

HSCOrgRefData XML to CSV

Java and .NET Support Documentation

Date: 30th November 2018
Version: 3.0

Information and technology
for better health and care

Contents

Table of Contents

Contents	2
1. Introduction	4
2. Package Content	5
2.1. Root Folder:	5
2.2. Saxon folder	5
2.3. Java folder	5
2.4. dotNET folder	6
3. Source Code	7
4. How to Use	8
4.1. Pre-requisites	8
4.2. Using the Software to Transform the XML	8
4.3. Download the XML release file that needs to be transformed	8
4.4. Download and Unpack the Compiled Saxon Package	8
5. Executing the Transform Process using Java	10
6. Execute the Transform Process using .NET	12
7. Writing Output to Servers and Share Names	14
7.1. Parameters	14
7.2. Java Example	15
7.3. .NET Example	15
8. XSLT Output	16
9. Output File Descriptions	17
Document Payload Content	17
9.1. Organisation_Details.csv	17
9.2. Contact_Details.csv	17
9.3. Roles_Details.csv	18
9.4. Relationships_Details.csv	18
9.5. Successor_Details.csv	18
9.6. Additional_Attributes_Details.csv	19

9.7. OtherID_Details.csv	19
Document Metadata Content	19
9.8. Manifest_Details.csv	19
9.9. PrimaryRole_Details.csv	20
9.10. Code_System_Details.csv	20

1. Introduction

In order to support the transition of existing consumers of Organisation reference Data from CSV to XML NHS Digital has developed an XSLT that splits out an XML file into its component parts. The approach reads an XML file and parses the content into 9 separate CSV files. The files include “foreign key” references to allow consumers to link the data once it has been loaded into a set of landing tables.

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.

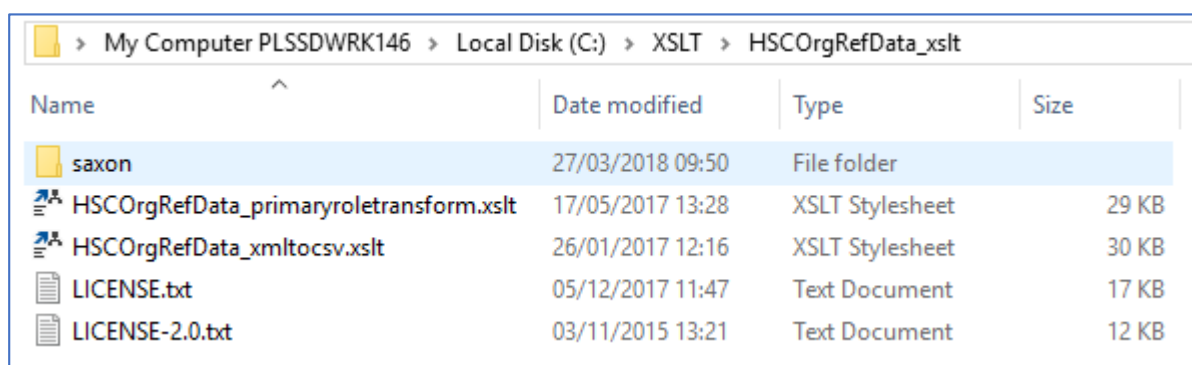
2. Package Content

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:

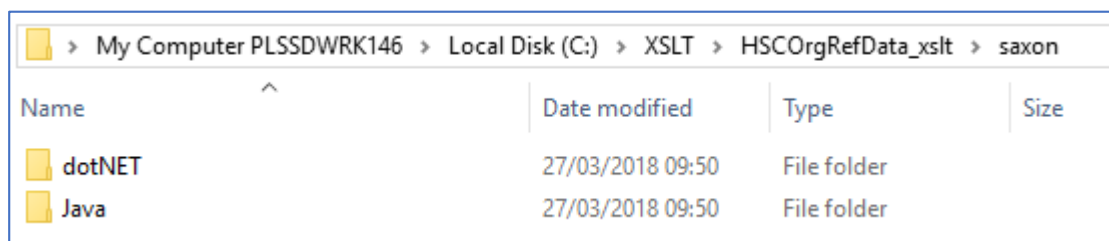
The root folder contains the saxon library folder and the transformation scripts as well as the relevant licences.



