

Thread Dump - Intelligence Report

File: noLineGap-single.txt
Timestamp: 2016-09-07 14:41:03
JVM Version: 64-Bit Server VM (24.101-b14 mixed mode)

Thread Count Summary

To learn about different thread states through real-life example, check out this [video tutorial](https://www.youtube.com/watch?v=fzYLtYa_J_D0) (https://www.youtube.com/watch?v=fzYLtYa_J_D0)

Total Threads count: 489

272

WAITING

[View Details](#)

174

RUNNABLE

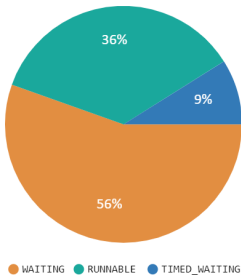
[View Details](#)

43

TIMED_WAITING

[View Details](#)

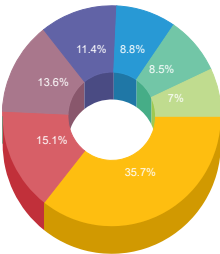
Thread state %



WAITING RUNNABLE TIMED_WAITING

Thread Group

Threads with similar names are grouped in this section



Legends	Thread Group	Count	States
	pool (same-state-threads.jsp?state...	97 threads	WAITING:88 RUNNABLE:9
	[ACTIVE] requestURI=/docRest/be...	41 threads	WAITING:26 RUNNABLE:15
	[ACTIVE] requestURI=/docRest/be...	37 threads	RUNNABLE:29 WAITING:8
	[ACTIVE] requestURI=/docRest/be...	31 threads	RUNNABLE:23 WAITING:5 TIMED_WAITING:3
	[ACTIVE] requestURI=/docRest/be...	24 threads	RUNNABLE:15 WAITING:9
	Gang worker (same-state-threads.j...	23 threads	RUNNABLE:23

Daemon vs non-Daemon

Learn more about [daemon and non-daemon \(i.e. user threads\)](https://blog.fastthread.io/2017/05/25/less-known-facts-about-daemon-and-non-daemon-threads/) (https://blog.fastthread.io/2017/05/25/less-known-facts-about-daemon-and-non-daemon-threads/)



343

DAEMON

[View Details](#)

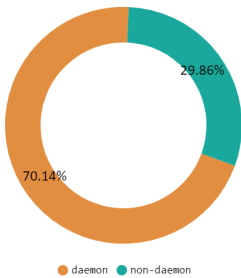


146

NON-DAEMON

[View Details](#)

Daemon vs non-Daemon

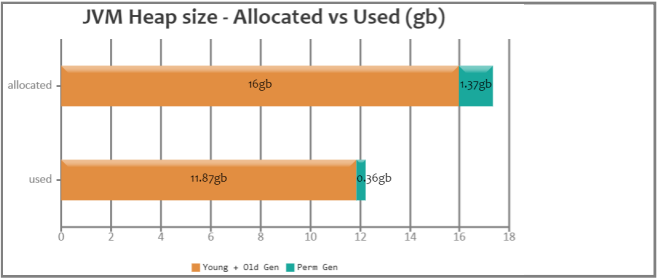


daemon non-daemon

Heap size

Heap Size of the JVM, as reported in thread dump

Generation	Allocated 📊	Used 📊
Young + Old Gen	16 gb	11.87 gb
Perm Generation	1.37 gb	368.84 mb
Total	17.37 gb	12.23 gb



Weblogic Thread Activity

Weblogic threads activity are presented in this section. Learn about [Weblogic threads activity](https://blog.fastthread.io/2018/05/13/weblogic-thread-stuck-active-standby/) (https://blog.fastthread.io/2018/05/13/weblogic-thread-stuck-active-standby/)

⚙️
219
ACTIVE

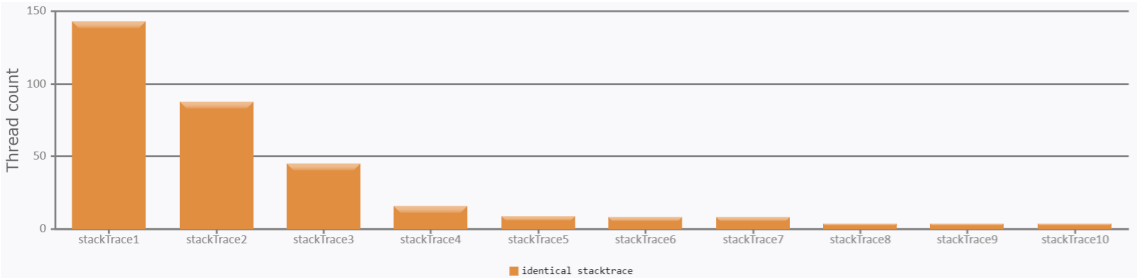
[View Details](#)

