

## Saxon/C - Bug #4302

### High memory usage with Saxon/C + PHP

2019-08-29 09:17 - Alf Eaton

<b>Status:</b>	Closed	<b>Start date:</b>	2019-08-27
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	O'Neil Delpratt	<b>% Done:</b>	100%
<b>Category:</b>	JET	<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>		<b>Spent time:</b>	0:00 hour
<b>Found in version:</b>	1.1.2	<b>Fixed in version:</b>	1.2.0

#### Description

I'm using a large XSL file (see attached, generated from Schematron rules) to process an XML file, and finding that the PHP extension is using 10x the amount of memory that Saxon uses to process the same file in Java.

When running the conversion with regular Saxon-HE on the command line, the memory usage of the Java process reaches 400MB and the processing takes ~20s.

When running the same conversion via Saxon-C as a PHP extension (using the patched catalog-aware files from <https://saxonica.plan.io/issues/4274>), the memory usage of the Apache instance reaches 4GB and the processing takes ~40s.

While there are obviously optimisations that can be made to the generated XSL to improve processing times, is it expected that the memory usage would be so much higher in the PHP extension?

#### History

##### #1 - 2019-08-29 09:40 - O'Neil Delpratt

- Assignee set to O'Neil Delpratt

Thanks for reporting your issue. To help in the investigation please can you supply us your PHP script you use or a snippet of it.

Thanks

##### #2 - 2019-08-29 09:54 - Alf Eaton

Here's a PHP snippet, it's quite simple:

```
$saxonProcessor = new Saxon\SaxonProcessor();

$catalog = getenv('XML_CATALOG_FILES');
$saxonProcessor->setCatalog($catalog, true);

$processor = $saxonProcessor->newXsltProcessor();
$processor->setSourceFromFile($_FILES['xml']['tmp_name']);
$processor->compileFromFile(__DIR__ . '/generated.xsl');
$result = $processor->transformToString();
```

I haven't yet tried to reproduce this by running the PHP script on the command line, only as a web service in Apache.

**#3 - 2019-08-29 12:09 - O'Neil Delpratt**

hi,

I am having trouble running the transformation. The input.xml file seems to have dependencies. Such as the reference to JATS-archivearticle1.dtd. I will download this file too.

**#4 - 2019-08-29 12:14 - Alf Eaton**

Apologies for not yet providing a complete repository for reproducing the issue - I can try to put one together in the next few days.

If you do want to try using the DTDs, you can download <https://github.com/JATS4R/jats-dtds/archive/master.zip> and point the catalog resolver to schema/catalog.xml in the unzipped archive.

**#5 - 2019-08-29 14:33 - O'Neil Delpratt**

Thanks. I have now got it to run.

Saxon on Java 9.9.1.4: 16 seconds

Saxon/C 1.1.2: PHP command-line takes 26 seconds.

Saxon/C 1.2.0 (pre-release): PHP on the command-line takes around 23 seconds.

Saxon/C 1.1 or 1.2 in the browser: Terminated maximum execution time 30 seconds for PHP exceeded.

Performance issues are sometimes difficult to get to the bottom of the cause. The memory should not blow up on Saxon/C as it does, maybe there are hotspots in the stylesheet which is causing the memory problem. I am investigating it further.

**#6 - 2019-08-29 19:24 - O'Neil Delpratt**

Update:

I managed to run the PHP script in the browser using a pre-release of Saxon/C on Excelsior JET 15.3 (MP1) enterprise which shows some improvements with the running time and memory. We are now turning our attentions to the profiling of the memory usage.

**#7 - 2019-09-03 12:56 - O'Neil Delpratt**

- *Tracker changed from Support to Bug*
- *Category set to JET*
- *Status changed from New to Resolved*
- *Priority changed from Low to Normal*
- *Found in version set to 1.1.2*

As mentioned in comment #6, there seems to be an issue with how Excelsior Jet 15.3 professional is handling the heap memory and the garbage collection.

The latest release of Excelsior Jet 15.3 (MP1) Enterprise, which will be used in Saxon/C 1.2 shows much better memory management of the heap and the GC

Details of experiment (command-line only): Java: Final memory used 841MB, Time: 33 seconds

Jet XJava (before optimizations): Memory goes over 2GB before final GC state of 255MB. Time= 1 minute, 21seconds

Jet 15.3 (MP1) Enterprise (with optimizations): Final memory 95MB, memory does go up to 600MB during transformation. Time: 49 seconds The memory used with the supplied stylesheet and document goes up to 2GB on my local machine.

Therefore the latest Jet resolves this issue, which will be available in the next major release of Saxon/C.

Marking this bug issue as resolved.

**#8 - 2019-10-16 00:49 - O'Neil Delpratt**

- Status changed from Resolved to Closed

- % Done changed from 0 to 100

- Fixed in version set to 1.2.0

Big fix applied in the Saxon/C 1.2.0 release.

**Files**

---

generated.xsl	1.27 MB	2019-08-29	Alf Eaton
input.xml	234 KB	2019-08-29	Alf Eaton