Vantage
Application Note

Using Vantage to Ingest XML into a Label

This App Note applies to Vantage versions 6.2 and later

Synopsis

Media files are sometimes accompanied by XML text attachments. For example, a media file containing a news story might be accompanied by the text of the story with timecode markings in the text for use during editing by an editor operator. Another scenario might include a media file destined for posting to a media store accompanied by XML files containing metadata about the media.

Vantage can process the media file, and the accompanying XML document can be transformed with a user-defined XSLT style sheet. Together, the input XML document transformed by an XSLT style-sheet, apply their results to a stage of the workflow or the workflow output.

In the example presented in this app note, we supply bit rate and file name data in an XML file accompanying ingested media. We create an XSLT style sheet to transform the XML so that a Vantage workflow can extract the data to a label and variables.

The workflow then transcodes the media and sets the transcoded output bit rate and file name parameters based on the variable data extracted from the XML.

Version Requirements

The following version is required:

- Vantage 6.2 or later.

If you do not have a supported version, please contact Telestream to upgrade your software.

Licensing

The Transcode Connect or Transcode Pro Connect license is required.

Example Project

When a media file is ingested into Vantage, an associated XML text file can also be ingested using an Associate action. This file typically provides information about the media that Vantage can pass along with the media at the output of the Vantage workflow.

Additionally, a standard XSLT style sheet file can be imported into Vantage via the *Add new style sheet* button in the Vantage Management Console Style Sheets panel. The XSLT style sheet provides information about how to format or extract the contents of the XML file.

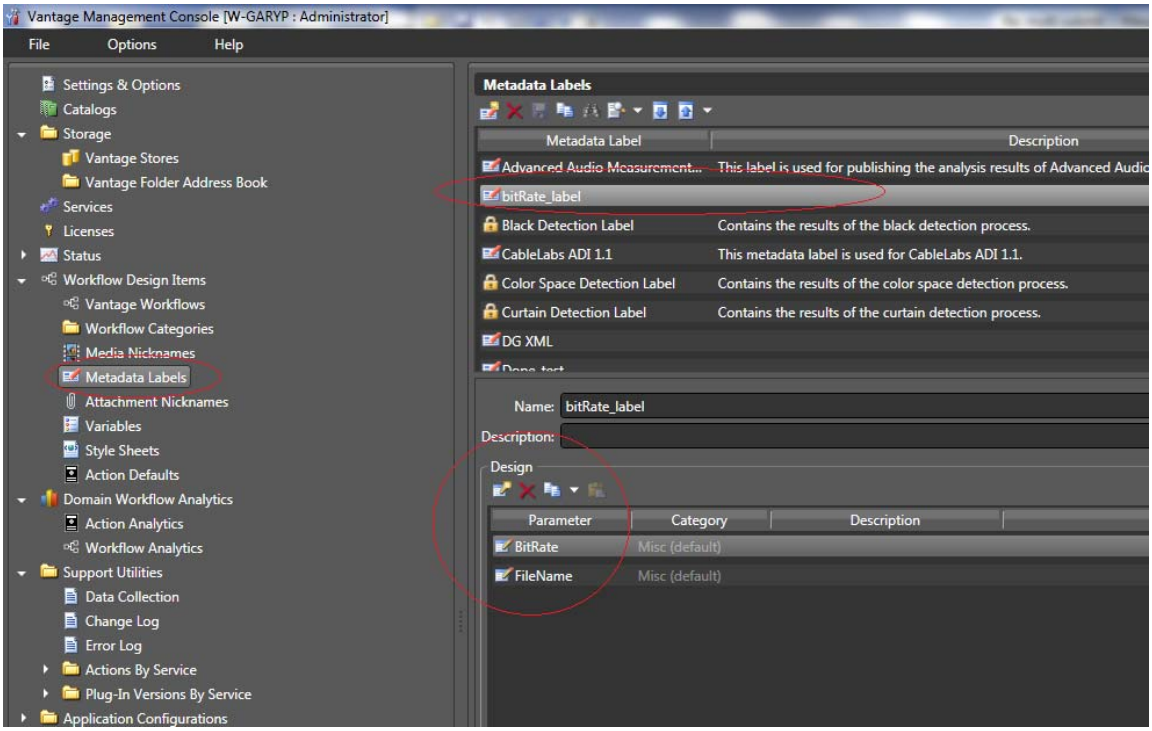
The following procedure describes how to create an XSLT style sheet to match and transform XML files that contain data related to ingested media. Once you have a style sheet, you can use it in a Vantage workflow that ingests media and its accompanying XML data file, and transforms the XML data into a Vantage label. The label can be used to create variables which hold the captured XML data and forward it to parameters included with the transcoded media output.

