

# Blackboard

## Accessing JVisualVM in Order to Take CPU Sample Screenshots

**Date Published:** Feb 17,2017 **Category:** Product:Application\_and\_Webserver\_Learn;  
Version:Learn\_9\_1\_Q4\_2016\_3100\_0\_0\_rel\_107\_401e,Learn\_April\_2014\_9\_1\_201404\_160205,Learn\_October\_2014\_9\_1\_201410\_160373,Learn\_9\_1\_Q4\_2015\_9\_1\_201510\_1  
**Article No.:** 000064203

**Product:** Blackboard Learn

**Release:** 9.1

**Introduction:** When troubleshooting Blackboard Learn issues where performance is in question, you may find jVisualVM gives you a much clearer picture of where your application time is going than normal thread dumps or monitoring software such as Nagios or NewRelic. Here is a quick guide for using jVisualVM when you have a Windows workstation, and you're connecting to a Unix-based app server.

**Note:**

If you have Windows application servers, you can simply run jvisualvm from Windows Explorer, and skip the PuTTY/VcXSrv. You can also skip the PuTTY/VcXSrv installation if you have another SSH client that supports X11 and another X Windows server for Windows (there are many free versions available).

**Functionality:**

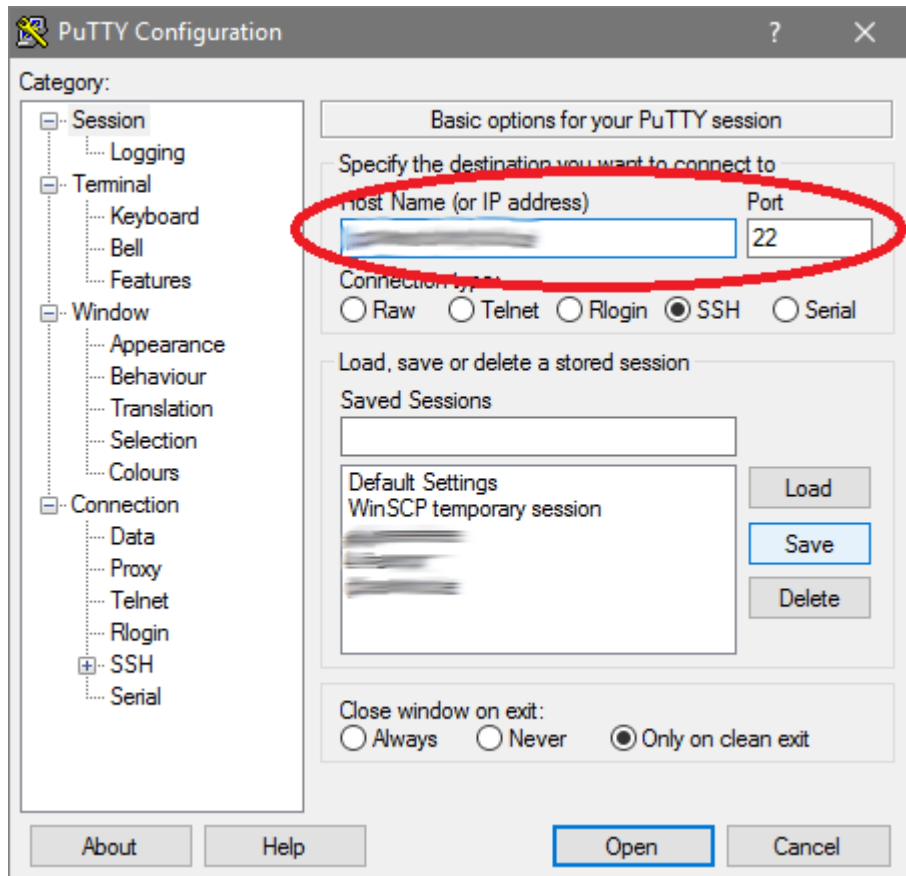
### jVisualVM - The easy way

You need two software packages, which may or may not already be installed:

