

XML HSCOrgRefData XSLT Primary Role Constrain

Java and .NET Support Documentation

Date: 25th September 2018

Version: 3.0

Information and technology
for better health and care

Contents

Table of Contents

Contents	2
1. Introduction	3
2. Package Content	4
2.1. Root Folder:	4
2.2. Saxon folder	4
2.3. Java folder	4
2.4. Saxon dotNET folder	5
3. Source Code	6
4. How to Use	7
4.1. Pre-requisites	7
4.2. Using the Software to Transform the XML	7
4.3. Download the XML release file that needs to be transformed	7
4.4. Download and Unpack the Compiled Saxon Package	7
5. Execute the Transform Process using Java	9
6. Execute the Transform Process using .NET	11
7. Writing Output to Servers and Share Names	13
7.1. Parameters	13
7.2. Java Example	14
7.3. .NET Example	14
8. XSLT Output	15

1. Introduction

In order to support consumers of Organisation Reference Data NHS Digital has developed an XSLT that produces an XML file containing only organisations that have one or more Primary Roles that matches a user-supplied list of role codes. Organisations that have a “refOnly” value of TRUE are automatically excluded from the file; both in appearing themselves or being present in the relationships of any included organisation.

This document summarises the salient points regarding the XSLT itself and runs through the process of actually running the transform. The issuing authority doesn’t mandate use of the XSLT or the supporting Saxon libraries by consumers; rather the artefacts have been made available as a set of support tools which can be used to transform information contained in Health and Social Care Organisation Reference Data XML files. It is expected that consumers will make use of the artefacts and extend them as required to suit local requirements – they serve as a “starter for ten” from which consumers can gain an understanding of XSLT.

The transform itself is written in raw XSLT and is therefore technology agnostic. It is released under the Apache 2.0 license. Consumers MUST check the conditions of the Apache 2.0 license prior to deployment and use of the software.

The two Saxon libraries which are included in the package have been compiled using Java and .NET are freely available under the Mozilla 2.0 license from the following location:

Java and .NET (version 9.0 or later):

<https://sourceforge.net/projects/saxon/files/Saxon-HE/>

Also see <http://www.Saxonica.com/products/products.xml> for details regarding Saxon Home Edition.

Consumers MUST check the conditions of the relevant Mozilla license prior to deployment and use of the Saxon parsing software. The issuing authority doesn’t mandate use of the version of the Saxon library at the download location above. Consumers are free to choose their own preferred tools to transform the data, however the library specified above has been tested with the steps described in this document.

The guidance provided in this document applies to Windows 10 only.

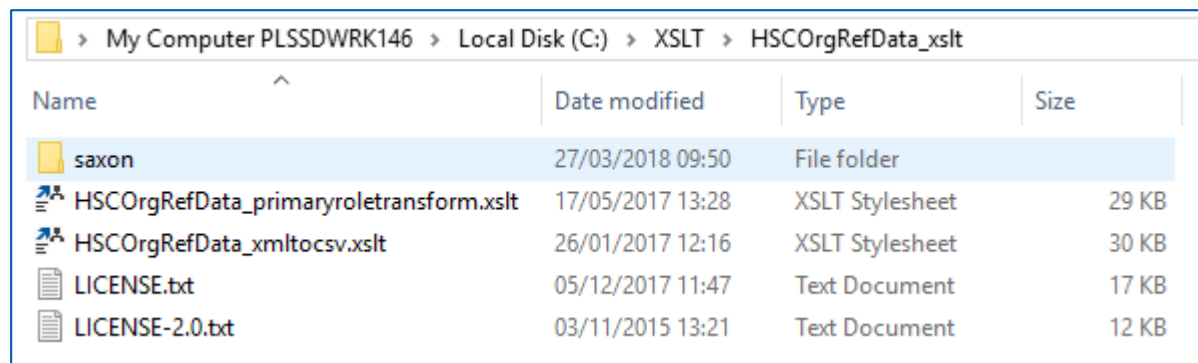
2. Package Content

The compiled Java package (HSCOrgRefData_xslt.zip) has the following structure.