⏸️
27
STANDBY

[View Details](#)

Threads with identical stack trace

Threads with identical stack traces are grouped here. If lot of threads start to exhibit identical stack trace it might be a concern, learn [RSI Pattern](https://blog.fastthread.io/2016/02/22/thread-dump-analysis-pattern-repetitive-strain-injury-rsi/) (https://blog.fastthread.io/2016/02/22/thread-dump-analysis-pattern-repetitive-strain-injury-rsi/)



Thread Count	Identical Stack trace
143 WAITING threads	<pre>java.lang.Thread.State: WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x000000064b7003f0> (a weblogic.work.ExecuteThread) at java.lang.Object.wait(Object.java:503) at weblogic.work.ExecuteThread.waitForRequest(ExecuteThread.java:248) ... See complete stacktrace. (threads.jsp?traceId=155&pc&dumpld=1&type=rs)</pre>
88 WAITING threads	<pre>java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000003e4c1ca60> (a java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject) at java.util.concurrent.locks.LockSupport.park(LockSupport.java:186) at java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject.await(AbstractQueuedSynchronizer.java:2043) ... See complete stacktrace. (threads.jsp?traceId=9&pc&dumpld=1&type=rs)</pre>
45 RUNNABLE threads	<pre>stacktrace See complete stacktrace. (threads.jsp?traceId=1&pc&dumpld=1&type=rs)</pre>
16 WAITING threads	<pre>java.lang.Thread.State: WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x00000003d4ef47d8> (a atg.service.scheduler.ScheduledJob) at java.lang.Object.wait(Object.java:503) at atg.service.scheduler.ScheduledJob.waitForSignal(ScheduledJob.java:526) ... See complete stacktrace. (threads.jsp?traceId=74&pc&dumpld=1&type=rs)</pre>
9 TIMED_WAITING threads	<pre>java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x00000003d546b108> (a java.lang.Object) at atg.dms.sql.SqlMessageConsumer.receive(SqlMessageConsumer.java:267) - locked <0x00000003d546b108> (a java.lang.Object) ... See complete stacktrace. (threads.jsp?traceId=37&pc&dumpld=1&type=rs)</pre>

8 WAITING threads	java.lang.Thread.State: WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x00000003d546af50> (a atg.service.datacollection.DataCollectorQueue) at java.lang.Object.wait(Object.java:503) at atg.service.queue.EventQueue.getElement(EventQueue.java:237) ... See complete stacktrace . (threads.jsp?traceld=66&pc&dumpld=1&type=rs)
8 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x00000003d4ef4690> (a com.quest.pas.agent.util.HiResTimerThread) at com.quest.common.util.TimerThread.waitForStart(TimerThread.java:503) - locked <0x00000003d4ef4690> (a com.quest.pas.agent.util.HiResTimerThread) ... See complete stacktrace . (threads.jsp?traceld=124&pc&dumpld=1&type=rs)
4 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x0000000388057280> (a java.util.TaskQueue) at java.util.TimerThread.mainLoop(Timer.java:552) - locked <0x0000000388057280> (a java.util.TaskQueue) ... See complete stacktrace . (threads.jsp?traceld=4&pc&dumpld=1&type=rs)
4 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method) at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269) at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:79) at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:87) ... See complete stacktrace . (threads.jsp?traceld=122&pc&dumpld=1&type=rs)
4 RUNNABLE threads	java.lang.Thread.State: RUNNABLE See complete stacktrace . (threads.jsp?traceld=137&pc&dumpld=1&type=rs)

Most used methods

Frequently executed methods are reported. If lot of threads executes same method, it may be a concern. Learn [All roads lead to Rome pattern](#) (<https://blog.fastthread.io/2016/02/01/thread-dump-analysis-pattern-all-roads-lead-to-rome/>)

Thread Count	Method	Percentage
208 threads	java.lang.Object.wait(Native Method). To see stack trace click here . (threads.jsp?traceld=2&dumpld=1&type=tm)	43%
101 threads	sun.misc.Unsafe.park(Native Method). To see stack trace click here . (threads.jsp?traceld=30&dumpld=1&type=tm)	21%
27 threads	atg.adapter.gsa.GSAPropertyDescriptor.getPropertyValue(GSAPropertyDescriptor.java). To see stack trace click here . (threads.jsp?traceld=10&dumpld=1&type=tm)	6%
11 threads	sun.nio.ch.EPollArrayWrapper.epollWait(Native Method). To see stack trace click here . (threads.jsp?traceld=47&dumpld=1&type=tm)	2%
10 threads	java.net.SocketInputStream.socketRead0(Native Method). To see stack trace click here . (threads.jsp?traceld=7&dumpld=1&type=tm)	2%

