Running the executeable

the following command(s) will convert a pml file to pdf

```
./pml2pdf.sh --cli --infile /path/to/file.pml --outfile /path/to/file.pdf

./pml2pdf.cmd --cli --infile /path/to/file.pml --outfile /path/to/file.pdf
```

Prerequisites

- Java Runtime, Version 8 or higher
- properly setup JAVA_HOME environment variable

Linux Recommendations

for Linux it is recommanded to install java privately using SDKMAN [https://sdkman.io/]

Page Markdown Layout (PML)

Basic Document Structure

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <?...instruction... ?>
  <page ...attributes...>
    ...page level objects...
  </page>
</document>
```

Attribute Types

Boolean

usually boolean can take the following values "true|false|on|off|yes|no" and are specified like below:

```
"_boolean_"
```

Factors ie. Percentages

usually Factors are integers in the rangs of 0 ... 100, sometimes -1 means diable feature and are specified like below:

```
"_factor_"
```

File/Path

usually Files/Paths must be actully exist on the filesystem and are specified like below:

```
"_file_"
"_path_"

Examples:

"file:///tmp/some/path/to/file.ext"
"./some/path/to/file.ext"
"../some/path/to/file.ext"
"/some/path/to/file.ext"

"./some/path"
"../some/path"
"/some/path"
```

Resources

usually Resource can be Files or references to mounts and the classpath and are specified like below:

```
"_resource_"

Examples:

"mnt:/tmp/some/path/to/file.ext"
"mnt:tmp/some/path/to/file.ext"
"zip:/tmp/some/path/to/file.ext"
"zip:tmp/some/path/to/file.ext"
"cp:tmp/some/path/to/file.ext"
```

Color

- (string) a html named color (https://en.wikipedia.org/wiki/Web_colors#HTML_color_names)
- (string) a X11 named color (https://en.wikipedia.org/wiki/Web_colors#X11_color_names)
- Material Design Flat UI Colors
    - md-flat-ui-turquoise = #1ABC9C
    - md-flat-ui-emerland = #2ECC71
    - md-flat-ui-peterriver = #3498db
    - md-flat-ui-amethyst = #9b59b6
    - md-flat-ui-wetasphalt = #34495e
    - md-flat-ui-greensea = #16a085
    - md-flat-ui-nephritis = #27ae60
    - md-flat-ui-belizehole = #2980b9
    - md-flat-ui-wisteria = #8e44ad
    - md-flat-ui-midnightblue = #2c3e50
    - md-flat-ui-sunflower = #f1c40f
    - md-flat-ui-carrot = #e67e22
    - md-flat-ui-alizarin = #e74c3c
    - md-flat-ui-clouds = #ecf0f1
    - md-flat-ui-concrete = #95a5a6
    - md-flat-ui-orange = #f39c12
    - md-flat-ui-pumpkin = #d35400
    - md-flat-ui-pomegranate = #c0392b
    - md-flat-ui-silver = #bdc3c7
    - md-flat-ui-asbestos = #7f8c8d
- (string) a hex-triplet (RGB) prefixed with '#' eg. '#ff0000' aka red (https://en.wikipedia.org/wiki/Web_colors#Hex_triplet)
- (string) a shorthand hex-triplet (RGB) prefixed with '#' eg. '#f00' aka red (https://en.wikipedia.org/wiki/Web_colors#Shorthand_hexadecimal_form)
- (string) a CSS3 rgb-function eg. 'rgb(255,0,0)' or 'rgb(100%,0%,0%)' aka red.
- (string) a hex-quadlet (CMYK) prefixed with '#' eg. '#000000ff' aka black.
- (string) a shorthand hex-quadlet (CMYK) prefixed with '#' eg. '#000f' aka black.
- (string) a cmyk-function eg. 'cmyk(0,0,0,255)' or 'cmyk(0%,0%,0%,100%)' aka black.
- HSL-Function Colors --- eg. "hsl(360,1.0,0.5)" or "hsl(360,50%,25%)"
- HSV-Function Colors --- eg. "hsv(360,1.0,0.5)" or "hsv(360,50%,25%)"
- HWB-Function Colors --- eg. "hwb(360,1.0,0.5)" or "hwb(360,50%,25%)"
- HCG-Function Colors --- eg. "hcg(360,1.0,0.5)" or "hcg(360,50%,25%)"
- XYZ-Function Colors --- eg. "xyz(0,1.0,-0.5)"
- L*a*b-Function Colors --- eg. "lab(100,100,-50)"
- Lch-Function Colors --- eg. "lch(100,100,360)"

Document Level Elements

Instructions

Options

JPEG Compression

specifies the jpeg compression factor in percent (1 = max compression, min quality; 100 = min compression, max quality)

```
<?option jpeg-compression="_factor_" ?>
```

Attribute lookup debug

```
<?option print-attribute-access="_boolean_" ?>
```

include

include the specified file for document processing.

