AFP Support for TrueType/Open Type Fonts and Unicode

Reinhard Hohensee
Distinguished Engineer

October 24, 2003

Ricoh
Topics

- What is Unicode?
- What are TrueType and OpenType fonts?
- Why have we extended the AFP Architecture to support TrueType and OpenType fonts?
- TrueType/OpenType font installation
- TrueType/OpenType font references in an AFP (MO:DCA) data stream
- TrueType/OpenType fonts in an AFP resource group
- Support for legacy encodings
- Unicode complex text
What is Unicode?

- A standard developed by the Unicode consortium that defines a universal character encoding for the characters in all the major scripts in the world.

- Basic encoding is a double-byte encoding (UTF-16) that supports 64K characters:
  - Addresses plane 0 (basic multilingual plane –BMP)

- Encoding can be extended to a four-byte encoding using the surrogate concept:
  - High-surrogate in range X’D800’ – X’DBFF’
  - Low surrogate in range X’DC00’ – X’DFFF’
  - Addresses additional 1024x1024 = 1,048,576 characters which covers planes 1-16 (supplemental planes)
Current Environment

The German and French words for beauty are Schönheit and beauté, respectively.

***Application needs to understand scripts and code pages***

Unicode Environment

The German and French words for beauty are Schönheit and beauté, respectively.

***Application only needs to understand Unicode***
What are TrueType and OpenType Fonts?

- TrueType technology developed by Apple and Microsoft in late 1980s and early 1990s
- OpenType technology is an extension of TrueType with better support for international characters and complex scripts; developed by Microsoft and Adobe
- TrueType/OpenType is an open font standard that defines a scalable outline technology; in addition the OpenType format also supports Adobe Type 1 outlines
- Great majority of TrueType/OpenType fonts support Unicode encoding

*** TrueType/OpenType is the base font technology for the Windows platform, therefore it is the most prevalent font technology in the industry ***
TrueType/OpenType Font File

Table Directory
- Directory Entries
  - Table Tag
  - Offset
  - Length

Naming Table
- Name Records
  - Language & Locale info
  - Name ID
  - String length
  - String offset

Encoding Subtable
- Character Mappings
  - U(ffff) -> glyph index
  - U(ffff) -> glyph index
  - ........
  - U(ffff) -> glyph index

Glyph Table
- glyph (0) data
- glyph (1) data
- ........
- glyph (n) data

CMAP Table
- Encoding subtable entries
  - Platform ID = X’0003’
  - Encoding ID = X’0001’
  - Offset

Full Font Name
- e.g. US - English
- Times New Roman Bold
Why Extend the AFP Architecture to Support TrueType/OpenType Fonts?

- Provide customers with much greater choice of typefaces, particularly non-Latin typefaces

- Provide a multilingual presentation environment through the support of Unicode
  - By using the Unicode encoding, print data can be generated in many languages without requiring font switching

- Allow customers to migrate towards the same single font technology on traditional host platforms and the Windows platform
  
- Provide ability to print from Windows applications with the same fonts used on the display, thereby eliminating font substitutions
  
- Provide the basis for future, more advanced support of complex non-Latin scripts, e.g. Arabic, Hebrew, Indic, Thai
Installing TrueType/OpenType Fonts (TTF/OTFs) in an AFP System

- TTF/OTFs are installed using the **Font Installer for AFP Systems**
  - Runs as a Windows application
  - Installs fonts in remote print server resource libraries using FTP protocol
  - Builds a Resource Access Table (RAT) for the fonts in the resource library
  - RAT is indexed with a **full font name**

- Fonts must be ‘**Unicode enabled**’:
  - Contain Microsoft Unicode cmap subtable (platform ID = 3, Encoding ID = 1)
  - Specify a full font name (Name ID = 4) using the Microsoft Unicode encoding

- **Fonts are installed as is** – they are not altered and they are not wrapped in a container
Installing TrueType/OpenType Fonts (TTF/OTFs) in an AFP System (contd)

• Fonts can be installed as part of a **TrueType Collection file (TTC)**
  – File with common header and multiple TTF/OTFs that share tables
  – File size reduction with TTC if fonts share large number of glyphs
• Fonts can be installed with **linked fonts:**
  – Fonts “font1”, “font a”, “font b”, and “font c” are treated logically as a single font
  – The base font (font1) is always processed first, followed by the first linked font (font a), then the second linked font (font b), etc.
  – Supports ability to add user-defined characters (UDCs) to a font
  – Supports ability to extend font with additional characters (TrueType/OpenType file format restricts number of characters in a given font to 64K)

ffn (font1)  (base font)
ffn (font a)  (first linked font)
ffn (font b)  (second linked font)
ffn (font c)  (third linked font)

ffn = full font name
Resource Access Table (RAT)