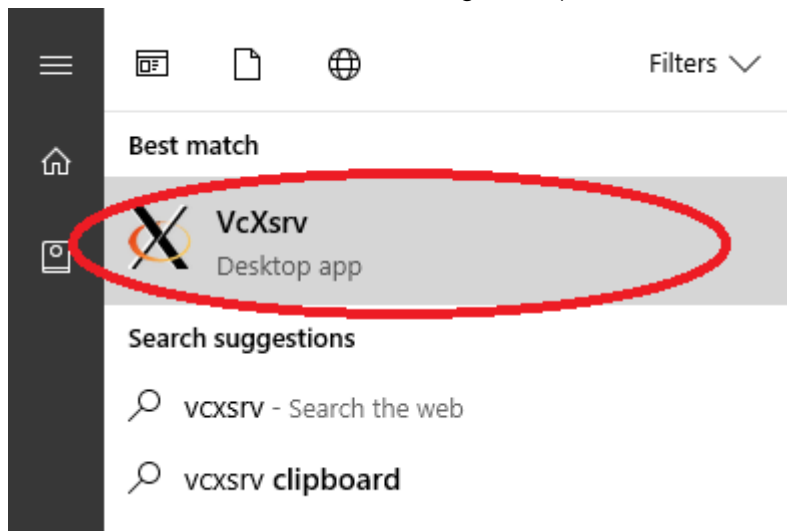
1. SSH client like PuTTY (or another PuTTY-based client, like SuperPuTTY)
2. X Server package for Windows.

If you don't already have both, and/or are not familiar with these packages, Blackboard Learn Client Support recommends using the links below to download and install PuTTY and VcXSrv, as they require very little configuration, and are both very small:

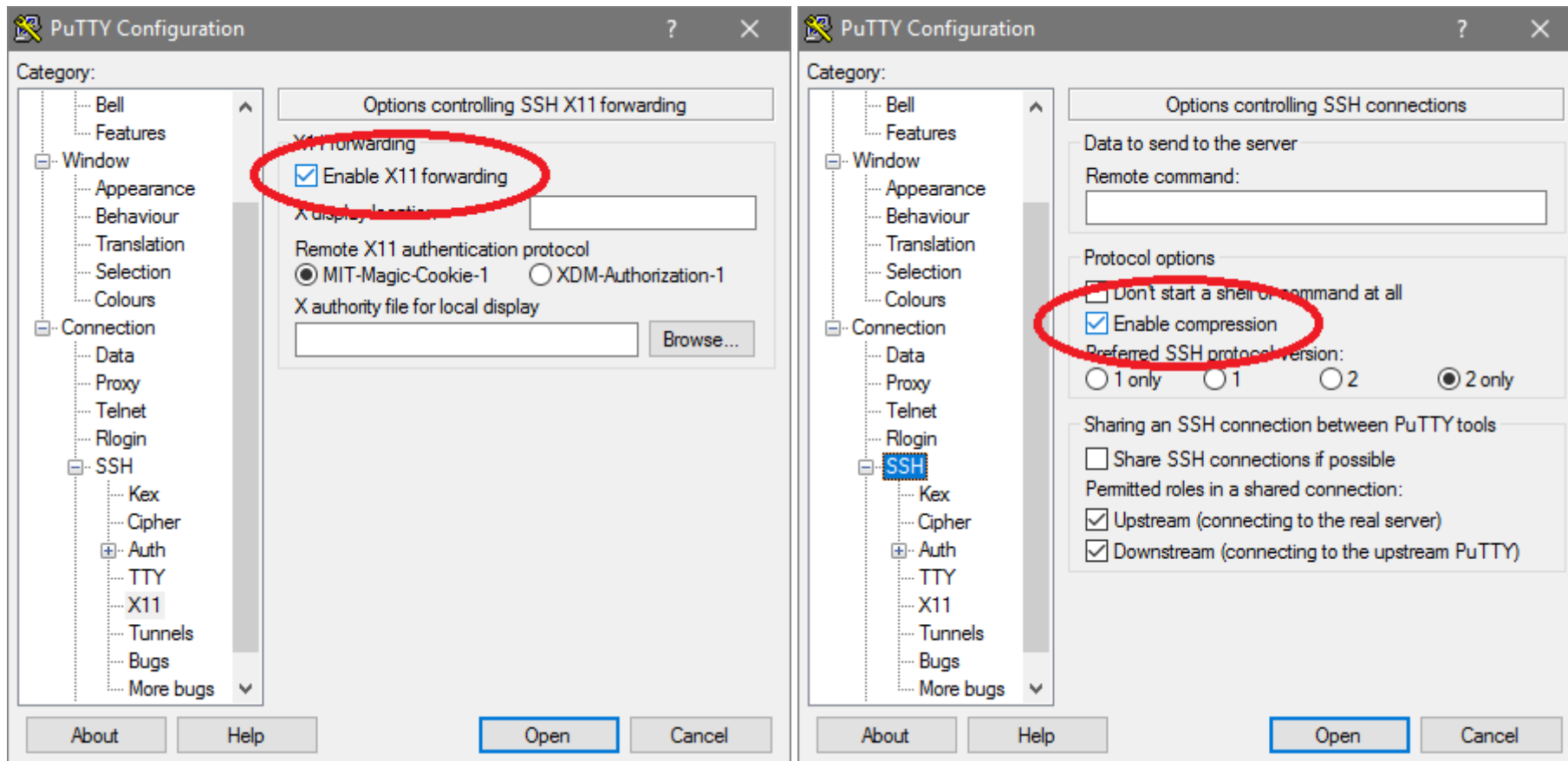
1. Install PuTTY client for your Windows workstation (<http://www.putty.org/>)
2. Install Visual C++ Redistributable 2015 (08 or newer works, if you already have it): (<https://www.microsoft.com/en-us/download/details.aspx?id=48145>)
  - If you have issues installing Visual C++ Redistributable on Win7, see the 'Common Issues' section below.
3. Install VCXSrv (Visual C X Server) for Windows (<https://sourceforge.net/projects/vcxsrv/>)
4. Restart your machine after installing both (VcXsrv needs to set system variables)
5. Open PuTTY and enter the app server host name you want to profile



