

## Saxon-JS - Bug #4588

### xsl:initial-template seems to be executed by fn:transform called from SaxonJS.XPath.evaluate although options set initial-match-selection and global-context-item

2020-06-14 20:27 - Martin Honnen

<b>Status:</b>	Resolved	<b>Start date:</b>	2020-06-14
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Lumley	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>		<b>Spent time:</b>	0:00 hour
<b>Applies to JS Branch:</b>	2.0, Trunk	<b>Fixed in JS Release:</b>	
<b>Fix Committed on JS Branch:</b>	Trunk	<b>SEF Generated with:</b>	

#### Description

I am trying to call fn:transform from XPath, providing the result of a parse-json call as both the initial-match-selection and the global-context-item.

When the XSLT used has an xsl:template match=".", the template seems to be applied using apply-templates, however, when the template additionally has xsl:template match="." name="xsl:initial-template", the template is executed but I get an error about the focus being absent.

With the Java version of Saxon it appears that my combination of options to fn:transform always results in the expected apply-templates invocation, I think Saxon-JS 2's behaviour is a bug.

Javascript code showing the error is

```
require('saxon-js');

const xslt = `<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="3.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:map="http://www.w3.org/2005/xpath-functions/map"
  exclude-result-prefixes="#all"
  expand-text="yes">

  <xsl:output method="adaptive"/>

  <xsl:template match="." name="xsl:initial-template">
    <result>
      <xsl:comment>
        Run with {system-property('xsl:product-name')} {system-property('xsl:product-version')}
      </xsl:comment>
      <xsl:sequence select="."/>
    </result>
  </xsl:template>

</xsl:stylesheet>`;

const json = `[1, 2, 3]`;

function transform(input, xslt, inputType, resultsSelect) {
  var transformationResult;

  if (inputType === 'XML') {
    transformationResult = SaxonJS.XPath.evaluate(
      "
transform(map {'stylesheet-text': $stylesheet-text , 'source-node' : parse-xml($source-text), 'delivery-format' : 'serialized' })",
```

```

    [],
    {
      params: {
        'stylesheet-text': xslt,
        'source-text': input
      }
    }
  );
}
else if (inputType === 'JSON') {
  transformationResult = SaxonJS.XPath.evaluate(
    "
let $json-input := parse-json($json-input-string) return transform(map {'stylesheet-text': $stylesheet-text, 'global-context-item' : $json-input, 'initial-match-selection' : $json-input, 'delivery-format' : 'serialized' })",
    [],
    {
      params: {
        'stylesheet-text': xslt,
        'json-input-string': input
      }
    }
  );
}
else if (inputType === 'None') {
  transformationResult = SaxonJS.XPath.evaluate(
    "
transform(map {'stylesheet-text': $stylesheet-text, 'delivery-format' : 'serialized' })",
    [],
    {
      params: {
        'stylesheet-text': xslt
      }
    }
  );
}
console.log(transformationResult);
}

transform(json, xslt, 'JSON', null);

```

while a change of the inlined XSLT to

```

const xslt = `<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="3.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:map="http://www.w3.org/2005/xpath-functions/map"
  exclude-result-prefixes="#all"
  expand-text="yes">

  <xsl:output method="adaptive"/>

  <xsl:template match=".">
    <result>
      <xsl:comment>
Run with {system-property('xsl:product-name')} {system-property('xsl:product-version')}
</xsl:comment>
      <xsl:sequence select="."/>
    </result>
  </xsl:template>
</xsl:stylesheet>`;

```

works as expected, the passed in XDM array is the context item in the template.

The error I get is

```
message: 'Focus for . (dot) is absent',  
code: 'XPDY0002',  
xsltLineNr: '14',
```

which I assume indicates that call template invocation on xsl:initial-template is done. As I said, with the same fn:transform options and Saxon Java I don't get any interference by the name="xsl:initial-template" attribute, both XSLT versions use apply templates.

## History

---

### #1 - 2020-06-15 12:08 - Michael Kay

- Assignee set to John Lumley
- Priority changed from Low to Normal

### #2 - 2020-06-15 14:07 - John Lumley

- Status changed from New to Resolved
- Applies to JS Branch Trunk added

It turns out that in `internalTransform()`, `call-template` mode was being chosen when there was `name="xsl:initial-template"` present, checking only for the absence of (the internal forms of) a source-node, not that and `initial-match-selection`. Simple patch fixes this.

### #3 - 2020-06-15 17:46 - John Lumley

- Fix Committed on JS Branch Trunk added