Standard FOP Extensions

Version 628775

Table of contents

By "extension", we mean any data that can be placed in the input XML document that is not addressed by the XSL-FO standard. By having a mechanism for supporting extensions, FOP is able to add features that are not covered in the specification.

The extensions documented here are included with FOP, and are automatically available to you. If you wish to add an extension of your own to FOP, please see the Developers' Extension Page.

> Note:
> All extensions require the correct use of an appropriate namespace in your input document.

1 SVG

Please see the SVG documentation for more details.

2 FO Extensions

2.1 Namespace

By convention, FO extensions in FOP use the "fox" namespace prefix. To use any of the FO extensions, add a namespace entry for `http://xml.apache.org/fop/extensions` to the root element:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format"
              xmlns:fox="http://xmlgraphics.apache.org/fop/extensions">
```

2.2 PDF Bookmarks

In previous versions of Apache FOP there was a `fox:outline` element which was used to create outlines in PDF files. The redesigned code makes use of the new bookmark feature defined in the latest XSL 1.1 working draft.

2.3 Anchors or Named Destinations

Use the fox:destination element to define "named destinations" inside a PDF document. These are useful as fragment identifiers, e.g. "http://server/document.pdf#anchor-name". fox:destination elements can be placed almost anywhere in the fo document, including a child of root, a block-level element, or an inline-level element. For the destination to actually work, it must correspond to an "id" attribute on some fo element within the document. In other words, the "id" attribute actually creates the "view" within the PDF document. The fox:destination simply gives that view an independent name.

```
<fox:destination internal-destination="table-of-contents"/>
...
<fo:block id="table-of-contents">Table of Contents</fo:block>
```

> Warning:
> It is possible that in some future release of FOP, *all* elements with "id" attributes will generate named-destinations, which will eliminate the need for fox:destination.

2.4 Table Continuation Label

This extension element hasn't been reimplemented for the redesigned code, yet.

2.5 fox:orphan-content-limit and fox:widow-content-limit

The two proprietary extension properties, fox:orphan-content-limit and fox:widow-content-limit, are used to improve the layout of list-blocks and tables. If you have a table with many entries, you don't want a single row to be left over on a page. You will want to make sure that at least two or three lines are kept together. The properties take an absolute length which specifies the area at the beginning (fox:widow-content-limit) or at the end (fox:orphan-content-limit) of a table or list-block. The properties are inherited and only have an effect on fo:table and fo:list-block. An example: fox:widow-content-limit="3 * 1.2em" would make sure the you'll have at least three lines (assuming line-height="1.2") together on a table or list-block.

2.6 fox:external-document

> **Note:**
> This feature is incomplete. Support for multi-page documents will be added shortly. At the moment, only single-page images will work. And this will not work with RTF output.

This is a proprietary extension element which allows to add whole images as pages to an FO document. For example, if you have a scanned document or a fax as multi-page TIFF file, you can append or insert this document using the `fox:external-document` element. Each page of the external document will create one full page in the target format.

The `fox:external-document` element is structurally a peer to `fo:page-sequence`, so wherever you can put an `fo:page-sequence` you could also place a `fox:external-document`. Therefore, the specified contents for `fo:root` change to:

```
(layout-master-set, declarations?, bookmark-tree?, (page-sequence|page-
sequence-wrapper|fox:external-document|fox:destination)+)
```

2.6.1 Specification

The `fox:external-document` extension formatting object is used to specify how to create a (sub-)sequence of pages within a document. The content of these pages comes from the individual subimages/pages of an image or paged document (for example: multi-page TIFF in the form of faxes or scanned documents, or PDF files). The formatting object creates the necessary areas to display one image per page.

In terms of page numbers, the behaviour is the same as for `fo:page-sequence`. The placement of the image inside the page is similar to that of `fo:external-graphic` or `fo:instream-foreign-object`, i.e. the viewport (and therefore the page size) is defined by either the intrinsic size of the image or by the size properties that apply to this formatting object.

Content: EMPTY

The following properties apply to this formatting object:

- (Common Accessibility Properties) (not implemented, yet)
- (Common Aural Properties) (not implemented, yet)

- block-progression-dimension
- content-height
- content-type
- content-width
- display-align
- height
- id
- inline-progression-dimension
- overflow
- pages: <page-set> (see below) (not implemented, yet)
- reference-orientation
- scaling
- scaling-method
- src
- text-align
- width

Datatype "page-set": Value: auto | <integer-range>, Default: "auto" which means all pages/subimages of the document. <integer-range> allows values such as "7" or "1-3"

> **Note:**
> `fox:external-document` is not suitable for concatenating FO documents. For this, XInclude is recommended.

2.7 Free-form Transformation for fo:block-container

For `fo:block-container` elements whose `absolute-position` set to "absolute" or "fixed" you can use the extension attribute `fox:transform` to apply a free-form transformation to the whole block-container. The content of the `fox:transform` attribute is the same as for [SVG's transform attribute](). The transformation specified here is performed in addition to other implicit transformations of the block-container (resulting from top, left and other properties) and after them.

Examples: `fox:transform="rotate(45)"` would rotate the block-container by 45 degrees clock-wise around its upper-left corner. `fox:transform="translate(10000,0)"` would move the block-container to the right by 10 points (=10000 millipoints, FOP uses millipoints internally!).

> **Note:**
> This extension attribute doesn't work for all output formats! It's currently only supported for PDF, PS and Java2D-based renderers.

2.8 Color functions

XSL-FO supports specifying color using the rgb(), rgb-icc() and system-color() functions. Apache FOP provides additional color functions for special use cases. Please note that using these functions compromises the interoperability of an FO document.

2.8.1 cmyk()

```
color cmyk(numeric, numeric, numeric, numeric)
```

This function will construct a color in device-specific CMYK color space. The numbers must be between 0.0 and 1.0. For output formats that don't support device-specific color space the CMYK value is converted to an sRGB value.