

stxx.properties Configuration

by Don Brown

Advanced configuration

Table of contents

1 General Properties.....	2
2 Transformers.....	3
3 Transform Selection.....	3
4 Serialization.....	4
5 XML Building Rules.....	4

1. General Properties

The stxx.properties file is where all of the stxx specific configuration properties are kept. There are a number of general properties that can be assigned in this file, they are:

Property	Value
stxx.allowURLDebug	true false

This determines whether "debug=true" can be appended to a query string and debug information will be sent back (if the transformer supports it).

`http://localhost:8080/stxx/index.do?debug=true`

Would return the XML data without transforming it. While this is useful for development, it is probably better to disable this for production use.

Property	Value
stxx.writeXMLDebug	true false

When debugging information is requested either by use if the previous method or setting the debug property for your transform, this value determines whether that same information will be written to disk. Unless you need the debugging files, turn this off to prevent your disk being filled up with debug files during development/testing.

Property	Value
stxx.debugXMLPath	Path to log file location

By default, the debugging files mentioned above are written to the /WEB-INF directory. If you wish for them to be written elsewhere, uncomment this property and set it to an absolute path of a directory on your server. If you are using a version of Windows, remember to escape your backslashes, for example ex: `c:\\Tomcat4\\logs`

Property	Value
stxx.alwaysUseChainedRoot	true (default) false

By default, when action chaining has NOT resulted in multiple XML documents, the document created in a Action is modified by stxx to add other elements like request attributes. By setting this property to true, stxx will always use the chained root node as the root node of the document and add the action-created document to it.

Property	Value
stxx.chainedRootNode	The name of the chained root node

When action chaining has resulted in multiple XML documents or the `stxx.alwaysUseChainedRoot` property is true, stxx places their root nodes under one new root node. This property determines the name of that node.

Property	Value
<code>stxx.useCSVParameterFormat</code>	true false (default)

In previous versions of stxx, multiple values for a request parameter were serialized as comma-separated values, however, the current behavior is separate "value" elements for each value. To revert to the previous CSV behavior, set this property to true.

2. Transformers

Property	Value
<code>stxx.transformer.TYPE.class</code>	The name of the transformer class
<code>stxx.transformer.TYPE.cache.class</code>	The name of the cache class
<code>stxx.transformer.TYPE.cache.name</code>	The unique name of the cache
<code>stxx.transformer.TYPE.cache.size</code>	The maximum number of elements allowed in the cache
<code>stxx.transformer.TYPE.cache.expiryTime</code>	Expiration time of the elements in the cache in seconds
<code>stxx.transformer.TYPE.CUSTOM</code>	Custom properties specific to the transformer

The preceding properties deal with what types of transformation stxx will recognize. It also allows you to customize the cache that is available to the Transformer. Only the first property is required for a transformer type. If you do not specify the next three properties, values of properties starting with `stxx.default` will be used. You can further customize each transformer by including additional properties unique to the transformer class by using the CUSTOM properties that will be read in by the transforming class

This Transformer, `CachedXSLTransformer`, uses a custom property named "mimeType" that is used to set the mime type of the transformation. This is how it is implemented in the `stxx.properties` file:

```
stxx.transformer.xml.class = com.oroard.stxx.transform.CachedXSLTransformer
stxx.transformer.xml.mimeType = text/xml
```

3. Transform Selection

Each transform element is given a name. You can specify how stxx will use that name to choose the correct transform. The default transform selector is the UserAgentSelector which will try to match the browser type to choose the appropriate stylesheet customized for the browser type.

Property	Value
<code>stxx.transformSelector.class</code>	The chosen Selector class

4. Serialization

To serialize a request parameter, request attribute, action error, or message resource, stxx calls an implementation of the serializer interface for the appropriate XML format. For example, the CachedXSLTransformer uses an implementation of the SAXSerializer to serialize information. Other transformers could use other types of XML formats that might have their own Serializer interfaces.

The default serializer uses a custom serializer that recognizes JavaBeans, JDOM, DOM, arrays, collections, and primitive wrappers. To have more control over how an object is serialized, extend these classes and override the methods of your choice. For example, if you wanted to use Castor XML to serialize request attributes, override the `serializeRequestAttribute` method.

Property	Value
<code>stxx.serialize.sax.class</code>	The class that serializes information as SAX events. Used by all built-in stxx transformers.
<code>stxx.serialize.FORMAT_TYPE.OPTION_NAME</code>	A custom serializer property where <code>FORMAT_TYPE</code> is the XML format type and <code>OPTION_NAME</code> is the name of the custom option.

5. XML Building Rules

You can configure what information will be attached to the XML in which situations.

Property	Value
<code>stxx.attach.rulesClass</code>	The implementation of BuilderRules that controls what XML is attached in what circumstances. The default class uses the following properties.
<code>stxx.attach.resources.include</code>	Matches which requests to attach resource XML
<code>stxx.attach.resources.exclude</code>	Matches which requests not to attach resource

	XML
stxx.attach.requestParameters.include	Matches which requests to attach request parameters XML
stxx.attach.requestParameters.exclude	Matches which requests not to attach request parameters XML
stxx.attach.requestParameters.ignore	Matches which request parameters to ignore
stxx.attach.requestAttributes.include	Matches which requests to attach request attributes XML
stxx.attach.requestAttributes.exclude	Matches which requests not to attach request attributes XML
stxx.attach.requestAttributes.ignore	Matches which request attributes to ignore
stxx.attach.requestParameters.exclude	Matches which requests not to attach request parameters XML
stxx.attach.errors.include	Matches which requests to attach action errors XML
stxx.attach.errors.exclude	Matches which requests not to attach action errors XML
stxx.attach.form.include	Matches which requests to attach action form XML
stxx.attach.form.exclude	Matches which requests not to attach action form XML
stxx.attach.messages.include	Matches which requests to attach action messages XML
stxx.attach.messages.exclude	Matches which requests not to attach action messages XML

XML attachment configurations allow you to have more control over what extra information is serialized and added to your XML document before transformation. For all information types, there will be two lines of configuration - include, and exclude. For the request attributes and parameters, there will be the additional "ignore" configuration line.

In order for information to be attached as XML, the absolute path of the requested URL must match the regular expression defined for the "include" and not the "exclude". Additionally, in the case of request attributes and parameters, the name of the parameter/attribute must not match the regular expression defined for "ignore".

The regular expressions are compatible with Perl5, for example:

```
stxx.attach.requestParameters.include=.*public/.  
stxx.attach.requestParameters.exclude=.*public/other/.  
stxx.attach.requestParameters.ignore=ID_.*
```

In this case, the transformation for the URL `"/stxx/public/index.do"` will have all request parameters that start with `"ID_"` attached to the XML but the URL `"/stxx/public/other/index.do"` will not. In forming your regular expressions, please note to use a `"\"`, you must type `"\"` due to the properties format. Therefore to match a space, type `"\\S"`.