

Saxon-JS - Bug #4829

Static variable initialised to node values cause run-time failure

2020-11-18 15:10 - John Lumley

Status: New	Start date: 2020-11-18
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0:00 hour
Sprint/Milestone:	Spent time: 0:00 hour
Applies to JS Branch: 2	Fixed in JS Release:
Fix Committed on JS Branch:	SEF Generated with:

Description

When compiling with the XJ compiler a static variable initialised to a node such as:

```
<xsl:variable name="t" static="yes" select="doc('testStatic.xml')"/>
```

where testStatic.xml is:

```
<foo a="1" xmlns:b="BBB" b:attribute="" xmlns="Flibbertygibbet">
  <bar/>
</foo>
```

the generated package contains a component definition:

```
<co id='0' binds=''>
  <globalVariable name='Q{}t' as='1ND' line='7' ... visibility='PRIVATE' flags='s'>
    <node kind='9' content='&lt;foo xmlns="Flibbertygibbet" xmlns:b="BBB" a="1" b:attribute=""&gt;
&#xA;  &lt;bar/&gt;&#xA;&lt;/foo&gt;' baseUrl='...testStatic.xml' />
  </globalVariable>
</co>
```

that is the serialization of the document is held as the content property of a node instruction, with the kind of the node indicated.

If the selection is doc('testStatic.xml')/(*,*/@*) the compilation becomes:

```
<globalVariable name='Q{}t' as='*N' line='7' ... visibility='PRIVATE' flags='s'>
  <literal count='3'>
    <node kind='1' content='&lt;foo xmlns="Flibbertygibbet" xmlns:b="BBB" a="1" b:attribute=""&gt;
&#xA;  &lt;bar/&gt;&#xA;&lt;/foo&gt;' baseUrl='.../testStatic.xml' />
    <node kind='2' localName='a' content='1' />
    <node kind='2' localName='attribute' prefix='b' ns='BBB' content='' />
  </literal>
</globalVariable>
```

Execution of a reference to this variable in SaxonJS2 throws an error as there is currently no support for a node instruction.

There appear to be two routes to solution:

1. Converting the export to a nested set of already supported node construction instructions elem, att, text etc, which will be very costly and increase the export size considerably.
2. Add support for a node instruction, using the implementation code for parse-xml() when necessary

The latter seems much more sensible.

Related issues:

Related to Saxon - Bug #4035: Exporting a stylesheet containing node-valued s...

Closed

2018-11-19

History

#1 - 2020-11-18 16:00 - Michael Kay

I thought we had an open issue on this, but I can't find it.

I think we should probably ban it. While we can handle some simple cases by putting a serialization of the node into the SEF file, it doesn't handle the general case, where for example \$x is bound to a document node, and \$y is bound to \$x*. We then get to the kind of problem that Java serialization faces, of serializing a general graph with arbitrary relationships between the nodes. Unless we're going to do that properly, I don't think we should do it at all.

A simple rule (you can't export a stylesheet if there are static variables bound to nodes) is easier to understand than a more complex set of restrictions.

#2 - 2021-01-14 17:07 - Michael Kay

- Related to Bug #4035: Exporting a stylesheet containing node-valued static variables added

#3 - 2021-01-14 17:07 - Michael Kay

This was actually covered on the Saxon-J side by bug [#4035](#). To resolve that bug (on the Java side) we added capability to export node-valued static variables to a SEF file in Saxon-J, but we didn't add the corresponding code to import them in Saxon-JS.

#4 - 2021-02-11 13:28 - Community Admin

- Applies to JS Branch 2 added

- Applies to JS Branch deleted (2.0)