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. 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_xmltocsv.xslt) and the compiled Java software 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 jre-8u161 (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;

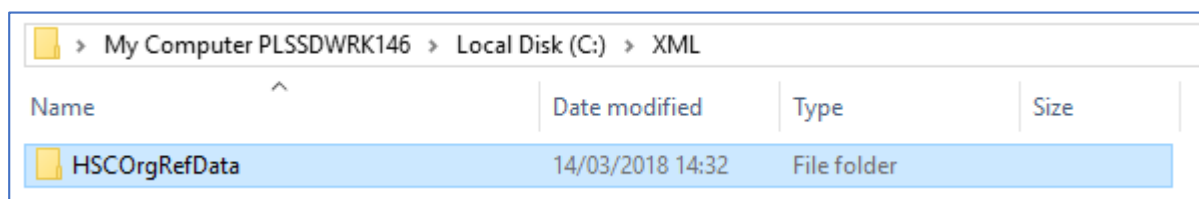
- Windows Server 2008 R2 (64 Bit)
- Intel Xeon CPU X5660 2.80GHz (2 processors)
- 12GB RAM

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

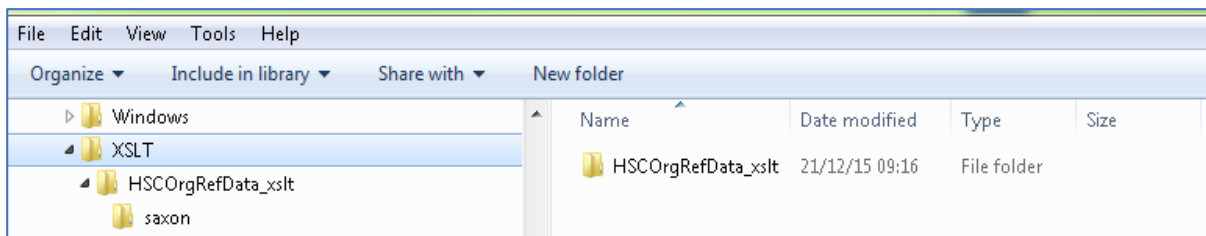
Unpack the file and make a note of where it has been saved, e.g. c:\XML\



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. Executing 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>
<params>	See section 7 below.

Tip! The java heap memory allocation provided via the Xms (JVM start memory allocation) and Xmx (JVM maximum memory allocation) flags are system dependent. The value specified for the Xmx MUST be free during the WHOLE process.

On a PC with 4GB of RAM setting the values for Xms and Xmx to 1300MB has allowed the process to complete successfully. The recommended value (by Oracle) is that the maximum heap size is set to 25% of the total amount of physical RAM but tests proved that allocations of 1024MB and 1100MB are not sufficient for a file of around 400MB in size. Setting these values to 1300MB should provide sufficient headroom for the larger XML release files in the Health and Social Care Reference Data product set. For machines with 8GB of RAM or more the values should be set to at least 2048MB.

For the file layout shown in previous steps the following command would execute the transform of a file called **HSCOrgRefData_Full_20170120.xml** to the set of CSV files defined within the rules **HSCOrgRefData_xmltocsv.xslt**.

```
java -Xms1300M -Xmx1300M -jar
c:\XSLT\HSCOrgRefData_xslt\Saxon\Saxon9he.jar -t -
s:c:\ls_xml_files\HSCOrgRefData_Full_20180828.xml -
xsl:HSCOrgRefData_xmltocsv.xslt
```

The full process of navigating to the HSCOrgRefData_xslt folder and running the command can be seen below:

```
C:\>
C:\>cd XSLT
C:\XSLT>cd HSCOrgRefData_xslt
C:\XSLT\HSCOrgRefData_xslt>java -Xms1300M -Xmx1300M -jar c:\XSLT\HSCOrgRefData_xslt\Saxon\Saxon9he.jar -t -s:c:\s_xml_files\HSCOrgRefData_Full_20180828.xml -xsl:HSCOrgRefData_xslt\tocsv.xslt
```

On a machine with the specification listed in the Pre-requisites section the process takes 55 to 65 seconds to transform a file of around 400MB in size. However, if the process is using files that are across a network then the speed of the network will impact upon the performance of the transform, decreasing the speed of the process.

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.

