Saxon - Bug #2934
Saxon 9.7 hangs when processing XSLT 3.0 stylesheet using streaming and xsl:iterate/xsl:break with early exit

2016-09-12 15:43 - Martin Honnen

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
<th>Start date:</th>
<th>2016-09-12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Michael Kay</td>
<td>% Done:</td>
<td>100%</td>
</tr>
<tr>
<td>Category:</td>
<td>Streaming</td>
<td>Estimated time:</td>
<td>0:00 hour</td>
</tr>
<tr>
<td>Sprint/Milestone:</td>
<td></td>
<td>Spent time:</td>
<td>0:00 hour</td>
</tr>
<tr>
<td>Legacy ID:</td>
<td></td>
<td>Fix Committed on</td>
<td>9.7, trunk</td>
</tr>
<tr>
<td>Applies to branch:</td>
<td>9.7</td>
<td>Branch:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fixed in Maintenance</td>
<td>9.7.0.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Release:</td>
<td></td>
</tr>
</tbody>
</table>

Description
When trying the example within http://saxonica.com/html/documentation/sourcedocs/streaming/partial-reading.html Saxon-EE 9.7.0.8J hangs when run from the command line and processing an input document with streaming that causes an early exit.

The current XSLT I have is a bit different from the example in the documentation as there the use of lt seems to only make sense if a schema is present and as Saxon told me to move the xsl:on-completion@. I also used @xsl:source-document instead of @xsl:stream@:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:math="http://www.w3.org/2005/xpath-functions/math" exclude-result-prefixes="xs math"
    version="3.0">
    <xsl:output method="text"/>
    <xsl:param name="input-uri" as="xs:string" select='"test2016091202.xml"'/>
    <xsl:variable name="contains-debit" as="xs:boolean">
        <xsl:source-document href="{$input-uri}" streamable="yes">
            <xsl:iterate select="./transaction">
                <xsl:on-completion select="false()"/>
                <xsl:if test="@value &lt; 0">[transaction value="5.13" date="2015-12-01"/>-->
                <xsl:break select="true()"/>
            </xsl:if>
        </xsl:iterate>
    </xsl:source-document>
    </xsl:variable>
    <xsl:template name="xsl:initial-template">
        <xsl:value-of select="$contains-debit"/>
    </xsl:template>
</xsl:stylesheet>
```

Now when I run it from the command line with -it against an input document as input-uri not having a transaction with a negative @value@, as for instance:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
    <transaction value="15.1" date="2015-09-12"/>
    <transaction value="1.0" date="2016-01-01"/>
</transactions>
```

2019-10-13
then transformation works fine and outputs false (but indicates a streamed processing when using the -t option).

However when I run it against an input-uri having a transaction with a negative value

```xml
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
  <transaction value="15.1" date="2015-09-12"/>
  <transaction value="-5.13" date="2015-12-01"/>
  <transaction value="1.0" date="2016-01-01"/>
  <transaction value="20.35" date="2016-02-01"/>
</transactions>
```

then Saxon hangs and does not output anything at all (no -t option) or only the messages about streaming and early exit (with -t option):

```
Processing (no source document) initial template = xsl:initial-template
URIResolver.resolve href="test2016091202.xml" base="file:/C:/SomePath/test201609120201.xsl"
Streaming input document file:/C:/SomePath/test2016091202.xml
Using parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Streaming test2016091202.xml : early exit
```

**History**

#1 - 2016-09-12 23:34 - Michael Kay

Thanks. Problem reproduced.

The key thing here is that the xsl:stream instruction is being evaluated within a variable (or function), which means it's in "pull mode" - the instruction is evaluated using an iterate() method where the consumer is in control. Because we always use a push parser, this leads to a push-pull conflict which we resolve by using two threads. Up until 9.6, we had our own class for managing the flow of data between these threads (called a Conduit); in 9.7, we switched to using a BlockingQueue provided by the Java concurrency framework.

The xsl:break is issued on the push side and successfully causes the push side thread to quit parsing, but I think it's failing to notify the pull thread of the fact that it has done so. I think the code is written on the assumption that the decision to exit early will always be taken on the pull side.

#2 - 2016-09-12 23:45 - Michael Kay

It seems we can solve the problem by ensuring that when the push-side code (PushToPull.provider()) catches a QuitParsingException, it writes the "stopper" item indicating end of sequence to the shared queue; the pull-side code recognises this and closes down tidily.

But although it solves this example, I need to check carefully that it doesn't stop other paths working correctly.

#3 - 2016-09-19 13:42 - Michael Kay

I've applied this change.

I'm currently getting 56 failures running the XSLT 3.0 test suite against Saxon 9.7. Of course it's likely that none of these failures have anything to do with this patch, but I need to do more work studying the failures before I can be sure. Most of them are likely to be caused by recent minor spec and/or test changes that have misaligned the product, the tests, and the spec.
#4 - 2016-10-03 19:15 - Michael Kay
- Category set to Streaming
- Status changed from New to Resolved
- Assignee set to Michael Kay
- Priority changed from Low to Normal
- Applies to branch 9.8 added
- Fix Committed on Branch 9.7, 9.8 added

I have satisfied myself that this solution is sound and have committed the change to the 9.7 and 9.8 branches.

#5 - 2016-10-21 17:58 - O'Neil Delpratt
- Status changed from Resolved to Closed
- % Done changed from 0 to 100
- Fixed in Maintenance Release 9.7.0.10 added

Bug fix applied in the Saxon 9.7.0.10 maintenance release

#6 - 2017-02-05 16:11 - Michael Kay
I have belatedly added this test case as si-iterate-099

#7 - 2017-05-23 16:05 - O'Neil Delpratt
- Applies to branch deleted (9.8)
- Fix Committed on Branch trunk added
- Fix Committed on Branch deleted (9.8)