

## Saxon - Bug #4074

### Try/Catch overhead because of potential multithreading

2018-12-19 18:29 - Michael Kay

<b>Status:</b>	Closed	<b>Start date:</b>	2018-12-19
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Michael Kay	<b>% Done:</b>	100%
<b>Category:</b>	Multithreading	<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>	9.9
<b>Applies to branch:</b>	9.9	<b>Fixed in Maintenance Release:</b>	9.9.1.1
<b>Description</b>			
<p>Under Saxon-EE, when try/catch is executed, the <code>xsl:try</code> code prepares for the eventuality that an <code>xsl:result-document</code> instruction will be executed asynchronously during the <code>xsl:try</code>, in which case the <code>xsl:try</code> cannot complete until the <code>xsl:result-document</code> instruction has successfully completed.</p> <p>This turns out to impose a significant overhead (about 50ms on the cost of the <code>xsl:try</code> instruction) which can rapidly accumulate if the <code>try/catch</code> encloses only a small amount of code that is frequently executed.</p> <p>It appears to be possible to avoid this overhead by having the Saxon ThreadManagerEE class create the underlying ExecutorService and CompletionService only when they are first needed, that is, when a result-document instruction is actually executed.</p>			

#### History

##### #1 - 2018-12-20 19:33 - Michael Kay

- Category set to Multithreading
- Status changed from In Progress to Resolved
- Priority changed from Low to Normal
- Fix Committed on Branch 9.9 added

Actually the 50ms figure isn't for one execution of `xsl:try`, it is for many executions, but the fact remains that it's a significant cost (in this use case, about 5% of the overall stylesheet execution cost).

##### #2 - 2018-12-21 17:54 - Michael Kay

- Status changed from Resolved to In Progress

Reopening.

This change caused a couple of tests to hang, for example result-document-1006.

It no longer hangs if we remove the "synchronized" keywords from `ThreadManagerEE.getExecutorService()` and `getCompletionService()`. However, if these methods aren't synchronized, then it's not clear that they are safe.

##### #3 - 2018-12-21 18:32 - Michael Kay

- Status changed from In Progress to Resolved

Fixed by synchronizing at the level of the `activate()` method instead -- though it's not entirely clear why this makes a difference.

##### #4 - 2019-01-21 19:34 - O'Neil Delpratt

- % Done changed from 0 to 100
- Fixed in Maintenance Release 9.9.1.1 added

Bug fix applied to the Saxon 9.9.1.1 maintenance release.

##### #5 - 2019-01-21 19:38 - O'Neil Delpratt

- Status changed from Resolved to Closed