

FOP FAQ

Table of contents

1 Questions.....	2
1.1 1. General questions.....	2
1.2 2. Problems running FOP.....	3
1.3 3. Problems with FOP output.....	7
1.4 4. Embedding FOP. Using FOP in a servlet.....	10
1.5 5. SVG specific questions.....	11
1.6 6. PDF specific (includes Acrobat peculiarities).....	13
1.7 7. IEx specific stuff.....	14
1.8 8. General questions regarding XSLT, XSLFO, and basic XML.....	14
1.9 9. Development-related Questions.....	16
1.10 10. General suggestions. How to solve problems.....	16

Questions

1. General questions

1.1. What is FOP?

FOP is a print formatter for [XSL formatting objects](#).

It can be used to render an XML file containing XSL formatting objects into a page layout. The main target is PDF but other rendering targets are supported, such as AWT, PCL, text and direct printing.

1.2. What can I do with FOP?

FOP provides both an application and a library that converts an XSL FO document into paginated output.

The FOP command line application can be directly used to transform XML into PDF, PostScript, PCL and other formats, there is also an AWT based viewer integrated.

The library can be used in servlets and other Java applications.

1.3. What does "FOP" stand for?

FOP is an acronym for **F**ormatting **O**bject **P**rocessor

1.4. How does FOP interact with other Apache Projects?

FOP is distributed with [Cocoon](#) as a PDF serializer for XSL (FO) documents.

[Batik](#) can be used with FOP to [transcode an SVG image](#) into a PDF document.

1.5. What is XSL?

XSL is a W3C standard concerned with publishing XML documents. It consists of two parts: [XSLT](#) and [XSL-FO](#). The acronym expands to eXtensible Stylesheet Language.

1.6. What is XSL-FO?

XSLFO (aka XSL-FO) is an XML vocabulary that is used to specify a pagination and other styling for page layout output. The acronym "FO" stands for **F**ormatting **O**bjects. XSLFO can be used in conjunction with [XSLT](#) to convert from any XML format into a paginated layout ready for printing or displaying.

XSLFO defines a set of elements in XML that describes the way pages are set up. The contents of the pages are filled from flows. There can be static flows that appear on every page (for headers and footers)

and the main flow which fills the body of the page.

Synonyms: XSL FO, XSL (FO), XSL:FO, XSL-FO, Formatting Objects

1.7. What is XSLT?

XSLT describes the transformation of arbitrary XML input into other XML (like XSLFO), HTML or plain text. The “T” comes from Transformation. For historical reasons, a transformation is often also called a “style sheet”.

Synonyms: XSL transformation, XSL:T, XSL style sheet.

1.8. When will feature X be implemented?

The short answer is "Don't ask." For more details, see [Understand FOP's Limitations](#).

1.9. When is the next released planned?

When it's ready and the committers have enough time to go through the time-consuming process of creating a release. We released version 1.0 in July 2010, and will release version 1.1 when we accumulated enough additions and bug fixes. If you want to speed up the process, consider [contributing](#) to FOP.

1.10. Why did you redesign FOP?

The code redesign of FOP was necessary in order to be able to adress following issues:

- Keeps and Breaks
- XSL-FO Compliance in general
- Process Files of Arbitrary Size
- Minimize Memory Use
- Table Layout

1.11. How can I contribute?

There are numerous ways that you can help. They are outlined in the [Developer's Introduction](#) page.

2. Problems running FOP

2.1. I get the error: [ERROR]: 'master-reference' for 'fo:page-sequence' matches no 'simple-page-master' or 'page-sequence-master'

FOP was changed to be in accordance with the latest standard (see [XSL standard](#)). The page master for a fo:page-sequence is now referenced by the master-reference attribute. Replace the master-name attributes of your fo:page-sequence elements by master-reference attributes. You have to do this also for fo:single-page-master-reference, fo:repeatable-page-master-reference and

fo:conditional-page-master-reference elements in you page master definitions.

Update your FO documents and style sheets.

2.2. I get a NoClassDefFound exception.

This is typically a problem with your classpath.

If you are running FOP from the command line:

- Use the command file `fop.bat`, `fop.cmd` or `fop.js` on MS Windows, or `fop` on Unix/Linux from the FOP distribution.
- If this doesn't help, check whether still all the jar files mentioned in the classpath in the `fop.bat` file are in their respective places.

