

## Saxon - Bug #4983

### Unable to read categories.xml file with Saxon HE 10.5 .NET Query.exe

2021-05-04 23:36 - Martin Honnen

<b>Status:</b>	Closed	<b>Start date:</b>	2021-05-04
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	O'Neil Delpratt	<b>% Done:</b>	100%
<b>Category:</b>	Build and release	<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>		<b>Spent time:</b>	0:00 hour
<b>Legacy ID:</b>		<b>Fix Committed on Branch:</b>	10
<b>Applies to branch:</b>	10	<b>Fixed in Maintenance Release:</b>	10.5.1
<b>Description</b>			

## History

### #1 - 2021-05-04 23:38 - Martin Honnen

The code

```
let $xsl :=
  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="3.0"
    xmlns:functx="http://www.functx.com"
    xmlns:mf="http://example.com/mf"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    exclude-result-prefixes="#all">
    <xsl:import href="http://www.xsltfunctions.com/xsl/functx-1.0-nodoc-2007-01.xsl"/>
    <xsl:function name="mf:words-to-camel-case" as="xs:string" visibility="public">
      <xsl:param name="input" as="xs:string"/>
      <xsl:sequence select="functx:words-to-camel-case($input)"/>
    </xsl:function>
  </xsl:stylesheet>
return
  transform(
    map {
      'stylesheet-node' : $xsl,
      'initial-function' : QName('http://example.com/mf', 'words-to-camel-case'),
      'delivery-format' : 'raw',
      'function-params' : ['This is a text with some words']
    }?output
```

when run through Saxon 10.5 HE .NET Query causes the error

```
<?xml version="1.0" encoding="UTF-8"?>
Unbehandelte Ausnahme: System.TypeInitializationException: Der Typeninitialisierer für "net.sf.saxon.regex.charclass.Categories" hat eine Ausnahme verursacht. ---> java.lang.RuntimeException: Unable to read categories.xml file
bei net.sf.saxon.regex.charclass.Categories.build()
bei net.sf.saxon.regex.charclass.Categories.getCategory(String cat)
bei net.sf.saxon.regex.charclass.Categories..cctor()
--- Ende der internen Ausnahmestapelüberwachung ---
bei net.sf.saxon.regex.charclass.Categories.getCategory(String cat)
bei net.sf.saxon.regex.RECompiler.escape(Boolean )
bei net.sf.saxon.regex.RECompiler.parseTerminal(Int32[] )
bei net.sf.saxon.regex.RECompiler.piece(Int32[] )
bei net.sf.saxon.regex.RECompiler.parseBranch()
bei net.sf.saxon.regex.RECompiler.parseExpr(Int32[] )
bei net.sf.saxon.regex.RECompiler.parseTerminal(Int32[] )
bei net.sf.saxon.regex.RECompiler.piece(Int32[] )
bei net.sf.saxon.regex.RECompiler.parseBranch()
bei net.sf.saxon.regex.RECompiler.parseExpr(Int32[] )
bei net.sf.saxon.regex.RECompiler.compile(UnicodeString pattern)
bei net.sf.saxon.regex.ARegularExpression..ctor(CharSequence pattern, String flags, String hostLanguage, List warnings, Configuration config)
bei net.sf.saxon.dotnet.DotNetPlatform.compileRegularExpression(Configuration config, CharSequence regex, S
```

```

tring flags, String hostLanguage, List warnings)
    bei net.sf.saxon.Configuration.compileRegularExpression(CharSequence regex, String flags, String hostLanguage, List warnings)
    bei net.sf.saxon.functions.RegexFunction.tryToBindRegularExpression(Expression[] )
    bei net.sf.saxon.functions.RegexFunction.makeFunctionCall(Expression[] arguments)
    bei net.sf.saxon.functions.Replace.makeFunctionCall(Expression[] arguments)
    bei net.sf.saxon.functions.RegexFunctionSansFlags.makeFunctionCall(Expression[] arguments)
    bei net.sf.saxon.functions.registry.BuiltInFunctionSet.bind(F symbolicName, Expression[] staticArgs, StaticContext env, List reasons)
    bei net.sf.saxon.functions.FunctionLibraryList.bind(F functionName, Expression[] staticArgs, StaticContext env, List reasons)
    bei net.sf.saxon.expr.parser.XPathParser.parseFunctionCall(Expression prefixArgument)
    bei net.sf.saxon.expr.parser.XPathParser.parseBasicStep(Boolean firstInPattern)
    bei net.sf.saxon.expr.parser.XPathParser.parseStepExpression(Boolean firstInPattern)
    bei net.sf.saxon.expr.parser.XPathParser.parseRelativePath()
    bei net.sf.saxon.expr.parser.XPathParser.parsePathExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parseSimpleMappingExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parseUnaryExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parseExprSingle()
    bei net.sf.saxon.expr.parser.XPathParser.parseFunctionArgument()
    bei net.sf.saxon.expr.parser.XPathParser.parseFunctionCall(Expression prefixArgument)
    bei net.sf.saxon.expr.parser.XPathParser.parseBasicStep(Boolean firstInPattern)
    bei net.sf.saxon.expr.parser.XPathParser.parseStepExpression(Boolean firstInPattern)
    bei net.sf.saxon.expr.parser.XPathParser.parseRelativePath()
    bei net.sf.saxon.expr.parser.XPathParser.parsePathExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parseSimpleMappingExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parseUnaryExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parseExprSingle()
    bei net.sf.saxon.expr.parser.XPathParser.parseExpression()
    bei net.sf.saxon.expr.parser.XPathParser.parse(String expression, Int32 start, Int32 terminator, StaticContext env)
    bei net.sf.saxon.expr.parser.ExpressionTool.make(String expression, StaticContext env, Int32 start, Int32 terminator, CodeInjector codeInjector)
    bei net.sf.saxon.style.StyleElement.makeExpression(String expression, AttributeInfo att)
    bei net.sf.saxon.style.XSLSequence.prepareAttributes()
    bei net.sf.saxon.style.StyleElement.processAttributes()
    bei net.sf.saxon.style.StyleElement.processAllAttributes()
    bei net.sf.saxon.style.StyleElement.processAllAttributes()
    bei net.sf.saxon.style.PrincipalStylesheetModule.processAllAttributes()
    bei net.sf.saxon.style.PrincipalStylesheetModule.preprocess(Compilation compilation)
    bei net.sf.saxon.style.Compilation.compilePackage(Source source)
    bei net.sf.saxon.style.StylesheetModule.loadStylesheet(Source styleSource, Compilation compilation)
    bei net.sf.saxon.style.Compilation.compileSingletonPackage(Configuration config, CompilerInfo compilerInfo, Source source)
    bei net.sf.saxon.s9api.XsltCompiler.compile(Source source)
    bei net.sf.saxon.functions.TransformFn.getStylesheet(Map , XsltCompiler , String , XPathContext )
    bei net.sf.saxon.functions.TransformFn.call(XPathContext context, Sequence[] arguments)
    bei net.sf.saxon.expr.FunctionCall.iterate(XPathContext context)
    bei net.sf.saxon.expr.Expression.evaluateItem(XPathContext context)
    bei net.sf.saxon.expr.LookupExpression.iterate(XPathContext context)
    bei net.sf.saxon.expr.Expression.process(Outputter output, XPathContext context)
    bei net.sf.saxon.query.XQueryExpression.run(DynamicQueryContext env, Result result, Properties outputProperties)
    bei net.sf.saxon.s9api.XQueryEvaluator.run(Destination destination)
    bei net.sf.saxon.Query.runQuery(XQueryExecutable exp, XQueryEvaluator evaluator, Source input, Destination destination)
    bei net.sf.saxon.Query.doQuery(String[] args, String command)
    bei Saxon.Cmd.DotNetQuery.Main(String[] args)

