



AGETOR[®]

AXT XML Formatting
Filter

Table of contents

1	Introduction	3
2	Supported output formats	3
3	Using the AXT XML Formatting filter	4
4	Installation	4
5	Configuration	5
6	Introduction to XSL-FO	6
6.1	What is XSL-FO?	6
6.2	Examples	7
6.3	Links for further information about XSL-FO	7

1 Introduction

The AGETOR® AXT XML Formatting filter transforms XML business documents into formatted output for different types of media such as PDF-documents. The transformed document can be sent to customers, suppliers and partners by email or FTP. It can also be printed or saved to disc for future auditing and transaction traceability.

The AGETOR® AXT XML Formatting filter is based on the XSL-FO W3C recommendation. This standard defines a styling page-layout specifying how XML-documents are transformed into any given output format – in this case PDF, PostScript, PCL or SVG. XSL transformations generating XSL-FO are either written from scratch using a text/XML editor or using a design tool, for example “Stylesheet Designer” found in the product XML Spy 2004.



Besides a few examples, this guide is not intended as a complete guide on using XSL-FO. For more information about the use of XSL-FO go to the W3C website at <http://www.w3.org>.

2 Supported output formats

This version of the AXT XML Formatting filter supports 4 different output medias. This is provided using the Apache implementation of XSL-FO standard, namely FOP version 0.20.5 (read more at the website <http://xml.apache.org> and in section 6, Introduction to XSL-FO).



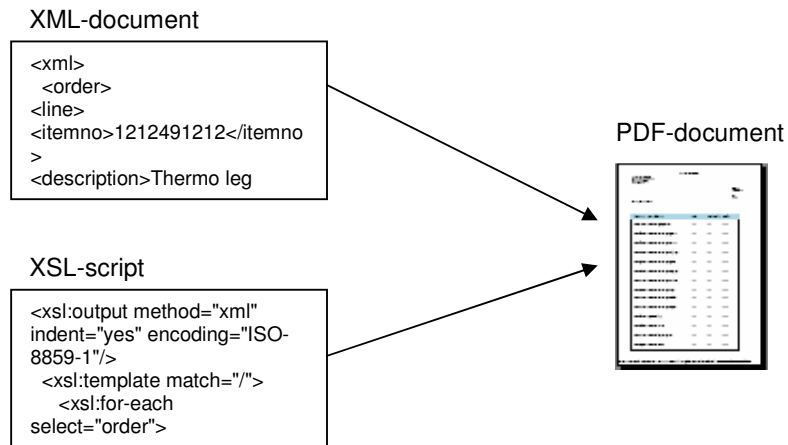
Please note the capabilities of the different formats are not the same either because of inherent limitations in the media itself or lack of development resources (See the website <http://xml.apache.org/fop/output.html>).

Media	Description
PDF	The best supported output media.
PostScript	PostScript is used for printing graphics and text.
PCL5	A printer language. The output is PCL 5 as documented in the "HP PCL 5 Printer Language Technical Reference Manual" (copyright 1990). Supported by many printer manufactures.
SVG	Scalable Vector Graphics is a W3C standard for describing two-dimensional graphics in XML.

3 Using the AXT XML Formatting filter

The AGETOR® AXT XML Formatting filter transforms XML business documents into formatted output for different types of media such as PDF-documents (See the section “Supported output formats”).

To use the AXT XML Formatting filter you plug it into the transformation process configured in “AXTConfiguration.xml” file as any other AXT filter. The filter takes the input document and makes an XSL-transformation producing in a XSL-FO-document. The resulting document will be rendered into a specified output media such as a PDF-document.



Key concepts:

- Easy-to-use AXT filter which is plugged into the AXT transformation chain.
- XSL-scripts transform XML-documents into the desired layout, which is hereafter transformed to the chosen output media automatically.
- Merge XML data with existing PDF designs – for example your existing standard forms for orders, invoices etc.
- The separation of presentation and data provides a flexible structure insuring reuse of working effort. This means you are able to transform the same XML-document into different layouts, potentially different customers, just by changing the XSL-script.
- No programming effort necessary – it is a matter of defining standard XML and XSL scripts – typically using a professional XML/XSL editor/designer.
- Experienced XSL writers can produce output after only one day of training.

4 Installation

This section gives you step-by-step information on how to install the XML Formatting filter.



The newest version of AXT XML Formatting filter is always available at the AGETOR download center at <http://www.agetor.com>

Before you begin the installation of the AXT XML Formatting filter, please make sure you have the following prerequisites installed:

- Java Development Kit (JDK) version 1.3 or newer.
- AGETOR Development Kit (ADK) version 2.0.0 or newer.
- AXT Basic version 2.0.0 or newer.
- Webserver with servlet runner e.g. Jakarta-Tomcat or Apache with Apache JServ.

Download the newest version of the AXT XML Formatting to your "AGETOR_HOME/install/packages" directory. If you are upgrading an existing installation the existing configuration will be retained.

- Start the AGETOR Install Tool from the Programs Menu (Windows) or by running "installer.bat" from an AGETOR prompt.
- If runned from an AGETOR prompt open an Internet browser and point to "http://localhost:8020".
- Type your login and password.
- Under "Product(s) ready to install" click on "AXT XML Formatting filter" and answer a few questions.

