AFP Fonts

AFP Unicode Migration Fonts
Product Summary

PRPQ 8A8087 (5799-GHJ)
AFP Unicode Migration Fonts for MVS and OS/390

PRPQ 8A8090 (5799-GJJ)
AFP Unicode Migration Fonts for AIX, AS/400, Windows NT, and Windows 2000

IBM Printing Systems

Latest Update: December 30, 1999 at 2:48:32 PM

First Edition (December 1999)

Introduction

The AFP Unicode Migration Font PRPQs provide Unicode access to all Unicode Version 2.0 characters currently provided in the AFP Font Collection 2.1 single-byte and double-byte font products. Additional support has been added for Vietnamese, Windows Glyph List 4.0, changes made to ISO 8859-7, and recent enhancements made to the AFP Font Collection double-byte fonts. The two PRPQs provide 34 AFP double-byte outline fonts and an AFP Unicode-encoded code page to be used with them. When presented with a Unicode-encoded data stream, the fonts and code page in these products provide access to 35,411 characters as defined in Version 2.0 of The Unicode Standard.

Product Packaging

The AFP Unicode Migration Fonts are packaged as two PRPQs identified by the operating systems on which they can be used.

- PRPQ 8A8087, product number 5799-GHJ, is *AFP Unicode Migration Fonts for MVS and OS/390*. The product is SMP/E installable and is available on 3480 tape cartridge. Price: $4000.
- PRPQ 8A8090, product number 5799-GJJ, is *AFP Unicode Migration Fonts for AIX, AS/400, Windows NT, and Windows 2000*. Two CD-ROMs are available: one for AS/400 and one for AIX, Windows NT, and Windows 2000. The AS/400 CD-ROM contains fonts packaged as AS/400 libraries. The AIX, Windows NT, and Windows 2000 CD-ROM contains directories and installation procedures for the AIX, Windows NT, and Windows 2000 operating systems. Price: $1500 per CD-ROM.

Product Rationale

Background

Unicode is becoming a necessary and desirable component of the emerging global society. As more and more business transactions occur across country boundaries, the ability to communicate through Unicode's common encoding system will prove to be beneficial both economically and socially. Combining the written scripts of the world into a single encoding provides a uniform method of character identification that is far more flexible than previous encoding systems.

Encoding systems were established many years ago for the purpose of providing vehicles for the *interchange* and *presentation* of data. Two of the most common encoding systems in use today are

ASCII and EBCDIC, both of which provide single-byte addressing. Single-byte addressing limits the number of characters that can be selected to 256. In order to provide a means to access more than 256 characters, the concept of a *code page* evolved.
Code pages are 16 x 16 matrices of 256 addressable segments called *code points*. Any character can theoretically be assigned to any code point and many standards and conventions have been established to control their propagation and use. Encoding systems such as ASCII and EBCDIC provide conventions for the placements of commonly used characters, but code pages are often populated with other character-to-code-point combinations that are not necessarily regulated by the encoding system. Therefore, to accurately identify a character it is necessary to know the encoding system, the code page defined within the encoding system, and the code point used within the code page.

The need to specify a code page tends to inhibit the ability to interchange data because both sending and receiving applications must know which code page is being used. This can be extremely difficult if the data stream does not provide a method for specifying code pages, or worse yet, "assumes" that a particular code page will be used. *The desire to have one universally accepted code page was part of the motivation for the creation of Unicode.*

A Word About Unicode

Unicode was conceived as a data interchange mechanism whereby a common encoding system would be used to access the vast majority of characters defined by the world's written languages. By using the universal Unicode encoding, applications would be able to create information that could be interchanged anywhere in the world with any application fluent in Unicode, thus eliminating the need for managing multiple code pages. In order to provide access to so many characters, Unicode utilizes two-byte addressing which increases the number of unique code points within the encoding from 256 to 65,535.

The Unicode Standard was first published in 1991 by The Unicode Consortium and is in continual refinement as more languages (scripts) are added. The Unicode 2.0 character list, which defines 38,885 characters, was used as a design point for the AFP Unicode Migration Fonts. Glyph support for 35,411 of the 38,885 characters is provided.