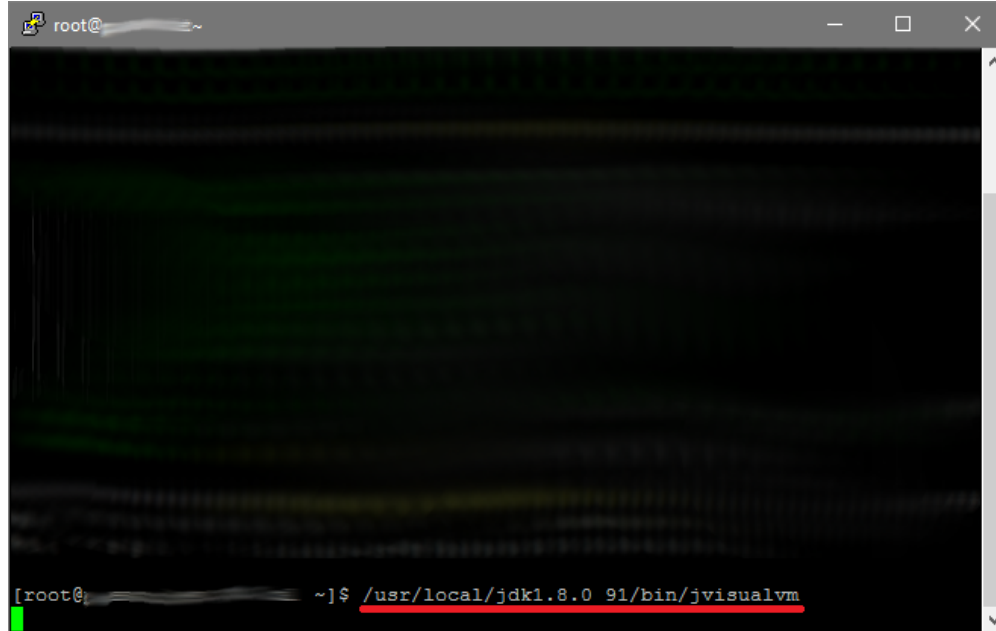
6. Start the VcXServ server in the background (**Start > Search > VcCSrv > Enter**)



