AFP: Advances and Directions
XPLOR Document University

Reinhard Hohensee
Distinguished Engineer
IBM Printing Systems Division
March 1, 2007
Agenda

- AFP – an open architecture
  - AFP Color Consortium (AFPCC) and scope
  - Expansion of scope – AFP Consortium (AFPC)
  - Current membership
  - Status of work

- Recent advances in AFP
  - TTF/OTF Fonts
  - Unicode
  - Resource Access Tables (RATs)
  - Color Management

- Future directions

- Summary
AFP Color Consortium (AFPCC)

- Formed in October 2004
- Initial membership was 16 companies
- Scope was limited to developing a comprehensive color management capability within the AFP architecture
- Work progressed very well - first release of AFP Color Management Architecture (ACMA) was formally approved by AFPCC on January 20, 2006 and has been published; see www.afpcolor.org:
  - Overview document - *AFP Color Management Architecture (ACMA) Release 1*
  - Significant updates to the MO:DCA, IPDS, and AFP Line Data data streams; see updated references
  - New architecture reference – *Color Management Object Content Architecture (CMOCA) Reference*
    - Provides syntax for wrapping ICC profiles and other color management objects so they can be processed as AFP resources
AFP Consortium (AFPC)

- In May, 2006, IBM announced proposal to open development of complete AFP architecture to the AFPCC; this was accepted by all members by September, 2006
- ***AFP is now a completely OPEN architecture and the AFPCC has become the AFP Consortium (AFPC)***
  - Customer benefits
    - More consistent implementations, therefore better interoperability of products
    - Broader choice of vendors and products
    - Better investment protection - standards have industry staying power
- New scope of AFPC work:
  - Data streams - MO:DCA, IPDS, AFP Line Data
  - Object Architectures:
    - AFP GOCA (vector graphics)  BCOCA (bar codes)
    - CMOCA (color management resources)  FOCA (fonts)
    - IOCA (image)  PTOCA (text)
  - **Develop AFP into best-of-breed presentation architecture**
Current AFP Consortium Membership (31)

**Document Composition & Resource Creation:**
- Cincom
- COPI
- DocuCorp
- Document Sciences
- Elixir Technologies
- Exstream Software
- FormsPath
- GMC Software Technology
- Group 1
- Invaris
- Inventive Designers
- Isis-Papyrus
- Metavante
- PrintSoft
- StreamServe

**Servers & Transforms**
- CDP Communications
- Compart Systemhaus
- Crawford Technologies
- Emtex
- Intermate
- LRS
- MPI
- Solimar Systems
- Xenos Group Inc.

**Printers & Controllers**
- IBM
- Kodak
- Lexmark
- Oce
- Ricoh
- Xeikon International
- Xerox
Recent AFP Advances: TrueType/OpenType Font Support

- Scalable outline technology used on the Windows and Mac platforms; most prevalent font technology in the industry

- Benefits for AFP customers:
  - Provides greater choice of typefaces, particularly non-Latin typefaces
  - Allows migration to the same single font technology on all presentation platforms, e.g. Windows, zOS, AIX
  - Provides ability to print from Windows and Mac applications without font substitutions
  - Provides ability to transform to/from other PDLs, e.g. PS/PDF, without font substitutions
  - Provides the basis for future, more advanced support of complex non-Latin scripts, e.g. Arabic, Hebrew, Indic, Thai

- Support added to AFP architecture and to AFP/IPDS products starting in 2003; rollout of support on printers, servers, and formatters continues
Installing TrueType/OpenType Fonts (TTF/OTFs) in an AFP System

- TTF/OTFs are installed using an Installer such as the IBM Font Installer for AFP Systems
  - Windows application that 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 from the data stream with a full font name, which matches the name specified within the font

- Fonts must support Unicode encoding
- Fonts are installed as is – they are not altered and they are not wrapped in an AFP container; therefore they can be shared with non-AFP applications
Installing TTF/OTFs (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; base font (font1) is always processed first, followed by the first linked font (font a), then the second linked font (font b), etc.