AFP Unicode Fonts: A Little Ahead of Their Time...

While operating systems and data handling applications tend to concentrate on the data *interchange* aspects of an encoding system, text formatters, printers, and fonts deal with data *presentation*. (As will be shown later, presenting the data that is being interchanged offers some significant challenges.)

The AFP Unicode Migration Fonts address the presentation aspect of Unicode and are a significant step toward the goal of printing and displaying Unicode-encoded data streams. Fonts, however, are only part of the story. Effective use of Unicode in the presentation environment requires that Unicode encoding be used throughout the environment. Operating systems, applications that run on the operating

systems (particularly text generators), applications that handle Unicode data, displays, and printers must all be fluent in Unicode. Today, not all the pieces are in place to offer a complete Unicode solution in AFP; however, over time, the need for Unicode-encoded data will likely drive the operating systems, applications, et al, toward full Unicode support.

In some ways the AFP Unicode Migration fonts are a little ahead of their time since they precede Unicode enablement in other areas key to Unicode presentation; however, they do provide part of the package: 34 AFP outline fonts and an AFP Unicode code page that can be used to print text encoded in Unicode.

Unicode and the AFP Unicode Migration Fonts

There are several things to understand about Unicode and the Unicode support provided by this product.

Data Interchange vs. Data Presentation

As mentioned earlier, Unicode was established to provide a vehicle for interchanging data. It must be understood that there is a difference between *interchanging* data and *presenting* the data that is being interchanged. *Data interchange* is the ability to exchange information in such a way that the meaning is not lost or altered when passed between operating system environments, programming applications, and encoding systems. *Data presentation* is the ability to print, display, or otherwise render the data that is being interchanged. Unicode does a fine job of addressing data interchange; however, significant challenges exist when Unicode-encoded text is presented. The next section explains why the selection of a character shape (glyph) to represent a Unicode character may not be as easy it appears.

Characters and Glyphs

Understanding the difference between a *character* and a *glyph* is key to understanding the challenges faced when presenting Unicode data. A *character* is the **description** of a particular language element. A *glyph* is the **shape** used to represent the *character*. *Characters* are defined in Unicode, but *glyphs* are dependent upon language.

Disclaimer: In the past, this author and others familiar with AFP architecture, tended to use the terms *character* and *glyph* interchangeably because, within IBM, each character/glyph was assigned a unique Global Character IDentifier (GCGID). Unicode separates the definition (character) from the shape used to represent the character (glyph) and our thinking must change accordingly. The author has tried to be consistent in using these terms correctly; however, old habits die hard and inconsistencies may be found!

In most cases, the glyphs used to represent the characters are universally accepted. For example, the glyph used to represent “Latin small letter a” will be the same in any language that uses “Latin small letter

a". Of course typographic variants will exist, but the shape of the glyph will most always be recognizable. However, for many characters common to the Chinese, Japanese, and Korean (CJK) languages, this is not always the case. For these characters, the **meaning** of the characters is universally understood; however, the **shapes** (glyphs) used to represent the characters can be entirely different, so different in fact, that the shapes used to represent the characters in one CJK language may be totally unrecognizable to someone reading another CJK language *even though the Unicode definitions of the characters are the same for both languages.* Determining which glyph to present for Unicode characters such as these is the challenge faced in the presentation environment.

To accommodate the requirement for language-specific glyphs, fonts in this product are arranged by CJK language with each CJK font containing language-specific glyphs. This means that to see the correct glyph rendered, ***the user must anticipate the target language and select an appropriate font for the language with which they are familiar.***

CJK Considerations

The vast majority of glyphs in the CJK fonts are pictorial characters called ideographs which account for the majority of CJK textual characters. In addition to ideographs, each CJK font contains a number of non-pictorial glyphs that are used for printing Latin text, symbols, mathematical operators, and so forth. These glyphs have been designed specifically for the CJK font in which they have been placed and are considered the "correct" glyphs to be used for non-ideographic presentation. The problem is that many of these glyphs are associated with General Scripts characters found in Unicode outside the CJK ranges. When the combined CJK + General Scripts fonts are built, glyphs from the General Scripts fonts replace like-named glyphs in the CJK fonts thus eliminating access to some of the specially designed glyphs.