CPU consuming threads

If application is consuming high CPU, investigate below threads. Learn [Athlete pattern](#) (<https://blog.fastthread.io/2016/06/23/really-running/>)

Thread	CPU consuming thread's stacktrace
pool-141272-thread-4 nativeld: 7498	java.lang.Thread.State: RUNNABLE at com.ctc.wstx.sr.InputElementStack.getLocalName(InputElementStack.java:945) at com.ctc.wstx.sr.BasicStreamReader.getLocalName(BasicStreamReader.java:736) at com.sun.xml.ws.util.xml.XMLStreamReaderFilter.getLocalName(XMLStreamReaderFilter.java:168) at org.eclipse.persistence.internal.xml.record.XMLStreamReaderReader.parseEvent(XMLStreamReaderReader.java:139) ... See complete stacktrace . (thread.jsp?tid=MHgvZDRh&dumpld=1)
[ACTIVE] requestURI=/docRest/bean/com... nativeld: 8457	java.lang.Thread.State: RUNNABLE at atg.adapter.gsa.GSAPropertyDescriptor.getPropertyValue(GSAPropertyDescriptor.java) at atg.adapter.gsa.GSAItem.getPropertyValue(GSAItem.java:1483) at atg.repository.RepositoryItemImpl.getPropertyValue(RepositoryItemImpl.java:151) at com.gfs.ecom.common.TranslatedPropertyDescriptor.getPropertyValue(TranslatedPropertyDescriptor.java:176) ... See complete stacktrace . (thread.jsp?tid=MHgyMTA5&dumpld=1)
[ACTIVE] requestURI=/docRest/bean/com... nativeld: 8456	java.lang.Thread.State: RUNNABLE at atg.adapter.gsa.GSAPropertyDescriptor.getPropertyValue(GSAPropertyDescriptor.java) at atg.adapter.gsa.GSAItem.getPropertyValue(GSAItem.java:1483) at atg.repository.RepositoryItemImpl.getPropertyValue(RepositoryItemImpl.java:151) at com.gfs.ecom.distribution.catalog.util.ProductValidationUtil.isProductActive(ProductValidationUtil.java:83) ... See complete stacktrace . (thread.jsp?tid=MHgyMTA4&dumpld=1)

[ACTIVE] requestURI=/docRest/bean/com... nativeId: 8453	java.lang.Thread.State: RUNNABLE at atg.adapter.gsa.GSAPropertyDescriptor.getPropertyValue(GSAPropertyDescriptor.java) at atg.adapter.gsa.GSAItem.getPropertyValue(GSAItem.java:1483) at atg.repository.RepositoryItemImpl.getPropertyValue(RepositoryItemImpl.java:151) at com.gfs.ecom.distribution.common.ui.model.ProductInfo.populate(ProductInfo.java:164) ... See complete stacktrace . (thread.jsp?tid=MHgyMTA1&dumpId=1)
[ACTIVE] requestURI=/docRest/bean/com... nativeId: 8450	java.lang.Thread.State: RUNNABLE at com.endeca.navigation.OptInputBuffer.read(Unknown Source) at com.endeca.navigation.OptInputBuffer.readFully(Unknown Source) at com.endeca.navigation.OptInputBuffer.readUInt(Unknown Source) at com.endeca.navigation.OptiBinaryInterp.FormatCat(Unknown Source) ... See complete stacktrace . (thread.jsp?tid=MHgyMTAy&dumpId=1)

Blocking Threads - Transitive Graph

Threads that block other threads are displayed here. Blocking threads makes application unresponsive, learn [Traffic Jam pattern](#) (<https://blog.fastthread.io/2016/03/07/thread-dump-analysis-pattern-ripple-effect/>)

☒ No transitive blocks found

GC Threads

Garbage collection threads count reported. Learn [Scavengers pattern](#) (<https://blog.fastthread.io/2015/09/02/thread-dump-analysis-pattern-several-scavengers/>)

44

GC threads

View Details

GC Thread type	Count
Concurrent Mark GC Thread	1
Gang worker	23
GC Daemon	1
G1 Concurrent Refinement Thread	19
Total	44

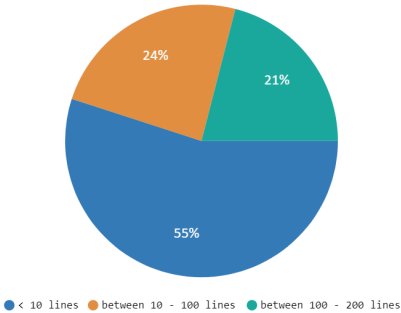
☒ GC thread count is normal

Threads Stack Length

Lengthy stacks can cause `StackOverflowError`. [Learn more](#) (<https://blog.fastthread.io/2018/09/24/stackoverflowerror/>)

☒ No Problem in Stack trace length.

