



## Print Service Fonts

(i.e. who supplies the font); the character encoding identifier and the common typographic name of the font. This same convention is used in the Interpress master to specify the fonts to be used to render the document strings. Font name examples follow. **[Font name char encoding???**

Xerox OIS Classic

Xerox OIS Modern-Bold-Italic

Xerox XC82-0-0 Gacha

Xerox XC1-1-0 Autologic-Times-Roman

The font file names may contain periods (".") in place of the space characters for compatibility with other file systems. For the third identifier only, minus characters are preferred but periods are treated equivalently (for backward compatibility). As a second identifier, "OIS" refers to the earlier character encoding definition [3]; "XC82-0-0" refers to the Xerox Standard encoding [4] and "XC1-1-0" identifies the later standard [5].

*For the Print Service, a font file is exactly a PrePress CD dictionary file, with field variations defined herein. A CD file is a collection of AC files, both of which are defined in [1]. The format of the CD file is formally defined in Appendix A. Font file fields defined in Appendix A are used within the following text in **Modern-Bold** font.*

### Font file installation and identification

Font files are installed on the print server by either loading them from floppies (the standard, supported method) or by ethernet transfer via use of the Filing protocol and a coresident file service. In either case, they must be copied into the <SystemFiles>PS: Fonts> directory which is created upon Print Service initialization. The Print Service provides a command for loading fonts from floppies.

In loading files from floppies (which is how, for instance, test patterns are loaded), the print service distinguishes the font files from other files by means of their NSFile.Type which are defined in the PSFileTypes definitions module from the range [ServicesFileTypes.firstPSType.. ServicesFileTypes.lastPSType]. These same file types must be assigned to files installed via the net. The definitions for Print Service font files are repeated here for completeness.

<u>NSFile.Type assignment</u>	<u>Description</u>
4290	Raster, 300 spi, scan lines parallel to long edge of paper
4291	Raster, 200 spi, scan lines parallel to short edge of paper
4296	Autologic font descriptors and character mappings

On the print server, all font files are cataloged by taking the file names found in the font directory, replacing all sequences of space, greater-than or periods with the space character and caching the resultant identifier for use in the Interpress FINDFONT operation. **[We should use dashes/minuses instead!]** Fine point: the last identifier is the one actually compared; the second identifier determines whether the incoming Interpress master's character encoding should be translated before performing a character lookup.

In cataloging the font files, the Print Service notes the second identifier which defines the character encoding used in producing the font. If this second identifier is "OIS" and the second identifier of the font call-out in the Interpress master is, for instance, "XC82-0-0", then the Print Service software

## Print Service Fonts

uses the `NSStringConvert` interface to map the Interpress master's character encodings (in this case [4]) into the OIS encoding ([3]) before looking up the character in the font file.

### AC format variation

The AC component of the CD file for the Print Service requires one important variation from the standard PrePress AC file: character subsets definition. These are arranged as follows.

The various character subsets are stored in the font file along with the various font sizes and orientations. Each separate subset (that is, each separate high-order eight bit combination) is a separate AC. To distinguish amongst the subsets with the same size and orientation, we need some info in the AC font segment.

Fortunately, there are two words in the AC font header that are not needed at the printer. These are the words that define the device resolution in x and y (units are pixels per ten inches). They are redundant because, at the print server, we can assume that font files will be distinguished by their `NSFileType`, as defined above.

*The character subset is defined by overwriting the "Y-resol" (resolutionB) word in the AC header. This word is overwritten with the value (character subset \* 10). (Why times ten? Because PrePress asks you for the resolution in pixels-per-inch and then multiplies by ten to get pixels-per-ten-inches, and the Prepress menu doesn't like to accept fractions. Earlier definitions allowed a Y-resol value greater than 2550 to indicate subset zero, letting the normal Y-resol = 3000 stand for subset zero, but the current requirement is to have CS0 be zero so that there would be no conflict with lower resolution fonts.)*

This variation is defined in Appendix B.

When the font file is first installed on the Print Service, the cataloging software determines the *area* required for the raster bitmap of each printable character and stores it into either `CharacterData.wx.fraction` (for non-rotated fonts) or `.wy.fraction` (rotated).

### CD format variation for phototypesetter fonts

The Print Service requires CD-like files for the APS phototypesetter. *The CD files required do not contain raster masks for the fonts but, instead, contain the APS-specific scaling factor, font numbers and character code required to identify the equivalent APS character.* Also, these CD font files contain *character width metrics in micas*, instead of rasters and scan-lines.