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

```
Saxon-HE 9.4.0.9J from Saxonica
Java version 1.8.0_181
Stylesheet compilation time: 407 milliseconds
Processing file:/c:/s_xml_files/HSCOrgRefData_Full_20180828.xml
Using parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Building tree for file:/c:/s_xml_files/HSCOrgRefData_Full_20180828.xml using class net.sf.saxon.tree.tiny.TinyBuilder
Tree built in 17778 milliseconds
Tree size: 10195625 nodes, 15124758 characters, 7810883 attributes

### User parameters interpreted as:

### server-name: C:/HSCOrgRefData/
### share-name:
### interpreted output-path: file://C:/HSCOrgRefData//

Writing to file:/C:/HSCOrgRefData/Organisation_Details.csv
Writing to file:/C:/HSCOrgRefData/Contact_Details.csv
Writing to file:/C:/HSCOrgRefData/Role_Details.csv
Writing to file:/C:/HSCOrgRefData/Relationship_Details.csv
Writing to file:/C:/HSCOrgRefData/Successor_Details.csv
Writing to file:/C:/HSCOrgRefData/Additional_Attributes_Details.csv
Writing to file:/C:/HSCOrgRefData/OtherID_Details.csv
Writing to file:/C:/HSCOrgRefData/Manifest_Details.csv
Writing to file:/C:/HSCOrgRefData/PrimaryRole_Details.csv
Writing to file:/C:/HSCOrgRefData/Code_System_Details.csv
Execution time: 1m 11.355s (71355ms)
Memory used: 877203288
NamePool contents: 75 entries in 70 chains. 7 URIs
```

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

- The specification of the machine running the XSLT

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>

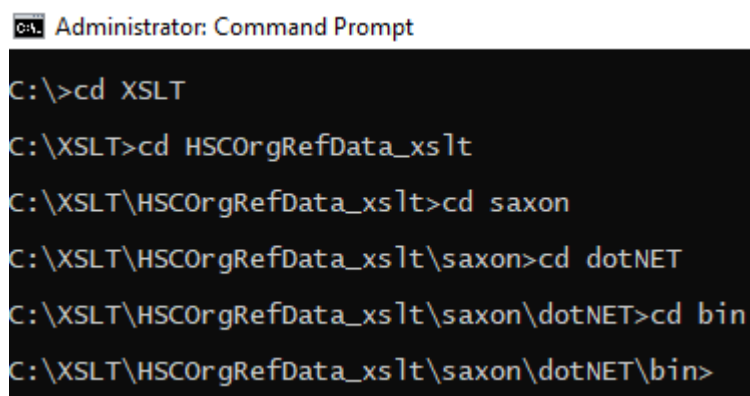
Command element	Description
Transform	Invoke the interpreter
<Saxon flags>	<p>-t = Display version and timing information to the standard error output. The output also traces the files that are read and writing, and extension modules that are loaded (see http://www.saxonica.com/html/documentation9.7/using-xsl/commandline/)</p> <p>-s: = Identifies the source XML file.</p> <p>-xsl: = Specifies the file containing the principal stylesheet module (XSLT).</p>
<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 the set of CSV files defined within the rules **HSCOrgRefData_xmltocsv.xslt**.

Please note this command is all on one line with elements separated by a single space:

Transform -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_xmltocsv.xslt

The full process of navigating to the HSCOrgRefData_xslt folder and running the command can be seen below:



```

C:\>cd XSLT
C:\XSLT>cd HSCOrgRefData_xslt
C:\XSLT\HSCOrgRefData_xslt>cd saxon
C:\XSLT\HSCOrgRefData_xslt\saxon>cd dotNET
C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET>cd bin
C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET\bin>
  
```

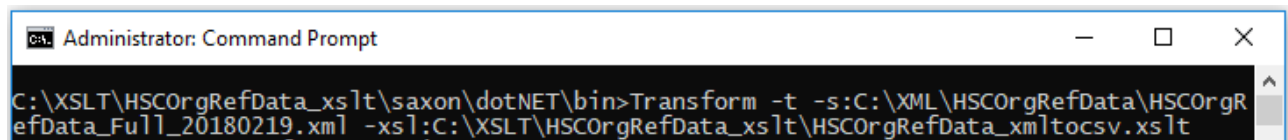
Or from the initial prompt from C:

'cd C:\XSLT\HSCOrgRefData_xslt\saxon\dotNET\bin'

Key in '**Transform -t -s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml -xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_xsltocsv.xslt**' and press enter.