```
<?include src="_file_" ?>
```

info

set pdf document info attributes

```
<?info
  author="_text_"
  creator="_text_"
  producer="_text_"
  subject="_text_"
  keywords="_text_"
  title="_text_"  ?>
```

mount

adds zip files or directories to the global search path.

```
<?mount src="_file_" ?>
<?mount dir="_path_" ?>

Examples:

<?mount src="/path/to/file.zip" ?>

<?mount dir="/some/path" ?>
```

define

defines values for attribute lookup

```
<?define key="_value_" ?>
```

defaults

like define but loads from a properties file

```
<?defaults src="_resource_" ?>

Examples:

<?defaults src="/path/to/attr.properties" ?>
<?defaults src="./path/to/attr.properties" ?>
<?defaults src="../path/to/attr.properties" ?>
<?defaults src="cp:path/to/attr.properties" ?>
<?defaults src="mnt:path/to/attr.properties" ?>
<?defaults src="zip:path/to/attr.properties" ?>
```

font

loads a font and defines id for referencing in attributes

```
<?font id="_id_"
  name|src="_resource_|_ref_"
  charset|encoding="_charset_"
  icon-map="_resource_" ?>
```

the following special references can be used as resources:

- "pdf:*name*" --- one of the base 14 pdf font names:
  - helvetica, helvetica-bold, helvetica-boldoblique, helvetica-oblique
  - courier, courier-bold, courier-oblique, courier-boldoblique
  - times-roman, times-bold, times-bolditalic, times-italic
  - symbol
  - zapf-dingbats
- "awt:*name*" --- a registered name of a awt font resource linked to a system or user installed font

the following character maps are defined (default: pdfdoc):

- adobe-standard, adobe-symbol, adobe-zapf-dingbats
- cp1250, cp1251, cp1252, cp1253, cp1254, cp1255, cp1256, cp1257, cp1258
- hp-roman8
- ibm437, ibm850, ibm851, ibm852, ibm855, ibm857
- iso-8859-1, iso-8859-2, iso-8859-3, iso-8859-4, iso-8859-5, iso-8859-6, iso-8859-7, iso-8859-8, iso-8859-9, iso-8859-13, iso-8859-15
- koi8-r, koi8-ru, koi8-u (cyrillic)
- macintosh (mac roman)
- microsoft-dingbats (0xF0xx)
- pdfdoc (pdf doc encoding extension to latin1)
- texnansi (TeX ansi encoding extension to latin1)
- text (groff ascii encoding extension to latin1)

the following aliases for encodings are supported:

- hp-roman, hproman
- mac-roman, macroman
- latin1 ... latin9, latin-1 ... latin-9
- iso-cyrillic, iso-arabic, iso-greek, iso-hebrew, iso-thai, iso-celtic
- ms-dingbats

the following special encoding can be used:

- unicode, allows ttf/otf fonts to use the entire 16-bit unicode glyph-space
- icons, allows ttf/otf based icon fonts for entity-lookup (see below)

the attribute "icon-map" allows loading a icon-entity specifier map

entity/icon

allows defining ids for referencing entities/icons

```
<?entity name="_name_" code="_code_" ?>
<?icon font="_ref_" name="_name_" code="_code_" ?>

Examples:

<?entity name="non-breaking-space" code="0xA0" ?>

<?icon font="fa" name="refresh" code="0xf123" ?>
```

class

allows specifying predefined classes of attribute sets for referencing

```
<?class id="_id_" key1="_value1_" ... keyN="_valueN_" ?>

Examples:

<?class id="a4-portrait" page-mediabox="595.842" ?>
<?class id="us-letter" page-mediabox="612,792" ?>
<?class id="a4-landscape" page-mediabox="842,595" ?>
```

image

