

LSim LOGGING FUNCTIONALITY MANUAL

Authors: Joan Manuel Marquès, Sergi Gea

Distributed Systems course

Spring 2017

1. Introduction

The logging functionality of LSim allows you to write log messages for each instance while the code is running. That is, in real time. You can use it to trace or debug your code and it is useful to get information about how your code works.

You can use the logging functionality both when running the practical assignment locally and when running it in DSLab:

- (a) locally: logs will be stored in a file on your own file system.
- (b) DSLab: logs will be stored in DSLab.

2. How to generate the logs.

To use the LSim logging you should write the following sentence in your code:

```
LSimFactory.getWorkerInstance().log(<logging_level>, <a String containing the message to log>);
```

Next example logs a message of level INFO (go to section 4 for more details about levels):

```
LSimFactory.getWorkerInstance().log(Level.INFO, "Put your message here");
```

That's all.

To facilitate the readability of the code we propose the use of a variable instead of `LSimFactory.getWorkerInstance()`, like in the next example:

```
// Declare lsim
private LSimWorker lsim = LSimFactory.getWorkerInstance();
```

Then, in any line of the class, a log can be easily generated calling `lsim.log`, as follows:

```
// Use lsim to log messages instead of LSimFactory.getWorkerInstance()
lsim.log(Level.TRACE, "Put your message here");
```

Another example: generates a log output inside a catch block to report an error:

```
catch (ArrayIndexOutOfBoundsException e) {
    lsim.log(Level.ERROR, "Array is out of Bounds.");
}
```

3. Message and file name format

All the log messages have the same format:

<Date> <Time with milliseconds precision> <Type><instance Id> <Level> : "Your message"

Local exectutions

A log line example:

01-04-2017 01:20:57:343 groupXX@127.0.1.1:35000 [TRACE] : [TSAESessionOriginatorSide] [session: 1] TSAE session

Fields:

- Date: 01-04-2017
- Time (with milliseconds precision): 01:20:57:343
- Type: groupXX@ 127.0.1.1
- Instance:35000
- Level : [TRACE]
- Message: [TSAESessionOriginatorSide] [session: 1] TSAE session

The file that keeps the logs for the instance with Id "*groupXX@ 127.0.1.1:35000*" (file with extension .log) has the following name:

2017-04-01_012057343_groupXX@127.0.1.1:35000.log

Remote executions (DSLlab)

The log messages in the DSLab have the same pattern:

```
02-04-2017    TRACE    Student    0    [TSAESessionPartnerSide] [session: 1] TSAE session
02:36:50:938
```

Logs are associated to each execution (in the '**Results**' tab, see section 6.2)

4. Levels

There are 6 levels. Each Level controls which logs will be written. When a level is selected (see section 6 *Configuration parameters* for more information on how to select a level) the logging system only writes those messages with equal or higher level than the selected level.

Levels in descending order from higher to lower severity:

Level	Description
FATAL	Recommended for high critical events that will lead to abort the application instance.
ERROR	An error event that can affect the program but might still allow to continue running the application instance.
WARN	An event that might have an adverse effect.
INFO	Informational messages about the progress of the application instance at coarse-

	grained level.
DEBUG	Informational messages useful to debug your code (more fine-grained than INFO level.). This level must be deactivated when you finish the debugging process.
TRACE	Informational messages more detailed than the DEBUG level (finest-grained)

In addition you can use:

Level	Description
OFF	To turn off the log. This log will not be written.

For example, if you have a log like this:

```
lsim.log(Level.DEBUG, "Your debugging message");
```

and you do not want neither use it nor to delete it, you can change the level to **OFF** and the log will not be written.

```
//This log will not be written!!
lsim.log(Level.OFF, "Your debugging message");
```

This allows you to reduce the number of logs that you must review later and also reduce the size of the log file generated.

5. Local executions


5.1. How to use the logs

We added some log lines in `TSAESessionOriginatorSide` and `TSAESessionPartnerSide` classes in order to facilitate the understanding of how the practical assignment actually runs.

For example, In the class `TSAESessionOriginatorSide` you can find a line of code as example:

```
lsim.log(Level.TRACE, "[TSAESessionOriginatorSide][session:"
+current_session_number+"] TSAE session");
```

When running your code some log files will be created in the default folder (`../lsimLogs`). A log file will be created for each instance executed. For example, 3 instances running will create 3 different log files:

	2017-01-04_012057314_groupXX@127.0.1.1:35000.log	9,5 MB	Text
	2017-01-04_012057371_groupXX@127.0.1.1:35001.log	9,5 MB	Text
	2017-01-04_012057414_groupXX@127.0.1.1:35002.log	5,9 MB	Text

If you run your practical assignment with **TRACE** logging level, when opening the first file, you will see that the following message has been written each time a TSAE session has occurred:

```
01-04-2017 01:20:57:343 groupXX@127.0.1.1:35000 [TRACE] : [TSAESessionOriginatorSide]
[session: 1] TSAE session
```

5.2. Folder where logs will be stored

The log files created will be kept in a local folder (for more information see section 5.3, *Configuration parameters*).