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

- Provides ability to add additional characters (TTF/OTF file format restricts number of characters in a given font to 64K) and user-defined characters (UDCs)
TTF/OTF Reference in AFP Data Stream

**AFP (MO:DCA-P) Data Stream**

- BAG
- MDR
  - <EBCDIC character encoding>
  - <FFN = SymbolPropBT>
  - <Object-type OID = TTF/OTF>
  - <Font LID n>
  - <Font Descriptor>
- EAG
- BPT
- Set Font LID n PTX
  - u{xxxx}

**Resource Access Table (RAT)**

<table>
<thead>
<tr>
<th>Full Font Name</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>SymbolPropBT</td>
<td>Z0F48HOH</td>
</tr>
</tbody>
</table>

**TrueType/OpenType Font**

- **Encoding subtable (Unicode)**
  - Code | glyph index
  - xxxx | index i
  - ..... | ..... 

- **Glyph table (TrueType outlines)**
  - ..... 
  - ..... 

**TrueType/OpenType Rasterizer (Printer-resident)**

- Rasterized glyph bit map

**One-to-one rendering**
Recent AFP Advances: Unicode Support

- Unicode support is an integral part of TrueType/OpenType font support.
- Unicode is 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.
- 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’
  - Four-byte code consists of high-surrogate code followed by low-surrogate code.
  - Addresses additional 1024x1024 = 1,048,576 characters.
# Character Groupings in Unicode Base Plane

<table>
<thead>
<tr>
<th>Name</th>
<th>Code Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Scripts</td>
<td>0000-1FFF</td>
<td>Phonetic scripts, e.g. Latin</td>
</tr>
<tr>
<td>Symbols</td>
<td>2000-2DFF</td>
<td>Punctuation, math symbols</td>
</tr>
<tr>
<td>CJK Symbols</td>
<td>2E00-3FFF</td>
<td></td>
</tr>
<tr>
<td>CJK Ideographs</td>
<td>3400-9FFF</td>
<td>Unified CJK ideographs</td>
</tr>
<tr>
<td>Yi Syllables</td>
<td>A000-A4CF</td>
<td></td>
</tr>
<tr>
<td>Hangul Syllables</td>
<td>AC00-D743</td>
<td>Korean Hangul syllables</td>
</tr>
<tr>
<td>Surrogates</td>
<td>D800-DFFF</td>
<td>High/low surrogates</td>
</tr>
<tr>
<td>Private Use</td>
<td>E000-F8FF</td>
<td></td>
</tr>
<tr>
<td>Compatibility and Specials</td>
<td>F900-FFFFD</td>
<td></td>
</tr>
</tbody>
</table>
Why Unicode?

Current Environment

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

FS = font switch

*** 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 ***
Recent AFP Advances: Resource Access Table (RAT)

- Architected table (defined in MO:DCA Reference) for processing resources installed in resource libraries
  - Allows referencing of resources in data stream by natural, platform-independent name, e.g. TTF/OTF full font name, CMR name
  - Allows association of additional processing parameters with resource, e.g. public/private flag and object OID, in platform-independent manner

- Table consists of common header, followed by repeating groups that are defined for specific object types
  - Repeating groups consist of a header with length and object-type identifier, followed by MO:DCA triplet-like table vectors (TVs)
  - TVs are specific to object-type
Currently supported object types

- TrueType/OpenType fonts (TTF/OTFs)
- Color Management Resources (CMRs)
- Data Objects, e.g. IOCA, EPS, PDF, TIFF, GIF, JFIF (JPEG)

File name for RATs is fixed:

- TTF/OTFs: IBM_DataObjectFont.rat
- CMRs: AFP_ColorManagementResource.rat
- Data Objects: AFP_DataObjectResource.rat
**Resource Access Table – TTF/OTF RAT**

**RG Header:** Length, Flags (private/public, capture,…), identifier for TrueType/OpenType font object-type

<table>
<thead>
<tr>
<th><strong>Table Vector Type</strong></th>
<th><strong>Table Vector Content</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Font name</strong></td>
<td>Full font name in UTF-16BE; can be repeated in multiple languages</td>
</tr>
<tr>
<td><strong>File name</strong></td>
<td>File name of font or font collection</td>
</tr>
<tr>
<td><strong>Font OID</strong></td>
<td>Object OID for font or font collection</td>
</tr>
<tr>
<td><strong>Index into font collection</strong></td>
<td>Ignored if not a collection</td>
</tr>
<tr>
<td><strong>Linked font names</strong></td>
<td>Full font names of linked fonts; can be repeated</td>
</tr>
</tbody>
</table>
Resource Access Table – CMR RAT