```

## #2 - 2021-05-04 23:39 - Martin Honnen

Somehow I have managed to fumble a shortened title into this issue, I can't seem to edit it: should say: "Unable to read categories.xml file with Saxon HE 10.5 .NET Query.exe"

## #3 - 2021-05-04 23:42 - Martin Honnen

Works fine with 10.3 .NET.

## #4 - 2021-05-04 23:44 - Michael Kay

- Subject changed from *categ* to *Unable to read categories.xml file with Saxon HE 10.5 .NET Query.exe*

## #5 - 2021-05-04 23:48 - Michael Kay

Thanks for reporting it. Looks like a build error - we've rewritten all the build scripts in Gradle. The scripts do a lot of self-testing (a lot more than we

used to do), so the failure is certainly surprising, but I think it must be in that area.

## #6 - 2021-05-05 21:12 - Martin Honnen

Much shorter, reduced example triggering the exception is

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="3.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="#all">

  <xsl:param name="input-string" as="xs:string">This is a text with some words</xsl:param>

  <xsl:template name="xsl:initial-template">
    <xsl:value-of select="tokenize($input-string, '\s+')"/>
  </xsl:template>

</xsl:stylesheet>
```

when run with Saxon Java HE 10.5 and e.g. `-it -xsl:above-code.xml` it works fine, with Saxon .NET HE 10.5 Transform.exe it fails with

Unbehandelte Ausnahme: System.TypeInitializationException: Der Typeninitialisierer für "net.sf.saxon.regex.charclass.Categories" hat eine Ausnahme verursacht. ---> java.lang.RuntimeException: Unable to read categories.xml file