There are 22 fonts provided for printing CJK text. 11 fonts are provided for use where the best possible rendition of CJK is required and contain only the ideographs and specially designed non-ideographs. These fonts are referred to as the **CJK fonts** and using them will result in the best possible CJK presentation.. The other 11 fonts contain the same ideographs as first 11; however, they also contain all of the 1267 General Scripts characters from either Helvetica or Times New Roman. These fonts are referred to as the **CJK + General Scripts fonts**. Since the addition of the General Scripts characters replaced some of the specially designed non-ideographs, presentation of CJK text containing non-ideographs will not be as good as that achieved with the CJK Fonts.

Using the CJK + General Scripts fonts will provide access to many more Unicode characters because languages such as Arabic, Cyrillic, Greek, Hebrew, and all Latin languages have been added to the fonts. By gaining access to these characters you lose some of the specially designed non-ideographic glyphs.

Character Composition and Decomposition

In most cases, there is a one-to-one relationship between a character and a glyph. For example, in a given font there is usually one glyph used to present the “Latin small letter a”. Although a one-to-one relationship between character and glyph exists for many characters, it is not always the case.

Unicode provides extensive rules for character composition, such as characters created by the combination of two or more individual characters. The most common composed characters are ligatures, diphthongs, and diacritic (accented) characters. Since composed characters do not exhibit a one-to-one relationship between character and glyph, there must be a way to create the composed string of characters. Similarly, the rendering engine must be able to interpret this string and position the glyphs on the page in correct sequence, with correct alignment, and in the same presentation space defined by the text formatter. Composing (and decomposing) characters is the work of a “layout engine”, a data stream processor that is sensitive to the combining rules defined by Unicode.

Since character composition and decomposition are outside the context of font data, it must be understood that the fonts in this product provide one-to-mappings so that each supported Unicode code point will print or display exactly one glyph. It should also be noted, however, that Unicode provides extensive support for pre-composed characters, that is characters that are already "put together". For example, many accented characters such as “n tilde”, which is composed of “n” and “~”, are assigned Unicode values and are available through a one-to-one mapping. *The AFP Unicode Migration Fonts provide glyphs for many such characters.*

Big-endian and Little-endian

When Unicode values are used as code points, they are composed of two bytes and the question of how to order the bytes arises. Byte ordering is the way operating systems view the significance of each byte. If the first byte is most significant, the ordering is said to be *big-endian*. If the second byte is most significant, the ordering is said to be *little-endian*.

Unicode code points used to represent Unicode values can be either big-endian or little-endian. For example, the big-endian code point representation of Unicode value U+20AC is ‘20’x followed by ‘AC’x, or ‘20AC’x. The little-endian code point for U+20AC is ‘AC’x followed by 20’x, or ‘AC20’x. The way the code page is constructed determines which “endian” can be used for text generation.

*The code page provided in this product, T11200, is **big-endian** and code points will look exactly as the Unicode values look in The Unicode Standard. Text created to use this code page must also be big-endian.*

AFP Unicode Migration Fonts Description

The AFP Unicode Migration Fonts provide AFP outline font character sets and a Unicode code page that can be used to print, display, and format Unicode data streams. The characters supported by this product represent many characters currently provided in the AFP Font Collection V2.1 Single-Byte Character Set (SBCS) and Double-Byte Character Set (DBCS) font products plus other characters added to support Vietnamese, Windows Glyph List 4.0, new additions to ISO 8859-7, and additions to the AFP Font Collection DBCS fonts. *A total of 35,411 Unicode characters are supported by 179,979 individual glyphs.*

AFP Font Collection has long been recognized as a leader in providing fonts that make printing solutions possible for many of the world's languages, and it should come as no surprise that the glyphs found in these fonts form the backbone of the AFP Unicode Migration Fonts. Providing access to these characters (and many more that have been added) through a common Unicode encoding is the intent of this product. It is a "migration" product in the sense that it provides a way to migrate from familiar font families associated with EBCDIC and ASCII to identical font families structured around Unicode. All fonts in this product are AFP CID-keyed double-byte outline fonts. *No raster fonts are provided.*

In order to control development expenses, CID outline font technology was chosen over other technologies because existing applications, tools, and fonts that support CID technology were readily available. *IBM is currently evaluating other outline font technologies and the fact that this product utilizes CID outline font technology does not infer that this technology will be used in future offerings.*

Character Set Descriptions

There are 34 character sets in this product. Each character set is a CID-keyed double-byte outline font designed specifically to be used with the AFP Unicode code page, T11200, also provided in this product.