If you run FOP embedded in your servlet, web application or other Java application, check the classpath of the application. Check the also the information pertaining to [servlet engines](#) for further hints.

If you downloaded the source distribution, or a snapshot from the repository, remember you have to build the FOP jars first.

2.3. I get a NoSuchMethodException or a NoSuchFieldException exception.

This is usually caused by an older version of one of the FOP jars or old XML tools in the classpath. Check in particular for `parser.jar`, `jaxp.jar`, `xml4j.jar` or `lotusxsl.jar`.

Incompatible versions of Batik may also cause this problem. Use the version of Batik that comes with FOP. It is not always possible to use a more recent version of Batik.

2.4. I get an OutOfMemoryException.

See [FOP Memory](#).

2.5. I get a MalformedURLException.

What you probably think of as "file names" are usually URLs, in particular the `src` attribute of `fo:external-graphic`.

Because usage of URLs is growing, you should make yourself familiar with it. The relevant specification is [RFC 2396](#).

In a nutshell, the correct syntax for an absolute file URL is `file:///some/path/file.ext` on Unix and `file:///z:/some/path/file.ext` on Windows systems. Note the triple slash, and also that only forward slashes are used, even on windows.

A relative file URL starts with anything but a slash, and doesn't have the `file:` prefix, for example `file.ext`, `path/file.ext` or `../file.ext`. The string `file:path/file.ext` is *not* a relative URL, in fact, it isn't a valid URL at all. A relative URL is subject to a resolving process, which transforms it into an absolute URL.

2.6. I get an "[ERROR]: null", or a NullPointerException.

This is often caused by an invalid FO document. Currently only very common errors are intercepted and produce a comprehensible error message. If you forgot container elements like `fo:page-sequence` or `fo:flow` and put blocks and inline elements directly as children of `fo:root` or `fo:page-sequence`, you'll only get a `NullPointerException`. Check whether your FO file has a proper structure. In some cases there are mandatory properties, like the master-reference in `fo:conditional-page-master-reference`, check also whether you got them right.

You may find it helpful to use the [validation tools](#) to validate your FO document. This will catch most problems, but should not be relied upon to catch all.

If you use XSLT, problems in your style sheet and in your source XML also can produce a `NullPointerException`. [Run the transformation separately](#) to check for this, usually you'll get a detailed error message from the XSLT processor.

If you turn on debugging with the `-d` option you may be able to see more detailed information.

2.7. FOP returns the message "[ERROR] object|property - 'x' is not implemented yet." When will it be implemented?

See FOP's Standards [Compliance](#) page.

2.8. FOP hangs. FOP does not exit.

The most likely reason is a known problem with the Java run time environment which is triggered by rendering SVGs. Sun's JDK 1.4 does not have this problem. See also [FOP does not exit if a SVG is included](#).

Another possibility is that FOP went into a non terminating loop. Usually this is indicated by lots of log messages of the form "[INFO]: [NNNN]" which indicate a new page has been started or [box overflows](#). After some time, FOP will crash with an `OutOfMemoryException`.

If you called the FOP command line application from some other program, for example from Java using `Runtime.exec()`, it may hang while trying to write log entries to the output pipe. You have to read the FOP output regularly to empty the pipe buffer. It is best to avoid `exec`'ing FOP, use the library interface instead.

2.9. FOP runs forever, writing lots of ">" to the log.

There is something too large to fit into the intended place, usually a large image, a table whose rows are kept together or a block with a space-before or space-after larger than the page size. Catch the first page showing this phenomenon and check it. If it is not obvious which element causes the trouble, remove stuff until the problem goes away. Decrease the dimensions of the offending element or property, or increase the dimension of the enclosing element or container, or remove `keep-with-*` properties.

2.10. FOP cannot find a file for fo:external-graphics.

The `src` attribute of the `fo:external-graphics` element requires a URI, not a file name. See [External Resources](#) for more information about specifying URIs.

2.11. FOP does not find my fonts.

Did you get: “Failed to read font metrics file C:\foo\arial.xml : File "C:\foo\arial.xml" not found”? The value for the `metrics-file` attribute in the user config file is actually an URL, not a file name. Use `"file:///C:/foo/arial.xml"` instead.