Building an XSLT to Match Ingested XMLs

1. Start by identifying an XML document to serve as a template. This must be an XML file that contains the same elements in the same format as all the other XML documents you plan to ingest into Vantage. Our example for this procedure is an XML file with *bit rate* and *file name* elements that contain pre-defined data to be captured by a Vantage workflow. Here are the contents of our XML template file:

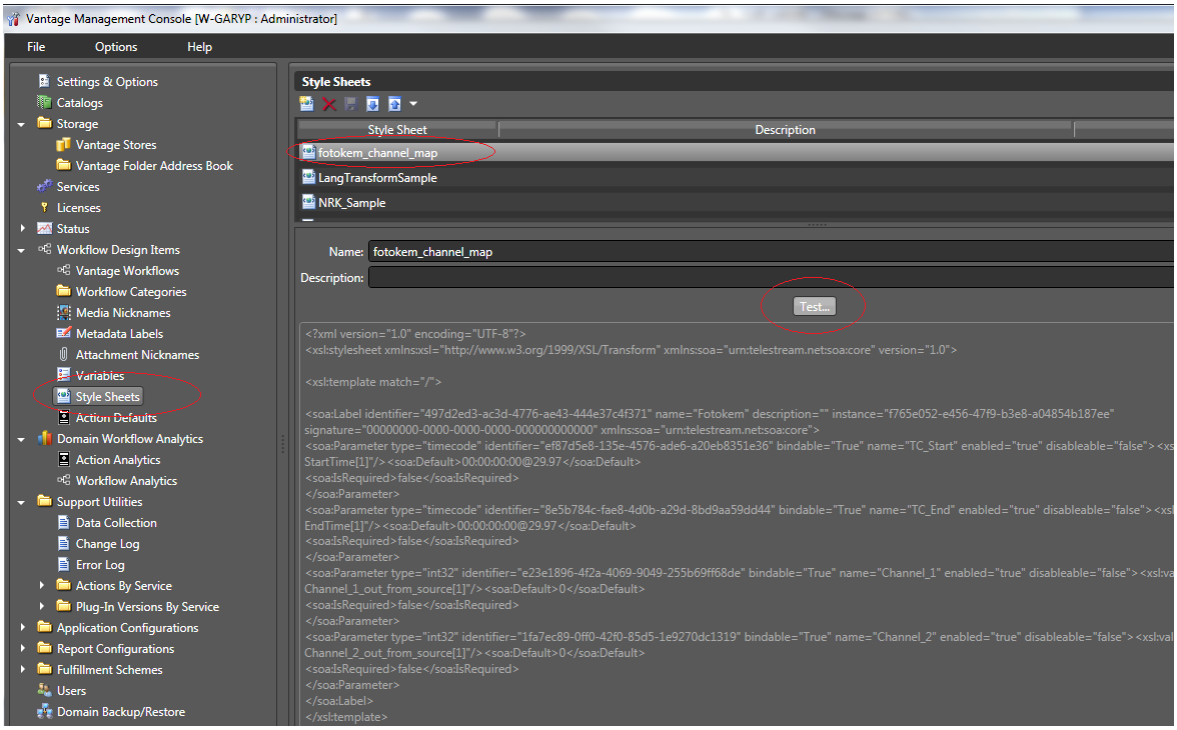
```
<Clip>
  <Bitrate>2000000</Bitrate>
  <FileName>SuperMan.mov</FileName>
</Clip>
```

2. In the Vantage Management Console, create a Vantage label that will hold the respective values from your input XML file. In this example the label holds a bit rate (integer) and a file name (string). You must create your own labels to meet your particular needs.