The character sets are grouped as follows:

- 12 Courier, Helvetica, and Times New Roman typefaces each containing 1267 non-CJK characters; referred to as *General Scripts* fonts.
- 11 CJK fonts containing glyphs designed specifically for the Chinese, Japanese, and Korean languages.

- 11 CJK plus General Scripts fonts add to each CJK font one General Scripts typeface thus increasing the Unicode support available in each CJK font.

The following licensed type families were included in the General Scripts fonts:

- For Courier: Boutros Typing and Shalom Hebrew
- For Helvetica: Boutros Modern Rokaa, Narkiss Tam, Pusuwan, and Thonburi
- For Times New Roman: Boutros Setting, Burirum, Kaewfah, and Narkissim

The character sets are summarized as follows:

- **Courier Unicode** General Scripts fonts in roman medium, italic medium, roman bold, and italic bold

  - CZ420U - Courier Unicode roman medium
  - CZ430U - Courier Unicode italic medium
  - CZ440U - Courier Unicode roman bold
  - CZ450U - Courier Unicode italic bold

- **Helvetica Unicode** General Scripts fonts in roman medium, italic medium, roman bold, and italic bold

  - CZH20U - Helvetica Unicode roman medium
  - CZH30U - Helvetica Unicode italic medium
  - CZH40U - Helvetica Unicode roman bold
  - CZH50U - Helvetica Unicode italic bold

- **Times New Roman Unicode** General Scripts fonts in roman medium, italic medium, roman bold, and italic bold

  - CZN20U - Times New Roman Unicode roman medium
  - CZN30U - Times New Roman Unicode italic medium
  - CZN40U - Times New Roman Unicode roman bold
  - CZN50U - Times New Roman Unicode italic bold

- **Japanese Unicode** fonts

  - CZJHKGU - Japanese Heisei Kaku Gothic (Japanese Extended)
  - CZJHMNU - Japanese Heisei Mincho (Japanese Extended)
  - CZJHMGU - Japanese Maru Gothic (Japanese Base)

- **Korean Unicode** fonts

  - CZHKG2U - Korean Gothic
  - CZHSM2U - Korean Myengjo

- **Simplified Chinese Unicode** fonts

  - CZSFSGU - Simplified Chinese Fang Song (GB)
  - CZSHEIU - Simplified Chinese Hei (GB)
  - CZSKAIU - Simplified Chinese Kai (GB)
  - CZSSNGU - Simplified Chinese Song (GBK)

- **Traditional Chinese Unicode** fonts

  - CZTKAIU - Traditional Chinese Kai
  - CZTSNGU - Traditional Chinese Sung

- **Japanese Unicode** fonts + General Scripts

  - CZH2JHKG - Japanese Heisei Kaku Gothic + Helvetica
  - CZN2JHMN - Japanese Heisei Mincho + Times New Roman
  - CZH2JHMG - Japanese Maru Gothic + Helvetica

- **Korean Unicode** fonts + General Scripts

  - CZH2HKG2 - Korean Gothic + Helvetica
  - CZN2HSM2 - Korean Myengjo + Times New Roman

- **Simplified Chinese Unicode** fonts + General Scripts

  - CZN2SFSG - Simplified Chinese Fang Song + Times New Roman
  - CZH2SHEI - Simplified Chinese Hei + Helvetica
  - CZN2SKAI - Simplified Chinese Kai + Times New Roman
  - CZN2SSNG - Simplified Chinese Song + Times New Roman

- **Traditional Chinese Unicode** fonts + General Scripts

  - CZN2TKAI - Traditional Chinese Kai + Times New Roman
  - CZN2TSNG - Traditional Chinese Sung + Times New Roman

All General Scripts fonts include support for 1267 Unicode characters included in the Unicode ranges shown below. "GS" indicates the number of characters supported by the General Scripts fonts and "Unicode" indicates the number of characters defined by Unicode 2.0.