If you used a relative URL, make sure your application has the working directory you expect. Currently FOP does not use the `baseDir` for resolving relative URLs pointing to font metric files.

Try also setting the [font-base configuration](#).

2.12. I get a SAXException: Mismatch: page-sequence vs. root (or similar).

The full exception usually looks similar to this:

```
Mismatch: page-sequence (http://www.w3.org/1999/XSL/Format) vs. root
(http://www.w3.org/1999/XSL/Format)
```

This exception is usually a follow-up error after another exception. Sometimes the original exception gets swallowed by Xalan's default `ErrorListener` (should be fixed in the latest Xalan release).

The work-around is to set an explicit `ErrorListener` on the `Transformer`. The `ErrorListener` can be as simple as this:

```
import javax.xml.transform.ErrorListener;
import javax.xml.transform.TransformerException;

public class DefaultErrorListener implements ErrorListener {

    public void warning(TransformerException exc) {
        System.err.println(exc.toString());
    }

    public void error(TransformerException exc)
        throws TransformerException {
        throw exc;
    }

    public void fatalError(TransformerException exc)
        throws TransformerException {
        throw exc;
    }
}
```

2.13. The following message appears after recently upgrading FOP: "I/O exception while reading font cache (org.apache.fop.fonts.EmbedFontInfo; local class incompatible: stream classdesc serialVersionUID = -9075848379822693399, local class serialVersionUID = 8755432068669997367). Discarding font cache file."

This message is a warning that FOP failed to read from the Font cache. Which means any Font auto detection or Font directories will be re-scanned. So this failure doesn't break anything.

To avoid the warning you can simply delete the old Font Cache file, which lives in `${base}\conf\font.cache` (see [font-base configuration](#)). Or you can disable Font Caching altogether using the option "use-cache."

3. Problems with FOP output

3.1. Leaders don't work in current FOP. Instead of filling the line, only three dots or a short ruler is output.

Leaders still work, in fact they work better than ever before. You'll just have to add `text-align="justify"` and/or `text-align-last="justify"` to the block with the leader. Be sure you haven't accidentally overridden the `leader-length.maximum="100%"` default value.

Earlier versions of FOP used to expand a leader to fill the rest of the line unconditionally, anything following it, like page numbers in a TOC, was actually shifted beyond the right margin.

The new implementation uses `leader-length.optimum` to determine where to break the line, and expands the leader only further if the line should be filled, as indicated by the `text-align` and `text-align-last` properties.

Actually due to the fuzzyness of the specification both the old and the new method are conformant (although adding text after the expanded leader in the old variant never was).

If you want to have a longer ruler or space in a non-justified line, you have to increase the `leader-length.optimum` property.

3.2. Why does FOP insert a blank page between my page sequences?

This is because spec conformance has been improved.

The [force-page-count property](#) controls how a FO processor pads page sequences in order to get certain page counts or last page numbers. The default is "auto". With this setting, if the next page sequence begins with an odd page number because you set the `initial-page-number`, and the current page sequence also ends with an odd page number, the processor inserts a blank page to keep odd and even page numbers alternating (similar for the case the current page sequence ends with an even page number and the next page sequence starts with an even page number).

If you don't want to have this blank page, use `force-page-count="no-force"`.

3.3. Why is my graphic not rendered?

Most commonly, the external file is not being found by FOP. Check the following:

- Empty or wrong `baseDir` setting.
- Spelling errors in the file name (including using the wrong case).

- Security Problems (i.e. the image could not be accessed because FOP is not allowed to read the file). This is especially a problem if the external file is retrieved over HTTP. Possible issues include security settings on the server, server configuration, and missing cookies or other authorization information. Any easy way to check this is to cut&paste the source URL from the `fo:external-graphic` into the Location field of a browser **on the machine where the FOP process will be running**.

Other possibilities:

- The image format is not supported or not supported completely. See [FOP Graphics Formats](#) for a list of supported formats and related issues.
- The graphic may be too large to fit into the intended space.
- There may be something (static content) that is obscuring the graphic. (This is very rare).

3.4. My graphical images do not work properly.

See [FOP Graphics Formats](#) for a list of supported graphics formats and related issues.

3.5. Why is my graphic rendered at a different resolution than it was created?

See [Graphics Resolution](#).

