

SAXON XSLT Processor Service

Welcome to the Saxon XSLT Processor Service

This local service provides an XSLT Transformation Engine ^[1] for transforming XML-encoded source documents using a supplied XSLT stylesheet. The service functions as a Java servlet and can be invoked using the following syntax:

```
hostname:[port-number]/saxon/SaxonServlet?source=[xml-source]&style=[xsl-stylesheet]&clear-stylesheet-cache=yes
```

Where:

- [hostname] – The hostname of the Fedora server; default is localhost.
- [port-number] – The port number on which the Fedora server is running; default is 8080.
- [xml-source] – The URL of the XML source file.
- [xsl-stylesheet] – The URL of the XSLT stylesheet file.
- clear-stylesheet-cache – An optional parameter indicating if the stylesheet cache is to be cleared. A value of "yes" signals to clear the stylesheet cache. If the parameter is omitted or has any other value, the stylesheet cache remains untouched.

Example:

```
http://localhost:8080/saxon/SaxonServlet?source=http://dl.lib.virginia.edu/data/xmltext/ead/viu03270&style=http://dl.lib.virginia.edu/bin/ead/xsl/document.xml
```

Configuring Authentication

If the Saxon Servlet needs to retrieve a stylesheet or source xml that is protected by basic authentication, it can be configured to do so by setting the parameters in the servlet's `web.xml` file. In the Fedora server distribution, this file can be found under the tomcat directory in `webapps/saxon/WEB-INF`.

The example below is what needs to be added for each host:port/path combination that the servlet needs to provide credentials for. The param-value is a colon-delimited username and password pair. When the Saxon Servlet needs to access a URL starting with the param-name, it will provide these credentials. Multiple init-param elements may be provided.

```
<init-param>
  <param-name>credentials for localhost:8080/fedora/getDS</param-name>
  <param-value>backendUser:backendPass</param-value>
</init-param>
```

Note that the text `credentials for` must be present at the start of the param-name, and the URL should be supplied without the protocol part (ie `localhost:8080/fedora/path` rather than `http://localhost:8080/fedora/path`).

The Saxon Servlet will need to be restarted in order for new values to take effect.

Passing parameters

Any URL query parameters passed to the Saxon servlet are available to your stylesheet (so long as you don't use one of the standard parameter names such as `source`, `style`, `clear-stylesheet-cache`). You need to declare a parameter using `xsl:param` in your stylesheet, which you can then reference as eg `$myParam`.

Example

URL

```
http://localhost:8080/saxon/SaxonServlet?source=http://dl.lib.virginia.edu/data/xmltext/ead/viu03270&style=http://dl.lib.virginia.edu/bin/ead/xsl/document.xml&param1=SomeText
```

Stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="param1"/>
  <xsl:template match="/">
    <output>
      <xsl:value-of select="$param1"/>
    </output>
  </xsl:template>
</xsl:stylesheet>
```

Passing additional XML documents

If a URL query parameter specifies the URL of an XML document, the contents of the document can be processed using the `xslt:document(...)` function.

For example, if you construct an SDef object that binds to a datastream and passes this to the Saxon servlet as a URL query parameter, you can access this additional datastream content (in addition to the `source` parameter as the main xslt input) using this method.

Example

URL

```
http://localhost:8080/saxon/SaxonServlet?source=http://dl.lib.virginia.edu/data/xmltext/ead/viu03270&style=http://dl.lib.virginia.edu/bin/ead/xsl/document.xml&param1=http://dl.lib.virginia.edu/data/xmltext/another-input
```

Stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="param1"/>
  <xsl:template match="/">
    <output>
      <xsl:value-of select="document($param1)/path/to/element"/>
    </output>
  </xsl:template>
</xsl:stylesheet>
```

Setting the Internet Media Type (MIMETYPE)

By default the Saxon servlet will return a media type of `text/html` if none has been defined in your stylesheet. If your stylesheet is generating XML you may find it useful to set the media type so the correct type will be returned - for instance to a browser via an object method (disseminator). To do this include an `xsl:output` element as a top-level element in your stylesheet, with a `media-type` attribute defining the media type.

Internet Media Type example

```
<xsl:output indent="yes" method="xml" media-type="text/xml"/>
```

^[1] This service uses the SAXON XSLT Processor from Michael Kay which is licensed under the Mozilla Public License (MPL). For additional information regarding the SAXON XSLT Processor, please refer to the Saxon project web site on SourceForge at: <http://saxon.sourceforge.net/>.