```
<?image id="_id_"
  src="_resource_"
  compress="jpg|jpeg|index|indexed|mono|grey|dct-gray"
  transparency="alpha|_boolean_"
  alpha="_factor_" ?>
```

svg

```
<?svg id="_id_"
  src="_resource_"
  render="_factor_"
  compress="jpg|jpeg|index|indexed|mono|grey|dct-gray"
  transparency="alpha|_boolean_"
  alpha="_factor_" ?>
```

header/footer

```
<?header* ... ?>

<?footer* ... ?>
```

page element

```
<page mediabox="_bbx_" rotate="_rot_">
  ... page level elements ...
</page>
```

Page Level Elements

image/svg

```
<svg id="_id_" ref="_id_" pos="_x_,_y_" width="_pts_" height="_pts_" background="_boolean_" />

<svg id="_id_" pos="_x_,_y_" width="_pts_" height="_pts_" background="_boolean_"
  src="_resource_"
  render="_factor_"
  compress="jpg|jpeg|index|indexed|mono|grey|dct-gray"
  transparency="alpha|_boolean_"
  alpha="_factor_" />
```

```
<image id="_id_" ref="_id_" pos="_x_,_y_" width="_pts_" height="_pts_" background="_boolean_" />

<image id="_id_" pos="_x_,_y_" width="_pts_" height="_pts_" background="_boolean_"
  src="_resource_"
  compress="jpg|jpeg|index|indexed|mono|grey|dct-gray"
  transparency="alpha|_boolean_"
  alpha="_factor_" />
```

h/heading

layouts heading(s) at position and creates outlines for the pdf outline tree (see outline/chapter/section)

```
<heading id="_id_" pos="_x_,_y_" level="_level_" ...attributes... >
...heading text...
</heading>

<h id="_id_" pos="_x_,_y_" level="_level_" outline="_text_" ...attributes... >
...heading text...
</h>
```

text/label

layouts simple text(s)/label(s) at position

```
<label id="_id_" link="_id_" pos="_x_,_y_" ...attributes... >
...simple label text...
</label>

<text id="_id_" link="_id_" pos="_x_,_y_" ...attributes... ><![CDATA[
...simple text line 1...
...simple text line 2...
...simple text line 3...
]]></text>
```

p/paragraph

layouts simple paragraph(s) justified at position with width

```
<p id="_id_" pos="_x_,_y_" width="_pts_" ...attributes... ><![CDATA[
...simple text paragraph 1...

...simple text paragraph 2...

...simple text paragraph 3...
]]></p>
```

table

layouts a table at position with width

```
<table id="_id_" pos="_x_,_y_" width="_pts_" align="_,_,_" widths="_,_,_" >
<cell>...</cell>
<markdown>...</markdown>
<cell>...</cell>
</table>
```

markdown

layouts flexmark/commonmark/markdown at position with width, optionally loading from resource.

```
<markdown id="_id_" pos="_x_,_y_" width="_pts_" src="_res_" ...attributes... />

<markdown id="_id_" pos="_x_,_y_" width="_pts_" ...attributes... ><![CDATA[
   ...flexmark syntax...
]]></markdown>
```

outline/part/chapter/section/subsection/subsubsection/reference

creates outlines for the pdf outline tree and references the actual page

```
<outline text="_text_" level="_level_" />
<part text="_text_" />
<chapter text="_text_" />
<section text="_text_" />
<subsection text="_text_" />
<subsubsection text="_text_" />
<reference text="_text_" />
```

- chapter implies level 0
- section implies level 1
- subsection implies level 2
- subsubsection implies level 3
- outline implies level 4 unless given

outline/... is a sort of "invisible" heading.

reference is sort of an out-of-tree heading.

draw-rect

draws a rectangle at position with width and height

```
<draw-rect id="_id_"
   pos="_x_,_y_"
   width="_pts_"
   height="_pts_"
   fill-color="_color_"
   stroke-color="_color_"
   stroke="_pts_"
   background="_boolean_" />
```

Layers

the following page elements support the **layer** attribute for putting the object on the specified *pdf-layer*:

- text, label, p, h
- image, svg
- markdown
- draw-rect
- table

Examples