3.6. Keep-with-next, keep-with-previous, keep-together don't work.

Current FOP releases have extensive support for these properties. The current release, FOP 1.0, still supports the values "always" and "never" only, no numerical values. There may be a few places where `keep-*` still don't work, this should be very rare.

The concept is called “blind table”. The table is used for pure layout reasons and is not obvious in the output.

An example of an image and the image caption to be kept together:

```
<fo:table table-layout="fixed" width="100%">
  <fo:table-column column-width="proportional-column-width(1)"/>
  <fo:table-body>
    <fo:table-row keep-with-next="always">
      <fo:table-cell>
        <fo:block>
          <fo:external-graphic src="foo.jpg"/>
        </fo:block>
      </fo:table-cell>
    </fo:table-row>
    <fo:table-row>
      <fo:table-cell>
        <fo:block>Image Caption</fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-body>
</fo:table>
```

Consider upgrading to the latest version which supports keeps.

3.7. My tables are missing, or missing their content.

Check for `fo:table-body` around the rows. Usually FOP will catch this problem.

Also, the `fo:table-with-caption` element is not implemented, tables within such an element are dropped too. FOP generates an error message for this problem. Older DocBook style sheets generate `fo:table-with-caption` elements, so watch out.

3.8. Text overflowing table cells and the like is not clipped. Long text flows into adjacent cells/block, obscuring stuff there.

Since the `overflow` property doesn't apply to `table-cell`, you can wrap the cell content in a block-container and specify `overflow="hidden"` there. Alternatively, if you have long words overflowing table cells, try to get them hyphenated. Artificial names like product identifications or long numbers usually aren't hyphenated. You can try special processing at XSLT level, like

- clip long text,
- explicit wrapping+clipping,
- insert zero width spaces (U+200B or `​`) to allow FOP to wrap.

Check the [XSL FAQ](#) and the [XSL list archive](#) for how to perform these tasks.

3.9. The contents of table cells don't wrap

You probably have `keep-together="always"` set on the table cell. See [next question](#).

3.10. FOP behaves differently from earlier versions when `keep-together="always"` is set on table cells

Support for inline-level keeps has been added in FOP 0.95, and setting `keep-together="always"` also implicitly sets `keep-together.within-line="always"`, which forbids FOP to break the text into multiple lines. Set `keep-together.within-column="always"` on table-cell instead. It's a good idea not to use the shorthand `keep-together="always"` at all!

3.11. FOP tells me: "The contents of row 1 are taller than they should be". What does that mean?

This is usually caused by setting a "height" on a table-row or table-cell and when the content is higher than the specified height. By setting "height" (a so-called corresponding property) you [implicitly set](#) `block-progression-dimension.minimum`, `block-progression-dimension.optimum` and `block-progression-dimension.maximum` to the same value. You'll get some information about that in the warning message. Look for something like: "MinOptMax[min=opt=max=14000]".

Assuming you set the height on the table-row to 14pt and your content is 75pt high, a constraint (maximum=14pt) is violated. Normally, you just want to make sure with the "height" property that the row has a minimum height. If that is so, the right solution is to specify `block-progression-dimension.minimum="14pt"` instead of `height="14pt"`.

3.12. Page numbers are not properly right aligned.

This happens for `fo:page-number-citation` elements if the citation occurs before FOP formatted the requested page, usually in TOC or index pages. It is caused by the problem that FOP has to guess how much space the yet unknown page number will occupy, and usually the guesses are somewhat off.

The most recent FOP releases should have this problem fixed. Check whether you can upgrade.

3.13. Hyphenation does not work.

Make sure you have set the language and optionally the country attributes for an appropriate XSL-FO element (`fo:page-sequence`, `fo:block` or `fo:character`):

```
<fo:page-sequence language="fi">
```

See [Hyphenation Support](#) for details and instructions on using hyphenation with FOP.

Explicitly enable hyphenation for an appropriate XSL-FO element (`fo:block`, `fo:character`):

```
<fo:block hyphenate="true">
```

3.14. When I use margins, my content in a nested table or block-containers gets indented twice. Is this a bug?

No, although you might easily think so. The problem has to do with property inheritance of the `start-indent` and `end-indent` properties to which the margin properties are mapped. Apache FOP strictly adheres to the XSL-FO specification here which many other commercial FO implementations don't do to better meet end-user expectations. You can make FOP behave like these if you set `setBreakIndentInheritanceOnReferenceAreaBoundary(true)` on the `FOUserAgent`. The better way is to reset `start-indent` and `end-indent` to "0pt" on table-body or block-container. For further details, please consult the [Wiki page on Indent Inheritance](#).

