How to evaluate a LIXI2 message against business rules written in [Schematron](http://schematron.com/) using C# and .Net Framework

# Prerequisites

* Basic knowledge of C#
* Basic knowledge of XSLT
* Some level of familiarities with visual studio

# How Schematron works

Below is a diagram that shows how different part of the Schematron come together to evaluate an xml message against rules written in Schematron

**Schematron.sch**

**LIXI2\_Message.xml**

**Saxon xslt2 engine**

**Iso\_svrl\_for\_xslt2.xsl**

**Iso\_** **schematron \_for\_saxon.xsl**

Internally calls

**Output.xml**

**Saxon xslt2 engine** is where all the magic happens. Saxon engine relies on [iso\_svrl\_for\_xslt2.xsl](https://github.com/Schematron/stf/blob/master/iso-schematron-xslt2/iso_svrl_for_xslt2.xsl)stylesheet which provides the ISO implementation of Schematron Validation Report Language (SVRL). The engine first uses this stylesheet ([iso\_svrl\_for\_xslt2.xsl](https://github.com/Schematron/stf/blob/master/iso-schematron-xslt2/iso_svrl_for_xslt2.xsl))to transform **Schematron.sch** file to **Schematron.xsl** stylesheet**.** Then the same engine will use **Schematron.xsl** stylesheetto transform LIXI\_Message.xml document and to output **output.xml** file. Following are the summary of steps that validates an xml document against Schematron document.

1. XSLT engine Uses **Iso\_svrl\_for\_xslt2.xsl** to transform **Schematron.sch 🡪 Schematron.xsl**
2. XSLT engine Uses **Schematron.xsl** to transform **LIXI2\_Message.xml 🡪 Output.xml**

## C# .net implementation

To implement this solution using C# and .net framework we do need to have access to **Iso\_svrl\_for\_xslt2.xsl** stylesheet and the engine that dose the transformation.

Luckily there is an open source NuGet package called [Saxon-HE](https://www.nuget.org/packages/Saxon-HE/) that we will use in our solution to do the transformation. However as you can see from the previous diagram, [iso\_svrl\_for\_xslt2.xsl](https://github.com/Schematron/stf/blob/master/iso-schematron-xslt2/iso_svrl_for_xslt2.xsl) stylesheet internally relies on [iso\_schematron\_skeleton\_for\_saxon.xsl](https://github.com/Schematron/stf/blob/master/iso-schematron-xslt2/iso_schematron_skeleton_for_saxon.xsl)stylesheet. Therefore we need them both.

The following is the class that does the transformation

public class XSLTransform

{

public Stream Transform(Uri xmluri, Uri xsluri)

{

// Create a Processor instance.

Processor processor = new Processor();

// Load the source document

XdmNode input = processor.NewDocumentBuilder().Build(xmluri);

// Create a transformer for the stylesheet.

var compiler = processor.NewXsltCompiler();

compiler.ErrorList = new List<StaticError>();

XsltTransformer transformer = compiler.Compile(xsluri).Load();

if (compiler.ErrorList.Count != 0)

throw new Exception("Exception loading xsl!");

// Set the root node of the source document to be the initial context node

transformer.InitialContextNode = input;

// Create a serializer

Serializer serializer = processor.NewSerializer();

MemoryStream results = new MemoryStream();

serializer.SetOutputStream(results);

// Transform the source XML to System.out.

transformer.Run(serializer);

//get the string

results.Position = 0;

return results;

}

public Stream Transform(Stream xmlstream, Stream xslstream)

{

// Create a Processor instance.

Processor processor = new Processor();

// Load the source document

var documentbuilder = processor.NewDocumentBuilder();

documentbuilder.BaseUri = new Uri("file://c:/");

XdmNode input = documentbuilder.Build(xmlstream);

// Create a transformer for the stylesheet.

var compiler = processor.NewXsltCompiler();

compiler.ErrorList = new List<StaticError>();

compiler.XmlResolver = new XmlUrlResolver();

XsltTransformer transformer = compiler.Compile(xslstream).Load();

if (compiler.ErrorList.Count != 0)

throw new Exception("Exception loading xsl!");

// Set the root node of the source document to be the initial context node

transformer.InitialContextNode = input;

// Create a serializer

Serializer serializer = processor.NewSerializer();

MemoryStream results = new MemoryStream();

serializer.SetOutputStream(results);

// Transform the source XML to System.out.

transformer.Run(serializer);

//get the string

results.Position = 0;

return results;

}

Step 1:

schematronTransform = XSLTransform.Transform(

Uri.Path.to.Schematorn.sch,

Uri.Path.to.iso\_svrl\_for\_xslt2.xsl

)

Step 2:

FileStream xmlstream = new FileStream(Uri.Path.to.Input.xml, FileMode.Open, FileAccess.Read, FileShare.Read);

Step 3:

Then resultOutput = XSLTransform.Transform(

xmlstream,

schematronTransform

)

From here you could further process the resultOutput and extract the necessary information. Please check out the following repository for a complete solution and step by step implementation in .net framework using C#