Stack Length	Thread count
< 10 lines (same-state-threads.jsp?state=length&pc&length=10)	269
between 10 - 100 lines (same-state-threads.jsp?state=length&pc&length=100)	118
between 100 - 200 lines (same-state-threads.jsp?state=length&pc&length=200)	102



Complex DeadLocks

Learn more about [Complex Deadlock](#) (<https://blog.fastthread.io/2016/05/18/circular-deadlock/>)

☒ No Complex Deadlocks found

Dead Lock

Learn more about [Deadlock](#) (<https://blog.fastthread.io/2016/04/25/deadlock/>)

☒ No Deadlock found

Finalizer Thread

If finalizer thread is BLOCKED or WAITING for a prolonged period, it can result in OutOfMemoryError, to learn more visit [Leprechaun Trap pattern](https://blog.fastthread.io/2015/11/20/thread-dump-analysis-pattern-leprechaun-trap/) (https://blog.fastthread.io/2015/11/20/thread-dump-analysis-pattern-leprechaun-trap/)

☒ No problem with Finalizer Thread.

Exception

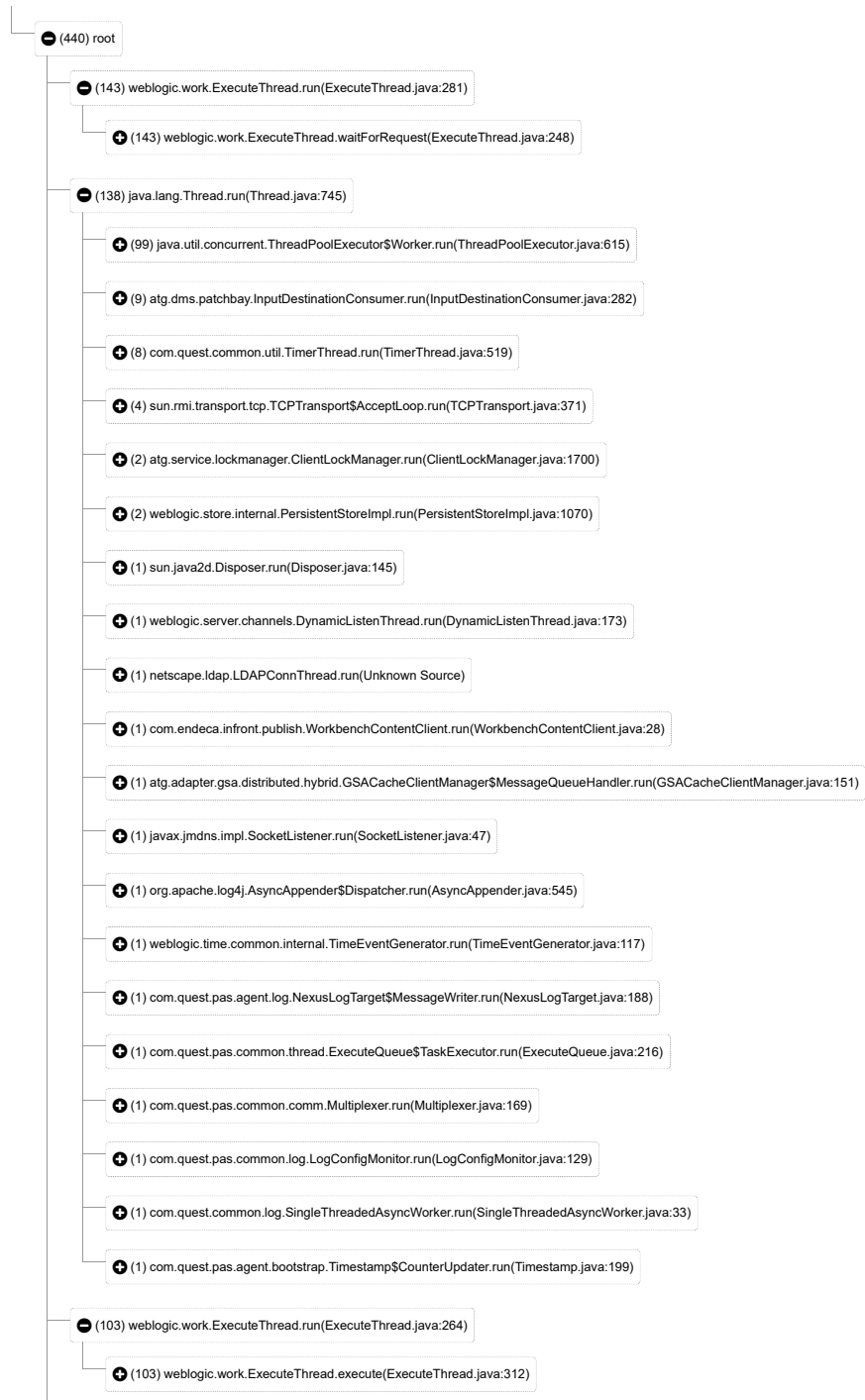
Threads throwing commonly known Exceptions/Errors are reported here. [Learn more](https://blog.fastthread.io/2020/06/10/threads-throwing-exception/) (https://blog.fastthread.io/2020/06/10/threads-throwing-exception/)

