Using Vantage to Ingest XML into a Label

This App Note	Synopsis	2
applies to	Version Requirements	2
Vantage versions	Licensing	2
6.2 and later	Example Project	3
	Building an XSLT to Match Ingested XMLs	3
	Using the XSLT in an Ingest Workflow	10
	Copyright and Trademark Notice	12



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.

Y	Van	tage Management Console [W-GARYP : Administrator]					the state state of the
	File	Options Help					
	8	Settings & Options	Metadata Labels				
	Ţ	Catalogs	🥪 🗙 🗏 🍡 🗛	🖻 🔹 💀 🔹			
	Ê	Storage	Metadata	Label	r -		Description
		🗾 Vantage Stores	Advanced Audio	Measurement	This lab	el is used for publishing the analys	sis results of Advanced Audio
		📮 Vantage Folder Address Book	≝bitRate label				
	đ	Services	Black Detection	Ishel	Contain	s the results of the black detectior	
		Licenses					••••
		Status	CableLabs ADI 1		This met	tadata label is used for CableLabs	ADI 1.1.
	ન્	Workflow Design Items	🔒 Color Space Det	ection Label	Contain	s the results of the color space de	tection process.
		ංසි Vantage Workflows	🔒 Curtain Detectio	n Label	Contain	s the results of the curtain detection	on process.
		Workflow Categories	I DG XML				
		Media Nicknames	Tons tort				
		Attachment Nicknames			_		
			Name: bitRat	e_label			
		Style Sheets	Description:				
		Action Defaults	Design				
	1	Domain Workflow Analytics	🛃 🗙 🌬 🔻 🛍				
		Action Analytics	Parameter	Categ	огу	Description	
		R Workflow Analytics	BitRate	Misc (defau	lit)		
	1	Support Utilities	FileName	Misc (defau			
		Data Collection					
		Change Log					i .
		📮 Error Log					i .
		🗎 Actions By Service					
		🚞 Plug-In Versions By Service					
Þ	1	Application Configurations					

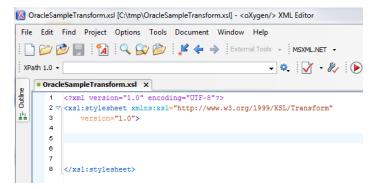
- **3.** Next, you need to select an XSLT style sheet to use in transforming the XML contents of the label. In the VMC, select Style Sheets and select any properly composed XSLT style sheet (for this purpose, the style sheet you use doesn't matter).
- 4. Select the Test button. This opens the Test window for testing your label.

Vantage Management Console [W-GARYP : A	(aministrator)	
File Options Help		
Settings & Options Catalogs	Style Sheets ■ × 示 □ ■ ×	
👻 🛅 Storage	Style Sheet Description	
🗾 Vantage Stores	P fotokem_channel_map	
🚞 Vantage Folder Address Book		
🛷 Services	PangTransformSample	
Y Licenses	MRK_Sample	
🕨 🌌 Status	-	
🗸 🕫 Workflow Design Items	Name: fotokem channel map	
ਾ ^C Vantage Workflows		
🛱 Workflow Categories	Description:	
Media Nicknames	(Test.,)	
Metadata Labels		
Attachment Nicknames		
Style Sheets		
Action Defaults	<soa:label 00000000-0000-0000-0000-000000000"="" description="" identifier="497d2ed3-ac3d-4776-ae43-444e37c4t371" instance="f765e052-e456-47f9-b3e8-
signature=" name="Fotokem" xmlnssoa="urn:telestream.netsoaxcore"></soa:label>	
👻 👘 Domain Workflow Analytics	signature= 00000000-0000-0000-0000-00000000000 xminsisoa= urn:telestream.netisoa:core > <soa:parameter bindable="True" disableable="false" enabled="tru</p></th><th>ie" identifier="ef87d5e8-135e-4576-ade6-a20eb8351e36" name="TC Start" type="timecode"><x< th=""></x<></soa:parameter>	
Action Analytics	StartTime[1]'> <soa:default>00:00:00@29.97 </soa:default>	
여당 Workflow Analytics		
👻 🛅 Support Utilities		
Data Collection	<soa:parameter bindable="True" enabled="true
EndTime[1]" identifier="8e5b784c-fae8-4d0b-a29d-8bd9aa59dd44" name="TC_End" type="timecode"></soa:parameter> <soa:default>00:00:00:00:00:00:00:00:00:00:00:00:00:</soa:default>	e disableable= taise > <x< th=""></x<>
Change Log		
Error Log		
 Actions By Service 		disableable="false"> <xsl:v< th=""></xsl:v<>
Elug-In Versions By Service	<pre>channel_iout_rrom_source[i] /><soa:derauit>U</soa:derauit> <soa:isrequired>false</soa:isrequired></pre>	
Application Configurations		
 Report Configurations 		
 Fulfillment Schemes 	<soalsrequired>false</soalsrequired> 	
a. Users		
📲 Domain Backup/Restore		