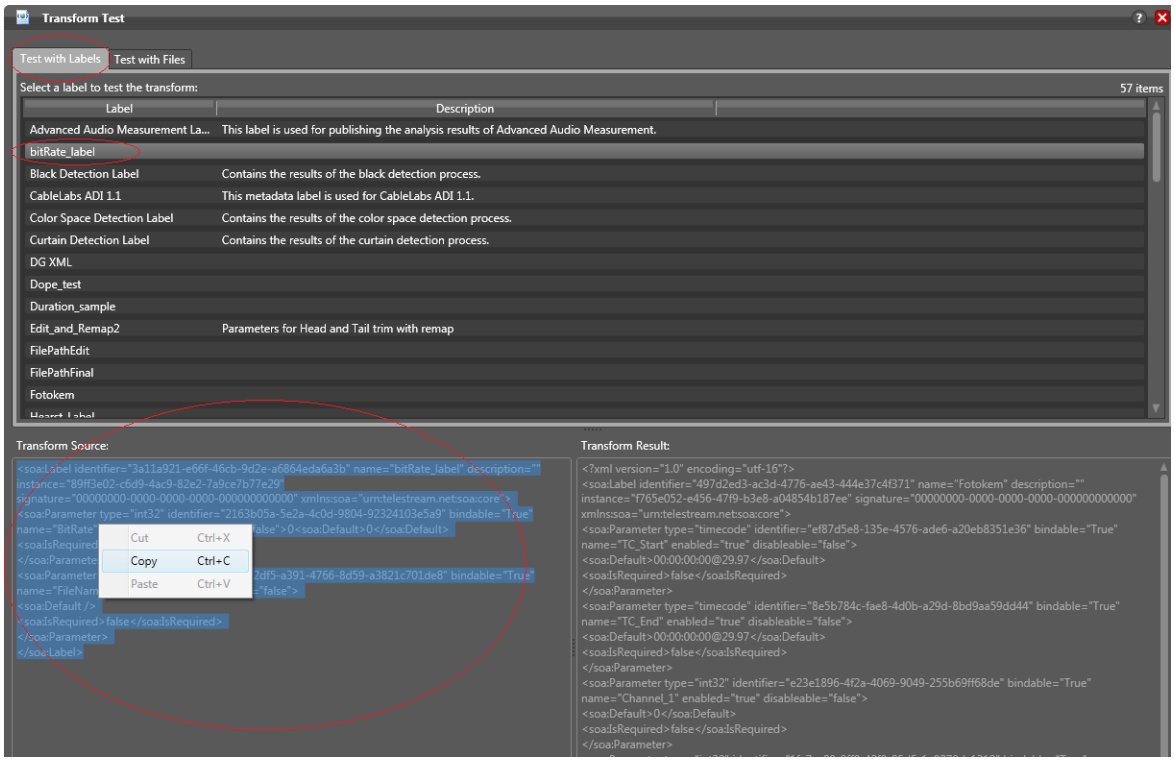


3. Next, you need to select an XSLT style sheet to use in transforming the XML contents of the label. In the VMC, select Style Sheets and select any properly composed XSLT style sheet (for this purpose, the style sheet you use doesn't matter).