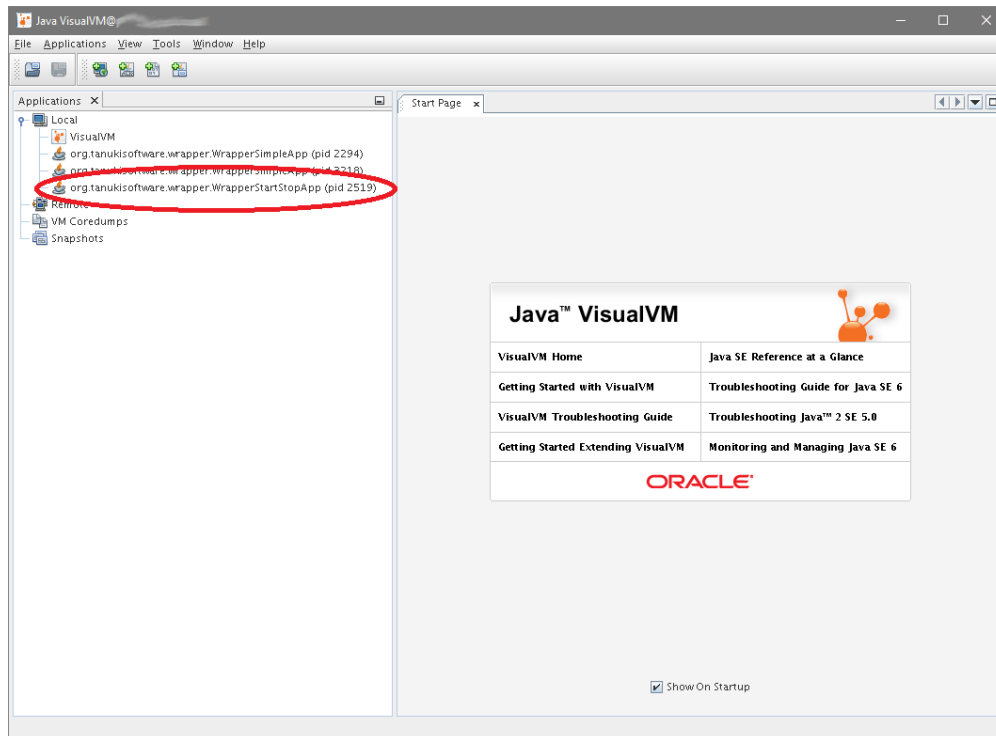
7. In PuTTY, click **Connection > SSH > X11** and check **Enable X11 Forwarding**. Under **Connection > SSH** click **Enable Compression**



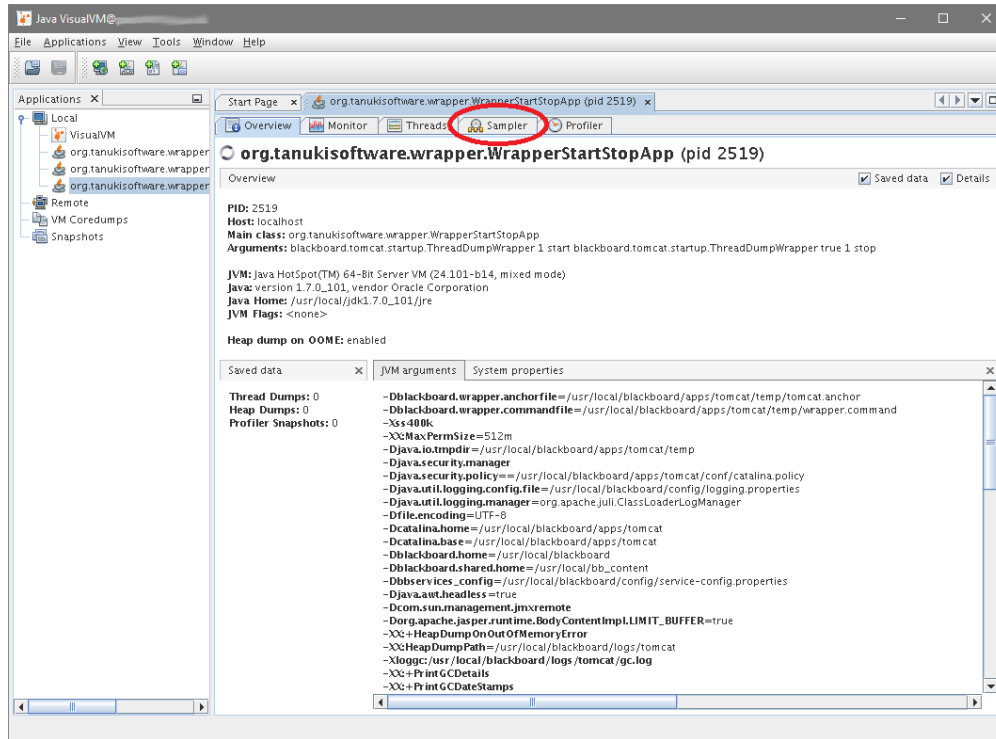
8. Open your SSH session, then run `/usr/local/jdk1.8_0_91/bin/jvisualvm` (replace with whichever JDK Learn is using)