The APS phototypesetter has the character bitmaps loaded directly into its system and these are referenced with a font number (in the range of 1 through 32577), character code (in the range of 1 through 128) and scaling (in tenths of a point). Each APS font is provided in several ranges, each of which has a unique number and represents a single master size from which smaller sizes can be obtained by optical scaling. Since the resolution of the APS 57 pica and 70 pica models are different and thus have different effective master font sizes, two entries are provided in the APS's CD file for the font number allowing reference to different font ranges where required. (Specifically, the 57 pica model has effective master sizes of 10, 20, 40, 80 and 160 points and the 70 pica model has master sizes of 12, 24, 48, 96 and 192. A 12 pt character would be scaled from the 20 pt font on the /57 but could use the 12 pt master on the /70.)

This version of the CD file uses the variant required for raster fonts to distinguish character subsets, as described above, as well as providing the character mapping from the Xerox encoding (of characters in the Interpress file) to equivalent characters on the APS, using the portion of the CD file which would otherwise contain the raster mask (bits) to provide the point size, font numbers and character code. Thus, the "raster" definition for all characters which can be printed will have a 5 word entry in place of the rasters: 6001B (representing 4 words\*1 scan-

## Print Service Fonts

line), the size of the character in 10ths of a point in the second word, the APS font number to use in the 57 pica (723 spi) model in the third word, the font number to use in the 70 pica (600 spi) model in the fourth word and the APS character code in the fifth. The traditional CD conventions apply for character entries which have no printable symbol.

To provide APS-equivalent character metrics, all **CharacterData** metrics will be encoded in micas (10<sup>-5</sup> meters).

These variations are formally defined in Appendix C.

### Raster font file creation

To create a simple raster font file for "Xerox OIS Timesroman", one might:

Obtain, via Prepress spline conversion or whatever, a "Timesroman12.ac" "Timesroman10.ac" "Timesroman8.ac" and "Timesroman6.ac" *for each character subset that is of interest.*

If interested in landscape printing, make a rotated version of each with Prepress.

Using the PrePress "rename" command, alter the "Y resol" word in each font by supplying a "Y resol" equal to the decimal subset number.

Merge all the AC files into one dictionary named "Xerox OIS Timesroman".

Some minor arrangements within the PrePress ".cd" file:

The 8044's software uses the "bbdx" and "bbdy" to calculate the area of the character's raster. The actual raster dimensions must match those actual character bounding box dimensions. Running PrePress in "verify" will ensure this.

It is desirable that the character rasters be stored in ascending order. Doing the Prepress "merge" in "slow mode" will ensure this.

## APPENDIX A. CD file format definitions

The following is an attempt to formally define the construct of a CD font file.

```
IX: TYPE = MACHINE DEPENDENT RECORD |
    type(0:0..3): [0..17B],
    length(0:4..15): [0..7777B] -- length of entry in words, including this one
    |;
```

ix0: IX ← [type: 0, length: 1]; -- end of indexes

ix1: IX ← [type: 1, length: 12]; -- for IXN

ix3: IX ← [type: 3, length: 11]; -- for CharIndexEntry

```
IXN: TYPE = MACHINE DEPENDENT RECORD |
    ix: IX ← ix1,
    code: CARDINAL ← 0, -- the numeric code for this name
    name: PACKED ARRAY [0..20] OF Byte -- 1 byte of len, 19 bytes of char
    |; -- establishes correspondence between code and name string
```

```
STDIX: TYPE = MACHINE DEPENDENT RECORD |
    ix: IX, -- header with various types
    family: Byte, -- Family name, using name code (see IXN)
    face: Byte, -- Encoded face
    bc: CHARACTER, -- Beginning character in this file
    ec: CHARACTER, -- Ending character in this file
    size: CARDINAL, -- Font size in micras of segment pointed to by segAddr
    rotation: CARDINAL, -- Rotation of the font segment
    segAddr: LongCardinal, -- relative addr of character segment in file
    segLength: LongCardinal -- Length of the character segment
    |;
```

```
CharIndexEntry: TYPE = MACHINE DEPENDENT RECORD |
    stdix: STDIX ← [ix:[ix3]],
    resolutionX: CARDINAL, -- Resolution in scan-lines/inch * 10
    resolutionY: CARDINAL -- Resolution in bits/inch * 10
    |;
```

