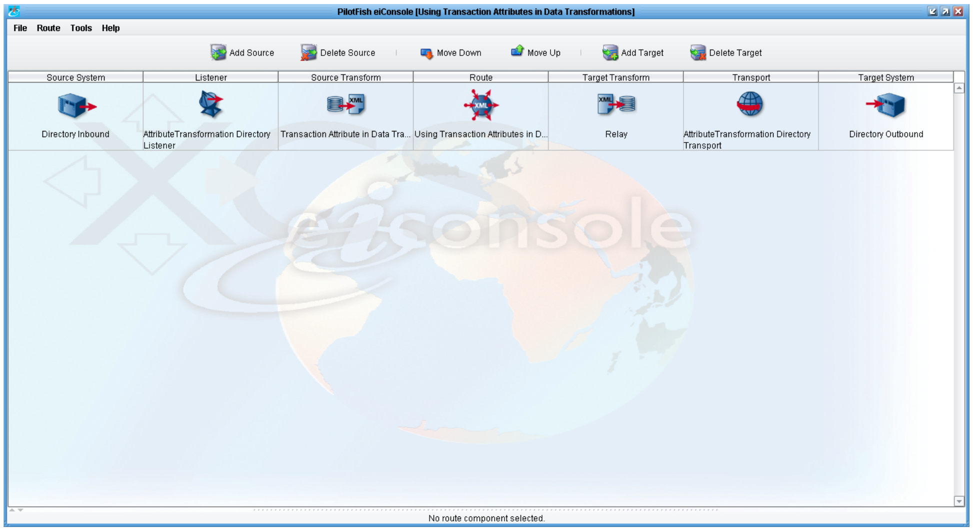
Data Mapping – Using Attributes

Overview

This interface demonstrates how to utilize transaction attributes in data transformation; this is useful for getting meta-information from listeners, such as when you need to know the filename a file was picked up as. It can also be used to store information that cannot be stored in the current transaction data, or for handling things such as images.

Steps

We'll start by using the simple Directory to Directory route created in previous exercises:

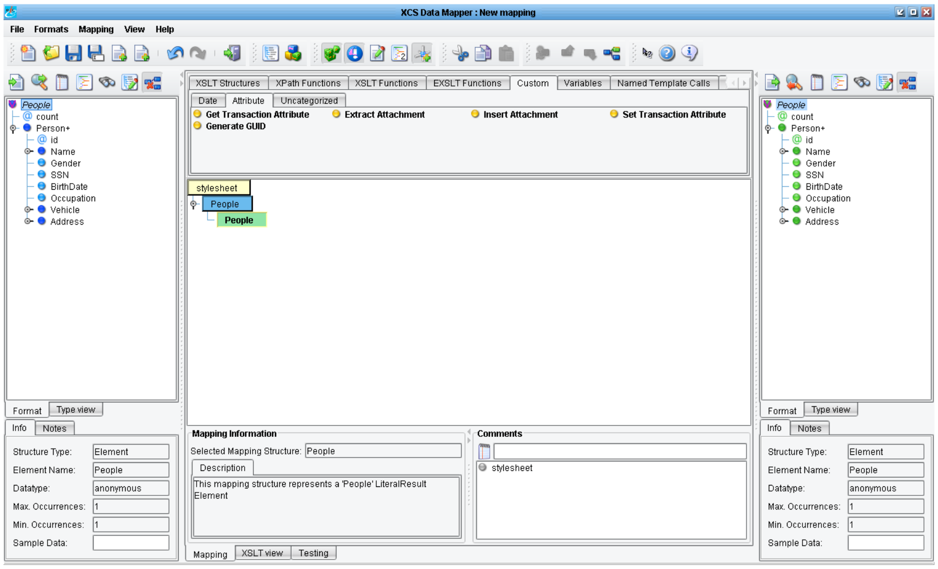


Source System	Listener	Source Transform	Route	Target Transform	Transport	Target System
Directory Inbound	AttributeTransformation Directory Listener	Transaction Attribute in Data Tra...	Using Transaction Attributes in D...	Relay	AttributeTransformation Directory Transport	Directory Outbound

No route component selected.