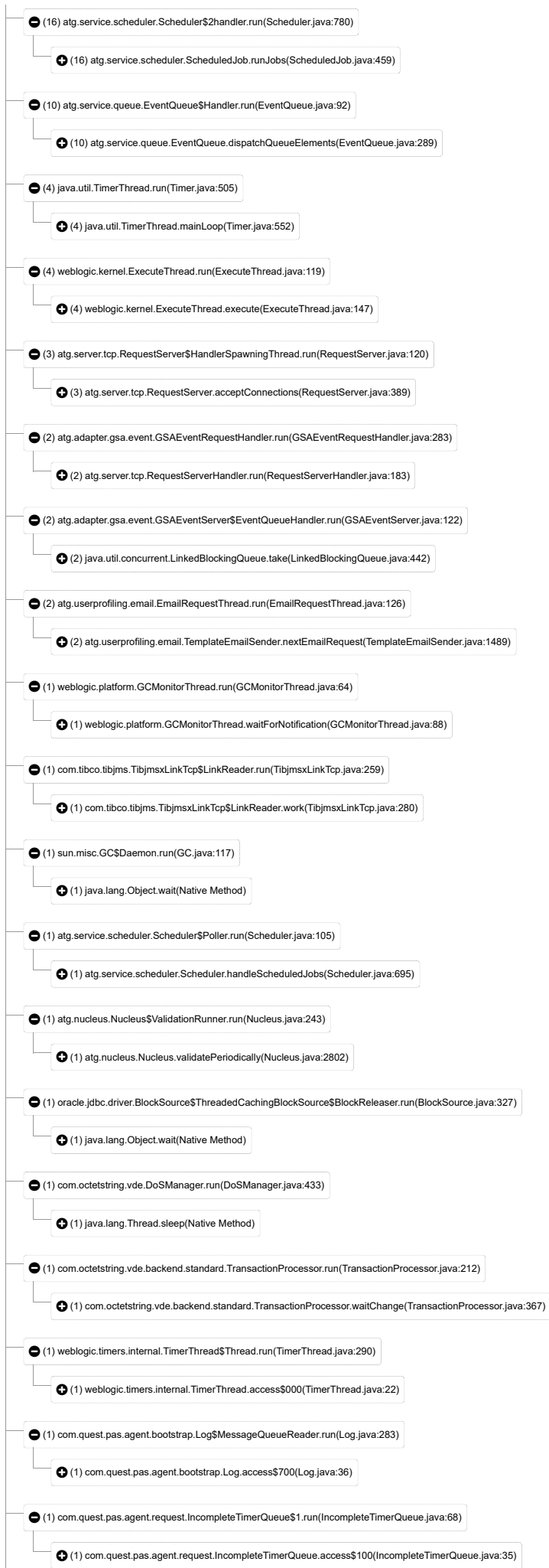
☒ No known exceptions are reported.

Bottom up Call Stack Tree

⬇ Show Top Down call stack

All threads stacktrace are combined in to one single tree. [Learn it's benefits](https://blog.fastthread.io/2018/04/27/benefits-of-call-stack-tree/) (https://blog.fastthread.io/2018/04/27/benefits-of-call-stack-tree/).





➡ (1) java.lang.ref.Finalizer\$FinalizerThread.run(Finalizer.java:209)

➡ (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:151)

➡ (1) java.lang.ref.Reference\$ReferenceHandler.run(Reference.java:133)

➡ (1) java.lang.Object.wait(Object.java:503)