3.15. Why is the output of Java2D/AWT-based renderers different than, for example, PDF or PS?

If you render the same document once to a PNG or TIFF and once into a PDF, the output may not be the same, i.e. line breaks are different or lines may have different heights. The reason for this: The Java2D-based renderers use the font subsystem of Java2D/AWT. The PDF and PS renderers use FOP's own font subsystem which provides much better font metrics than Java2D. These can lead to different layout decisions when the same document is rendered with different renderers. An alternative approach to fix this problem might be available but it hasn't been tested, yet. See also the [notes on fonts in the various output formats](#).

4. Embedding FOP. Using FOP in a servlet.

4.1. How do I use FOP in a servlet?

See [Using FOP in a Servlet](#).

4.2. How do I use FOP in a servlet with an XSLT transformation?

See [Using FOP in a Servlet with XSLT Transformation](#).

4.3. How do I pass parameters to the XSLT transformation?

See [Using FOP in a Servlet with XSLT Transformation](#).

4.4. How do I use my own fonts when running FOP from a servlet?

Declare the fonts in the `userconfig.xml` file as usual. See [loading the user configuration file](#) for further details.

4.5. How do I set the `baseDir` property in a servlet environment?

See [Setting the Configuration Programmatically](#).

4.6. How do I use a user configuration file from a servlet?

See [Using a Configuration File in an Embedded App](#).

4.7. I keep getting `NoClassDefFound` and other exceptions. How do I get FOP working for various servlet engines?

See [Servlet Engines](#).

4.8. Can FOP be used in multithreaded environments?

See [Multithreading FOP](#).

5. SVG specific questions

5.1. The rendering of SVG text in my PDF is of poor quality. Can I control this?

See [Placing SVG Text into PDF](#).

5.2. How do I use FOP with SVG on headless servers?

See [FOP: Graphics \(Batik\)](#).

5.3. FOP does not exit if a SVG is included.

Applies to older FOP versions and JDK 1.3 and older. That's because there is an AWT thread hanging around. The solution is to put a `System.exit(0)` somewhere.

5.4. I have problems with SVG referring to gradients etc. using "uri(#stuff)". I get a `MalformedURLException`.

This is really a "resolving relative URI" problem with some twists. The problem is that the `#stuff` URL fragment identifier is resolved within the current SVG document. So the reference must be valid within the XML subset and it cannot reference other SVG documents in the same XML file. Some options to try:

- Put the SVG into a separate file and use it with `fo:external-graphics`.
- Use a separate SVG file which contains only the gradient (and perhaps other SVG stuff you want to reference) and point an absolute URL to it:
`fill="url(file:///c:/refstuff/grad.svg#PurpleToWhite)"`.
- Same as above but use a relative URL: `fill="url(grad.svg#PurpleToWhite)"`. This may be easier to deploy.
- Make sure that the reference is valid in the current SVG document.

In any case, the referenced stuff has to be pointed to by an URL. It doesn't necessarily have to be a file URL, HTTP should also work. Also, expect a performance hit in all cases, because another XML file has to be retrieved and parsed.

Ultimately, both FOP and especially Batik should be fixed to make your code work as expected, but this will not only take some time but also some effort by a standard committee in order to make the semantics of this kind of references in embedded SVG clearer.

See also [MalformedURLException](#)

5.5. Why is my SVG rendered at a size different from that specified in my `fo:external-graphic` element?

See [SVG Scaling](#).

5.6. FOP fails with something like "The attribute "width" of the element `<rect>` is required". What's wrong?

This phenomenon is not a bug in FOP, but rather in the dependent packages: Apache Batik, Apache Xalan-J and SAXON. The bug in Apache Batik will be fixed in the next release. In the latest release of Apache Xalan-J, the bug is already fixed. It only occurs in the Xalan version bundled with Sun's JVM, because Sun uses a rather old version.