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

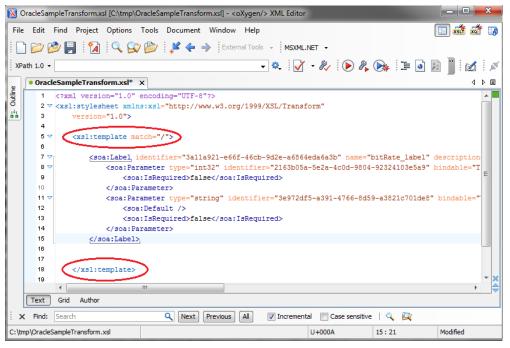
Transform Test		? 🗴
Test with Labels Test with Files		
Select a label to test the transform:		57 items
Label	Description	5/ rtems
	This label is used for publishing the analysis results of Advanced Au	udio Measurement.
bitRate label	······································	
Black Detection Label	Contains the results of the black detection process.	
CableLabs ADI 1.1	This metadata label is used for CableLabs ADI 1.1.	
Color Space Detection Label	Contains the results of the color space detection process.	
Curtain Detection Label	Contains the results of the curtain detection process.	
DG XML		
Dope_test		
Duration_sample		
Edit_and_Remap2	Parameters for Head and Tail trim with remap	
FilePathEdit		
FilePathFinal		
Fotokem		
Hoaret Labol		x
Transform Source:		Transform Result:
Instruce="89ff3e02-c6d9-4ac9-82e2-7 granture="00000000-0000-0000-0000 <soa:parameter identifier<br="" type="int32">name="BitRate" Cut Ctr <soa:is:required copy="" ctr<br=""><td>+0000000000000000000 vmlnssoa = 'urndtelestream netsoaccore' = 2163b05a-5e2a-4c0d-9804-92324103e5a9" bindable= "(ue' +X Abe" > 0<soadefault> 0</soadefault> +C 2df5_a391-4766-8d59-a3821c701de8" bindable= "True" = "false" ></td><td><pre><?zml version="10" encoding="tif15"> </pre></td></soa:is:required></soa:parameter>	+0000000000000000000 vmlnssoa = 'urndtelestream netsoaccore' = 2163b05a-5e2a-4c0d-9804-92324103e5a9" bindable= "(ue' +X Abe" > 0 <soadefault> 0</soadefault> +C 2df5_a391-4766-8d59-a3821c701de8" bindable= "True" = "false" >	<pre><?zml version="10" encoding="tif15"> </pre>

- **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.

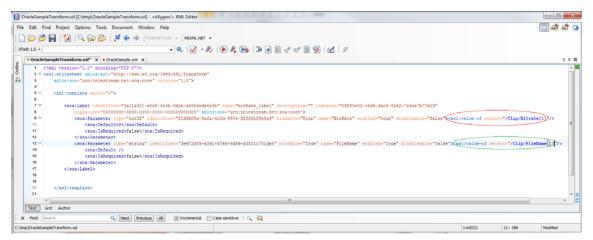


We have now told the XSLT what the output will be and added the name space, but we have not told the XSLT where to get the values that will get placed into the two elements within the label contents—the bit rate and the file name. These values are present in the XML file you are submitting to the workflow, 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>].