HSCOrgRefData_xslt (Root Folder)	LICENSE-2.0.txt (Apache 2.0 License)	
	LICENSE-1.0.txt (Mozilla 1.0 License)	
	HSCOrgRefData_xmltocsv.xslt (XSLT)	
	HSCOrgRefData_primaryroletransform.xslt (XSLT)	
	saxon(Folder)	Contains Java and dotNET libraries in separate folders

2.1. Root Folder:

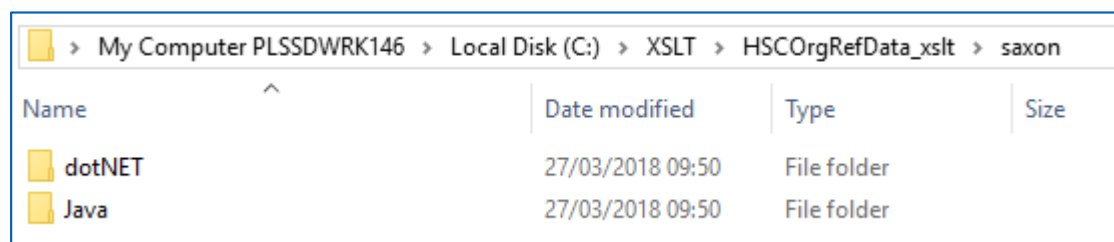
The root folder contains the saxon library folder and the transformation scripts as well as the relevant licences.



My Computer PLSSDWRK146 > Local Disk (C:) > XSLT > HSCOrgRefData_xslt				
Name	Date modified	Type	Size	
saxon	27/03/2018 09:50	File folder		
HSCOrgRefData_primaryroletransform.xslt	17/05/2017 13:28	XSLT Stylesheet	29 KB	
HSCOrgRefData_xmltocsv.xslt	26/01/2017 12:16	XSLT Stylesheet	30 KB	
LICENSE.txt	05/12/2017 11:47	Text Document	17 KB	
LICENSE-2.0.txt	03/11/2015 13:21	Text Document	12 KB	

2.2. Saxon folder

The saxon folder contains the two technology platform folders.



My Computer PLSSDWRK146 > Local Disk (C:) > XSLT > HSCOrgRefData_xslt > saxon				
Name	Date modified	Type	Size	
dotNET	27/03/2018 09:50	File folder		
Java	27/03/2018 09:50	File folder		

2.3. Java folder

Java platform folder containing the relevant Java archive file (jar) as well as supporting resources.

My Computer PLSSDWRK146 > Local Disk (C:) > XSLT > HSCOrgRefData_xslt > saxon > Java				
Name	Date modified	Type	Size	
doc	27/03/2018 09:50	File folder		
notices	27/03/2018 09:50	File folder		
LICENSE-1.0.txt	17/12/2015 16:01	Text Document	20 KB	
LICENSE-2.0.txt	03/11/2015 13:21	Text Document	12 KB	
saxon9he.jar	04/12/2017 16:00	Executable Jar File	4,678 KB	

2.4. Saxon dotNET folder

Folder containing the .NET components and resources.

My Computer PLSSDWRK146 > Local Disk (C:) > XSLT > HSCOrgRefData_xslt > saxon > dotNET				
Name	Date modified	Type	Size	
bin	27/03/2018 09:50	File folder		
notices	27/03/2018 09:50	File folder		
unins000.dat	14/03/2018 13:59	DAT File	4 KB	
unins000.exe	14/03/2018 13:58	Application	699 KB	

3. Source Code

The only source code released by HSCIC (under the Apache 2.0 license) within this package is **HSCOrgRefData_xmltocsv.xslt** and **HSCOrgRefData_primaryroletransform.xslt**.

Please refer to www.Saxonica.com for licenses and source code related to the Saxon library (released under the relevant Mozilla licenses).

4. How to Use

This guidance refers to the use of the XSLT file (HSCOrgRefData_primaryroletransform.xslt) and the compiled Java software the within the Saxon package.

4.1. Pre-requisites

The software has been tested successfully on a Windows Surface Pro PC with the following specification:

- Windows 10 (64 Bit)
- Core i5 3320M quad core processor running at 2.6GHz
- Microsoft .NET Framework 4.5
- 8GB RAM
- Java jdk-8u161 (it is advised to always have an up to date version of the JDK deployed)
- Java jre1.8.0_161 (it is advised to always have an up to date version of the JRE deployed)

The specification doesn't serve as a warranted environment specification, but may be useful in helping to define a minimum specification under which consumers are able to run the software.