4. Select the Test button. This opens the Test window for testing your label.

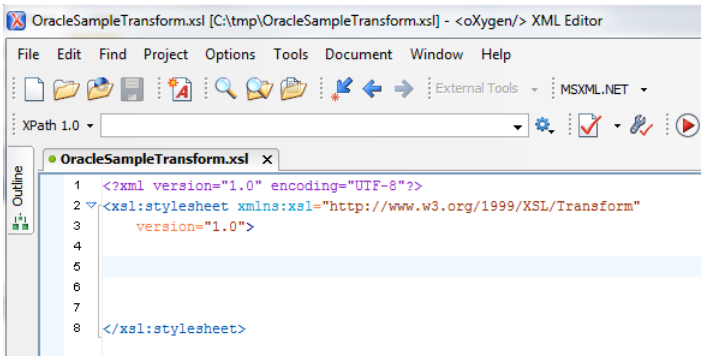


5. Select the label that you just created (bitRate_label in our example).



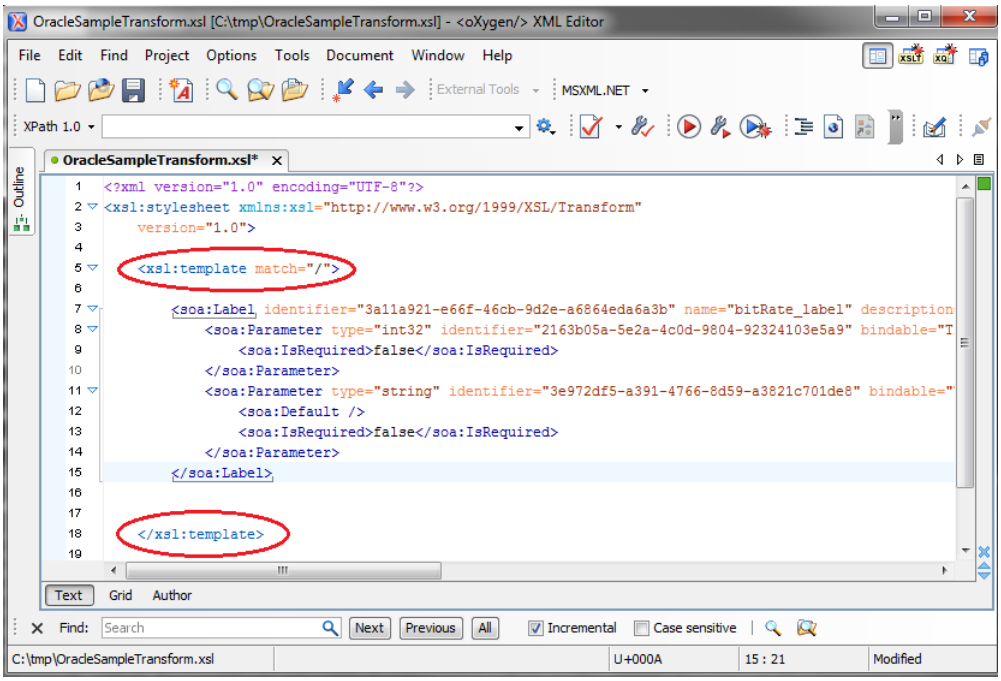
6. At the bottom of the Test screen, highlight and copy the Transform Source text. (Essentially, this process converts the label contents into the proper format for the body of an XSLT file.)

7. Use an XSLT editor to create a new XSLT document and paste the copied bitRate_label Transform Source contents into the middle of the document (after the xsl header). In our example, we used the Oxygen integrated development environment (IDE) to create a new project for an XSLT style sheet and save it as OracleSampleTransform.xsl.



8. Add an opening Template command and Match statement directly above the pasted label text and a corresponding closing statement directly after the text, as shown below.

   Typically, your transformation style sheet will start with a Template command and a Match statement. The Match statement identifies where to start a path search to locate a value in the XML document . In our example, we start our path search at the root element denoted by the "/". Thus, in the screen shot we have added the <Template Match="/"> commands (opening and closing).