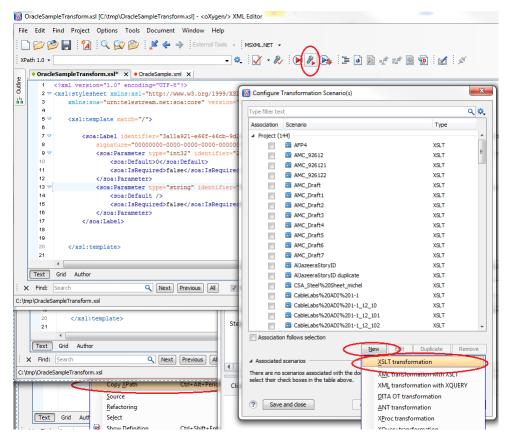
10. Open the XML file in the editor and copy the XPath of each value and place these values into your XSLT style-sheet as shown in the next step. In the example below, we right-clicked on the Bitrate element to copy its XPath.

🔀 C	DracleSar	mple.xml [C:\	tmp\(DracleSample.xml] - <oxy< th=""><th>gen/> XML Editor</th><th></th><th></th><th></th><th></th><th></th><th>x</th></oxy<>	gen/> XML Editor						x
File	Edit	Find Proje	ect (Options Tools Docum	nent Window Help					🔲 🛋 🛣	
	۱ 📂 🕯	• : 📄 🎓	A :	🔍 🎯 📂 🛛 🦊 🗸	External Tools	- Saxon-EE	-				
i vp:	ath 1.0 🗸) 🔊 : = 🗖) 🔝 🧏 🕫 🧕	a 📾 🗄 🚜 🗄	
: ***						••• : <u>V</u>	Ø √ : 🕑 Ø	s 😪 : 🖛 🖻			
Outline	Oracl 1			<pre>sl* x • OracleSample n="1.0" encoding="U]</pre>						4 Þ	
	1	xml ve:</td <td>13101</td> <td>h="1.0" encoding="0]</td> <td>12-8-2></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	13101	h="1.0" encoding="0]	12-8-2>						
11	3 🗸	<clip></clip>									
	4	<bit:< td=""><td>rate'</td><td>>2000000</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></bit:<>	rate'	>2000000							
	6	<fil< td=""><td></td><td><u>C</u>ut</td><td>Ctrl+X</td><td>]</td><td></td><td></td><td></td><td></td><td></td></fil<>		<u>C</u> ut	Ctrl+X]					
	7			С <u>о</u> ру	Ctrl+C						
	9		6	<u>P</u> aste	Ctrl+V						
	10		÷!	Toggle Comment	Ctrl+Shift+Comma	_					
				Co <u>m</u> pare							
				Op <u>e</u> n	•						
				Folding	•	-					
			ഷ	Go to Matching Tag	Ctrl+Shift+G						
				Go after <u>N</u> ext Tag	Ctrl+Close Bracket						
			_	Go after Previous Tag	Ctrl+Open Bracket	ī					
		\sim	-	Copy <u>X</u> Path	Ctrl+Alt+Period	Ρ					
				Source Refactoring	F						*
	Text	Grid Aut		Select	, ,						
: x	Find:	,	1	Show Definition	Ctrl+Shift+Enter	✓ Incremental	Case sensit	ive 🔍 🔯			
-		Sample.xml		■ Lea	rn completed			U+0061	5:10		

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.

New scenario					×
Name: OracleSampleTran	sform				
Storage: 🔘 Global Options	Project O	ptions			
XSLT FO Processor Out	tput				
XMLURL: file:/C:/tmp/Or	acleSample.xm	N		*0	•
XSL URL: file:/C:/tmp/Or	acleSampleTra	nsform.xsl	•	10	•
Use "xml-stylesheet"		out \${currentFileURL}			
	Transformer:	Saxon6.5.5 🔹	0 2		
		Parameters (0)			
		Extensions (0)			
	Additio	nal XSLT stylesheets (0)			
	Арре	end header and footer			
?			ОК	Ca	ncel

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

New scenario	×
Name: OracleSampleTransform	
Storage: 🔘 Global Options 💿 Project Options	
XSLT FO Processor Output	
Output file	
Prompt for file	
Save As	👻 📩 🥟
Open in Browser/System Application	
Saved file	
Other location	- ± 📂 -
Dpen in Editor	
Show As	
XHTML SVG	
Image URLs are relative to:	
?	OK Cancel



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.