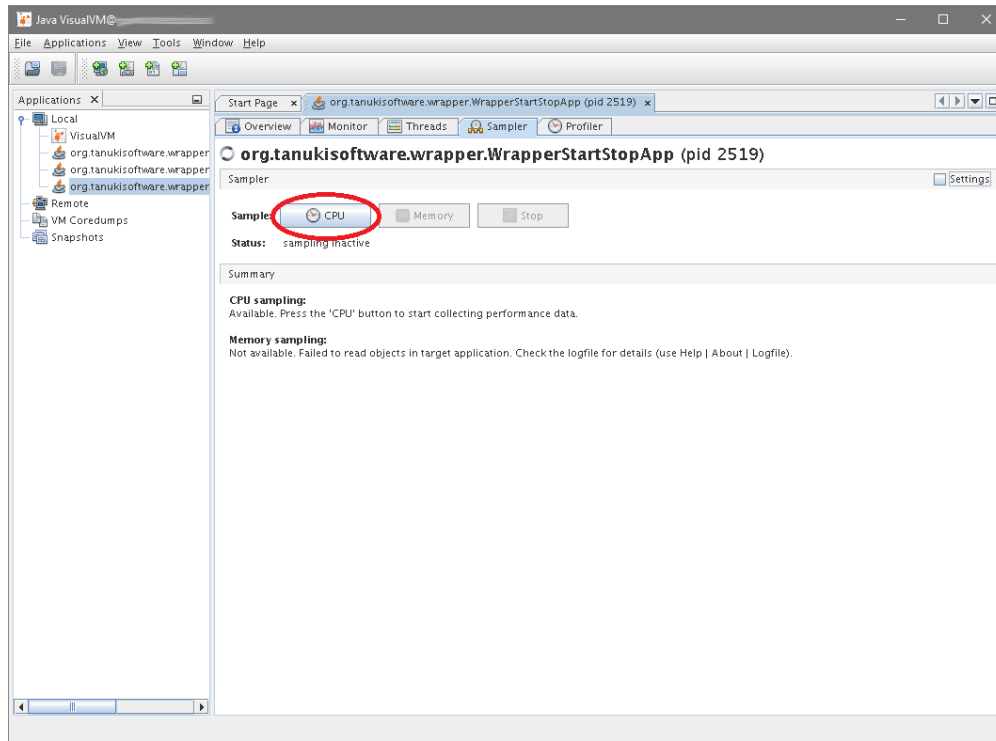
9. Since X11 is enabled, you should get a GUI window for jVisualVM. From here, find the main Tomcat JVM by looking for `org.tanukisoftware.wrapper.WrapperStartStopApp`:



10. Double click the WrapperStartStopApp, and you're in! You can now take CPU samples, memory samples, observe the live heap, take thread dumps or heap dumps, or anything else you might need to do. For example, we often use CPU Sample Snapshots to find out what classes are taking up the most time in a specific action. To get a snapshot, click the "Sampler" tab:



11. Then click "CPU" to start sampling (do this DURING an issue):



12. After you've been sampling the issue for enough time and can see some items may be problematic, take a snapshot by clicking the Snapshot button:

Java VisualVM

Applications: X

Local

- VisualVM
- org.tanukisoftware.wrapper
- org.tanukisoftware.wrapper
- org.tanukisoftware.wrapper
- Remote
- VM CoreDumps
- Snapshots

Start Page x org.tanukisoftware.wrapper.WrapperStartStopApp (pid 2519) x

Overview Monitor Threads Sampler Profiler

org.tanukisoftware.wrapper.WrapperStartStopApp (pid 2519)