```
<text layer="_layer_name_" ... > ... content ... </text>
<label layer="_layer_name_" ... > ... content ... </label>
<p layer="_layer_name_" ... > ... content ... </p>
<h layer="_layer_name_" ... > ... content ... </h>
```

```
<image layer="_layer_name_" ... />
<svg layer="_layer_name_" ... />
```

```
<markdown layer="_layer_name_" ... > ... content ... </h>
```

```
<draw-rect layer="_layer_name_" ... attributes ... />
```

```
<table layer="_layer_name_" ... > ... cells ... </h>
```

Entities & Font-Icons

TBD.

Drawing with Templating

Document Commands

settitle ***text***

setcreator ***text***

setauthor ***text***

setsubject ***text***

setkeywords ***text***

setproducer ***text***

Draw Commands

```
<draw templating="_boolean_" ... attributes ... >
   ... draw commands ...
</draw>
```

Generic Graphic State Commands

newpage ***w h*** **or** ***x1 y1 x2 y2***

starts a new page, optionally with a mediabox

mediabox ***w h*** **or** ***x1 y1 x2 y2***

sets the page mediabox

newcontent ***bool***

starts a new content-stream, if bool is true puts it into the background

content ***string***

inserts raw pdf language in the content-stream

gsave

Saves graphic state onto the stack (push).

grestore

Restores graphic state from the stack (pop).

startlayer ***name***

Starts a named graphics layout group.

endlayer

Ends a named graphics layout group.

matrix ***a b c d e f***

adds matrix to current matrix (use gsave/grestore to reset).

rotate *rot*

adds rotation to current matrix

skew *sx sy*

adds skew to current matrix

scale *sx sy*

adds scale to current matrix

translate *tx ty*

adds translation to current matrix

Specific Graphic State Commands

(PDF32000-1:2008 8.4.3)

linedash *d1 ... dN*

(PDF32000-1:2008 8.4.3.6)

linedashx *offs d1 ... dN*

(PDF32000-1:2008 8.4.3.6)

linewidth *n*

Thickness of the painted stroke (integer or float). (PDF32000-1:2008 8.4.3.2)

linecap *N*

Appearence style of open paths (and dashes if any). (PDF32000-1:2008 8.4.3.3)

- 0. butted cap
- 1. round cap
- 2. projected square cap

linejoin *N*

Appearence style of path corners (joins). (PDF32000-1:2008 8.4.3.4)

- 0. miter join
- 1. round join
- 2. bevel join

meterlimit *f*

Style factor of miter joins. (PDF32000-1:2008 8.4.3.5)

flatness *f*

flatness factor (1-100, 0 = default). (PDF32000-1:2008 10.6.2)

pen *color width* [*opacity*]

a shorthand for **strokecolor** and **linewidth**, and optional **strokealpha**

Opacity/Alpha Commands

fillalpha *N*

opacity for fill operations (0-100; 100 = default, fully opaque; 0 = fully transparent).

strokealpha *N*

opacity for stroke operations (0-100; 100 = default, fully opaque; 0 = fully transparent).

Color Commands

fillcolor *p1 ... pN*

strokecolor *p1 ... pN*

Color Command Parameters

One Parameter

Named Colors

see section Attribute / Color

Gray

- (float) a grey-level percentage (0-100; 0 = black, 100 = white).

Two Parameters

- (string,int) the string 'gray' and a grey-level (0-255; 0 = black, 255 = white).
- (string,float) the string 'gray%' and a grey-level percentage (0-100; 0 = black, 100 = white).

Three Parameters

- (float,float,float) rgb-levels as percentage (0-100).

Four Parameters

- (string,int,int,int) the string 'rgb' and rgb-levels (0-255).
- (string,float,float,float) the string 'rgb%' and rgb-levels as percentage (0-100).
- (float,float,float,float) cmyk-levels as percentage (0-100).

Five Parameters

- (string,int,int,int,int) the string 'cmyk' and cmyk-levels (0-255).
- (string,float,float,float,float) the string 'cmyk%' and cmyk-levels as percentage (0-100).

Text Commands

starttext

starts a text operations section

endtext

ends a text operations section

movetext *dx dy*

moves the text origin relative to the current position

hscale *pct*

sets text horizontal scaling in percent (100 = default;)

font *name size*

selects a text-font and size in points

text *string*

writes text at the actual position; html-entities are supported if the font supports the unicode-point.

rendertext *string*

like **text** above, but also recogizes font-icons, and advances to the next line.

Graphics Commands

moveto *x y*

move to a new position

movetox *x*

move to a new x-coordinate while keeping the y-coordinate

movetoy *y*

move to a new y-coordinate while keeping the x-coordinate

moverel *dx dy*

move relative to the actual position

movepolar *dL alpha*

move relative to the actual position, *alpha* in degrees, clock-wise.

lineto *x y*

line *x0 y0 x1 y1*

shorthand for moveto/lineto.

!!! line-rel line-polar hline-rel vline-rel

hline *x*

vline *y*

arrow *x0 y0 x1 y1 scale bFrom bTo*

arrowto *x1 y1 scale bFrom bTo*

arrowrel *dx dy scale bFrom bTo*

arrowpolar *dL alpha scale bFrom bTo*

