

CachedXSLTransformer

by Don Brown

Transformer features and configuration

Table of contents

1 Overview.....	2
2 Features.....	2
3 Configuration.....	2
3.1 Global.....	2
3.2 Transform-specific.....	3
3.3 Examples.....	4

1. Overview

This transformer uses a [JAXP 1.2](#) -implementing XSLT processor to transform the XML with one or more XSL files. It takes advantage of TRaX interfaces to compile and cache stylesheets to speed up the transformation process. It can handle both server and client side transformations.

2. Features

- Multiple XSL stylesheet support
- Configurable cache of compiled stylesheets
- Uses SAX to build the document
- Can be extended to add SAX filters before XSLT processing
- Supports client-side transformations
- Locates stylesheets from the servlet context
- Can filter what application resources are included in the transform
- Can auto-reload XSL files from disk when they have been modified since being cached

3. Configuration

3.1. Global

Property	Value	Status
contentType	The mime type of the transformed content	Required
autoReloadTemplates	true false (default). If true, the cached Templates will be checked to see if the stylesheet has been modified since and reload as necessary.	Optional
allowRenderParameter	true false (default). If true, the type of rendering can be specified as a request parameter. For example, "render=client" will force a client-side transformation.	Optional
resolveFromTemplatePath	false true (default). If true, URIs used in xsl:import, or xsl:include are resolved from the directory of the parent	Optional

	stylesheet. Otherwise, URIs are resolved from the root directory of the web application.	
transformerFactoryClass	The name of the JAXP transformer factory implementation to use.	Optional

Table 1: Global stxx.properties properties

Note:

If you are having problems with NullPointerExceptions when stxx is loading stylesheets, it could be because your servlet container doesn't implement request.getRealPath() in a useful manner. If this is the case, set both resolveFromTemplatePath and autoReloadTemplates to "false" and stxx will load the stylesheets using ServletContext.getResourceAsStream().

3.2. Transform-specific

These parameters are specified in stxx 1.2+ pipeline configuration format. See [Configuring with Struts 1.1](#) for more information.

Parameter	Value	Status
path	The path to the XSL file	Required and can be repeated
debug	true false (default)	Optional
render	client server (default)	Optional
messages	Matches which application resources will be included in the transformation.	Optional
attachRequestParameters	false true (default)	Optional
attachRequestAttributes	false true (default)	Optional
attachForm	false true (default)	Optional
attachResources	false true (default)	Optional
attachErrors	false true (default)	Optional
attachMessages	false true (default)	Optional
OTHER	Any other parameter will be passed to the XSLT as a stylesheet parameter.	Optional

Table 1: Transform-specific parameters

All transform parameters will be passed to the XSLT as stylesheet parameters

If you have a browser accessing stxx that can support rendering of XSL, you can off-load the transformation of the XML. If "client" is set, when stxx processes the XML file, it will add a processing instruction at the top of the XML file, pointing to the XSL stylesheet on the server running stxx. Your browser should be able to handle it from there.

To limit which application resources messages are included in the transformation, you can supply one or more messages parameters. They match application resource keys using Perl 5-compatible regular expressions.

3.3. Examples

Global configuration:

```
stxx.transformer.html.class=com.oroad.stxx.transform.CachedXSLTransformer
stxx.transformer.html.mimeType=text/html
```

Transform-specific configuration:

```
<pipeline match="simple/*.dox">
  <transform type="html">
    <param name="path"      value="/{1}.xsl" />
    <param name="render"    value="server" />
    <param name="messages" value="simple.*" />
  </transform>
</pipeline>
```