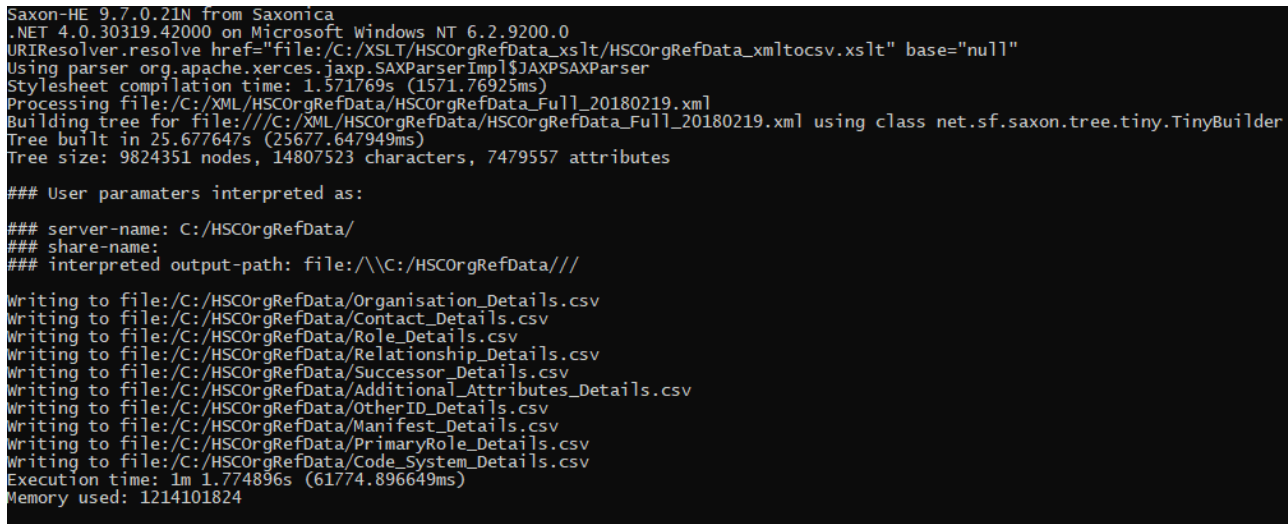
(NB quotes used for reference only)

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_xsltocsv.xslt
```

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



```
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_xsltocsv.xslt" base="null"
Using parser org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser
Stylesheet compilation time: 1.571769s (1571.76925ms)
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 25.677647s (25677.647949ms)
Tree size: 9824351 nodes, 14807523 characters, 7479557 attributes

### User paramaters interpreted as:
### server-name: C:/HSCOrgRefData/
### share-name:
### interpreted output-path: file://C:/HSCOrgRefData///

Writing to file:/C:/HSCOrgRefData/Organisation_Details.csv
Writing to file:/C:/HSCOrgRefData/Contact_Details.csv
Writing to file:/C:/HSCOrgRefData/Role_Details.csv
Writing to file:/C:/HSCOrgRefData/Relationship_Details.csv
Writing to file:/C:/HSCOrgRefData/Successor_Details.csv
Writing to file:/C:/HSCOrgRefData/Additional_Attributes_Details.csv
Writing to file:/C:/HSCOrgRefData/OtherID_Details.csv
Writing to file:/C:/HSCOrgRefData/Manifest_Details.csv
Writing to file:/C:/HSCOrgRefData/PrimaryRole_Details.csv
Writing to file:/C:/HSCOrgRefData/Code_System_Details.csv
Execution time: 1m 1.774896s (61774.896649ms)
Memory used: 1214101824
```

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

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_xmltocsv.xslt  
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"  
-s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml  
-xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_xmltocsv.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_xmltocsv.xslt  
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 "  
-s:C:\XML\HSCOrgRefData\HSCOrgRefData_Full_20180219.xml  
-xsl:C:\XSLT\HSCOrgRefData_xslt\HSCOrgRefData_xmltocsv.xslt
```

