

**Gustavo Chavez**  
Principal Consultant | Pre-Sales

DevTest Logging / Debugging

Quick guide v1.6

# Table of Contents

[Table of Contents 1](#_Toc480277023)

[1 DevTest Logging 2](#_Toc480277024)

[2 What file controls logging 2](#_Toc480277025)

[3 Where are the logs located 2](#_Toc480277026)

[4 What logs are useful 3](#_Toc480277027)

[5 What about issues with crashes 3](#_Toc480277028)

[6 VSE\_Matches file. 3](#_Toc480277029)

[7 VSE.log file. 4](#_Toc480277030)

[8 Set VSE recorder logging 4](#_Toc480277031)

[9 Run TCPMon 4](#_Toc480277032)

[10 Metrics. 5](#_Toc480277033)

[11 Running the ITR against the VMS. 5](#_Toc480277034)

[12 Other Tuning Recommendations. 6](#_Toc480277035)

[13 Other variable handling recommendations 6](#_Toc480277036)

[14 Reference 7](#_Toc480277037)

# DevTest Logging

As a rule of thumb the logging and tracing settings should be left as set by “default” for most of the time in the commonly used installations.

In case additional logging is needed or a suspected problem is detected after looking at main component logs contact your account pre-sales engineer for discussion or open a support ticket so a recommendation can be made on what is the best approach to get more granular information.

# What file controls logging

Logging is controlled by a main file located at your *LISA\_HOME* directory called  
   
logging.properties

And the descriptive detail can be found [here](https://docops.ca.com/devtest-solutions/8-0/en/reference/property-descriptions/custom-property-files/logging-properties)

This file is your main source for manipulating what is being logged across the multiple components of DevTest

# Where are the logs located

All logs are located under the “lisatmp” directory usually named with the version release number as part of the name *lisatmp\_x.x.x*

\ *LISA\_HOME* \lisatmp\_8.o.0

A list of the log files allocated can be found [here](https://docops.ca.com/devtest-solutions/8-0/en/administering/logging/log-file-overview)

NOTE: How can the default logging location be changed? Use the lisatmp.dir property defined as a system level variable

i.e.:

-Dlisa.tmpdir=c:/itko/lisa/server/lisa-logs/

# What logs are useful

It all depends on the component and task at hand but as a rule of thumb just by looking at the main log for the component in question or suspected to have an issue should be enough to get an idea on what is the problem.

The main component logs are:  
  
- Registry , Coordinator , Simulator and VSE

# What about issues with crashes

In cases when it’s not clear what the problem is a thread dump may be needed in order for the support team to analyze and diagnose, if that is the case then you will need to change the “threadDumpLogger” setting from WARN to INFO as described in the [documentation](https://docops.ca.com/devtest-solutions/8-0/en/administering/logging/automatic-thread-dumps) and echoed here.  
  
“You can use the **logging.properties** file to enable automatic thread dumps, which are helpful in debugging performance issues.

Locate the following property and change **WARN**to **INFO**.

log4j.logger.threadDumpLogger=WARN, THREAD\_DUMPS

To disable the thread dumps, change **INFO**back to **WARN**.

The default interval of the thread dumps is 30 seconds. You can change the interval by editing the **lisa.threadDump.interval** property in the **lisa.properties** file “

NOTE:   
Additionally you can change the setting from WARN to 1 for more granular logging without generating a dump

# VSE\_Matches file.

DevTest will generate a log file with the services matches into a file named “VSE\_matches” this log file can be used to debug errors at the request-response level, it is located at the “lisatmp” directory.

It will contain the matching characteristics for the request being processed.

# VSE.log file.

This file will contain the errors found when matching request-response.

# Set VSE recorder logging

In order to generate a log for the VSE recorder add the following to the logging. properties file, these properties enable DEBUG logging for all the events that happen during the VSE recorder:

# Keep a separate log for VSE recorder events, this makes debugging much easier.

log4j.logger.VSE\_RECORDER=DEBUG, VSE\_RECORDER

log4j.appender.VSE\_RECORDER=com.itko.util.log4j.TimedRollingFileAppender

log4j.appender.VSE\_RECORDER.File=${lisa.tmpdir}/vse\_recorder.log

log4j.appender.VSE\_RECORDER.MaxFileSize=10MB

log4j.appender.VSE\_RECORDER.MaxBackupIndex=20

log4j.appender.VSE\_RECORDER.layout=org.apache.log4j.EnhancedPatternLayout

log4j.appender.VSE\_RECORDER.layout.ConversionPattern=%d{ISO8601}{UTC}Z (%d{HH:mm})[%t] %-5p - %m%n

This will create a separate logging file in the tmp dir  ->  vse\_recorder.log

# DB Debugging

Sometimes you need to get some detail on database interaction, add these properties to logging.properties file.