X	Un	titled	d4.xml [C.\tmp\Untitled4.xml] - <oxygen></oxygen> XML Editor	
F	ile	Edit	t Find Project Options Tools Document Window Help	🔲 📸 📸 🗊
	D		r 💯 🗐 🗄 🕅 🔍 😰 💯 🕴 🦊 🜩 🗄 External Tools 👻 Saxon-EE 🗸	
		h 1.0		
-	٦.	• Ora	adeSampleTransform.xsl* x • OracleSample.xml x • Untitled4.xml* x	∢ ▷ 🗉
T. Outline	Г	1	<pre></pre> <pre><</pre>	
ō		2	2 🗸 <soa:label <="" identifier="3a11a921-e66f-46cb-9d2e-a6864eda6a3b" td="" xmlns:soa="urn:telestream.net:soa:core"><td></td></soa:label>	
15		з	name="bitRate_label" description="" instance="89ff3e02-c6d9-4ac9-82e2-7a9ce7b77e29"	
		4	signature="0000000-0000-0000-0000-0000000000">	
			<pre><soa:parameter <="" bindable="True" identifier="2163b05a-5e2a-4c0d-9804-92324103e5a9" pre="" type="int32"></soa:parameter></pre>	
		6	Man	
		7		
			<pre>style="styl</pre>	
		9		
		10		
		11 12		
		12	4	
				32
	0	Text	t Grid Author	*
1	×	Find	d: Search 🔍 Next) Previous All 📝 Incremental 📄 Case sensitive 🔍 🔀	
C:	\tmp	\Untit	tled4.xml Format and Indent successful U+0000 12 : 1	Modified

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.

		I The second sec	the same second s	and the second sec	m 🛄	Populat	No. of Concession, Name	Flip	¢	Сору	Ē
C Orga		Original	() mpst_mm		Caratan Ja.		. I LOAR ADD	ana series was		anian ana	
	-		El zone	C Owner	C rost.ent		A Dates	C Internet	1) restant	e seturi	C ADCAN
							111111		() and part		
											C Cogoner E content
First File and Utart Job	i i	find me	econg the	Transform attack	ments or metabora	Upente	110000	Corvet	maga 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*.

Transform: Transform at	tachments or metadata 🛛 M 🕐 🗙	*Transform: Transform attachments or metadata M	? X
Select what to do:			
Name	Description	Style sheet: 🧧 Ingest Bitrate	×
Attachment To Attachment Attachment To Label	Transform an attachment from one form to an Transform an attachment to a label using XSLT	Input attachment: 🌒 input_xml	~
Label To Attachment	Convert a label to an attachment	New label: 🗾 bitRate_label	\mathbf{v}
Label To Label	Convert one label to another label		
	Cancel Next >	Cancel < Previous	Next >

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.

Populate: Update metadata		M ? X	Populate: Update metadata	M ? X
Select what to do:			Select label:	
Name	Description		🖬 bitRate_label	\mathbf{v}
Item From File	Create a media version or attachment from a file			
Label From Variable(s)	Add a label using one or more variables as sou		Select the variables to add and populate from the label parameters:	
Variable From Attachment	Add a text variable to the job, from the conten		🗹 BitRate: 📕 BitRate 🗸 Integer Number variable to generate	
Variable From Entire Label	Add a text variable to the job, from an entire I		🗸 FileName: 📕 File Name 🗸 Text variable to generate	
Variables From Label Parameters	Add variables to the job, from parameters with			
Variables From System Informati	. Add variables to the job, from system informat			
		Cancel Next >	Cancel < Previous	Next >
			Cancel < Previous	wext >

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.



©2014 Telestream[®], Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, altered, or translated into any languages without written permission of Telestream, Inc. Information and specifications in this document are subject to change without notice and do not represent a commitment on the part of Telestream.

Telestream, CaptionMaker, Episode, Flip4Mac, FlipFactory, Flip Player, Lightspeed, ScreenFlow, Vantage, Wirecast, GraphicsFactory, MetaFlip, and Split-and-Stitch are registered trademarks and Pipeline, MacCaption, e-Captioning, and Switch are trademarks of Telestream, Inc. All other trademarks are the property of their respective owners.