Bug description: In a namespace-enabled Level 3 DOM, an attribute in the default namespace must be set with `"null"` as the value for the namespace URI. SAX, on the other side, uses an empty string (`""`) to designate the default namespace. Many packages appear to not properly handle this difference in which case they still use the empty string as the namespace URI parameter for `org.w3c.dom.Element.setAttributeNS()`.

Work-around: Use the latest version of Apache Xalan-J. Note that starting with JDK 1.4 it's not enough to replace Xalan-J on the normal application classpath and you need to override the Xalan-J version

bundled with the Sun JVM using the [Endorsed Standards Override Mechanism](#), i.e. you must place Xalan-J in the "lib/endorsed" directory of your JRE.

6. PDF specific (includes Acrobat peculiarities)

6.1. How do I embed fonts in PDF?

See the [Fonts](#) page for information about embedding fonts.

6.2. Some characters are not displayed, or displayed incorrectly, or displayed as “#”.

This usually means the selected font doesn't have a glyph for the character.

The standard text fonts supplied with Acrobat Reader have mostly glyphs for characters from the ISO Latin 1 character set. For a variety of reasons, even those are not completely guaranteed to work, for example you can't use the fi ligature from the standard serif font. Check the [overview](#) for the default PDF fonts.

If you use your own fonts, the font must have a glyph for the desired character. Furthermore the font must be available on the machine where the PDF is viewed or it must have been embedded in the PDF file. See [embedding fonts](#).

For most symbols, it is better to select the symbol font explicitly, for example in order to get the symbol for the mathematical empty set, write:

```
<fo:inline font-family="Symbol">&#x2205;</fo:inline>
```

The "#" shows up if the selected font does not define a glyph for the required character, for example if you try:

```
<fo:inline font-family="Helvetica">&#x2205;</fo:inline>
```

6.3. What tools are available for post-processing my PDF document?

See [PDF Post-processing](#).

6.4. How do I add security features (encryption, disable printing) to my PDF document?

See [PDF Encryption](#). See also [PDF Post-processing](#).

6.5. How do I add document properties (title, author, etc.) to my PDF document?

See [Metadata](#).

6.6. How do I add watermarks to my PDF document?

See [PDF Watermarks](#).

6.7. The PDF is printed contorted!

Check the paper size in Acrobat settings and the "fit to page" print setting. Contorted printing is often caused by a mismatched paper format, for example if the setting is "US Letter" but the PDF was made for A4. Sometimes also the printer driver interferes, check its settings too.

6.8. How do I control the Acrobat bookmark display?

FOP supports the starting-state property of the XSL 1.1 [fo:bookmark](#) element which can be used for this. The color, font-style and font-weight properties on [fo:bookmark-title](#) are not yet supported, though.

7. IEx specific stuff

7.1. The FOP servlet is called multiple times!

This is a problem of Internet Explorer requesting the content several times. Please see the [notes on Internet Explorer](#) for more information.

7.2. How do I print PDF directly from the browser?

It depends whether you mean "printing to a printer under control of the server" or "printing on the client's printer".

For the first problem, look at the print servlet in the FOP examples. You'll have to gather any printer settings in a HTML form and send it to the server.

For the second task, you can use some client side script to start Acrobat Reader in print mode, or use a Java applet based on the FOP print servlet. This depends heavily on the client installation and should not be relied on except in tightly controlled environments.

See also <http://marc.theaimsgroup.com/?l=fop-dev&m=101065988325115&w=2>

8. General questions regarding XSLT, XSLFO, and basic XML

8.1. (FO) How do I vertically center an image or a table (or whatever)?

See [XSL-FO Vertical Centering](#).

8.2. (FO) How do I center a table horizontally?

See [XSL-FO Horizontal Centering \(Tables\)](#).

8.3. (FO) How to get page numbers printed on the "outer side" of the page (for books, for example)?

See [Recto/Verso Static Content Differences](#).

8.4. (FO) How do I use different static content for odd and even pages?

See [Recto/Verso Static Content Differences](#).

8.5. (FO) How do I get a special header on the first page?

See [Making the First Page Special](#).

8.6. (FO) How do I omit my headers on a blank page? How do I write "This page is left blank" on an intentionally blank page?

See [Blank Pages](#).

8.7. (FO) How do I print an Euro sign, a checkbox or other some other special symbols?

See [Special Characters](#).

8.8. (FO) How do I keep linebreaks and hard spaces? How do I get preformatted text displayed as expected?