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

8. XSLT Output

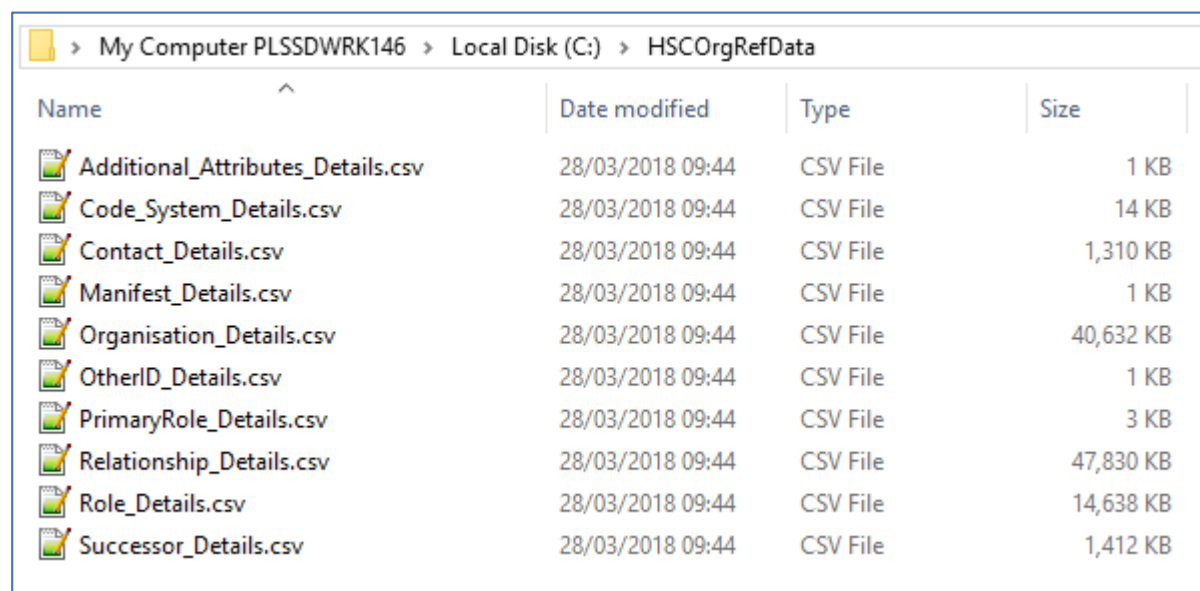
The XSLT process writes output to ten separate CSV files. The location and name of each output file is specified at appropriate points within the stylesheet definition. The sample output below illustrates how the name and location of the main organisation details are specified (see line 32 - file:///C:/HSCOrgRefData/Organisation_Details.csv).

```

22 <xsl:output method="text" encoding="utf-8"/>
23 <xsl:strip-space elements="*" />
24 <!-- Change the following separator variable value to the desired value.-->
25 <xsl:variable name="separator">","</xsl:variable>
26 <xsl:variable name="terminator">"</xsl:variable>
27 <!-- <xsl:variable name="booleanValue">>false</xsl:variable> -->
28
29 <xsl:template match="/">
30   <!-- Organisation Details -->
31   <!-- Header Row -->
32   <xsl:result-document method="text" href="file:///C:/HSCOrgRefData/Organisation_Details.csv">
33     <xsl:value-of select="$terminator" />
34     <xsl:text>OrgRecordClass</xsl:text><xsl:value-of select="$separator" />
35     <xsl:text>refOnly</xsl:text><xsl:value-of select="$separator" />
36     <xsl:text>Name</xsl:text><xsl:value-of select="$separator" />
37     <xsl:text>Operational Start Date</xsl:text><xsl:value-of select="$separator" />
38     <xsl:text>Operational End Date</xsl:text><xsl:value-of select="$separator" />
39     <xsl:text>Legal Start Date</xsl:text><xsl:value-of select="$separator" />

```

By default all files are written to a folder called “HSCOrgRefData” on the root of the C drive.



Name	Date modified	Type	Size
Additional_Attributes_Details.csv	28/03/2018 09:44	CSV File	1 KB
Code_System_Details.csv	28/03/2018 09:44	CSV File	14 KB
Contact_Details.csv	28/03/2018 09:44	CSV File	1,310 KB
Manifest_Details.csv	28/03/2018 09:44	CSV File	1 KB
Organisation_Details.csv	28/03/2018 09:44	CSV File	40,632 KB
OtherID_Details.csv	28/03/2018 09:44	CSV File	1 KB
PrimaryRole_Details.csv	28/03/2018 09:44	CSV File	3 KB
Relationship_Details.csv	28/03/2018 09:44	CSV File	47,830 KB
Role_Details.csv	28/03/2018 09:44	CSV File	14,638 KB
Successor_Details.csv	28/03/2018 09:44	CSV File	1,412 KB

9. Output File Descriptions

The tables below document the content of each output file. A description of the data mapped to each field can be found in the Data Item Catalogue. The data types specified below are generic data types which should align to any database.

Document Payload Content

Note the primary key for payload content is the OrganisationId (this is the value populated in the extension attribute of the Instance Identifier for the Organisation). All payload CSV files have the OrganisationId as a foreign key reference to enable consumers to build a set of related tables in landing tables.

