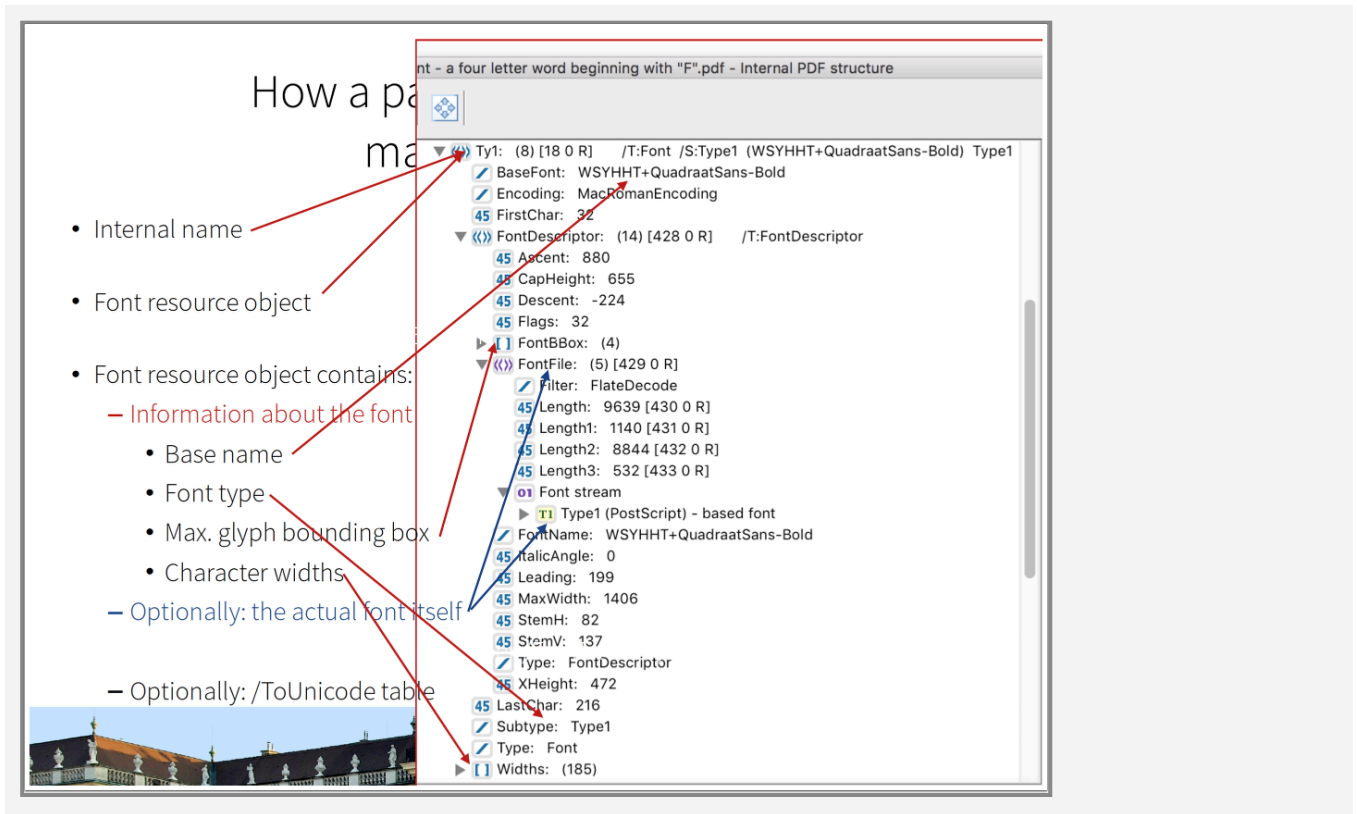
- Page description (content stream) selects font through an internal name: /F25 12 Tf
- Internal name points to font resource object (for current page):

```
/Resources << /Font << /F25 12 0 R >> >>
```

- Font resource object contains information about the font, for example:

1. Base name (example: /Courier)

2. Font type (example: /TrueType)
 3. Character widths (example: [593 615 572 ...])
- Optionally: the actual font itself (= “font is embedded”)
 - Data stream (for example in /FontFile2 ...)
 - Optionally: /ToUnicode table for mapping to Unicode



Mechanisms to include fonts in a PDF

By preference any fonts that are used in a layout are also included in the PDF file itself. This makes sure that the file can be viewed and printed as it was created by the designer. There are two mechanisms to include fonts in a PDF:

- Full embedding – A full copy of the entire character set of a font is stored in the PDF.
- Subsetting – Only those characters that are actually used in the layout are stored in the PDF. If the “\$” character doesn’t appear anywhere in the text, that character is not included in the font. This means that PDF files with sub-setted fonts are smaller than PDF files with embedded fonts.

Font types

- Simple fonts: 1 byte character codes that can address max. 256 glyphs
- 1. Type 1: 20 years old hence ancient in technology terms. Type 1 fonts are not cross-platform.
- 2. Compact Font Format (CFF font), very similar to Type 1
- 3. Type 3: Uses PDF drawing commands to define the glyph outlines. This font format allows greater flexibility over the appearance of the glyphs but does not include a **hinting** mechanism resulting in reduced visual quality for small text or low resolutions.
- 4. TrueType: Outline fonts which means that they can deliver output at any resolution or size.
- 5. OpenType: OpenType fonts internally resemble either TrueType fonts or Type 1 font data
- Composite fonts: 1 or 2 byte character codes that can address max. 65535 glyphs
- 1. Type 0: (Type 1 or OpenType/Type 1/CFF based)
- 2. Type 2: (TrueType or OpenType/TrueType based)

Mapping to Unicode

- Not required for displaying the page, but for text search or text copy.
- Options of mapping to Unicode:
- 1. Implicitly by standard (glyph selection) encodings in the PDF (example: MacRomanEncoding)
- 2. Encoding information in the actual font ('ToUnicode' cmap in TrueType fonts)
- 3. By glyph names in the actual font (in Type 1 fonts) via the Adobe Glyph List (maps glyph names to Unicode)
- 4. Explicitly by a ToUnicode table in the PDF data structure