5 Configuration

The following is an example of a document entry added to "AXTConfiguration.xml". It takes an input XML-document and makes a XSL-transformation before it is rendered into a PDF-document.

```
<axt-conf xmlns="http://www.bording.dk/2001/AXT/config">
  <doc name="xmlformatting">
    <key name="url" value="xmlformatting" />

    <filter class="dk.bording.axt.tc.xsl.XSLFopFilter">
      <param name="xsl-script"
value="{AGETOR_HOME}\doc\axt_formatting\examples\example1.xsl"/>
      <param name="media" value="pdf"/>
    </filter>
  </doc>
</axt-conf>
```

The filter takes two parameters:

Filter name	"dk.bording.axt.tc.xsl.XSLFopFilter"	
Input format	XML	
Output format	Postscript	Text (ASCII)
	PCL	Binary
	SVG	Text (ASCII)
	PDF	Binary

Parameter name	Mandatory	Default	Parameter description
xsl-script	no		The XSL-script used to transform the XML-document into an XSL-FO document. If this parameter is not supplied, the filter won't make an XSL-transformation before rendering the XML-document.
media	no	pdf	The output format. The possible values are: "postscript", "pcl", "svg" or "pdf".

basedir	no		Sets the base directory where XSL-FO is looking for images.
---------	----	--	---

The filter can throw a number of java exceptions if something goes wrong:

Error-type	Description
javax.xml.transform.TransformerConfigurationException	Thrown if the XSLT-script can't be found.
javax.xml.transform.TransformerException	This error occurs if something went wrong while doing the XSL-transformation.
org.apache.fop.apps.FOException	Thrown if something is wrong with the XSL-FO document.
java.io.IOException	Thrown if something went wrong writing the output to the output stream.

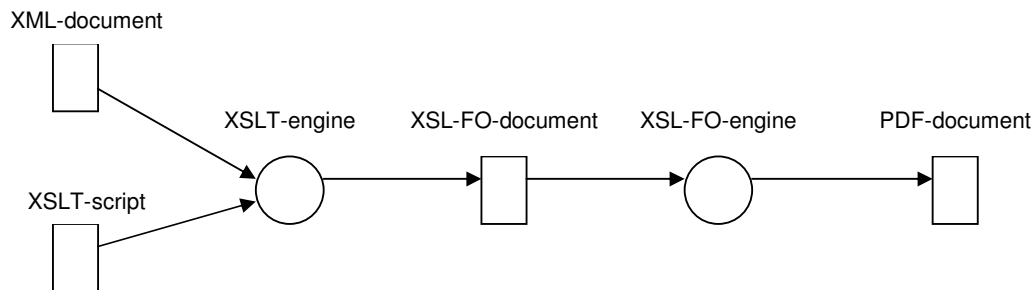
 Remember that generation of large PDF-documents can be rather memory intensive.

6 Introduction to XSL-FO

This section gives a quick introduction to using XSL-FO to format an XML-document with a few simple examples provided in the installation package. Please be aware that it is by no means a complete tutorial on the use of XSL-FO.

6.1 What is XSL-FO?

XSL-FO stands for Extensible Stylesheet Language Formatting Objects which is a standard defined by the World Wide Web Consortium (W3C – see their website at www.w3.org) for formatting XML-documents. The XML based markup language describes how XML data is converted into different output types, such as PDF, screen, printing or other medias.



XSL-FO is now an integrated part of the XSL standard also defined by the W3C but has its own language syntax. As XML has matured over the years there has been a growing need for a standard way to separate formatting of XML-documents from the contents of the XML-documents and their transformations. Together XSLT and XSL-FO describe a standard way transforming and styling information of XML-documents.

In the above figure, the XSLT-script is used to transform the contents of the XML-document into an XSL-FO document that describes both the formatting and content of the output. If the output document is rendered in an XSL-FO processor it can be outputted to, for example, a PDF-document.

6.2 Examples

The AXT XML Formatting filter package includes examples on the use of XSL-FO to format an XML-document. All the examples can be tested using the AXT XML Formatting filter with a document entry as described in section 5, Configuration.

For input to the examples, use the XML-document called “example1.xml” located in the folder, “AGETOR_HOME/doc/axt_formatting”.

Try transforming the input XML with the following style sheets using the filter.

Filename	Description
example1.xsl	The usually “hello world” example that only prints the text “Hello World” to the document.
example2.xsl	This example shows how to make a header and footer to a document and how to extract simple information from the XML-document.
example3.xsl	Shows how to make a table of all the lines in the order from the XML-document.

Please be aware that to fully understand the examples you need to have an understanding of XML, XML-namespaces and XSLT.

6.3 Links for further information about XSL-FO

Further information
http://www.w3.org provides information of the XSL-FO standard
http://xml.apache.org/fop contains some nice examples on how to use XSL-FO
http://www.w3schools.com contains a tutorial on how to use XSL-FO

It is highly recommend to use a tool to make the XSLT-script as they tend to become rather complex in their structure. One such tool is XMLSPY 2004 Enterprise (See <http://www.xmlspy.org>) that has a build in style sheet designer supporting XSL-FO.