

FOP

1 Extensions to FOP

Sometimes it is desirable to have extensions to xsl:fo in order to support some feature of the output format which isn't covered by the xsl:fo specification.

1.1 Default Extensions

These extension are available by default. They are automatically loaded and you only need to provide the correct namespace for your embedded xml markup.

1.1.1 SVG

Please see the [SVG page](#) for more details.

1.1.2 Bookmarks

To use this standard Fop extension, you need to add a namespace entry for <http://xml.apache.org/fop/extensions> on the root element.

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:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format"
         xmlns:fox="http://xml.apache.org/fop/extensions">
  <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 docs/examples/fo for a more complete example.

1.2 Adding Your Own

To add your own extension you need to do the following things.

Write code that implements your extension functionality. The easiest place to start is by looking at the code in `org.apache.fop.extension`.

Create a jar file with your classes, it must also include the following file `/META-INF/services/org.apache.fop.fo.ElementMapping`. In this file you need to put the fully qualified classname of your element mappings class. This class must implement the `org.apache.fop.fo.ElementMapping` interface.

Create your fo file with the extra xml data embedded in the file with the correct name space. The examples for `svg` and `pdfoutline.fo` show how this can be done.

Put your jar file in the classpath and then run fop over the fo file.