- Architected table used by print servers to locate and process TTF/OTFs

Table Header

- Repeating group for 1st font resource
- Repeating group for 2nd font resource
- Repeating group for nth font resource

ffn (font1) (English)

ffn (font1) (German)

file name (font or TTC)
object OID (font or TTC)
index into TTC
linked font ffn (font a)

linked font ffn (font x)
TrueType/OpenType font reference in an AFP (MO:DCA) data stream

- TTF/OTF reference specified with a Map Data Resource (MDR) structured field that contains:
  - **Full font name** of font (FQN type X’01’ triplet)
    - Can be specified in multiple languages
    - Can be specified with various encodings (EBCDIC, ASCII, Unicode);
      - encoding defined by X’01’ triplet
  - Font descriptive information – size, character rotation, cmap encoding; specified by X’8B’ triplet
  - Resource type definition (object-type OID for TTF/OTF); specified by X’10’ triplet
  - Local ID for font reference within text, graphics, bar code objects; specified by FQN type X’BE’ triplet
TrueType/OpenType Fonts and Collections in Print File Resource Groups

- TTF/OTF Fonts and Collections may be carried in an AFP print-file level resource group ("inline" resources)
- RAT not used to process inline resources; all required information specified on Begin structured fields of containers
  - Full font name of font on Begin Resource (BRS) structured field of font container (multiple languages)
  - Full font name of base fonts on BRS of collection container (multiple languages)
  - Full font names of linked fonts follow name of base font
  - Object OID of font or collection on Begin Object Container (BOC) of container causes print server to attempt to use a printer-captured version of the font or collection
Font Container

BRS
  <Object-type OID=TTF/OTF (X’10’ triplet)>
  <fnn (FQN type X’01’)>
    <fnn - linked font a (FQN type X’7E’)>
     ........................................
    <fnn - linked font x (FQN type X’7E’)>

BOC
  <Object-type OID=TTF/OTF (X’10’ triplet)>
  <Object OID for font (FQN type X’01’ OID fmt)>
  OCD (1)

     Native, unaltered
     font data
     (segment 1)

     .........

     OCD (n)

     Native, unaltered
     font data
     (segment n)

EOC

ERS

Font Collection Container

BRS
  <Object-type OID=TTC (X’10’ triplet)>
  <fnn - base font 1 (FQN type X’6E’)>
    <fnn - linked font a (FQN type X’7E’)>
     ........................................
    <fnn - linked font x (FQN type X’7E’)>

  <fnn - base font n (FQN type X’6E’)>
     ........................................
  <fnn - linked font b (FQN type X’7E’)>

     ........................................
  <fnn - linked font y (FQN type X’7E’)>

BOC
  <Object-type OID=TTC (X’10’ triplet)>
  <Object OID for TTC (FQN type X’01’ OID fmt)>
  OCD (1)

     Native, unaltered
     collection data
     (segment 1)

     .........

     OCD (n)

     Native, unaltered
     collection data
     (segment n)

EOC

ERS
Support for Legacy Encodings

*** Allow data that is encoded with current EBCDIC and ASCII code pages to be printed with Unicode-enabled TrueType/OpenType fonts ***

- Code page that matches encoding of print data (e.g. EBCDIC code page 500) is identified on the MDR for the TTF/OTF font reference
- Code page can be identified by name (FQN type X’85’) triplet or by global identifier – CPGID (X’20’ triplet)
- Printer uses resident conversion tables to convert encoding of print data to Unicode encoding required by font
*** Allow data that is encoded in UTF-8 (ASCII-friendly 1-4 byte Unicode encoding) to be printed with Unicode-enabled TrueType/OpenType fonts ***

- UTF-8 encoding of print data specified with Encoding Scheme Identifier (X’50’) triplet on the MDR for the TTF/OTF font reference
- Printer uses resident UTF-8 to UTF-16 conversion algorithm to convert encoding of print data to Unicode encoding required by font
What is Unicode Complex Text?

- Unicode text is **complex** if it requires
  - Bidirectional rendering
  - Contextual shaping
  - Combining characters that have no equivalent code point
  - Special word break and justification rules

- Major languages are Arabic, Hebrew, Hindi, Thai

- Complex text cannot be rendered in one-code-point-to-one-glyph fashion; it requires
  - OpenType fonts
  - Layout engine to process runs of code points

- Support for OpenType fonts in AFP architecture forms basis for any potential future support for Unicode complex text
*** The new TrueType/OpenType support in the AFP architecture allows customers to migrate to a single, standard font technology across all presentation environments, and enables them to globalize their applications and workflows with Unicode-based multilingual capability. ***