**RG Header:** Length, Flags (private/public, capture,… ) identifier for CMR object type

<table>
<thead>
<tr>
<th>Table Vector Type</th>
<th>Table Vector Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMR name</td>
<td>Name of CMR in UTF-16BE</td>
</tr>
<tr>
<td>File name</td>
<td>File name of CMR</td>
</tr>
<tr>
<td>CMR OID</td>
<td>Object OID for CMR</td>
</tr>
<tr>
<td>Mapped CMR name</td>
<td>Name of CMR that is “mapped” to this CMR; can be repeated</td>
</tr>
<tr>
<td></td>
<td>- Link CMR if this CMR is a CC CMR</td>
</tr>
<tr>
<td></td>
<td>- TTC or HT CMR if this CMR is generic</td>
</tr>
</tbody>
</table>
### Resource Access Table – Data Object RAT

**RG Header:** Length, Flags (private/public, capture,…), identifier for data object type

<table>
<thead>
<tr>
<th>Table Vector Type</th>
<th>Table Vector Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object name</strong></td>
<td>Name of object in UTF-16BE</td>
</tr>
<tr>
<td><strong>File name</strong></td>
<td>File name of object</td>
</tr>
<tr>
<td><strong>CMR OID</strong></td>
<td>Object OID for object</td>
</tr>
<tr>
<td><strong>Rendering Intent (RI)</strong></td>
<td>Rendering Intent for object</td>
</tr>
<tr>
<td><strong>CMR name</strong></td>
<td>Name of a CMR that is to be associated with this object; can be repeated</td>
</tr>
<tr>
<td><strong>CMR Descriptor</strong></td>
<td>Processing mode for CMR (audit/instruction); paired with CMR name</td>
</tr>
</tbody>
</table>
Recent AFP Advances: Color Management

- **Customer benefits**
  - Device-independent color rendering - accurate colors regardless of output device
  - **Consistent** colors across multi-vendor configurations and workflows
  - **Tunable** colors at system level (in contrast to dials on printer)

- **AFP Color Management is based on:**
  - ICC profiles and ICC color management concepts
  - A new AFP object – the AFP Color Management Resource (CMR):
    - Provides wrapper for color management constructs, e.g. wrapper for an ICC profile (which is not modified)
    - Defined in a new formal architecture under the AFP Architecture umbrella – the Color Management Object Content Architecture (CMOCA)
    - Published in the new CMOCA Architecture Reference
    - Allows color management constructs like ICC profiles, tone transfer curves, halftones, to be processed as AFP resources – download once, reference multiple times, capture in device, package in inline resource group, etc.
# CMR Structure

<table>
<thead>
<tr>
<th>Length (bytes)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Length</td>
<td>CMR length, including length field</td>
</tr>
<tr>
<td>4</td>
<td>Signature</td>
<td>Eye-catcher string</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td>Set to zero</td>
</tr>
<tr>
<td>16</td>
<td>CMR Alias</td>
<td>Human-readable name</td>
</tr>
<tr>
<td>4</td>
<td>CMR Type</td>
<td>CMR Type</td>
</tr>
<tr>
<td>14</td>
<td>CMR Version</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Device-specific fields</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Media-specific fields</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>CMR-type-specific fields</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Reserved</td>
<td>Set to “@...@”</td>
</tr>
<tr>
<td>8</td>
<td>Reserved</td>
<td>Set to zero</td>
</tr>
<tr>
<td>N</td>
<td>CMRData</td>
<td>CMR data</td>
</tr>
</tbody>
</table>

**Grayed fields in order shown define CMR name – 146 bytes = 73 characters, which is used to reference the CMR in the data stream**
CMR Types

- **Color Conversion (CC) CMRs**: contain ICC profiles (in their ICC-defined syntax) that convert a device-specific color to/from the device-independent ICC Profile Connection Space (PCS)

- **Tone Transfer Curve (TTC) CMRs**: one-dimensional curves that are used to modify the values of a particular color component, e.g. cyan

- **Halftone (HT) CMRs**: screens that are applied to multibit data

- **Indexed (IX) CMRs**: mappings of indexed (numbered) colors in the data stream to output device colors or colorant combinations