See [Preformatting Content](#).

8.9. (FO) How do I print the total number of pages, for example "page 1 of 12"

See [Total Document Pages](#).

8.10. (FO) The header overlaps body content. The body extends into the footer.

See [Aligning Regions](#).

8.11. (FO) How do I get lines in the document, as separators, side bars or folding marks?

See [Drawing Lines](#).

8.12. (FO) How do I validate my FO document?

See [Validating XSL-FO](#).

8.13. (XML) How do I get a non-breaking space in FO? There are complaints about .

See [XML Special Characters](#).

8.14. (XML) How do I enter special characters in XML? There are complaints about undefined

entities, such as ü, which work in HTML.

See [XML Special Characters](#).

8.15. (XML) There are complaints about illegal characters and entities in the input.

See [Using HTML Character Names](#).

8.16. (XML) There are complaints about illegal bytes or characters in the input. There are odd characters in the result.

See [XML Encoding Issues](#).

8.17. (XSLT) How can I use the current date and time in my document?

See [Current Date and Time](#).

8.18. Can I use FOP to convert XHTML or HTML files to PDF?

You will get better results if you transform your source XML directly into XSLFO. It is tempting to use the XML->XHTML->XSLFO approach, because this apparently means there's only one transformation to maintain, but XHTML and publishing grade PDF are dissimilar enough that an "generic" XHTML->XSLFO(PDF) usually won't cut it, because the XHTML already lacks information from the original XML. It may be worthwhile to go from the source XML to an intermediate custom XML which on one hand is close enough to representation to allow simple transformation to either XHTML or XSLFO for most of the structure, but on the other hand retains enough semantic from the original XML that elements which must be handled differently in the two representations can be handled differently. Setting this up requires experience and most likely extensive prototyping.

8.19. I know it may be a bad idea to use FOP to convert XHTML or HTML files to PDF, but I want to do it anyway. Are there any tools available to help me?

The [FOP Resources page](#) has links to a tool called [html2fo](#) which can be used to make such a transformation. In addition, there are tools on the Antenna House [XSL-FO Tutorial and Sample](#) page.

9. Development-related Questions.

9.1. Where can I learn how the FOP docs and web site are built?

See FOP [Doc Management](#). ;-)

10. General suggestions. How to solve problems.

10.1. I think I have found a bug in FOP. What should I do?

See [the Bugs page](#) for information about bugs already reported and how to report new ones.

10.2. I have questions that are not addressed in this FAQ. Where should I post them?

- If your question is a development-related question, please see the [Developer FAQs](#).
- If you have a runtime exception or other runtime problem:
 - double-check the [Runtime FAQs](#).
 - `ClassNotFoundException`, `NoSuchMethodException` and `NoSuchFieldException` problems are almost always a problem with the local environment. Try to get local help first.
 - Check [Reported Issues](#) to see if this is a known problem.
 - If none of the above apply, post a question to the fop-dev mailing list.
- In the case where something works properly with another formatter, (AntennaHouse, PassiveTex, etc.) but doesn't work with FOP, please check the Release Notes, the [FOP Standards Compliance](#) document, and the remaining FAQ in this document. If not found there, look at the list of [Bugs Already Reported](#). If not found there, please post a question on the fop-user mailing list or [Open a New Bug](#).
- Question about how to use FOP, how to perform certain tasks with FOP or how to integrate FOP into another application should be posted to fop-user.
- XSLT specific stuff should go to the [XSL list](#). This includes problems with the language and XSLT HOW-TOs.
- Problems specific to a certain XSLT processor, like Xalan, Saxon or MSXML, should be handled by processor specific lists. This includes problems with deployment, processor specific extensions, suspected bugs and processor specific APIs. Note that JDK 1.4 comes with an XML parser and an XSLT processor which are older versions of Xerces and Xalan, respectively, and both have a number of annoying bugs. See <http://java.sun.com/developer/EJTechTips/2003/tt0311.html> and <http://java.sun.com/j2se/1.4.2/docs/guide/standards/> (or similar pages for later Java releases) for some hints on how to use more recent versions or other packages instead.
- Problems with servlet containers should be asked on the vendor specific lists for these software packages.
- More general questions regarding Java, including deployment, Java APIs, classpath issues and property definitions should be redirected to an appropriate Java specific list.