Transformation Configuration

Start by creating a simple transform of "PeopleA.xml" to "PeopleA.xml":



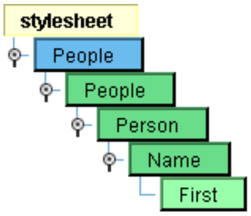
Structure Type: Element
Element Name: People
Datatype: anonymous
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data:

Mapping Information
Selected Mapping Structure: People
Description
This mapping structure represents a 'People' LiteralResult Element

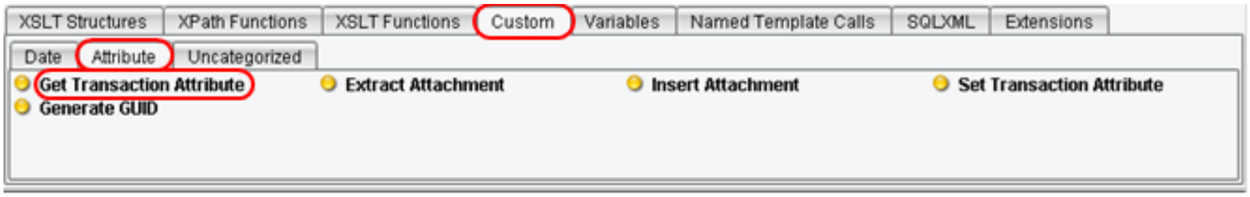
Comments
stylesheet

Structure Type: Element
Element Name: People
Datatype: anonymous
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data:

The transformation is relatively simple, performing a near one-to-one mapping. For the purpose of this exercise we will populate the /People/Person/Name/First with an “Attribute”:

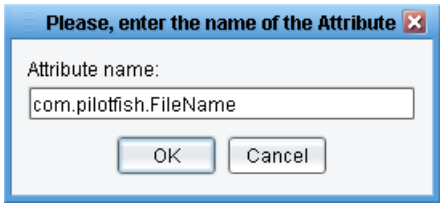


To add a transaction attribute, navigate to the “Custom” tab in your Data Mapper palette, and select “Attribute” - “Get Transaction Attribute” automatically generates the XSLT code necessary to read data in from a transaction attribute in your XSLT:



You will be asked to provide an attribute name:

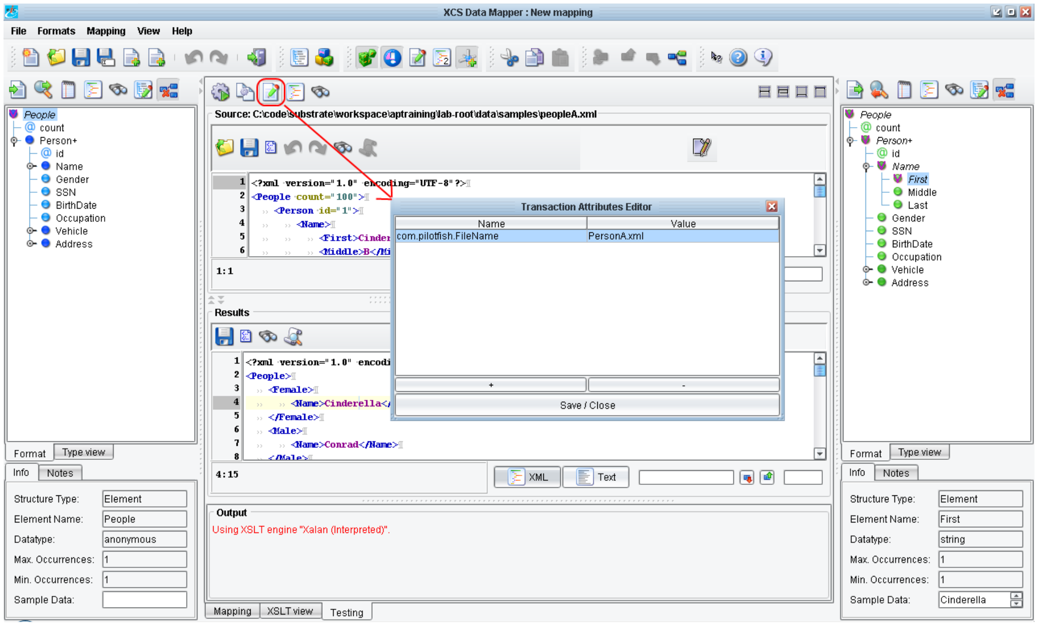
com.pilotfish.FileName



Please, enter the name of the Attribute

Attribute name: com.pilotfish.FileName

You can test this directly in your XSLT; go to the Testing tab, and on the testing toolbar is an icon of a green pencil and notepad. Clicking this will bring up an attribute editor. This data will be given directly to the transformer when this test is executed.



XCS Data Mapper : New mapping

Source: C:\code\substrate\workspace\aptraining\lab-root\data\samples\peopleA.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<People count="100">
    <Person id="1">
        <Name>
            <First>Cinder
            <Middle>B</Mi
```

1:1

Transaction Attributes Editor

Name	Value
com.pilotfish.FileName	PersonA.xml

```
<?xml version="1.0" encodi
<People>
    <Female>
        <Name>Cinderella</
    </Female>
    <Male>
        <Name>Conrad</Name>
    </Male>
```

4:15

Output

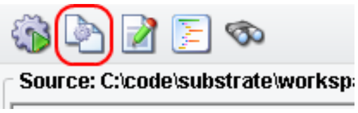
Using XSLT engine "Xalan (Interpreted)".

Structure Type:	Element
Element Name:	People
Datatype:	anonymous
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	

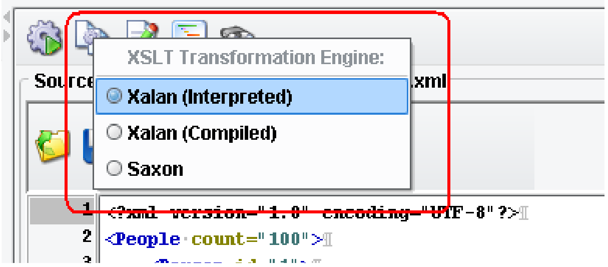
Structure Type:	Element
Element Name:	First
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	Cinderella

Remember to tab out of the input fields before hitting “Save/Close”.

If you receive an error, ensure that the XSLT engine is set to “Xalan (Interpreted)”:



Source: C:\code\substrate\worksp



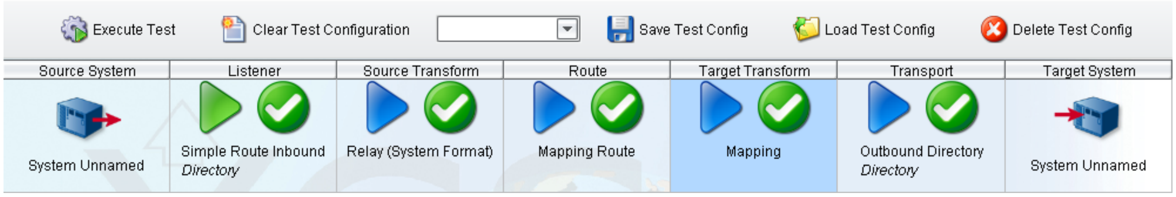
XSLT Transformation Engine:

- ● Xalan (Interpreted)
- ○ Xalan (Compiled)
- ○ Saxon

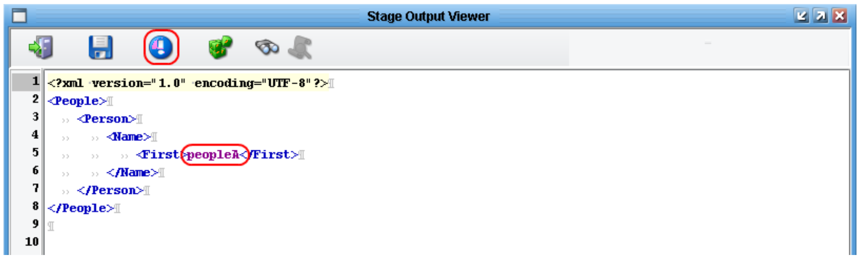
```
<?xml version="1.0" encoding="UTF-8"?>
<People count="100">
```

Route Testing

Save and close the XSLT editor. Save the route and switch to “Test Mode”:

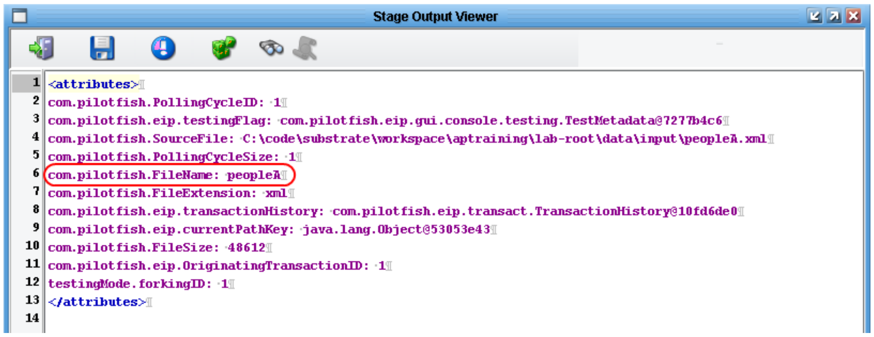


View the “Stage Output” after the “Target XSLT Transform” step. You should see something like the following:



