FPML - filePro Markup Language (v0.1)
==========

FPML is an "XML-like" markup language for filePro PDF output. Printer operations are defined using:

<TAG ATTRIB="value" ATTRIB="value" ...>

Unless otherwise noted, everything is case-insensitive. For example, these are equivalent:

X="CENTER"
X="Center"
x="center"

Values can be enclosed in single- or double-quotes:

FILE="foo.jpg"
file='foo.jpg'

Unknown tags are silently ignored.

========== NUL

<NUL [...] >

A no-op. This does nothing, and any attribute/values that are given are ignored.

========== PAGE

<PAGE ([SIZE="size"] [ORIENT="orientation"]) | (HEIGHT="height" WIDTH="width") >

Sets the page size and orientation. Note that, if data has already been sent to the current page, this will implicitly start a new page.

Size can be:

LETTER
LEGAL
A3
A4
A5
B4
B5
EXECUTIVE
4x6
4x8
5x7

Orientation can be:

LANDSCAPE
PORTRAIT

Alternatively, an explicit height and width can be specified.

The default is LETTER, PORTRAIT. This setting will be retained until explicitly changed.

Margins are currently fixed at 1/4 inch.

---------- Future enhancements

Specify margins.

Specify "page units" (defined elsewhere).

========== IMAGE

<IMAGE FILE="filename" [ROTATE="rotation"] [HEIGHT="height"] [WIDTH="width"] [X="x-pos"] [Y="y-pos"] >

Places an image on the document.

There is no way to explicitly set the Z-order of images. The image will be "above" any text that comes earlier on the page, and "under" any text that comes later on the page.

If no explicit path is given to the filename, filePro will search the following, in this order:

The main filePro file's default directory.

PFIMAGEDIR

PFDLDIR

The filename is case-sensitive if the underlying O/S uses case-sensitive filenames. (ie: Unix/Linux are case-sensitive, while Windows is not.)

The rotation is specified in degrees, counter-clockwise.

Height and width are specified in "page units" (to be described elsewhere). If you specify "scale", then that direction will scale proportionately, based on the size given the other direction. For example, if the image has

a height of 75, then specifying:

HEIGHT="150" WIDTH="scale"

will double the width as well. Specifying one direction and not the other will leave the other unchanged.

The position on the page (default: the current position) is done with the X/Y attributes. These can be a number, specified in "page units" (described elsewhere), or one of "LEFT"/"CENTER"/"RIGHT" for "X", and "TOP"/"CENTER"/"BOTTOM" for "Y".

When specifying an explicit location, the origin (0,0) is the top-left of the page (inside the margins), with X increasing to the right and Y increasing downwards.

Note that there is no resampling of the image. The image is printed at full resolution, just scaled to the specified size.

---------- Future enhancements

Specify relative positions. (ie: an offset from the current position.)

Scaling of images as a percentage. This could be either a percentage of the original image size, or as a percentage of the page size. (Or, even better, some way of specifying which method, on a per-image basis.)

Specifying size by other units, such as "3 inches".

Downsampling of large images when shown smaller, or when printed in a lower resolution.

========== FONT

<FONT [NAME="fontname"] [SIZE="size"] [BOLD="on|off"] [ENCODE="encoding"]
[UNDERLINE="on|off"] [COLOR="color"]
>

Sets the font as specified.

Currently, only the PDF built-in "base14" fonts are allowed. These font names are not-case sensitive:

"Courier" (plain, bold, italic, bold-italic)
"Helvetica" (plain, bold, italic, bold-italic)
"Times" (plain, bold, italic, bold-italic)

"Symbol"
"ZapfDingbats"

Size is specified in points.

