PStill 1.72.9 Structured Output Mode Tag Description

Create output:

Use command line option:

-m structured -o output.xml

Tag descriptions:

<PStillOutput>

Leads in PStill structured output

<EC Type="..." Value="..."/>

DSC comment transported to the XML section, Type is usually "ENDC" with an empty value. Used currently only for other PStill purposes (e.g. S2C), can be ignored for general parsing

<Page Number="...">

Leads in page section, number indicates current page number starting at 1

There are basically 4 graphics elements that are written by PStill, all are defined in the same sequence of tags. The 4 elements are 'stroke' (drawing of lines), 'fill' (drawing of areas), 'show'
(drawing of text) and 'image' (drawing of raster data).

Each element is defined by a sequence of structure tags, the first is the clip path definition

<Clip>

Leads in the clip definition section, all following path definitions now define the clip path. See path defintion below.

</Clip>

Leads out the clip definition section, the clip path is now in effect

Next follows the Gstate information

<Color Type="DeviceCMYK" Value="CCC,MMM,YYY,KKK" />

Type can be DeviceCMYK (4 numbers value), DeviceGray (one number value), DeviceRGB (3 number value). Note that spot colors are also written here in their alternate color spaces but later defined as spot. Nummeric values can be 0.0 to 1.0.

<GState Type="Overprint" Value="..." />

Defines overprinting, can be either 0 (no overprinting for element) or 1 (overprinting for element)
Only written if overprinting output is demanded by options.

<GState Type="Transparency" Value="..." />

Defines element transparency, can be either 1.0 (no transparency for element) or < 1.0

(transparency
for element). Only written if transparency output is demanded by options.

<GState Type="SpotColor" Value="CCC,MMM,YYY,KKK (NAME_OF_SPOTCOLOR)"/>

Defines the spot color in effect. Only written if spot color output is demanded by options.

<GState Type="Linewidth" Value="..." />

Defines the current linewidth. only written if different from default linewidth (1.0).

<GState Type="Linecap" Value="..." />

Defines the current linecap. only written if different from default linecap.

<GState Type="Linejoin" Value="..." />

Defines the current linejoin. only written if different from default linejoin.

<GState Type="Mitrelimit" Value="..." />

Defines the current mitrelimit. only written if different from default mitrelimit.

<GState Type="Flattness" Value="..." />

Defines the current flattness. only written if different from default flattness.

<GState Type="DashArray" Value="[...] ... d" />

Defines the current line dash. only written if different from default line dash (no dash).
[...] contains the dash array, the second value the offset

Path definitions:

A path is defined by a sequence of several path definition tags:

<Path Type="Moveto" Value="XXX,YYY" />

Defines a moveto to given postion

<Path Type="Lineto" Value="XXX,YYY" />

Defines a lineto to given position (from last postion)

<Path Type="Curveto" Value="XXX1,YYY1,XXX2,YYY2,XXX3,YYY3" />

Defines a cuveto, XXX1,YYY1 is control point 1, XXX2,YYY2 is control point 2, XXX3,YYY3 is end point, starting point it last position). Curveto are usually not used in a clip path defintion (because the path is already flattened for clip paths) but can be used in normal drawing.

<Path Type="Close" />

Closes the current path by applying an explicit lineto to the starting point.

Now follows the defintion of the graphic elements.

Simple elements:

Stroke element:

<Stroke>

Leads in the definition of the Stroke element, followed by a path definition (See below)

</Stroke>

Conculdes the definition of the Stroke element, at this point the defined path is drawn.

Fill element:

<Fill>

Leads in the definition of the Fill element, followed by a path definition (See below)

</Fill>

Conculdes the definition of the Fill element, at this point the defined path is drawn.

<EOFill>

Leads in the definition of the EOFill element, followed by a path definition (See below)
EOFill uses a different path winding rule to draw a fill area, see PS language manual.

</EOFill>

Conculdes the definition of the EOFill element, at this point the defined path is drawn.

Complex elements:

Text element:

Before a text element can be defined the details about the font used must be defined.

<TextEncoding Number="..." Hash="...">

Leads in the defintion of a Text Encoding, Number is the number assigned by PStill to the encoding and hash is a unique number for this encoding. Following are 256 names that define the encoding, each one starting by '/'

</TextEncoding>

Concludes the defintion of a text encoding.

Parser systems should store the encoding under the number given in the defintion for PStill may later recall encodings by number only:

<TextEncoding Recall="Yes" Number="..." />

<Font Name="Times-Roman"
Matrix="11.999999,0.000000,0.000000,11.999999,0.000000,0.000000" />

Defines the font in use by name (true PS name) and its matrix. The matrix defines the sizes, translation and
rotation, see PS Language Reference manual. For an unrotated font element 1 and 4 in the matrix defines
X_size and Y_size respectively.

Now a path definition follows, usually only a moveto to set the point where the text is drawn. See path definition below.

<Show Type="Tj" Text="..." />

Defines the text to be drawn, according to Encoding, Font name, Matrix and position defined before.
Currently only one type ("Tj") is used.

This concludes the text element

Image element:

<**ImageObj Count="1" Type="1" Multi="1" NumColors="1" Width="120" Height="143" Bps="2" Matrix="1.000,0.000,0.000,-1.000,0.000,143.000" Mode="-1" DpiX="72.00" DpiY="72.00" IsIndexedColor="0">**

Start of defintion of image object.

Count is the number assigned by PStill for the image.
Type can be 1 for plain image, 2 for image mask and 3 for color image
Multi defines if the image uses multiple data streams (one per each color)
Width givens the image width in pixel (raw width)
Height gives the image height in pixel (raw height)
Bps defines the BitsPerSample, can be 1,2,4 or 8
Matrix defines the image matrix, see PostScript reference manual
Mode is -1 for mask, 1 for plain image
DpiX gives image dpi for X axis when imaged on page
DpiY gives image dpi for Y axis when imaged on page
IsIndexedColor can be 0 for no indexed color or 1 for indexed color

<PageMatrix Value="1.000 0.000 0.000 1.000 216.000 326.000" />

Defines the page matrix (the matrix that is used to project the image data on the page). see PostScript reference manual

<Image Type="Mono" NumSections="3" SectionLength="30" ExpectedDataLength="4290">

Defines the start of the image data section, Type is either "Mono", "Mask" or "Color"
NumSections gives the number of individual data runs per scanline, if not given there is only one
SectionLength defines the number of bytes in each scanline.
ExpectedDataLength gibes the number of bytes to be read

<ImageData Type="Embedded">

Leads in an embedded data section

<ImageDataAttr BlockSize="..." />

Sets the block size for each data run to the given number of bytes. Runs are read in one block and
interpreted according to the NumSections value.

Data is now defined as hex bytes

Alternatively PStill will use precompressed external data files for large images (> 128 kByte)

<ImageData Type="GZFile" Name="...">

The external file defined by Name will contain the image data, use GZIP (www.libz.org) to decompress.

</ImageData>

Concludes the data defintion

</Image>

Concludes the image definition

</ImageObj>

Concludes the image obj defintion

End of Page:

Each page is concluded by

</Page>

End of Job:

At the end of the document the following extra info is given:

<SpotColorInfo Count="0" >

The number of spot colors seen. If there are more than 0 the colors will follow in the form

<SpotColor Value="CCC,MMM,YYY,KKK" Name="..." />

until

</SpotColorInfo>

<ExtraJobInfo PageSize="612,792" />

Will give the page size as defined by the job.

</PStillOutput>

Concludes the structured output defintion.