Saxon - Bug #2104

ArrayIndexOutOfBounds in Chain.itemAt()

2014-07-14 10:03 - Michael Kay

Status:	Closed	**Start date:**	2014-07-14
Priority:	Normal	**Due date:**	
Assignee:	Michael Kay	**% Done:**	100%
Category:	Internals	**Estimated time:**	0:00 hour
Sprint/Milestone:		**Spent time:**	0:00 hour
Legacy ID:		**Fix Committed on Branch:**	
Applies to branch:		**Fixed in Maintenance Release:**	

Description

Added by Vladimir Nesterovsky (on the help forum) 31 minutes ago

A while ago I already reported a problem "Saxon Exception during execution in Saxon-HE-9.5.1-2".

Now I got similar exception in Saxon-HE-9.5.1-6.

Please consider the code:

<xsl:stylesheet version="2.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns:t="this"

exclude-result-prefixes="xs">

<xsl:template match="/">

```
<xsl:param name="new-line-text" as="xs:string" select="'&#10;'"/>

<xsl:variable name="items" as="item()*" select="'select', $new-line-text"/>


<xsl:message select="t:string-join($items)"/>
```

</xsl:template>

<xsl:function name="t:string-join" as="item()*">

```
<xsl:param name="items" as="item()*"/>


<xsl:variable name="indices" as="xs:integer*" select="
  0,
  index-of
  (
    (
      for $item in $items return
        $item instance of xs:string
```

```
    ),
    false()
  ),
  count($items) + 1"/>

<xsl:sequence select="
  for $i in 1 to count($indices) - 1 return
  (
    $items[$indices[$i]],
    string-join
    (
      subsequence
      (
        $items,
        $indices[$i] + 1,
        $indices[$i + 1] - $indices[$i] - 1
      ),
      ''
    )
  )"/>
```

</xsl:function>

</xsl:stylesheet>

The output is:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1

```
at net.sf.saxon.om.Chain.itemAt(Chain.java:161)
at net.sf.saxon.om.SequenceTool.itemAt(SequenceTool.java:130)
at net.sf.saxon.expr.FilterExpression.iterate(FilterExpression.java:1143)
at net.sf.saxon.expr.LetExpression.iterate(LetExpression.java:365)
at net.sf.saxon.expr.instruct.BlockIterator.next(BlockIterator.java:49)
at net.sf.saxon.expr.MappingIterator.next(MappingIterator.java:70)
at net.sf.saxon.expr.instruct.Message.processLeavingTail(Message.java:264)
at net.sf.saxon.expr.instruct.Block.processLeavingTail(Block.java:660)
at net.sf.saxon.expr.instruct.Template.applyLeavingTail(Template.java:239)
at net.sf.saxon.trans.Mode.applyTemplates(Mode.java:1057)
```

```
at net.sf.saxon.Controller.transformDocument(Controller.java:2088)

at net.sf.saxon.Controller.transform(Controller.java:1911)

...
```

History

#1 - 2014-07-14 13:16 - Vladimir Nesterovsky

This refactoring, which does not throw, might give more hints:

```
  <xsl:function name="t:string-join" as="item()*">
    <xsl:param name="items" as="item()*"/>

    <xsl:variable name="indices" as="xs:integer*" select="
      0,
      index-of
      (
        (
          for $item in $items return
            $item instance of xs:string
        ),
        false()
      ),
      count($items) + 1"/>

    <xsl:sequence select="
      for
        $i in 1 to count($indices) - 1,
        $s in $indices[$i],
        $e in $indices[$i + 1]
      return
      (
        $items[$s],
        string-join(subsequence($items, $s + 1, $e - $s - 1), '')
      )"/>
  </xsl:function>
```

#2 - 2014-07-14 20:55 - Michael Kay

The problem is caused by the expression $items[$indices[$i]], which is the kind of expression affected by the changes for bug https://saxonica.plan.io/issues/2077.

Although surprisingly, the problem in this case occurs only when there is no bytecode generation.

In this case $indices[$i] is zero, which should mean that the expression returns an empy sequence, but the code is subtracting 1 and attempting to access item -1 in the underlying array, hence the AAIOB with index -1.

#3 - 2014-07-14 21:09 - Michael Kay

The problem actually seems to have been introduced by the change for

https://saxonica.plan.io/issues/1924

The itemAt() method is expected to return null if the index is out of range, but Chain.itemAt is not performing this check.

#4 - 2014-07-14 21:15 - Michael Kay

- Status changed from In Progress to Resolved

I have committed a patch to Chain.itemAt() on the 9.5 and 9.6 branches to check for index<0.

#5 - 2014-08-06 14:43 - O'Neil Delpratt

- Status changed from Resolved to Closed

- % Done changed from 0 to 100

- Fixed in version set to 9.5.1.7

Bug fix applied in Saxon maintenance release 9.5.1.7