PLEASE NOTE: While the transform will complete with the aforementioned PC specification it is much quicker when performed on a server with a minimum specification as follows;

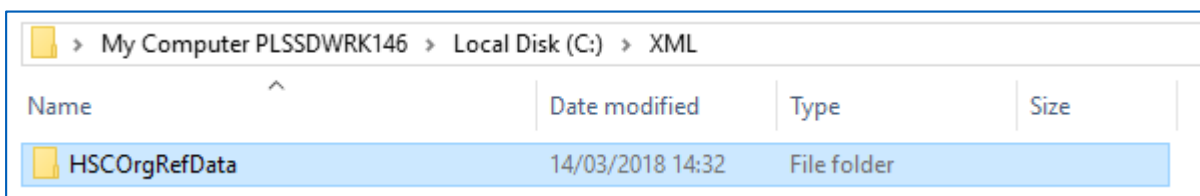
- Windows Server 2008 R2 (64 Bit)
- Intel Xeon CPU X5660 2.80GHz (2 processors)
- 12GB RAM

4.2. Using the Software to Transform the XML

Please note that the example folders included in the steps below are not mandated but they are consistent with the command line arguments show in the examples.

4.3. Download the XML release file that needs to be transformed

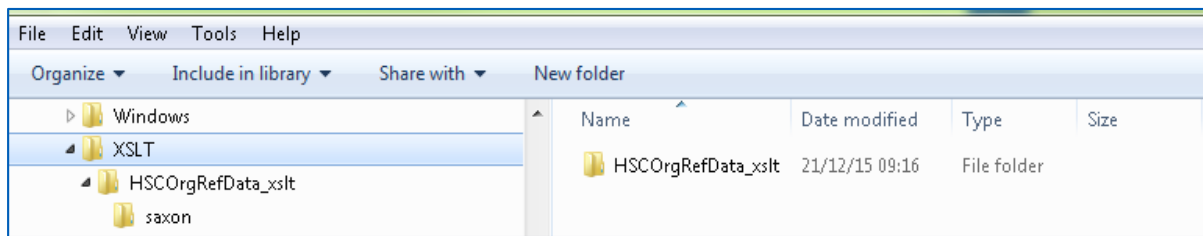
Unpack the file and make a note of where it has been saved, e.g. c:\xml_files\



My Computer PLSSDWRK146 > Local Disk (C:) > XML			
Name	Date modified	Type	Size
HSCOrgRefData	14/03/2018 14:32	File folder	

4.4. Download and Unpack the Compiled Saxon Package

The Compiled Saxon Package is available from the NHS Digital website. Download the package and unzip the archive. Make a note of where the files within the package have been extracted, e.g. c:\XSLT\



5. Execute the Transform Process using Java

Open a command prompt (run as administrator). Navigate to the directory that contains the XSLT file (for example c:\XSLT\HSCOrgRefData_xslt).

The command must follow the syntax and structure below:

Java -<memory allocation flags> -jar Saxon9he.jar -t -s:<xml file> -xsl:<xslt file>

Command element	Description
Java	Invoke the Java interpreter
<memory allocation flags>	Instructs the Java interpreter to allocate sufficient heap memory to run the process
-jar	Java command line argument informing Java that it will be handling a jar file
Saxon9he.jar	The compiled Java file to be executed
<Saxon flags>	<p>-t = Display version and timing information to the standard error output. The output also traces the files that are read and writing, and extension modules that are loaded (see http://www.saxonica.com/html/documentation9.7/using-xsl/commandline/)</p> <p>-s: = Identifies the source XML file.</p> <p>-xsl: = Specifies the file containing the principal stylesheet module (XSLT).</p>
<role-ids>	Comma separated list of Primary Role IDs
<params>	See section 7 below.

For the file layout shown in previous steps the following command would execute the transform of a file called **HSCOrgRefData_Full_20180219.xml** to a role-constrained XML file containing only organisations with a primary role code of RO180 or RO157 which would be created on the server "ODSDATA" in a folder named "DATA\Test\XML".