```
bei net.sf.saxon.regex.charclass.Categories.build()
bei net.sf.saxon.regex.charclass.Categories.getCategory(String cat)
bei net.sf.saxon.regex.charclass.Categories..ctor()
--- Ende der internen Ausnahmestapelüberwachung ---
bei net.sf.saxon.regex.RECompiler.escape(Boolean )
bei net.sf.saxon.regex.RECompiler.parseTerminal(Int32[] )
bei net.sf.saxon.regex.RECompiler.piece(Int32[] )
bei net.sf.saxon.regex.RECompiler.parseBranch()
bei net.sf.saxon.regex.RECompiler.parseExpr(Int32[] )
bei net.sf.saxon.regex.RECompiler.compile(UnicodeString pattern)
bei net.sf.saxon.regex.ARegularExpression..ctor(CharSequence pattern, String flags, String hostLanguage, List warnings, Configuration config)
bei net.sf.saxon.dotnet.DotNetPlatform.compileRegularExpression(Configuration config, CharSequence regex, String flags, String hostLanguage, List warnings)
bei net.sf.saxon.Configuration.compileRegularExpression(CharSequence regex, String flags, String hostLanguage, List warnings)
bei net.sf.saxon.functions.RegexFunction.tryToBindRegularExpression(Expression[] )
bei net.sf.saxon.functions.RegexFunction.makeFunctionCall(Expression[] arguments)
bei net.sf.saxon.functions.RegexFunctionSansFlags.makeFunctionCall(Expression[] arguments)
bei net.sf.saxon.functions.registry.BuiltInFunctionSet.bind(F symbolicName, Expression[] staticArgs, StaticContext env, List reasons)
bei net.sf.saxon.functions.FunctionLibraryList.bind(F functionName, Expression[] staticArgs, StaticContext env, List reasons)
bei net.sf.saxon.expr.parser.XPathParser.parseFunctionCall(Expression prefixArgument)
bei net.sf.saxon.expr.parser.XPathParser.parseBasicStep(Boolean firstInPattern)
bei net.sf.saxon.expr.parser.XPathParser.parseStepExpression(Boolean firstInPattern)
bei net.sf.saxon.expr.parser.XPathParser.parseRelativePath()
bei net.sf.saxon.expr.parser.XPathParser.parsePathExpression()
bei net.sf.saxon.expr.parser.XPathParser.parseSimpleMappingExpression()
bei net.sf.saxon.expr.parser.XPathParser.parseUnaryExpression()
bei net.sf.saxon.expr.parser.XPathParser.parseExprSingle()
bei net.sf.saxon.expr.parser.XPathParser.parseExpression()
bei net.sf.saxon.expr.parser.XPathParser.parse(String expression, Int32 start, Int32 terminator, StaticContext env)
bei net.sf.saxon.expr.parser.ExpressionTool.make(String expression, StaticContext env, Int32 start, Int32 terminator, CodeInjector codeInjector)
bei net.sf.saxon.style.StyleElement.makeExpression(String expression, AttributeInfo att)
bei net.sf.saxon.style.XSLValueOf.prepareAttributes()
bei net.sf.saxon.style.StyleElement.processAttributes()
bei net.sf.saxon.style.StyleElement.processAllAttributes()
bei net.sf.saxon.style.StyleElement.processAllAttributes()
bei net.sf.saxon.style.XSLTemplate.processAllAttributes()
bei net.sf.saxon.style.PrincipalStylesheetModule.processAllAttributes()
bei net.sf.saxon.style.PrincipalStylesheetModule.preprocess(Compilation compilation)
bei net.sf.saxon.style.Compilation.compilePackage(Source source)
bei net.sf.saxon.style.StylesheetModule.loadStylesheet(Source styleSource, Compilation compilation)
bei net.sf.saxon.style.Compilation.compileSingletonPackage(Configuration config, CompilerInfo compilerInfo, Source source)
bei net.sf.saxon.s9api.XsltCompiler.compile(Source source)
bei net.sf.saxon.Transform.doTransform(String[] args, String command)
bei Saxon.Cmd.DotNetTransform.Main(String[] args)
```

**#7 - 2021-05-06 10:20 - O'Neil Delpratt**

- Status changed from New to In Progress
- Priority changed from Low to Normal
- Applies to branch 10 added

Hi Martin,

I ran the simple stylesheet you have supplied against all are .NET products:

- Saxon-HE - reproduced exception as described in comment #6
- Saxon-PE - Succeeds
- Saxon-EE - Succeeds

As Mike mentioned it is a build issue.

**#8 - 2021-05-06 10:37 - O'Neil Delpratt**

- Category set to Build and release
- Assignee set to O'Neil Delpratt
- % Done changed from 0 to 100
- Fix Committed on Branch 10 added

In the gradle task hen\_jar we are missing the following:

```
from ("src/main/resources") {
    include "net/sf/saxon/data/**"
}
```

Fixed in the file build-n.gradle and available for the next maintenance release. There seems to some gaps on the testing side on .NET for Saxon-HEN especially in the build scripts, which we will address.

**#9 - 2021-05-06 19:28 - O'Neil Delpratt**

- Status changed from In Progress to Closed
- Fixed in Maintenance Release 10.5.1 added

Bug fixed in the gradle build script.

We have made available on Sourceforge and Nuget a special Saxon-HEN 10.5.1 release with this fix.