Article
David.M · Sep 7, 2016 5m read

Running HealthShare XSLTs from Terminal

HealthShare uses a lot of XSLTs. These are used to convert IHE medical documents to SDA (internal HealthShare format) and back to IHE formats, to create summary reports, and to deal with IHE profiles (e.g., patient information query, document provide and register). Customers may customize the XSLTs to customize reports or for other reasons.

For debugging and development, it is very convenient to be able to run an XSLT from Terminal.

The Class Method

Following is a class that contains a class method that lets you do this on Windows. Create the class in HSREGISTRY or another HealthShare namespace (don't use HSLIB or VIEWERLIB) and compile it.

```
Class Local.XsltTransformer Extends %RegisteredObject
{

ClassMethod Transform(XslDirectory As %String, XslBaseFilename As %String, Directory
As %String, InputFilename As %String, OutputFilename As %String, byref Parameters = "
")
{

   // Run the XSLT transform with the base filename XslBaseFilename (i.e., without th
e .xsl
   // extension) that is in the XslDirectory. Run it on the input file with name Inpu
tFilename
   // and put the output in the file with name OutputFilename. The input file must be
 in the
   // directory Directory, and the output will be put in the same directory. The Para
meters
   // argument may be used to pass parameters to the transform (rarely needed). This
class
   // method should be run from Terminal in a HealthShare namespace other than HSLIB
or
   // VIEWERLIB. The method will write out the path and name of the transform and any
 error
   // messages.

   set In                          = ##class(%Stream.FileCharacter).%New()
   set In.Filename                 = Directory _ "\" _ InputFilename
   set Out                         = ##class(%Stream.FileCharacter).%New()
   set Out.Filename                = Directory _ "\" _ OutputFilename
   set Transformer                 = ##class(HS.Util.XSLTTransformer).%New()
   set Transformer.XSLTCacheMode = "N"
   set Transformer.XSLTDirectory = XslDirectory
   write !, XslDirectory _ "\" _ XslBaseFilename _ ".xsl"
   set Status = Transformer.Transform( .In, XslBaseFilename _ ".xsl", .Out, .Paramete
rs )
```

```
   if $system.Status.IsOK( Status ) {
      set Status = Out.%Save()
   }

   if $system.Status.IsError( Status ) {
      write $system.Status.GetErrorText( Status )
   }
}

}
```

Following are what the parameters to the Transform class method are: XslDirectory is the directory where the XSLT file is; the class will first try to append \Custom to the directory when looking for the transform, then try it without. XslBaseFilename is the filename of of the XSLT transform without the .xsl extension. Directory is the directory where you have the input document and where you want the transform to write the output of the transform. InputFilename is the name of the input document file including extension. OutputFilename is the name of the output file including extension. Parameters can be used to pass parameters to the transform, but this is rarely needed.

Running the Transform in Terminal

To run an XSLT transform, open Terminal, and change to the namespace and run the Transform method. For example, let's run the CCDA-to-SDA transform that comes with HealthShare. I have HealthShare installed in C:\InterSystems\HealthShare2016.1.1. I'll put a CCDA named "Sample_CCDA.xml" in my c:\Junk folder.

```
USER>zn "hsregistry"

HSREGISTRY>do ##class(Local.XsltTransformer).Transform( "C:\InterSystems\HealthShare2
016.1.1\CSP\xslt\SDA3", "CCDA-to-SDA", "c:\Junk", "Sample_CCDA.xml", "SDA_Out.xml" )

C:\InterSystems\HealthShare2016.1.1\CSP\xslt\SDA3\CCDA-to-SDA.xsl
HSREGISTRY>
```

The SDA_Out.xml output file is now in c:\Junk.

Note that you should not use the transforms that are in the CSP\xslt\SDA directory. Use the ones in the CSP\xslt\SDA3 directory. The ones in the SDA directory are for an old version of SDA (version 2).

Studio XSL Transform Wizard

An alternative to running your transforms from Terminal is to use the Studio XSL Transform Wizard. In Studio, select Tools > Add-Ins > XSL Transform Wizard. Enter your input file in the "XML File" box and the XSLT transform in the "XSL File" box. For "XSLT Helper Class", select "HSREGISTRY" and "HS.Util.XSLTHelper". Click "Finish". The output is displayed in the dialog box (you can copy and paste it):

Debugging

When debugging XSLTs, one approach is to add <xsl:comment> elements to the XSLTs so that you can see various items in the output. Here are some examples:

```
<xsl:comment>useFirstTranslation <xsl:value-of select="$useFirstTranslation"/>.
             referenceValue      <xsl:value-of select="$referenceValue"/>.
             displayName         <xsl:value-of select="@displayName"/>.
             originalText        <xsl:value-of select="hl7:originalText/text()"/>.
             descriptionValue    <xsl:value-of select="$descriptionValue"/>.
</xsl:comment>

<xsl:comment>Context node:</xsl:comment>
<xsl:copy-of select="." />
<xsl:comment>End of context node.</xsl:comment>
```

Several of the HealthShare XSLTs apply a "Canonicalize" template. For example, CCDA-to-SDA.xsl does this where you see the comment "Canonicalize the SDA output". Canonicalizing will remove comments from the output, so you may wish to comment it out while debugging.

Documentation

For more information on XSLTs in HealthShare, see the following chapters in the documentation:

- HealthShare 2020.1 > InterSystems Clinical Data Models > CDA Interoperability with SDA > CDA Documents and XSL Transforms
- HealthShare 2020.1 > InterSystems Clinical Data Models > CDA Interoperability with SDA > Customizing CDA XSL Transformations

In particular, the "Customizing CDA XSL Transformations" chapter has a few suggestions for debugging. Also see the following chapter:

- HealthShare 2020.1 > Running a HealthShare Unified Care Record System > Unified Care Record Registries > Managing XML Summary Report Types

#Terminal #Tips & Tricks #HealthShare

Source URL:https://community.intersystems.com/post/running-healthshare-xslts-terminal