

# Standard FOP Extensions

\$Revision: 490012 \$

## Table of contents

- 1 SVG.....2
- 2 FO Extensions.....2
  - 2.1 Namespace.....2
  - 2.2 PDF Bookmarks.....2
  - 2.3 Anchors or Named Destinations.....3
  - 2.4 Table Continuation Label.....3

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 required 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 identifier. 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://xml.apache.org/fop/extensions">
```

### 2.2. PDF Bookmarks

You can provide outlines inside the root object (but outside any page-sequences or other formatting objects). Here's an example of an outline entry:

```
<fox:outline internal-destination="sec3">
  <fox:label>Running FOP</fox:label>

  <fox:outline internal-destination="sec3-1">
    <fox:label>Prerequisites</fox:label>
  </fox:outline>
</fox:outline>
</fo:root>
```

It works similarly to a basic-link. There is also an external-destination property, but it isn't supported currently. See the pdfoutline.fo file in examples/fo/basic for a more complete example.

## 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

Use the `fox:continued-label` element to create content in table-header and table-footer cells that will appear only on pages after the first page that the table appears. `fox:continued-label` is itself inline content, and is a container of `fo:inline` content. This content will be laid out only if the table does not fit on a single page and flows to following pages. Here is an example of FO code creating such a table-header:

```
<fo:table-header>
  <fo:table-row>
    <fo:table-cell>
      <fo:block>Header column 1 with continued label
        <fox:continued-label><fo:inline>
(cont.)</fo:inline></fox:continued-label>
        </fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block>Header column 2 with no continued label</fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-header>
```