Unicode Range	Code Points	GS	Unicode
Basic Latin	0000-007F	95	95
Latin-1 Supplement	0080-00FF	96	96
Latin Extended-A	0100-017F	128	128
Latin Extended-B	0180-024F	12	149
Spacing Modifier Letters	02B0-02FF	10	57
Combining Diacritic Marks	0300-036F	8	72
Greek	0370-03FF	77	105
Cyrillic	0400-04FF	94	226
Hebrew	0590-05FF	27	83
Arabic	0600-06FF	16	193
Thai	0E00-0E7F	87	87
Lao	0E80-0EFF	65	65
Latin Extended Additional	1E00-1EFF	8	246
General Punctuation	2000-206F	33	77
Superscripts and Subscripts	2070-209F	23	28
Currency Symbols	20A0-20CF	7	14
Letterlike Symbols	2100-214F	25	57
Number Forms	2150-218F	4	48
Arrows	2190-21FF	20	91
Mathematical Operators	2200-22FF	79	242
Miscellaneous Technical	2300-23FF	37	122
Control Pictures	2400-243F	3	37
Optical Character Recognition	2440-245F	3	10
Box Drawing	2500-257F	44	128
Block Elements	2580-259F	9	22
Geometric Shapes	25A0-25FF	25	80
Miscellaneous Symbols	2600-26FF	12	106
CJK Symbols and Punctuation	3000-303F	1	57
Private Use Area	E000-F8FF	118	0
Alphabetic Presentation Forms	FB00-FB4F	5	57
Small Form Variants	FE50-FE6F	4	26
Arabic Presentation Forms-B	FE70-FEFE	91	141
Specials	FEFF,FFF0-FFFF	1	2

Chinese, Japanese, and Korean fonts provide support for Unicode characters as described in the following table.

Unicode Range	J-Base	J-Ext	Korea	S-Ch(GB)	S-Ch(GBK)	T-Ch	Unicode
3000-303F	27	27	26	23	34	32	57
4E00-9FFF	6723	12211	4953	6763	20902	13090	20902
AC00-D7A3	0	0	11172	0	0	0	11172
E000-F8FF	0	0	44	2	97	5	0
FB00-FB4F	0	0	0	0	0	0	57
FB50-FE6F	19	19	20	18	52	54	26
FE70-FEFE	0	0	0	0	0	0	141
FEFF,FFF0-FFFF	220	220	222	222	222	105	2

Code Page Description

There is one code page provided, T11200. T11200 is a Unicode-encoded, big-endian, double-byte code page containing 35,411 code points corresponding to the 35,411 Unicode characters supported by this product. T11200 can be used with any of the 34 AFP Unicode Migration font character sets.

Coded Fonts

No coded fonts are provided with this product. Font selection must be by character set and code page or by Global Resource IDentifier.

Global Resource IDentifiers (GRIDs)

The following Font Global IDentifiers (FGIDs), Graphic Character Set Global IDentifiers (GCSGIDs), and Code Page Global IDentifiers (CPGIDs) have been established for the AFP Unicode Migration Fonts. Also shown in this table are Typeface (Roman Medium, Italic Medium, Roman Bold, and Italic Bold), Character Spacing (TYPOgraphic, UNIform, and BiWidth), and LineSpace in relative units.