CharSubsetRange: TYPE = {0..0}; -- where range is determined  
-- ... by the total number of character subsets, sizes and rotations

CharIndex: TYPE = CharSubsetRange OF CharIndexEntry;

```
CharData: TYPE = MACHINE DEPENDENT RECORD |
    wx: Fraction, -- X width in scan lines (non-rotated)
    wy: Fraction, -- Y width in bits
    bbox: INTEGER, -- bounding box offsets from origin
    bboy: INTEGER, -- ...to bottom left of bounding box
    bbdx: INTEGER, -- width of bounding box (scan-lines)
    bbdy: INTEGER -- height of bounding box (bits) or -1 if no character
    |;
```

Print Service Fonts  
APPENDIX A. CID file format definitions

**CharArrayRange:** TYPE = [0..0]; -- where range is defined as [0..ec-bc) from CharIndexEntry

**CharMetrics:** TYPE = ARRAY CharArrayRange OF CharData;

**Directory:** TYPE = ARRAY CharArrayRange OF LongCardinal; -- relative offsets to Rasters or  
-- ... minus one (in both words) if there are no rasters

**RasterHdr:** TYPE = MACHINE DEPENDENT RECORD |  
    bbdyWords (0: 0..5): [0..77B],  
    bbdxBits (0: 6..15): [0..1777B]  
];

**Rasters:** TYPE = RECORD |  
    rasterHdr: RasterHdr,  
    rasters: ARRAY [0..0) OF CARDINAL -- the actual rasters, words = bbdyWords \* bbdxBits  
];

**CharSegment:** TYPE = RECORD | -- for one char subset, size and/or rotation  
    charMetrics: CharMetrics,  
    directory: Directory,  
    rasters: ARRAY [0..0) OF Rasters  
];

**Byte:** TYPE = [0..255];

**LongCardinal:** TYPE = MACHINE DEPENDENT RECORD |  
    high: CARDINAL,  
    low: CARDINAL]; -- NOTE: different from Mesa's LONG CARDINAL

**Fraction:** TYPE = MACHINE DEPENDENT RECORD |  
    bitWidth: INTEGER; -- Integer part  
    fraction: CARDINAL; -- fraction part

-- CD file format...

**cdFile:** RECORD |  
    ixn: **IXN**,  
    charIndex: CharIndex,  
    ix0: IX ← ix0,  
    charSubsetSegment: CharSubsetRange OF CharSegment  
];

## APPENDIX B. CD format for Raster fonts

The following is the principal variation required by the Print Service for raster fonts from the CD format definitions given in Appendix A. The principal difference is the CharIndexEntry.charSubsetx10 field in place of resolutionY.

```
CharIndexEntry: TYPE = MACHINE DEPENDENT RECORD |
  stdix: STDIX ← [ix:[ix3]],
  resolutionX: CARDINAL, -- Resolution in scan-lines/inch * 10
  charSubsetx10: CARDINAL -- characterSubSet*10
};
```

```
CharData: TYPE = MACHINE DEPENDENT RECORD |
  wx: Fraction, -- (the fraction of one of these will contain the raster area
  wy: Fraction, -- ... after being processed at the Print Service)
  bbox: INTEGER, -- bounding box offsets from origin
  bboy: INTEGER, -- ...to bottom left of bounding box
  bbdx: INTEGER, -- width of bounding box (scan-lines)
  bbdy: INTEGER -- height of bounding box (bits) or -1 if no character
};
```

## APPENDIX C. CD format for Autologic phototypesetter fonts

The following is the principal variation required by the Print Service for Autologic phototypesetter fonts from the CD format definitions given in Appendix A. The difference is the **CharIndexEntry.charSubsetx10** field in place of **resolutionY** (as required for raster fonts), the **CharData** in micas and the definition of **CharMapping** instead of **Rasters**.