Sampler

Settings

Sample: CPU Memory Stop

Status: CPU sampling in progress

CPU samples Thread CPU Time

Snapshot Thread Dump

| Thread  | Method | Self Time [%]     | Self Time | Self Time (CPU) | Total Time | Total Time (CPU) |
|---|--------|-------------------|-----------|-----------------|------------|------------------|
| blackboard.platform.queue.QueueTaskManager\$      |        | 95,267 ms (40.6%) | 0.000 ms  | 96,603 ms       | 1,336 ms   |                  |
| org.apache.tomcat.util.threads.TaskQueue\$take () |        | 72,002 ms (30.7%) | 0.000 ms  | 72,002 ms       | 0.000 ms   |                  |
| org.apache.activemq.store.kahadb.MessageDatabase  |        | 18,794 ms (8%)    | 0.000 ms  | 19,666 ms       | 0.000 ms   |                  |
| org.apache.activemq.store.kahadb.disk.journal.Da  |        | 18,794 ms (8%)    | 0.000 ms  | 19,666 ms       | 871 ms     |                  |
| org.bouncycastle.crypto.engines.DESEngine\$des R  |        | 16,845 ms (7.2%)  | 16,845 ms | 16,845 ms       | 16,845 ms  |                  |
| org.apache.tomcat.util.net.NioEndpoint\$KeyAttact |        | 5,045 ms (2.2%)   | 0.000 ms  | 5,045 ms        | 0.000 ms   |                  |
| org.bouncycastle.crypto.modes.CBCBlockCipher\$    |        | 3,897 ms (1.7%)   | 3,897 ms  | 20,742 ms       | 20,742 ms  |                  |
| oracle.net.ns.Packet\$receive ()                  |        | 1,336 ms (0.6%)   | 1,336 ms  | 1,336 ms        | 1,336 ms   |                  |
| org.apache.activemq.store.kahadb.MessageDatabase  |        | 871 ms (0.4%)     | 0.000 ms  | 871 ms          | 0.000 ms   |                  |
| org.apache.activemq.util.RecoverableRandomAcce    |        | 871 ms (0.4%)     | 871 ms    | 871 ms          | 871 ms     |                  |
| blackboard.util.CalendarUtil\$stringToCalendar () |        | 765 ms (0.3%)     | 765 ms    | 765 ms          | 765 ms     |                  |
| org.apache.tomcat.util.threads.TaskThread\$Wrap   |        | 0.000 ms (0%)     | 0.000 ms  | 97,790 ms       | 20,742 ms  |                  |
| net.sf.ehcache.distribution.jms.JMSCacheReplicato |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| net.sf.ehcache.distribution.jms.JMSCacheReplicato |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| net.sf.ehcache.distribution.jms.JMSCacheReplicato |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| blackboard.platform.log.impl.QueueLogger\$Que     |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| blackboard.platform.log.impl.QueueLogger\$rem     |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| blackboard.platform.log.impl.QueueLogger\$Que     |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| blackboard.platform.log.impl.FileLogSinkImpl\$run |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| blackboard.platform.messagequeue.impl.activemq    |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |
| org.apache.activemq.SimplePriorityMessageDispat   |        | 0.000 ms (0%)     | 0.000 ms  | 0.000 ms        | 0.000 ms   |                  |

Method Name Filter (Contains)

13. Save the snapshot by clicking the new snapshot-generated tab and selecting **Export to:**

Java VisualVM

File Applications View Tools Window Help

Applications x

Start Page x org.tanukisoftware.wrapper.WrapperStartStopApp (pid 2519) x

Overview Monitor Threads Sampler Profiler [snapshot] 11:29:19 AM x

org.tanukisoftware.wrapper.WrapperStartStopApp (pid 2519)