General Scripts		TFace	Space	LineSp	GCSGID	FGID
CZN20U	Times New Roman	RM	TYPO	1070	65030	2308
CZN30U	Times New Roman	IM	TYPO	1070	65030	2310
CZN40U	Times New Roman	RB	TYPO	1070	65030	2309
CZN50U	Times New Roman	RB	TYPO	1070	65030	2311
CZH20U	Helvetica	RM	TYPO	1070	65030	2304
CZH30U	Helvetica	IM	TYPO	1070	65030	2306
CZH40U	Helvetica	RB	TYPO	1070	65030	2305
CZH50U	Helvetica	IB	TYPO	1070	65030	2307
CZ420U	Courier	RM	UNI	1000	65030	416
CZ430U	Courier	IM	UNI	1000	65030	424
CZ440U	Courier	RB	UNI	1000	65030	420
CZ450U	Courier	IB	UNI	1000	65030	428

CJK Character Sets		TFace	Space	LineSp	GCSGID	FGID
CZJHKGU	J-Heisei Kaku Gothic Unicode	RM	BiWidth	1000	65025	53249
CZJHMNU	J-Heisei Mincho Unicode	RM	BiWidth	1000	65025	53248
CZJHMGU	J-Heisei Maru Gothic Unicode	RM	BiWidth	1000	65024	53250
CZHKG2U	Korean Gothic Unicode	RM	BiWidth	1000	65026	53816
CZHSM2U	Korean Myengjo Unicode	RM	BiWidth	1000	65026	53560
CZSFSGU	S.Chinese Fang Song Unicode	RM	BiWidth	1000	65028	54566
CZSHEIU	S.Chinese Hei Unicode	RM	BiWidth	1000	65028	54565
CZSKAIU	S.Chinese Kai Unicode	RM	BiWidth	1000	65028	54568
CZSSNGU	S.Chinese Song Unicode	RM	BiWidth	1000	65027	54567
CZTKAIU	T.Chinese Kai Unicode	RM	BiWidth	1000	65029	54568
CZTSNGU	T.Chinese Sung Unicode	RM	BiWidth	1000	65029	54563

Combined Fonts (General Scripts + CJK)		TFace	Space	LineSp	GCSGID	FGID
CZH2JHKG	J-Kaku Gothic Helvetica Unicode	RM	TYPO	1070	65032	53249
CZN2JHMN	J-Mincho TNR Unicode	RM	TYPO	1070	65032	53248
CZH2JHMG	J-Maru Gothic Helvetica Unicode	RM	TYPO	1070	65031	53250
CZH2HKG2	Korean Gothic Helvetica Unicode	RM	TYPO	1070	65033	53816
CZN2HSM2	Korean Myengjo TNR Unicode	RM	TYPO	1070	65033	53560
CZN2SFSG	S.Chinese Fang Song TNR Unicode	RM	TYPO	1070	65035	54566
CZH2SHEI	S.Chinese Hei Helvetica Unicode	RM	TYPO	1070	65035	54565
CZN2SKAI	S.Chinese Kai TNR Unicode	RM	TYPO	1070	65035	54568
CZN2SSNG	S.Chinese Song TNR Unicode	RM	TYPO	1070	65034	54567
CZN2TKAI	T.Chinese Kai TNR Unicode	RM	TYPO	1070	65036	54568

```
CZN2TSNG  T.Chinese Sung TNR Unicode      RM  TYPO    1070  65036  54563
```

Choosing the Right Font

The font chosen to print Unicode text is dependent upon the source of the text and the target viewing audience, neither of which may be known when the document is printed or displayed. This means a certain amount of intelligent guessing will have to be done when selecting the font. Some of the choices are rather obvious. For example, if a document is created in Japan for a Japanese speaking audience, the obvious choice will be a Japanese font. A document created in France for a French speaking audience will use a General Scripts font.

There is a certain amount of commonality between the fonts in that all fonts contain some Latin characters, all CJK fonts support the Unified CJK Ideographs, and all CJK + General Scripts fonts contain support for 1267 General Scripts characters. With one exception, there is the possibility that no matter which language was used when the document was created, any font used will provide some of the selected characters. The exception is using a General Scripts font to render a CJK document that only contains CJK ideographs, in which case nothing will print because none of the source characters are in the target font.

There are five language groups: Japanese, Korean, Simplified Chinese, Traditional Chinese, and the General Scripts fonts (non-CJK). When the combined CJK + General Scripts fonts are considered, there are actually 9 different source and target categories. Text can be created assuming any of the nine and, of course, any of the nine can be used to present the text. Presentation results will vary depending on how well the source and target fonts match.