```
<?xml version="1.0" encoding="UTF-8"?>
<People>
    <Person>
        <Name>
            <First>peopleA</First>
        </Name>
    </Person>
</People>
```

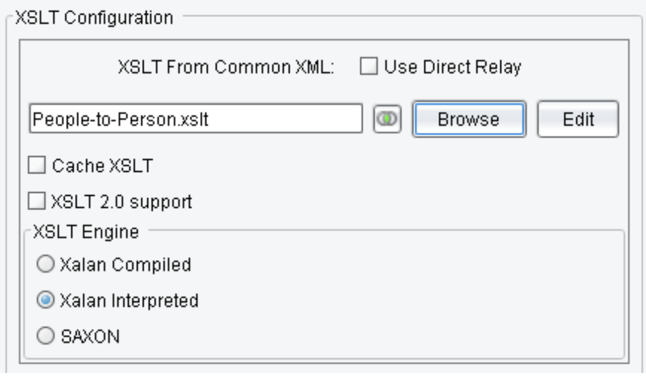
Switch to the “Attribute” view using the blue exclamation icon highlighted above, to see the attributes generated in the route:



```
<attributes>
com.pilotfish.PollingCycleID: 1
com.pilotfish.eip.testingFlag: com.pilotfish.eip.gui.console.testing.TestMetadata@7277b4c6
com.pilotfish.SourceFile: C:\code\substrate\workspace\aptraining\lab-root\data\input\peopleA.xml
com.pilotfish.PollingCycleSize: 1
com.pilotfish.FileName: peopleA
com.pilotfish.FileExtension: xml
com.pilotfish.eip.transactionHistory: com.pilotfish.eip.transact.TransactionHistory@10fd6de0
com.pilotfish.eip.currentPathKey: java.lang.Object@53053e43
com.pilotfish.FileSize: 48612
com.pilotfish.eip.OriginatingTransactionID: 1
testingMode.forkingID: 1
</attributes>
```

Notice that the value in the XML data matches the attribute highlighted above.

If you instead receive an error, ensure that the XSLT engine is set to “Xalan (Interpreted)”:



XSLT Configuration

XSLT From Common XML: □ Use Direct Relay

People-to-Person.xslt

□ Cache XSLT

□ XSLT 2.0 support

XSLT Engine

○ Xalan Compiled

● Xalan Interpreted

○ SAXON