```
  curveto() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.curveTo(_args); }},
  arc() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.arc(_args); }},
  pie() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.pie(_args); }},
  circle() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.circle(_args); }},
  ellipse() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.ellipse(_args); }},
  spline() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.spline(_args); }},
  closepath() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.closePath(); }},
  endpath() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.endPath(); }},
  rect() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.rect(_args); }},
  rectxy() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.rectXY(_args); }},


  stroke() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.stroke(); }},
  fill() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.fill(_args); }},
  fillstroke() {@Override public void execute(PmlScriptContext _ctx, String _cmd, List<String> _args) { _ctx.fillStroke(_args);
}},
```

Templating

you can switch on **jinjava/jinja2** templating by specifying the **templating=on** attribute.

```
<draw templating="_boolean_" ... attributes ... >
  ... draw commands ...
</draw>
```

by executing draw-script dircetly from the commandline, templating is always on and you can specify variables using the "-D [type:]name=value" option.

for further infomation on jinjava/jinja2 templating see the following links:

- https://product.hubspot.com/blog/jinjava-a-jinja-for-your-java
- https://jinja.palletsprojects.com/en/2.11.x/templates/

Scripting using LUA

the lua language as implemented by this processor is subject to the following limitations:

- the lua language is implemented at version 5.2
- all limitations of luaj 3.0.x apply, see http://www.luaj.org/luaj/3.0/README.html
- all limitations of the Java SE platform apply, see http://luaj.org/luaj/3.0/api/org/luaj/vm2/lib/jse/JsePlatform.html

for programming the lua 5.2 language please refer to "Programming in Lua 3"*" at https://github.com/xfbs/PiL3 which is the definite guide

Implementation specifics

the implementaion deviates from standard luaj at the following points:

- availability of the *jstring* module
- availability of the "_" context object.

jstring module

the jstring module implemnts a number of convenience functions pertaining to strings or thier production/processing.

split separator-regex, source-string

splits a string into a lua-table. is equivalent to java String.split(source, regex)

explode separator, source-string

splits a string into a lua-table. is equivalent to java StringUtils.split(source, sep)

implode separator, table

concats the values of a lua-table with separator

implode separator, arg1, ... argN

concats the arguments with separator

join separator, table

concats the values of a lua-table with separator, using lua-stringification on non-string values.

join separator, arg1, ... argN

concats the arguments with separator, using lua-stringification on non-string arguments.

concat arg1, ... argN

concats the arguments, using lua-stringification on non-string arguments.

mformat pattern, arg1, ... argN

formats the arguments according to pattern. equivalent to MessageFormat.format(pattern, arg1, ... argN), see https://docs.oracle.com/javase/8/docs/api/java/text/MessageFormat.html

sformat pattern, arg1, ... argN

formats the arguments according to pattern. equivalent to String.format(pattern, arg1, ... argN), see https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html#syntax

context object

the processor automatically imports the "_" context object into the lua namespace.

the "_" context object represents a simplified wrapper around the pdf scripting api.

_:newPage(); _:newPage(w,h); _:newPage(x0,y0,x1,y1);

initializes a new page for output.

_:endPage();

closes a page for output.

_:setMediabox(w,h); _:setMediabox(x0,y0,x1,y1);

sets the pages width/height or mediabox.

_:newContent();

initializes a new content-stream for output.

_:endPage();

closes content-stream for output.