XsltCrosswalk

XsltCrosswalk.zip

Configurable XSLT-driven Crosswalk

This page describes a powerful crosswalk plugin which is part of the Crosswalk Plugins family. The code was added in DSpace 1.4.

The XSLT Crosswalk classes let you can create many different crosswalks between DSpace internal data and any XML format. Just add another XSLT (XSL transformation) stylesheet to the configuration to create a new crosswalk. Each stylesheet appears as a new plugin name, although they all share the same plugin implementation class.

The XML transformation must produce (for submission) or expect (for dissemination) a document in DIM - DSpace Intermediate Metadata format. See DSpace Intermediate Metadata for details.

Configuration

The DSpace configuration is prepared as follows:

Configuring a Submission Crosswalk

A submission crosswalk is described by a DSpace configuration key like:

```
crosswalk.submission.<pluginName>.stylesheet = <path>
```

The **`pluginName`** is the plugin name given to the Plugin Manager, and the **`path`** value is the pathname (relative to `[dspace]/config/`) of the crosswalk stylesheet, e.g. "`mycrosswalk.xslt`".

For example, this configures a crosswalk named "LOM" using a stylesheet in

```
[dspace]/config/crosswalks/s-lom.xsl
```

(under the DSpace "home" directory):

```
crosswalk.submission.LOM.stylesheet = crosswalks/s-lom.xsl
```

Configuring a Dissemination Crosswalk

A dissemination crosswalk is described by a DSpace configuration key like

```
crosswalk.dissemination.<pluginName>.stylesheet = <path>
```

The **pluginName** is the plugin name given to the Plugin Manager, and the *path*value is the pathname (relative to `[dspace]/config/`) of the crosswalk stylesheet, e.g. "`mycrosswalk.xslt`"

The disseminator also needs to be configured with all of the XML Namespaces that might appear in its output, including the prefix and URI for each one. It also needs a value to include in the `schemaLocation` attribute. These are configured on additional properties in the DSpace Configuration, i.e.:

```
crosswalk.dissemination.<pluginName>.namespace.<prefix> = <namespace-URI>
crosswalk.dissemination.<pluginName>.schemaLocation = <Schema Location string>
crosswalk.dissemination.<pluginName>.preferList = <boolean>
```

`preferList` indicates whether the output should be a list of elements (like the Dublin Core XML formats) or a single element. All it does is change the value returned by the plugin's `preferList()` method. The default, which is correct in most cases, is `false`.

For example, a plugin named `MODS`:

```
crosswalk.dissemination.MODS.stylesheet = crosswalks/mods_out.xsl
crosswalk.dissemination.MODS.namespace.mods = http://www.loc.gov/mods/v3
crosswalk.dissemination.MODS.schemaLocation = http://www.loc.gov/mods/v3 http://www.loc.gov/standards/mods/v3
/mods-3-0.xsd
```

You can configure two (or more) names to point to the same crosswalk, just add two configuration entries with the same path, e.g.

```
crosswalk.submission.MyFormat.stylesheet = crosswalks/myformat.xslt
crosswalk.submission.almost_DC.stylesheet = crosswalks/myformat.xslt
```

Testing Dissemination Crosswalks

With DSpace up to version 1.8.2 you can test dissemination stylesheets very easily by configuring an OAI-PMH output format in the OAI configuration file, named the same as your dissemination plugin and implemented by `org.dspace.app.oai.PluginCrosswalk`. Then just request a document's metadata in the given format to see what the disseminator produces.

The new OAI in DSpace 3.0 uses the "xoai" format as input so stylesheets expecting the "DIM" format do not apply. From DSpace 3.1 on it is possible to test dissemination crosswalks by using a command line utility. To invoke the dissemination crosswalk tester, run the class XSLTDisseminationCrosswalk as follows:

```
[dspace]/bin/dspace dsrun org.dspace.content.crosswalk.XSLTDisseminationCrosswalk <plugin name> <handle>
[output-file]
```

For example, you can test the marc plugin on the handle 123456789/3 with:

```
[dspace]/bin/dspace dsrun org.dspace.content.crosswalk.XSLTDisseminationCrosswalk marc 12345678/3
```

Informations from the script will be printed to stderr while the XML output of the dissemination crosswalk will be printed to stdout. You can give a third parameter containing a filename to write the output into a file, but be careful: the file will be overwritten if it exists.

Testing Ingestion Crosswalks

To help you test submission stylesheets, the XSLT ingestion crosswalk includes a command-line utility, since otherwise you would have to try ingesting a new Item for each test. The utility transforms a metadata
document you supply in a file and returns the resulting DIM.

To invoke the crosswalk tester, run the class as follows:

```
[dspace]/bin/dspace dsrun org.dspace.content.crosswalk.XSLTIngestionCrosswalk <plugin name> <input-file>
```

For example, you can test the `LOM` plugin on the file `test.xml` with:

```
[dspace]/bin/dspace dsrun org.dspace.content.crosswalk.XSLTIngestionCrosswalk LOM test.xml
```

Add the `-l` option to pass the submission stylesheet a list of elements instead of a whole document, as if the List form of the `ingest()` method had been called, e.g.:

```
[dspace]/bin/dspace dsrun org.dspace.content.crosswalk.XSLTIngestionCrosswalk -l LOM test.xml
```

In any case, the output will be prettyprinted XML of the DIM document returned by the stylesheet.

Usage

You must use the `PluginManager` to instantiate an XSLT crosswalk plugin, e.g.

```
IngestionCrosswalk xwalk = PluginManager.getPlugin(IngestionCrosswalk.class, "LOM");
```

Since there is significant overhead in reading the properties file to configure the crosswalk, and a crosswalk instance may be used any number of times, we recommend caching one instance of the crosswalk for each alias and simply reusing those instances. The `PluginManager` does this automatically.

Auto-reloading

```
PluginManager
```

This plugin will automatically reload any XSL stylesheet that was modified since it was last loaded. This lets you edit and test stylesheets without restarting DSpace.