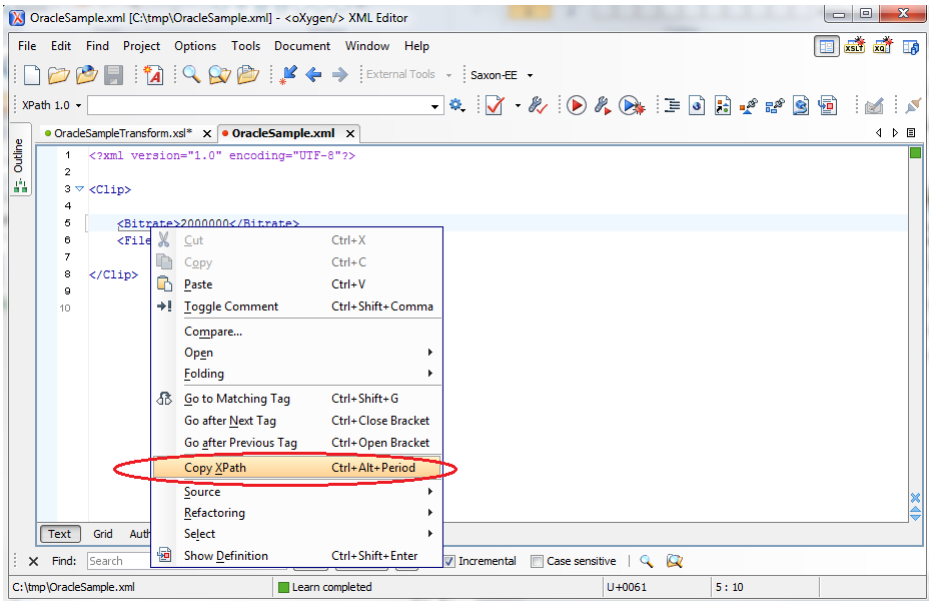


   If you look at the above screen shot of our Vantage label you'll see that we use the soa: namespace for all of our elements. In the next step, we need to make sure our style sheet is aware of this name space.

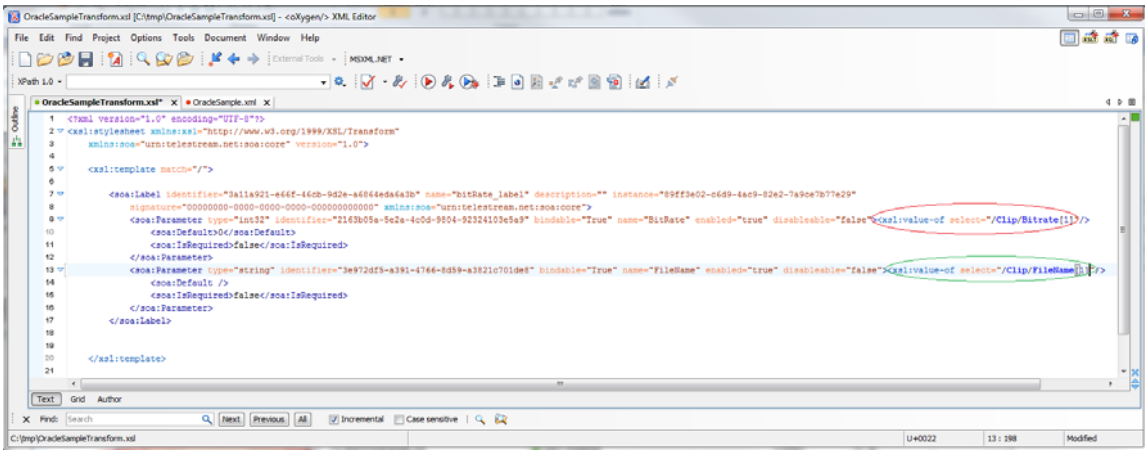
9. Copy the soa: namespace URL that is in the label we just pasted and place it in the header of the XSLT document.

We have now told the XSLT what the output will be and added the name space, but we have not told the XSLT where to get the values that will get placed into the two elements within the label contents—the bit rate and the file name. These values are present in the XML file you are submitting to the workflow, so the next step is to point this XSLT to the XPath of the input XML. You can do this with this XSLT command as shown in the following steps: ['value-of-select' = <some xpath>].