Java -Xms2000M -Xmx2000M -jar

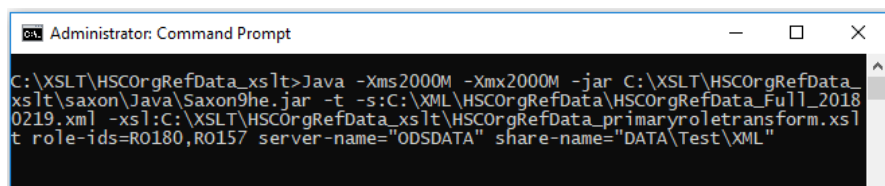
C:\XSLT\HSCOrgRefData_xslt\saxon\Java\Saxon9he.jar -t -

s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -

xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt role-ids=RO180,RO157 server-name="ODSDATA" share-name="DATA\Test\XML"

The output file will be named HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml, however this can be changed by using the '**-o:filename**' parameter.

The full process of running the command can be seen below:



```
Administrator: Command Prompt
C:\XSLT\HSCOrgRefData_xslt>Java -Xms2000M -Xmx2000M -jar C:\XSLT\HSCOrgRefData_xslt\saxon\Java\Saxon9he.jar -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt role-ids=RO180,RO157 server-name="ODSDATA" share-name="DATA\Test\XML"
```

After executing the command the Saxon process displays version and timing information as specified by the `-t` flag:

```
Saxon-HE 9.7.0.21J from Saxonica
Java version 1.8.0_161
Stylesheet compilation time: 648.071493ms
Processing file: C:/XML/HSCOrgRefData/HSCOrgRefData_Full_20180219.xml
Using parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Building tree for file: C:/XML/HSCOrgRefData/HSCOrgRefData_Full_20180219.xml using class net.sf.saxon.tree.tiny.TinyBuilder
Tree built in 16.255955s (16255.955632ms)
Tree size: 21594003 nodes, 14807523 characters, 7479557 attributes
### server-name: ODSDATA
### share-name: DATA\Test\XML
### output-path: file:///ODSDATA/DATA/Test/XML/
### filename: HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml
### role-ids list: RO180,RO157
### include-ref-only: no

Writing to file: ///ODSDATA/DATA/Test/XML/HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml
Execution time: 3m 15.522549s (195522.549382ms)
Memory used: 1447797136
```

The time taken for the transformation to run will be dependent on the following items;

- The specification of the machine running the XSLT
- The number of role ids specified by the user for inclusion in the file

Progress is written to the console whilst the transform takes place and returns to the DOS prompt on successful completion of the process.

If error(s) are found these are reported to the console and the transform is halted.

6. Execute the Transform Process using .NET

Open a command prompt (run as administrator).

Navigate to the directory that contains the (Saxon) Transform executable (for example C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET\bin).

The command must follow the syntax and structure below:

Transform -t -s:<xml file> -xsl:<xslt file>

Command element	Description
Transform	Invoke the interpreter
<saxon flags>	<p>-t = Display version and timing information to the standard error output. The output also traces the files that are read and writing, and extension modules that are loaded (see http://www.saxonica.com/html/documentation9.7/using-xsl/commandline/)</p> <p>-s: = Identifies the source file or directory.</p> <p>-xsl: = Specifies the file containing the principal stylesheet module.</p>
<role-ids>	Comma separated list of Primary Role IDs
<params>	See section 7 below.

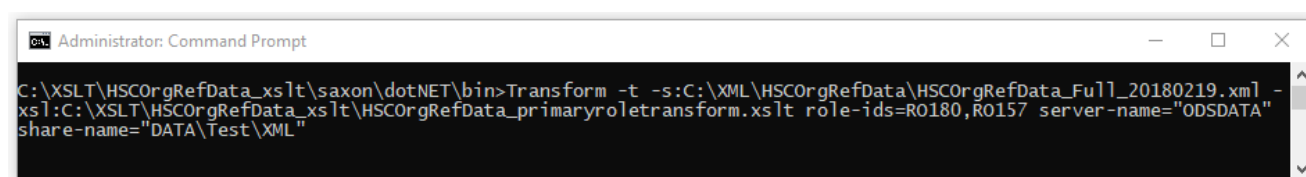
Navigate to the bin folder containing the Saxon 'Transform' executable (For example: C:\XSLT\HSCOrgRefData_xslt\Saxon\dotNET\bin)

For the file layout shown in previous steps the following command would execute the transform of a file called **HSCOrgRefData_Full_20170120.xml** to a role-constrained XML file containing only organisations with a primary role code of RO180 or RO157 which would be created on the server "ODSDATA" in a folder named "DATA\Test\XML".