Don’t forget to **remove these properties** after you are done debugging.

log4j.logger.com.mchange.v2.c3p0=DEBUG

log4j.logger.com.mchange.v2.resourcepool=DEBUG

# Run TCPMon

TCPMon is a utility that lets you monitor the messages that are passed in a TCP-based conversation.

TCPMon consists of the following elements:

For Windows: A .jar file, a .bat file For UNIX: A shell script

**To run TCPMon:**

Double-click the .bat file on Windows or execute the shell script on UNIX.

You can find a **tcpmon.bat** file in the **LISA\_HOME\bin** directory. You can get the latest version of TCPMon from: <http://ws.apache.org/commons/tcpmon/>.

# Metrics.

There are multiple metrics that you can observe within the product one of the most useful is the [JMX Metix](https://docops.ca.com/devtest-solutions/8-5/en/reference/metrics-descriptions/jmx-metrics)

The JMX metrics use the Java Management Extension (JMX) API to provide metrics.

These JMX connectors are provided for an easy setup:

* Any JSR 160 RMI connection
* JBoss 3.2-4.0
* JSE 5 Connector
* Oracle AS (OCJ4)
* Tomcat 5.0.28
* WebLogic 6.1-8.1
* WebLogic 9.x
* WebSphere 5.x
* ITKO JMX Agents

Each of these applications requires slightly different connection parameters. The values that you require for your server are available from your server administrator. Each provider provides slightly different metrics. To use other JMX features, you can invoke them as RMI steps

# Running the ITR against the VSE.

Sometimes you need to hit a virtual service and modulate the execution of the request and observe the response, in order to do this it is recommended to use the ITR to drive the test, there is a sample of this in the as illustrated in step 10 of the [DevTest tutorial](https://docops.ca.com/devtest-solutions/8-5/en/getting-started/ca-service-virtualization-tutorial).

# Other Tuning Recommendations.

Memory allocation for the DevTest components is modulated by a “.vmoptions” file associated with each component and located under the *LISA\_HOME/bin* directory.

For example:  
If you need to increase the memory permitted to the DevTest Workstation, you will find a file locate under *LISA\_HOME/bin*, called Workstation.vmoptions into this file, type a new line of:

-Xmx1024m

Save and quit, then restart your DevTest Workstation.

This increases the amount of memory available to the JVM in which DevTest Workstation is running to 1 GB.

 Here is a direct link to the DevTest documentation page that explains this [Change Memory Allocation](https://docops.ca.com/devtest-solutions/8-0-1/en/administering/general-administration/memory-settings)

NOTES

* 32-bit Windows has a maximum addressable memory of 2 GB per process, and Java takes a large part of this, so your maximum -Xmx size will be somewhere between 1.1 GB and 1.5 GB.
* Make sure the coordinator and simulator memory allocation matches each other in order to prevent overflowing.

# Other variable handling recommendations

This is not used very often but you can manipulate VM properties by defining a “LISA\_MORE\_VM\_PROPS” system variable.  
  
i.e. When using docker, use the **LISA\_MORE\_VM\_PROPS** property on the command line to point to a running Enterprise Dashboard:

  > docker run -P -e LISA\_MORE\_VM\_PROPS="-Dlisa.enterprisedashboard.service.url=tcp://dradishost

# Reference

[Property Descriptions](https://docops.ca.com/devtest-solutions/8-0/en/reference/property-descriptions/custom-property-files/logging-properties)

[DevTest Logging](https://docops.ca.com/devtest-solutions/8-0/en/administering/logging)

[Memory Setting](https://docops.ca.com/devtest-solutions/8-0-1/en/administering/general-administration/memory-settings)

[Metrics Descriptors](https://docops.ca.com/devtest-solutions/8-5/en/reference/metrics-descriptions/jmx-metrics)

[Event Descriptors](https://docops.ca.com/devtest-solutions/8-5/en/reference/event-descriptions)

[Service Virtualization Tutorial](https://docops.ca.com/devtest-solutions/8-5/en/getting-started/ca-service-virtualization-tutorial)

And as always, use our [community forum](https://communities.ca.com/community/ca-devtest-community/blog) ! ☺