9.1. Organisation_Details.csv

Field Name	Data Type	Example
OrganisationRecordClass	String	RC1
RefOnly	Boolean	true
Name	String	GREATER MANCHESTER STRATEGIC HA
OperationalStartDate	Date	2002-04-01
OperationalEndDate	Date	2006-06-30
LegalStartDate	Date	
LegalEndDate	Date	
OrganisationId (Primary Key)	String	Q14
OrganisationRoot	String	2.16.840.1.113883.2.1.3.2.4.18.48
AssigningAuthorityName	String	HSCIC
Status	String	Inactive
LastChangeDate	Date	2015-02-11
LocationId	Number	
AddrLn1	String	GATEWAY HOUSE
AddrLn2	String	PICCADILLY SOUTH
AddrLn3	String	
Town	String	MANCHESTER
County	String	GREATER MANCHESTER
PostCode	String	M60 7LP
Country	String	ENGLAND
UPRN	Number	976987654312

9.2. Contact_Details.csv

Field Name	Data Type	Example
OrganisationId (Foreign Key)	String	8DV59
Tel	String	01457850860

Fax	String	01457820017
MailTo	String	info@mh.surgery.nhs.net
Http	String	HTTP://WWW.AGEUK.ORG.UK

9.3. Roles_Details.csv

Field Name	Data Type	Example
OrganisationId (Foreign Key)	String	AKL
RoleId	String	RO172
UniqueRoleId (Primary Key)	Number	199678
OperationalStartDate	Date	2014-10-01
OperationalEndDate	Date	
LegalStartDate	Date	
LegalEndDate	Date	
Status	String	Active
PrimaryRole	Boolean	True

9.4. Relationships_Details.csv

Field Name	Data Type	Example
OrganisationId (Foreign Key)	String	RJX
RelationshipId	String	RE1
UniqueRelationshipId (Primary Key)	Number	73958
Derived	Boolean	true
OperationalStartDate	Date	1996-04-01
OperationalEndDate	Date	2002-03-31
LegalStartDate	Date	
LegalEndDate	Date	
Status	String	Inactive
TargetRootId	String	2.16.840.1.113883.2.1.3.2.4.18.48
TargetOrganisationId	String	QCX
TargetAssigningAuthorityName	String	HSCIC
TargetPrimaryRoleId	String	RO132
TargetUniqueRoleId	Number	87449

9.5. Successor_Details.csv

Field Name	Data Type	Example
OrganisationId (Foreign Key)	String	RVXKB

UniqueSuccessorId	Number	17104
Type	String	Predecessor
LegalStartDate	Date	1999-04-01
LegalEndDate	Date	1999-04-01
ForwardSuccession	Boolean	true
TargetRootId	String	2.16.840.1.113883.2.1.3.2.4.18.48
TargetOrganisationId	String	RRYKB
TargetAssigningAuthorityName	String	HSCIC
TargetPrimaryRoleId	String	RO198
TargetUniqueRoleId	Number	69915

Note – there is no single primary key on this table. A composite key can be built against a combination of OrganisationId and UniqueSuccessorId.

9.6. Additional_Attributes_Details.csv

Field Name	Data Type	Example
OrganisationId (Foreign Key)	String	11N
AttributeKey	String	SEVEN DAY OPENING
AttributeValue	String	YES
AttributeDate	Date	2016-04-01

9.7. OtherID_Details.csv

Field Name	Data Type	Example
OrganisationId (Foreign Key)	String	11N
OtherIdRoot	String	9.8.76.001
OtherOrganisationId	String	AY99VZZT
AssigningAuthorityName	String	COMPANIES HOUSE

Document Metadata Content

Note the primary key for metadata content is the Publication Sequence number populated in the Manifest.

9.8. Manifest_Details.csv

Field Name	Data Type	Example
OrgRefDataVersion	String	2.0.0
PublicationType	String	Full
PublicationSource	String	HSCIC
PublicationDate	Date	2015-11-30
PublicationSequenceNumber (Primary Key)	Number	203
FileCreationDateTime	TimeStamp	2018-09-24T20:00:00

FileExpiryDate	Date	
RecordCount	Number	192123
ContentDescription	String	HSCOrgRefData_Full_20180924

9.9. PrimaryRole_Details.csv

Field Name	Data Type	Example
RoleId (Primary Key)	String	RO197
DisplayName	String	NHS TRUST
PublicationSequenceNumber (Foreign Key)	Number	203

9.10. Code_System_Details.csv

Field Name	Data Type	Example
CodeSystemName	String	OrganisationRelationship
OID	String	2.16.840.1.113883.2.1.3.2.4.17.508
Id (Primary Key)	String	RE1
Code	Number	1
DisplayName	String	HAS A LEGACY RELATIONSHIP TO
PublicationSequenceNumber (Foreign Key)	Number	203