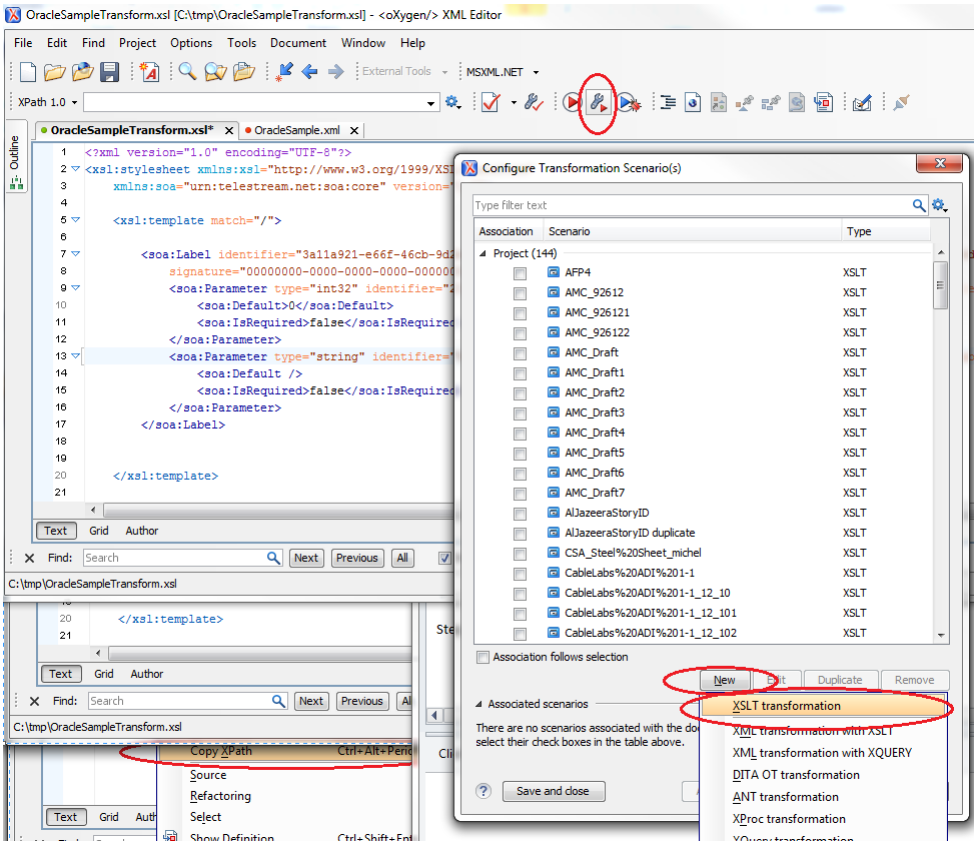
10. Open the XML file in the editor and copy the XPath of each value and place these values into your XSLT style-sheet as shown in the next step. In the example below, we right-clicked on the Bitrate element to copy its XPath.



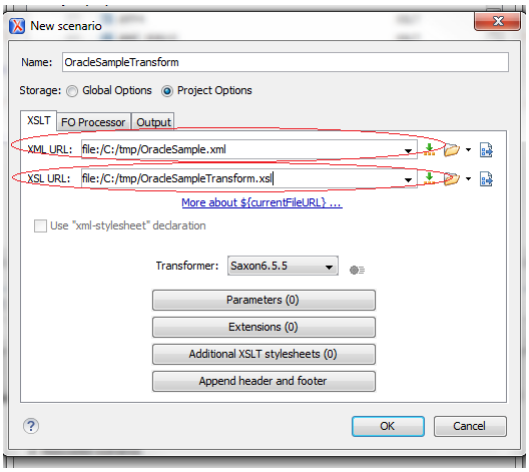
11. Use the command <value-of-select = xpath> to add the XPath values to the XSLT as shown below. In our example, we are telling the XSLT style sheet to go to the BitRate element in the OracleSample.xml document and place the value found into the Vantage Label BitRate element. We have done this for both the bit rate element and the file name in our example.



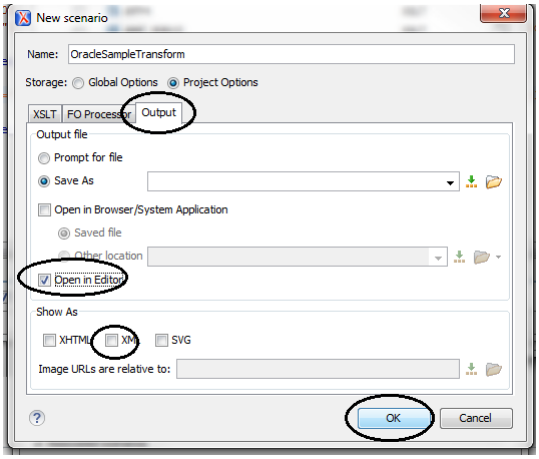
12. Now run a test transformation of the XML using the XSLT in the IDE. This will tell you if your XML and XSLT are correctly designed to work together. For example, in OXygen, click on the picture of the Wrench on the tool bar and select New and XSLT transformation.