Transform -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt role-ids=RO180,RO157 server-name="ODSDATA" share-name="DATA\Test\XML"

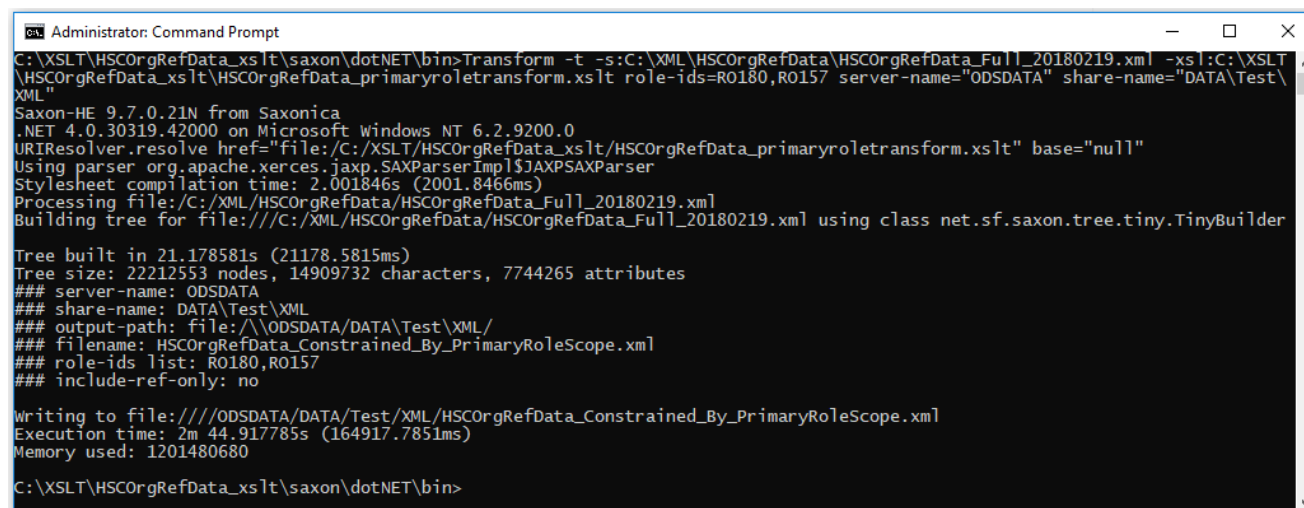
The output file will be named HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml, however this can be changed by using the '**-o:filename**' parameter.

The full process of running the command can be seen below:



```
Administrator: Command Prompt
C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET\bin>Transform -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt role-ids=RO180,RO157 server-name="ODSDATA" share-name="DATA\Test\XML"
```

After executing the command the Saxon process displays version and timing information as specified by the **-t** flag:



```
Administrator: Command Prompt
C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET\bin>Transform -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt role-ids=R0180,R0157 server-name='ODSDATA' share-name='DATA\Test\XML'
Saxon-HE 9.7.0.21N from Saxonica
.NET 4.0.30319.42000 on Microsoft Windows NT 6.2.9200.0
URIResolver.resolve href="file:/C:/XSLT/HSCOrgRefData_xslt/HSCOrgRefData_primaryroletransform.xslt" base="null"
Using parser org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser
Stylesheet compilation time: 2.001846s (2001.8466ms)
Processing file:/C:/XML/HSCOrgRefData/HSCOrgRefData_Full_20180219.xml
Building tree for file:///C:/XML/HSCOrgRefData/HSCOrgRefData_Full_20180219.xml using class net.sf.saxon.tree.tiny.TinyBuilder
Tree built in 21.178581s (21178.5815ms)
Tree size: 22212553 nodes, 14909732 characters, 7744265 attributes
### server-name: ODSDATA
### share-name: DATA\Test\XML
### output-path: file://\ODSDATA\DATA\Test\XML/
### filename: HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml
### role-ids list: R0180,R0157
### include-ref-only: no
Writing to file:///ODSDATA/DATA/Test/XML/HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml
Execution time: 2m 44.917785s (164917.7851ms)
Memory used: 1201480680
C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET\bin>
```

The output file will be named `HSCOrgRefData_Constrained_By_PrimaryRoleScope.xml`, however this can be changed by using the `'-o:filename'` parameter.