- **Link Color Conversion (LK) CMRs**: Look-up tables (LUTs) that convert directly from a device-specific input color space in the data to the device-specific output color space of the presentation device
CMR Processing Modes

- Provides context for how a CMR that is tied to a document component should be processed
- Is specified along with the CMR reference and is not part of the CMR

Processing modes

- **Audit**: an audit CMR specifies color-management processing that *has been done* on a document component; for example it may specify a color conversion that has been done on the data (similar to an ICC input profile)
- **Instruction**: an instruction CMR specifies color-management processing that *is to be done* on a document component; for example it may specify a color conversion that is to be applied to the data (similar to an ICC output profile)
- **Link**: a link CMR is used to convert an input color space in the presentation data directly to the output color space of the presentation device without going through the PCS (similar to an ICC device link profile)
CMR Installation

- CMRs can be installed in an AFP resource library with an application that:
  - uploads the CMR into the resource library of a print server
  - builds the CMR Resource Access Table (RAT) entry for the CMR

- CMRs may also be carried in a printfile-level resource group in a BRS - BOC/EOC – ERS container, and are then called inline CMRs
  - BRS specifies CMR name
  - Print server always searches for CMRs first in resource group, then in resource library via CMR RAT
Associating CMRs with a MO:DCA document component

- CMRs can be associated with document components at all levels of the MO:DCA document hierarchy:
  - Printfile (highest)
  - Document
  - Group of pages/sheets
  - Page/overlay
  - Data object (lowest)

- A CMR at a lower level always overrides – for that document component only – a conflicting CMR at a higher level
  - Example: CC CMR at printfile level defines RGB = scanner (y) RGB, CC CMR at page (n) level defines RGB = scanner (x) RGB. According to rule, since CC CMRs conflict, all RGB for page (n) is interpreted as scanner (x) RGB
CMR Type: Color Conversion (CC) CMR

- ICC Profile Connection Space (PCS) - CIELab D50 or CIEXYZ D50 (24 bit)
- CC CMR: scanner x ICC profile - RGB --> PCS
- Processing mode = audit
- CC CMR: Infocolor 130+ ICC profile - PCS --> CMYK
- Processing mode = instruction
- TIFF image 24 bit RGB scanner x
- Printer bit map 32 bit CMYK Infocolor 130+
Link Color Conversions (contd)

TIFF image
24 bit RGB
scanner x

LK CMR:
scanner x RGB ->
Infocolor 130+ CMYK

Processing mode = link

Printer bit map
32 bit CMYK
Infocolor 130+
CMR Type: Indexed (IX) CMR - named colorant output

Indexed (IX) CMR in MO:DCA

Document Hierarchy:
- printfile
- document
- page/sheet group
- page/overlay
- object

Named Colorant Output:

MO:DCA printfile
...
BPG/BMO (page/overlay)
Color specified using X’4E’ triplet syntax:
<PTOCA -SEC>
<GOCA - SPC>
<BCOCA - BDD>
<IOCA - IDD>
<PGD/OBD>
........
EPG/EMO

X’yzz’ = X’0100’ to X’FFFF’

ColSpce=X’06’
(Highlight)

Color=X’yzz’

Indexed (IX) CMR - named colorant output

CMR Header

Subset ID Tag - ID = X’01’

Number of Named Colorants Tag - “n” colorants (max = 15)

Color Palette Named Colorants Tag

<table>
<thead>
<tr>
<th>X’yzz’</th>
<th>CIE Lab value (6B)</th>
<th>Intensity of 1st colorant (1B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Intensity of 2nd colorant (1B)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intensity of nth colorant (1B)</td>
</tr>
</tbody>
</table>

Colorant Identification List Tag

<table>
<thead>
<tr>
<th>Length</th>
<th>Colorant 1 Name (UTF-16BE) (2-250B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Colorant 2 Name (UTF-16BE) (2-250B)</td>
</tr>
<tr>
<td>Length</td>
<td>Colorant n Name (UTF-16BE) (2-250B)</td>
</tr>
</tbody>
</table>
Associating CMRs with a Document Component:
Printfile

BFM (FormDef)
BDG (Document Environment Group (DEG))

MDR, CMRname = cmrxyz
X’10’ triplet: Object-type OID = CMR
X’91’ triplet: Processing mode = audit