**CharIndexEntry:** TYPE = MACHINE DEPENDENT RECORD |  
  stdix: **STDIX** ← [ix:[ix3]],  
  resolutionX: CARDINAL, -- *not used*  
  charSubsetx10: CARDINAL -- *characterSubSet\*10*  
|;

**CharData:** TYPE = MACHINE DEPENDENT RECORD |  
  wx: **Fraction**, -- *X width in micas*  
  wy: **Fraction**, -- *Y width in micas*  
  bbox: INTEGER, -- *bounding box offsets from origin, in micas*  
  bboy: INTEGER, -- *...to bottom left of bounding box, in micas*  
  bbdx: INTEGER, -- *width of bounding box, in micas*  
  bbdy: INTEGER -- *height of bounding box (bits) or -1 if no character*  
|;

**APSHdr:** TYPE = MACHINE DEPENDENT RECORD |  
  bbdyWords (0: 0..5): [0..77B] ← 4,  
  bbdxBits (0: 6..15): [0..1777B] ← 1  
|;

**CharMapping:** TYPE = RECORD |  
  rasterHdr: **APSHdr**,  
  pointSizex10: CARDINAL, -- *char size in 10th of points*  
  font57: CARDINAL, -- *font number on 57 pica (723 spi) model, in range [1..32755]*  
  font70: CARDINAL, -- *font number on 70 pica (600 spi) model*  
  charCode: CARDINAL[0..256] -- *actually, codes will be in [1..128]*  
|;

**CharSegment:** TYPE = RECORD | -- *for one char subset, size and/or rotation*  
  charMetrics: **CharMetrics**,  
  directory: **Directory**,  
  charMap: ARRAY [0..0] OF **CharMapping**  
|;