"Encoding" specifies the character set encoding for 128-255, and is one of the following:

http://libharu.org/wiki/Documentation/Encodings

StandardEncoding	It is the default encoding of PDF
MacRomanEncoding	The standard encoding of Mac OS
WinAnsiEncoding	The standard encoding of Windows
FontSpecific	Use the built-in encoding of a font.
ISO8859-2	Latin Alphabet No.2
ISO8859-3	Latin Alphabet No.3
ISO8859-4	Latin Alphabet No.4
ISO8859-5	Latin Cyrillic Alphabet
ISO8859-6	Latin Arabic Alphabet
ISO8859-7	Latin Greek Alphabet
ISO8859-8	Latin Hebrew Alphabet
ISO8859-9	Latin Alphabet No. 5
ISO8859-10	Latin Alphabet No. 6
ISO8859-11	Thai, TIS 620-2569 character set
ISO8859-13	Latin Alphabet No. 7
ISO8859-14	Latin Alphabet No. 8
ISO8859-15	Latin Alphabet No. 9
ISO8859-16	Latin Alphabet No. 10
CP1250	Microsoft Windows Codepage 1250 (EE)
CP1251	Microsoft Windows Codepage 1251 (Cyrl)
CP1252	Microsoft Windows Codepage 1252 (ANSI)
CP1253	Microsoft Windows Codepage 1253 (Greek)
CP1254	Microsoft Windows Codepage 1254 (Turk)
CP1255	Microsoft Windows Codepage 1255 (Hebr)
CP1256	Microsoft Windows Codepage 1256 (Arab)
CP1257	Microsoft Windows Codepage 1257 (BaltRim)
CP1258	Microsoft Windows Codepage 1258 (Viet)
KOI8-R	Russian Net Character Set

"Color" specifies the color. Currently, the only method of specifying the color is "#rrggbb" where "rr", "gg", and "bb" are two-digit hex values for red/green/blue, respectively.

The default is Courier 10, black, with "standard" encoding.

Note that specifying a font name will clear bold/italic/underline, unless explicitly set to "on" in the same code.

Note that PDF does not directly support underlined text. Instead, this

is accomplished by drawing a line underneath the text.

---------- Future enhancements

Text rendering mode.

http://libharu.org/wiki/Documentation/API/Graphics#HPDF_Page_SetTextRenderingMode.28.29

Some way of getting the line-drawing characters. I have checked all of the encodings for Courier, and I didn't see any of them available. It looks like they can be done using Unicode, but I don't know if there is a way to say "these characters are specified in Unicode" when the rest of the text isn't.

The location and thickness of the underline should be adjusted, based on the size and "boldness" of the current font.

========== MOVETO

<MOVETO [X="x-pos"] [Y="y-pos] >

Move the current text position as specified.

If one coordinate is not specified, the position on that axis remains unchanged.

Units can be absolute or relative. (If the value starts with "+" or "-", it is relative to the current position.)

---------- Future enhancements

Currently, the position is in "page units" (described elsewhere). A future enhancement would be the ability to specify other units, such as "em".

========== RECT

<RECT [X="x-pos"] [Y="y-pos"] WIDTH="width" HEIGHT="height"
[STROKE="line-width"] [COLOR="color"]
>

Draw a rectangle. If (X,Y) is not specified, the current position is used.

Note that a height of 0 draws a horizontal line, and a width of 0 draws a vertical line.

Defaults:
Color = black

Stroke = 1.0

========== FPML output

If you set the new "allow embedded FPML on form" option in dmoedef, you can include FPML codes directly on the output format.

You can also generate "on-the-fly" FPML by putting the FPML in a field on the form, prefaced with ESC. For example:

```
   Then: color = "black"

     If: value lt "0"
   Then: color = "red"

   Then: xx = chr("27") & "<FONT COLOR='" { color { "'>" { value
            { chr("27") & "<FONT COLOR='black'>"
```

Finally, note that you can put FPML directly on the form, and still fill in values at runtime, by putting something like this on the form:

```
   <FONT COLOR='*aa      '>*bb       <FONT COLOR='black'>
```

========== Notes

"Page units" currently means 1/72 of an inch. (The traditional typography "point".)