EDG

EFM

*** CMR scope = complete printfile ***
Associating CMRs with a Document Component: Group of pages/sheets

**CMR scope = group of sheets that contain pages (n+1) to (m)***
Associating CMRs with a Document Component: Page/Overlay

BPG/BMO (page/overlay)
BAG (Active Environment Group (AEG))

MDR, CMRname = cmrxyz
X’10’ triplet: Object-type OID = CMR
X’91’ triplet: Processing mode = audit
Scope = page/overlay

EAG
EPG/EMO

*** CMR scope = page or overlay ***
Associating CMRs with a Document Component: Data Object via IOB

BPG/BMO (page/overlay)
BAG (Active Environment Group (AEG))

MDR
  FQN type X’DE’ triplet: cmrxyz
  X’91’ triplet: Processing mode = audit
  Scope = data object

EAG

IOB
  FQN type X’01’ triplet: epslogo
  X’10’ triplet: Object-type OID = eps
  FQN type X’DE’ triplet: cmrxyz
  X’91’ triplet: Processing mode = audit
  Scope = data object

EPG/EMO

*** CMR scope = data object (eps) ***
Associating CMRs with a Document Component: Data Object via OEG

BBC/BGR/BIM/BOC (data object)
BOG (Object Environment Group (OEG))

MDR, CMRname = cmrxyz
X’10’ triplet: Object-type OID = CMR
X’91’ triplet: Processing mode = audit
Scope = data object

EOG

EBC/EGR/EIM/EOC

*** CMR scope = data object ***
Associating CMRs with a Data Object via Data Object RAT

BPG/BMO (page/overlay)
BAG (Active Environment Group (AEG))
...............MDR
FQN type X’CE’ triplet: epslogo
X’10’ triplet: Object-type OID = eps
...............EAG
...............IOB
FQN type X’01’ triplet: epslogo
X’10’ triplet: Object-type OID = eps
...............EPG/EMO

*** CMR scope = data object (eps) ***
Rendering Intent

To properly process a Color Conversion CMR (that contains an ICC profile) and a Link Color Conversion CMR (that effectively provides a conversion between two ICC profiles), system needs to know user rendering intent

- **Rendering intents defined by ICC**
  - **Perceptual**: Preserve appearance by mapping all colors into the device gamut while maintaining color relationships; typically used to render continuous-tone images
  - **Saturation**: Generate vivid, saturated colors, even at the expense of color accuracy; typically used for business graphics
  - **Media-relative colorimetric**: Maps white point of source color to white point of output media; reproduces all in-gamut colors accurately. Colors ‘look right’ but may not be accurate colorimetrically. Typically used for vector graphics or images.
  - **ICC-absolute colorimetric**: Does not map white point of source color to white point of output media; reproduces all in-gamut colors accurately. Colors are accurate colorimetrically but may not ‘look right’. Typically used for logos or proofing, i.e., simulating output of one device/media on another device/media.
Rendering Intent (contd)

- Associated with MO:DCA document components at the same levels of the document hierarchy as CMRs
  - Printfile (highest)
  - Document
  - Group of pages/sheets
  - Page/overlay
  - Data object (lowest)

- RI triplet specifies rendering intent independently for each major AFP object-type category
  - IOCA
  - Object containers (EPS/PDF/TIFF/JFIF/GIF)
  - PTOCA
  - GOCA
Future Directions

- Develop AFP as the best-of-breed presentation architecture for monochrome and color production variable-data printing
  - Based on customer requirements, fill-in “missing” functionality in AFP formats - PTOCA, GOCA, IOCA, BCOCA
- Develop infrastructure – architecture levels, testing, certification, etc. to ensure interoperability of AFP products developed by AFPC members
  - Formatting applications
  - Transforms
  - Print Servers
  - Printers
- Develop extensions to allow AFP environments/products to better interoperate with other environments – PS/PDF, XML, web, view, archive, sort/search, index, etc; requires better *metadata* support
Summary

- AFP is now an open industry standard, developed by the AFP Consortium (AFPC)
- AFP has been extended with a number of significant advances in the past few years:
  - TrueType/OpenType font support
  - Unicode encoding
  - Resource Access Tables (RATs)
  - ICC-based color management
- Future Directions
  - Best-of-breed presentation architecture
  - Interoperability of products
  - Interoperability with other environments