## APPENDIX D. Miscellaneous font information

### Handy constants

1 inch = 2.54 cm = 72.29 points = 6.024 picas

1 mica =  $10^{-5}$  meters = 1000 cm

10 pt = 0.13833 in = 0.351 cm

12 pt = 1 pica = 0.166 in = 0.422 cm

#### *300 spot per inch constants*

1 spot = 0.003333 in = 0.0084667 cm = 8.4667 micas

1 pt = 4.15 spots

1 cm = 118.11 spots

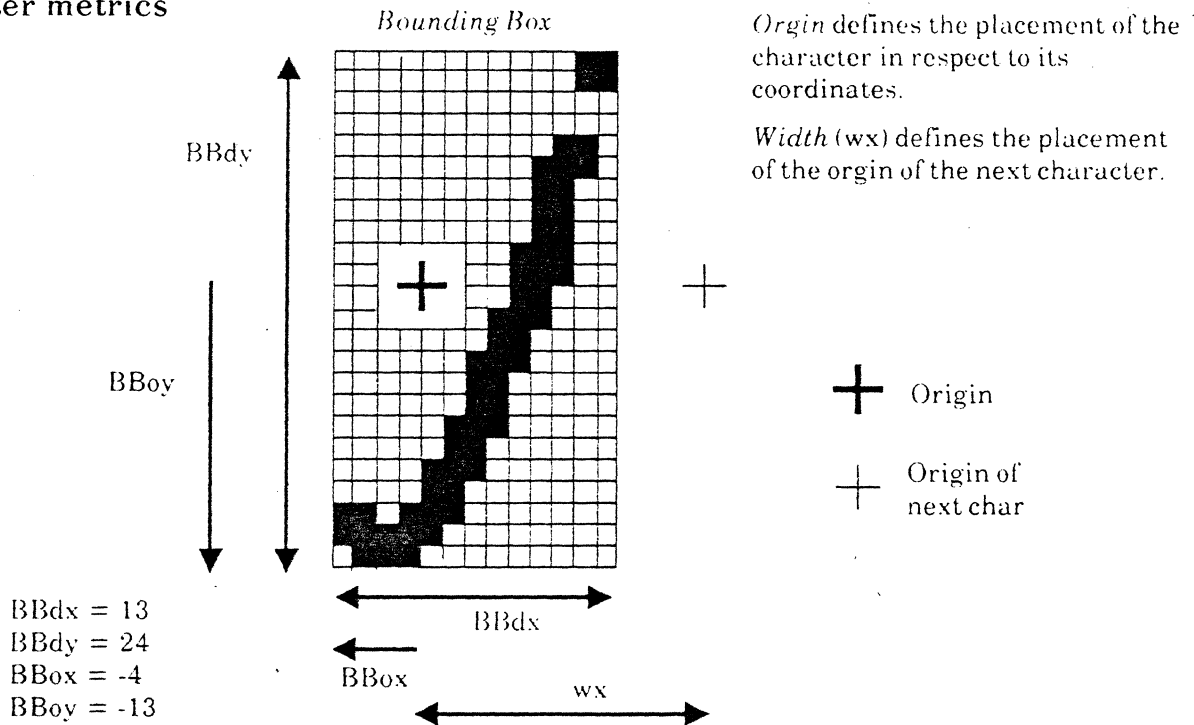
#### *200 spot per inch constants*

1 spot = 0.005 in = 0.0127 cm

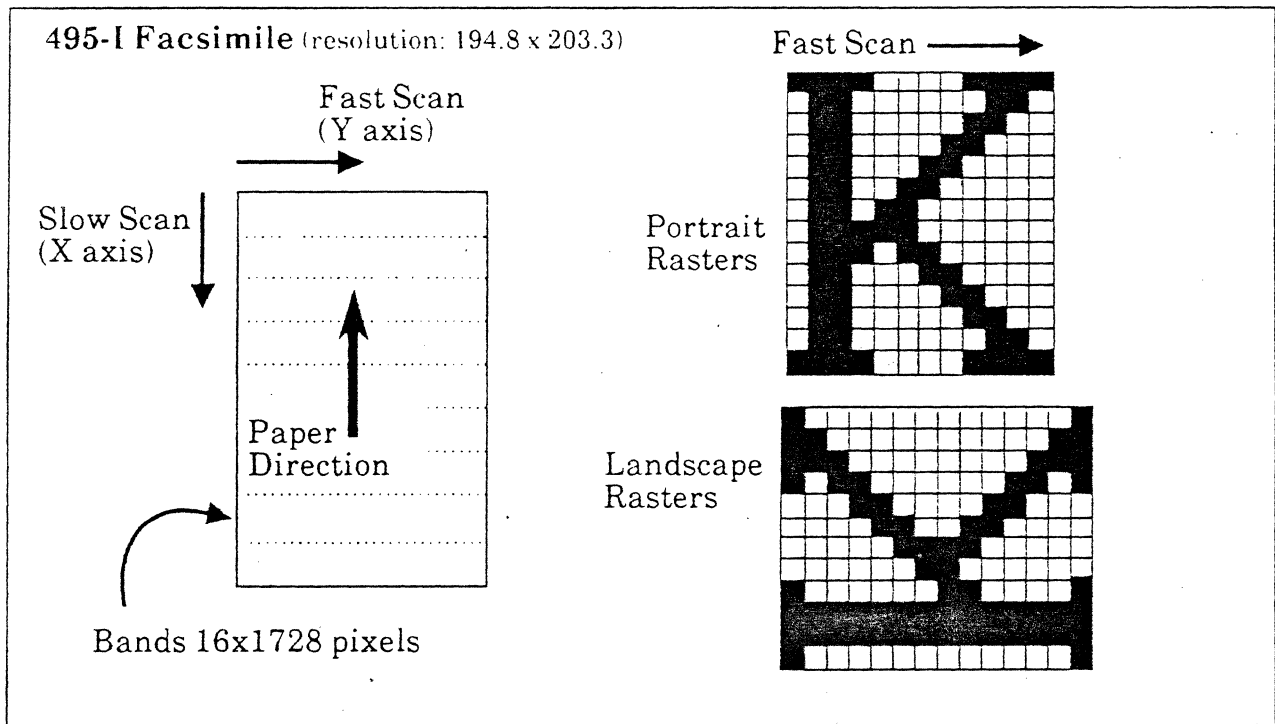
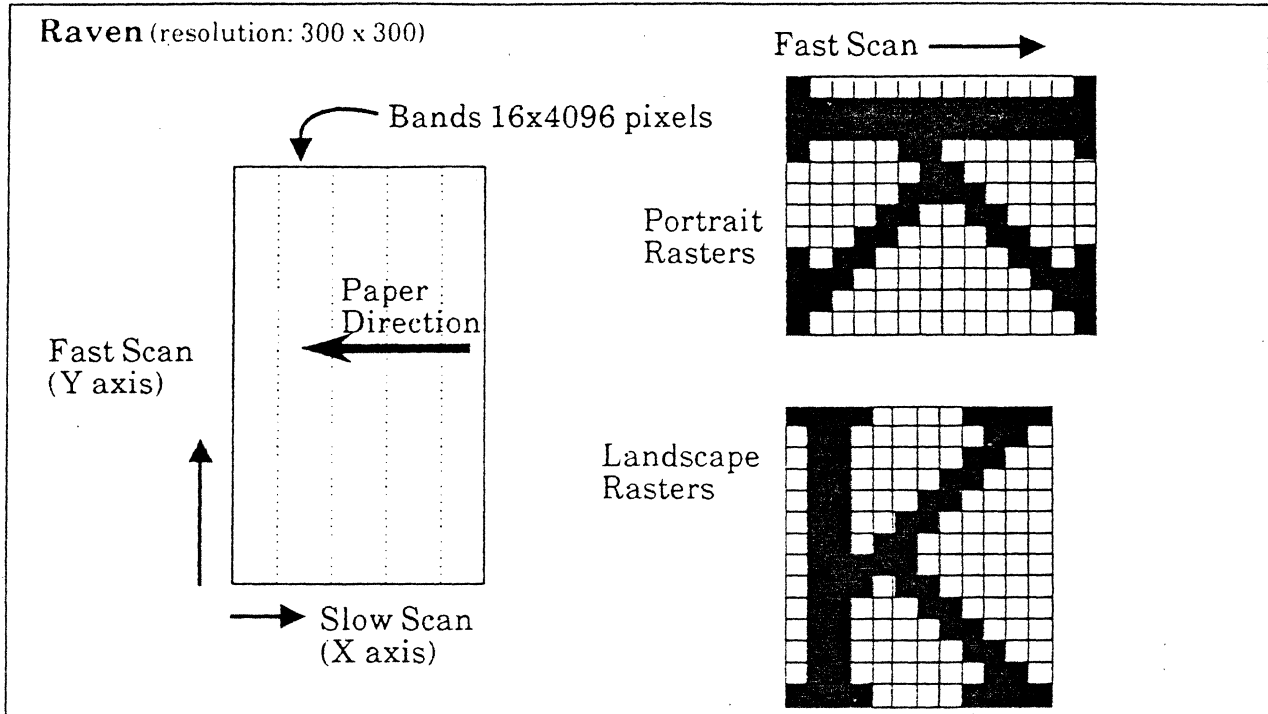
1 pt = 2.7666 spots