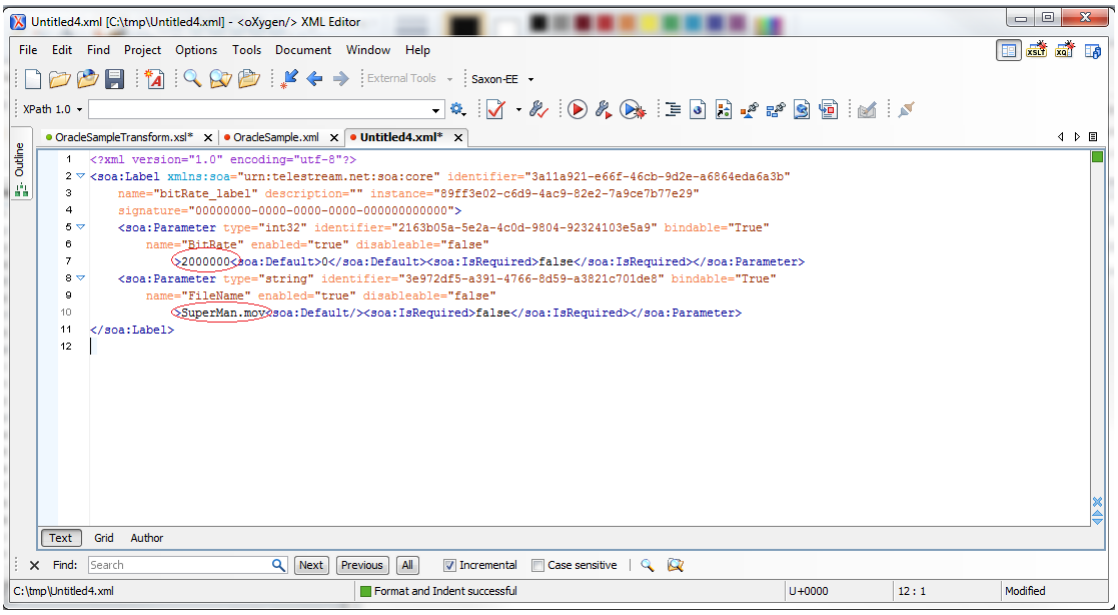
13. Point to the XML document you are going to transform and the XSLT style sheet that will be doing the transformation.



14. To run the transform, select the Output tab, unclick the Show As XML check box, select Open in Editor, and click OK.

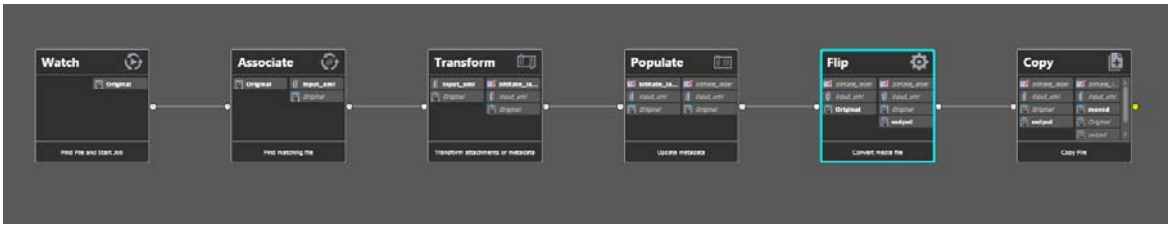


15. View and validate the output. We have now converted our example OracleSample.xml file into a Vantage label whose values contain the values that were present in the XML example file. The correctly transformed output proves that we can now use the XSLT style sheet in a Vantage workflow to transform XML files that follow the format for which the style sheet was created.