The following table shows on a 1 - 4 scale what can be expected from various source/target scenarios.

4 indicates the best possible match. The same language group was used for creation and presentation. All glyphs will be presented and the highest level of rendering will be achieved. For CJK fonts, non-ideographic glyphs will be those designed specifically for use with the CJK fonts in which they are found.

3 indicates good Unicode coverage, but loss of special non-ideographs. This would typically be the case where the source of the text is unknown, but is assumed to be one of the CJK languages and possibly contains non-Latin characters. The highest possible Unicode coverage is desired at the presentation end and, if not known, a guess is made as to which CJK language was used to create the text. The cost of expanded Unicode coverage is the loss of the non-CJK ideographs. This could also be a case where a CJK font was used to create the text and a CJK + General Scripts font is used to present the text.

2 indicates partial support. For non-CJK, it means that the source was created with a General Scripts font and is being presented with a CJK + General Scripts font. In this case, if the source General Scripts typeface happened to match the one in the CJK + General Scripts font, it would be a perfect match and the value would be 4. If the typefaces did not match, all the characters would print, but in the wrong typeface, thus a value of 2. In the case of CJK, it means one CJK language was used to create the document and another was used to present it. The common Unified CJK ideographs will print, but they will not be those designed for the language used by the creator of the document.

1 indicates minimal support. For non-CJK it means that the source text was created for General Scripts and presented with a CJK font. A few of the special non-ideographic characters found in the CJK fonts will be used, but the style will not match the source and a very limited set of characters will be available. For CJK it means that the source language and target language do not match and also that the target font is a CJK + General Scripts font. Not only will Unified CJK ideographs for another language be used, but also some of the special non-ideographs will have been replaced by General Scripts glyphs. If the source was CJK + General Scripts and the document was predominantly CJK, a General Scripts font used to present the text will contain few, if any, of the source characters, thus a value of 1.

	GS	J	K	CHS	CHT	J+GS	K+GS	CHS+GS	CHT+GS
Non-CJK	4	1	1	1	1	4/2	4/2	4/2	4/2
J	1	4	2	2	2	3	1	1	1
K	1	2	4	2	2	1	3	1	1
CHS	1	2	2	4	2	1	1	3	1
CHT	1	2	2	2	4	1	1	1	3
J+GS	4/2/1	3	2	2	2	4	1	1	1
K+GS	4/2/1	1	3	1	1	1	4	1	1
CHS+GS	4/2/1	1	1	3	1	1	1	4	1
CHT+GS	4/2/1	1	1	1	3	1	1	1	4

Print Scenarios

Similar in construct to existing AFP DBCS outline fonts, the AFP Unicode Migration fonts present few problems to existing AFP applications and printers capable of handling AFP DBCS outlines. Most of our testing was conducted in a production environment utilizing OS/390, AS/400, and AIX operating systems. The Infoprint 20 and Infoprint 60 printers were used for printing. Note that the Infoprint 20 (as well as Infoprint 32 and Infoprint 40) requires a microcode update in order to handle the code page associated with these fonts. *When the availability of this upgrade is available, it will be posted on this web page:* http://www.printers.ibm.com/R5PSC.NSF/Web/font.

When the fonts were tested, the biggest challenge was generating Unicode-encoded data. In order to verify that all characters could be printed, REXX EXECs were written to create Unicode text based on the contents of the fonts. The text was then treated as line data or put in overlays and sent to the printers.

The following AFP applications were used in printing Unicode text:

- ACIF - AFP Conversion and Indexing Facility (MVS)
- Infoprint Manager
- PPFA - Page Printer Formatting Aid (MVS)
- OGL - Overlay Generation Language (MVS; see “Limitations”)
- PSF - Print Services Facility (MVS, AS/400, and AIX)
- Infoprint Manager (AIX)

Here is a sample print scenario we used to print Unicode text. The font in this example is Times New Roman Unicode roman medium, CZN20U.