Profiler Snapshot

Export to... Call Tree - Method

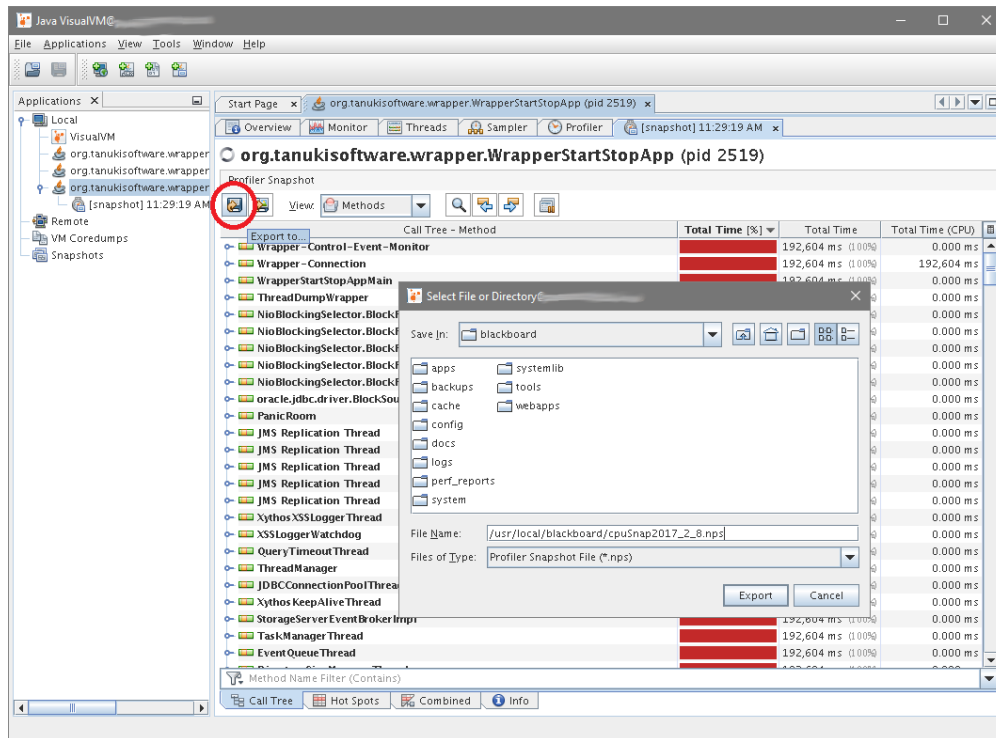
| Method  | Total Time [%]    | Total Time | Total Time (CPU) |
|---|-------------------|------------|------------------|
| Wrapper - Control - Event - Monitor                                     | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| Wrapper - Connection  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| WrapperStartStopAppMain   | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| ThreadDumpWrapper   | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| NioBlockingSelector.BlockPoller-1                                       | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| NioBlockingSelector.BlockPoller-2                                       | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| NioBlockingSelector.BlockPoller-3                                       | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| NioBlockingSelector.BlockPoller-4                                       | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| NioBlockingSelector.BlockPoller-5                                       | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| oracle.jdbc.driver.BlockSource.ThreadedCachingBlockSource.BlockReleaser | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| PanicRoom   | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| JMS Replication Thread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| JMS Replication Thread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| JMS Replication Thread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| JMS Replication Thread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| JMS Replication Thread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| Xythes.XSSLoggerThread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| XSSLoggerWatchdog   | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| QueryTimeoutThread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| ThreadManager   | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| JDBCConnectionPoolThread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| Xythes KeepAliveThread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| StorageServerEventBrokerImpl  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| TaskManagerThread   | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |
| EventQueueThread  | 192,604 ms (100%) | 192,604 ms | 0.000 ms         |

Method Name Filter (Contains)

Call Tree Hot Spots Combined Info

14. Save it to a location from which you can retrieve it, and use WinSCP or some other SFTP client of your choice to retrieve it:





- Repeat as needed for further snapshots, thread dumps, etc. and submit the requested CPU Snapshots, App Snapshots, Heap dumps, or Thread dumps to support for investigation.

## Common Issues:

If you have issues installing Visual C++ Redistributable on Windows 7, you may find the following resources useful:

- Visual C++ Stuck during install at Windows7\_MSU\_x64:  
<http://stackoverflow.com/questions/41233672/installation-microsoft-visual-c-redistributable-stuck-at-processing-window>
  - Install Update for Windows 7 for x64-based Systems (KB2999226)
  - Download the update KB2999226 for your OS edition from here: [Update for Universal C Runtime in Windows](#)
  - Manually install the KB2999226 as below:
    - Find the path to the file Windows6.1-KB2999226-x64.msu from the folder C:\ProgramData\Package Cache\xxxxx\packages\Patch\x64, where xxxxx is the update ID, similar to "42D5BEC7DDFBD49E76467529CBC2868987BF8460"
    - Create a folder named 'XXXX' in that and execute following commands from **Admin** command prompt (start > cmd.exe > right click > run as admin)
      - wusa.exe Windows6.1-KB2999226-x64.msu /extract:XXXX
      - DISM.exe /Online /Add-Package /PackagePath:XXXX\ Windows6.1-KB2999226-x64.cab
      - Run the Visual C++ Redistributable with the repair option: vc\_redist.x64.exe /repair