1 cm = 78.74 spots

### Raster metrics



### Raster Orientation



## APPENDIX E. Obsolete Strikeout variation

The initial definition of the CD font files provided for synthetic strikeout and underline capabilities, but this has never been used by clients and is no longer supported. The following definition and notes are retained for reference only.

Information on synthetic strikeout and underline capabilities may be stored in the *X-resol* (*resolutionS*) word in the AC segment header. The X-resol word is to be encoded as

```
IF (syntheticUnderline THEN 10000 ELSE 0)
+ (delta-in-pixels from baseline to bottom of strikeout)*100
+ (thickness of strikeout in pixels)*10
```

This allows for the strikeout to be up to 99/300 inch above the baseline (it must be above, not below) and up to nine pixels thick.

The absence of synthetic strikeout capability is indicated via a thickness of zero.

Conveniently, in the absence of font synthesis, one can leave the original X-resol of 3000 alone, as it indicates the absence of both underline and strikeout.

The metrics (thickness and distance from baseline) that apply to synthetic underline (if  $\text{resolutionS} \geq 10000$ ) are taken from the underline character [00314<sub>g</sub> = 204<sub>10</sub>] that matches this .ac font in size and orientation. Such a character must be present in the font file if synthetic underline is utilized.

If you wanted to implement "Xerox OIS Timesroman Strikeout" (and/or underline) synthetically -- that is without creating a true AC file of little pictures for the struckout letters -- you would

When using Prepress 'rename' to fix up the "Y resol" words for the character subsets, alter the "X resol" to match the formula for synthetic strikeout and underline given earlier.

After storing the file "<volume><sysdir>xerox>ois>timesroman.cd", create another directory entry named "<<volumesysdir>xerox>ois>timesroman>strikeout.cd" which points to the same file. [A fine point: this requires that the file not contain a Pilot leader page.] (The interim Press printing software looks at the trailing fields in the filename to see if it should actually use the information on the synthesizable strikeout and underline.)

Other notes:

For correct synthetic underlining, the bounding box information for the underline character (204<sub>10</sub>) must define the actual thickness of the line -- no white space within the supposed bounding box.

Synthetic strikeout ink is only supplied within the character's bounding box. This means that "strikeout"s of consecutive characters typically do not quite meet, and that high and low punctuation (e.g. ".") and blanks are not struck out. I claim that this is desirable behavior: it can be altered by including white space within the bounding box, but this may be quite hard to actually accomplish with PrePress.