1. Unicode text created as hex line data and saved in N20UDATA.
2. A pagedef source file was created as follows (using font CZN20U):

```
pagedef n20u width 8 in height 11 in replace yes;
font f1 cs n20u cp 1200 dbcs height 10;
pageformat n20u lineone .5 in .3 in;
printline repeat 60
position .5 in .3 in
font f1;
```

3. A pagedef P1N20U was created by running PPFA against the pagedef source file.
4. ACIF was run against N20UDATA using the following parameters: ‘page=P1N20U, cc=n, restype=all’.
5. The output from ACIF was routed to the printer.

Printers

Generally speaking, in order to print the AFP Unicode Migration Fonts, the printer must have the following:

- The ability to accept downloaded AFP outline fonts
- DBCS capability
- Adequate memory

The following printers may be used to print the AFP Unicode Migration Fonts:

3130-002 with DBCS feature
3160-001 at microcode level 6.10 (or higher)
Infoprint 60 with DBCS feature
Infoprint 62
Infoprint 4000
Infoprint 3000

Infoprint 20 with DBCS feature at microcode level IPDS 5.01F, CTL code level 2.40F, 64MB memory recommended
Infoprint 32 with DBCS feature at microcode level IPDS 5.01F, CTL code level 2.50F, 64MB memory recommended
Infoprint 40 with DBCS feature at microcode level IPDS 5.01F, CTL code level 2.50F, 64MB memory recommended

Problems and Limitations

There were two problems and one limitation discovered when attempting to print Unicode-encoded data in the AFP environment.

1. Overlay Generation Language (OGL) could not process fonts with segmented font index (FNI) records. The combined CJK + General Scripts fonts have more characters than can be contained in a single-record FNI structured field. In order to handle the remaining characters, the first index record was created with the Segmented FNI flag set and the remaining characters were added to a second index record. This occurred for each of the four rotations so instead of four FNIs, these fonts have 8. Segmented FNIs are architecturally correct; however, this was the first time we had created fonts that used them. During development it was discovered that OGL could not process Segmented FNIs, and therefore could not process the CJK + General Scripts fonts. *A fix for this problem is in the works and when available, will be identified on this web page:* http://www.printers.ibm.com/R5PSC.NSF/Web/font.

2. The Infoprint 20, 32, and 40 printers need a microcode update in order to handle the Unicode code page used with these fonts. The error presented by the printer is IPDS Exception Check 713. *When a fix for this problem is available, it will be posted on this web page:* http://www.printers.ibm.com/R5PSC.NSF/Web/font.

3. Printing speed varies dramatically from printer to printer and by the content of the print job. As a general rule, jobs containing many unique characters, lots of characters only used once or twice in the job, will cause the most significant throughput degradation.

Summary

The AFP Unicode Migration Fonts provide access to 35,411 characters when used with Unicode-encoded text. Although use of these fonts in the AFP environment is limited at the moment, we believe the future will see the evolution of Unicode-encoded text in many areas. As time passes and Unicode requirements become better defined, this product or its follow-on will be adjusted to meet the Unicode presentation needs of the Advance Function Presentation community.

Trademarks

The following terms are trademarks or registered trademarks of the International Business Machines Corporation:

AFP	Advanced Function Presentation
ACIF	AFP Conversion and Indexing Facility
AIX	
AS/400	
IBM	
Infoprint	
Infoprint Manager	
MVS	
OGL	Overlay Generation Language
OS/390	
PPFA	Page Printer Formatting Aid
PSF	Print Services Facility

The following terms are trademarks of other companies.

- Boutros Typing is a trademark of Applied Arabic Limited
- Heisei Kaku Gothic, Heisei Maru Gothic, and Heisei Mincho are trademarks of Japanese Standards Association
- Helvetica is a trademark of Linotype-Hell Company.
- ITC, ITC Boutros Modern Rokaa, and ITC Boutros Setting are trademarks of International Typeface Corporation
- Narkiss Tam and Narkissim are licensed to IBM by Zvi Narkiss.
- Times New Roman is a trademark of Monotype Typography, Inc.
- Unicode is a trademark of Unicode, Inc.
- Windows NT and Windows 2000 are trademarks of Microsoft Corporation