7. Writing Output to Servers and Share Names

By default the transform outputs the files to c:\HSCOrgRefData. The location can be overridden by passing in arguments from the command line as described below.

The XSLT takes the parameters **server-name** and **share-name** - usage is described further below.

7.1. Parameters

role-ids (Mandatory)	Example Entries
	role-ids=RO180,RO157
Note that if a role-ids value is not provided the file produced will have no organisations in it.	

server-name (Optional)	Example Entries
	Remote server -> server-name="INTEGRATION_SERVER "
	local machine -> server-name="C:"
	local machine (mapped network drive) -> "S:"
Note that if server-name is not provided the files will be saved to "C:\HSCOrgRefData\" by default.	
server-name can NOT be an empty string, "" will result in failure as the empty string replaces the default	

share-name (Optional)	Example Entries
	share-name="ORG_STAGING"
	share-name="ORG_STAGING/XMLOrgData"
Note that forward slashes '/' must be used to delineate directories (only applies to the share-name). The transform will fail if back slashes '\' are used due to the general use of backslashes to escape characters.	
share-name can be an empty string as the default value is already defined as an empty string. The empty string allows for instances where the output is written to the local machine or the root directory of mapped drives (also applies to remote servers although this is unlikely to be used due to permissions on the server's root folder).	

If both server-name and share-name are provided the file is written to <server-name>\<share-name>

e.g. server-name="INTEGRATION_SERVER" share-name="ORG_STAGING" would write the output to \\INTEGRATION_SERVER \ORG_STAGING\

If **share-name** is provided without server-name the files would be written to <Default server-name>\<share-name>, e.g. C:\HSCOrgRefData\ORG_STAGING\

If both parameters are omitted the files are saved to "C:\HSCOrgRefData\" by default

Further parameter examples can be found on the Saxon website:

<http://www.saxonica.com/documentation/index.html#!using-xsl/commandline>

7.2. Java Example

An example command line in java is shown below (lines have been wrapped to aid legibility):

```
java -Xms2000M -Xmx2000M -jar
C:\XSLT\HSCOrgRefData_xslt\saxon\Java\Saxon9he.jar -t
-s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml
-xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt
role-ids=RO180,RO157
server-name=" INTEGRATION_SERVER "
share-name=" ORG_STAGING "
```

Parameters can be specified at any point after the call to Saxon – the example below is also valid:

```
java -Xms2000M -Xmx2000M -jar
C:\XSLT\HSCOrgRefData_xslt\saxon\Java\Saxon9he.jar -t
server-name="INTEGRATION_SERVER"
share-name="ORG_STAGING"
role-ids=RO180,RO157
-s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml
-xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt
```

Note: Trailing slashes are not required on the parameter values provided. If provided they are silently ignored.

7.3. .NET Example

An example command line in java is shown below (lines have been wrapped to aid legibility):

```
Transform -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml
-xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt
role-ids=RO180,RO157
server-name="INTEGRATION_SERVER"
share-name="ORG_STAGING"
```

Parameters can be specified at any point after the call to Saxon – the example below is also valid:

```
Transform -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -t
server-name="INTEGRATION_SERVER "
share-name="ORG_STAGING "
role-ids=RO180,RO157
-s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml
-xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_primaryroletransform.xslt
```

Note: Trailing slashes are not required on the parameter values provided. If provided they are silently ignored.

8. XSLT Output

The XSLT processing writes output to a single XML file which follows the same schema as the full HSCOrgRefData XML file.

Only Organisations that have a PrimaryRole that matches one of the user-supplied RoleId values will be included in the file.

Organisations that have a “refOnly” value of TRUE are automatically excluded from the file; both in appearing themselves or being present in the relationships of any included organisation.