Using the XSLT in an Ingest Workflow

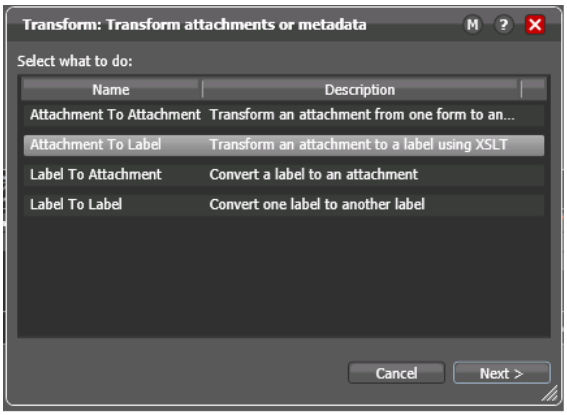
A workflow is shown below that ingests media and an associated XML file and uses the XSLT file to transform the XML data into a label whose values are then turned into variables, and the variables converted into parameters for the media file.

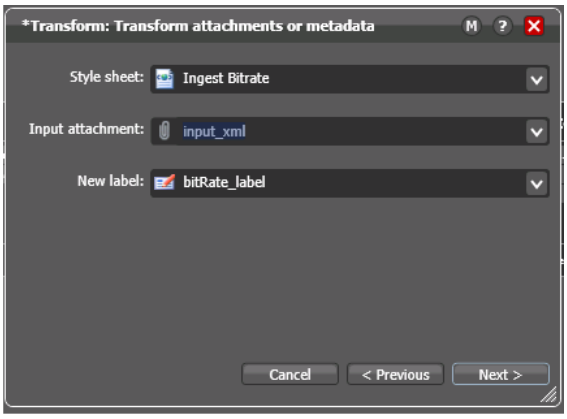


This workflow looks for a media file using the Watch action and then also waits for an associated XML file of the following format:

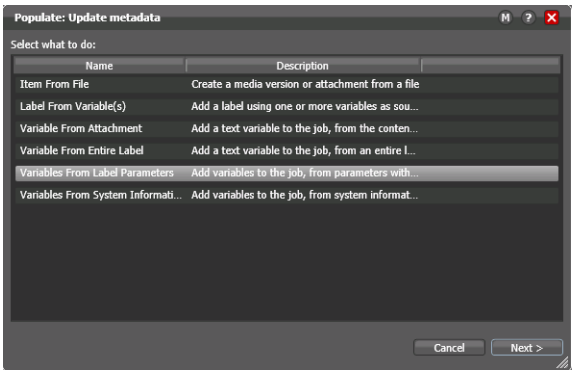
```
<Clip>
  <Bitrate>2000000</Bitrate>
  <FileName>SuperMan.mov</FileName>
</Clip>
```

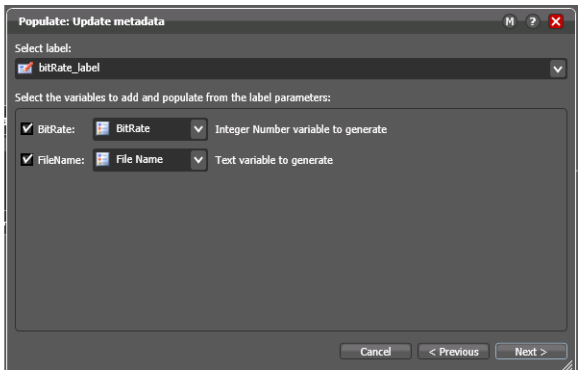
As shown below, the Transform action uses the Ingest Bitrate XSLT style sheet to transform the XML attachment *input_xml* into a label named *bitRate_label*.





The Populate action assigns variables from the label components. The BitRate and FileName fields of the label are assigned to two Variables named BitRate and File Name.





The BitRate and File Name variables are then used in the Flip and Copy actions by binding them to parameters. This allows the bit rate of the transcoded output and the output file name to be set dynamically based on data in the XML file associated with the ingested media.

Copyright and Trademark Notice