5.3. Configuration parameters

Use logging configuration parameters in order to change the behavior of the logging system. These parameters also allows you to control the maximum file size or the maximum message length that will be written in the local log file (DSLab also have a maximum message length and a maximum log size)

These parameters are defined in section 4 of the config.properties file:

```
# 4. LSim logging system parameters.
# LSimLogsFolder: path to the folder where the logs will be stored.
logsFolder=../lsimLogs
#Maximum number of characters for the log message field (250 characters by
default).
msgMaxLen=10000
#Maximum log file size in KBytes (10 KBytes by default).
logMaxSize=12228
#level: define which messages should be written to the log (INFO by
default). The levels allowed are: FATAL, ERROR, WARN, INFO, DEBUG, TRACE.
#In addition there are two special levels: OFF to turn off the logging and
ALL to log everything.
level=TRACE
```

Some aspects should be considered:

- **logsFolder:** result files are created in ../lsimLogs folder. If a wrong path or nonexistent folder is specified the log files will be stored in the current working directory (../scripts).
- **msgMaxLen:** this parameter contains the maximum message length in number of characters. If there is no value or a value under 250 this parameter will be set to the minimum length, 250 characters.

If the message exceeds the defined length it will be truncated to the first msgMaxLen value characters. For example, if msgMaxLen is established to 250 characters, this log:

```
lsim.log(Level.INFO, "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
ullamcorper orci non condimentum hendrerit. Donec pretium neque ligula, a semper
ligula bibendum posuere. Etiam dapibus iaculis elit. Nam fringilla mollis quam, et
egestas sem orci aliquam. THIS LINE WILL NOT BE PRINTED.");
```

Only prints the first 250 characters of the message added to a warning message:

```
12-03-2017 11:57:51:013 groupXX@127.0.1.1:35000 [WARN] : WARNING!!! This log
entry has exceeded the maximum allowed length. Log entry has been truncated to first
250 characters. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
ullamcorper orci non condimentum hendrerit. Donec pretium neque ligula, a semper
ligula bibendum posuere. Etiam dapibus iaculis elit. Nam fringilla mollis quam, et
egestas sem orci aliquam.
```

- **logMaxSize:** this parameter contains the maximum file size in KBytes. If there is no value or a value under 10 KBytes this parameter will be set to the minimum size, 10 KBytes.

If the size of the file exceeds the specified value prints a warning message at the end of the file and the next logs will be not printed. Example for logMaxSize equal to 10 Kbytes:

```
12-03-2017 12:02:50:824 groupXX@127.0.1.1:35000 [WARN] : FILE SIZE  
EXCEEDED!!! The log file has exceeded the 10.0 KBytes.
```

- **level:** the logging system only writes those messages with equal or higher level than the level set in this parameter. Only the levels of section 4 *Levels*, are allowed. If the level parameter have a wrong value, it will take as default the INFO level. To turn off the logging system use OFF in this parameter (the logs will not be written).

Note: If you use **TRACE** as level parameter instead of **ALL**, you will obtain the same result (log everything) because both are at the same hierarchy level.

Feel free to change these parameters at your convenience.

6. Remote executions (DSLAb)

6.1. Good practices

Once you have debugged your code locally and you have tested that it works properly is time to launch it in DSLAb. It is advisable to turn off the logs that you don't need anymore in order to reduce the amount of logs generated. A great amount of log messages can be difficult to read and work with it can be exhausting.

In order to turn off the logs you added in the classes that you upload to DSLAb use the level **OFF**, as we have explained in the section 4, *Levels*. You do not need to do anything in the logs in the rest of classes. They will implement the default logging behavior.

6.2. How to show the logs

In order to show the logs go to the experiment and click on the results tab and then click the Log Filter icon.

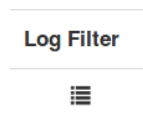


Image 1

It will show you a filter where you can select the logs that you want see.

6.3. Message and log storage maximum size

Log messages can have a maximum of 10000 characters. If a message exceed this length, it will be truncated. When it occurs a warning message is added at the beginning of the truncated

message:

02-04-2017 06:36:59:440	TRACE	Student	0	WARNING!!! This log entry has exceeded the maximum allowed length. Log entry has been truncated to first 10000 characters. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis
----------------------------	-------	---------	---	---

The logs storage capacity for each instance executed is also limited. Try to make a good use of the storage space. If the maximum capacity is reached a warning message will be showed at the end of the log and the next logs will not be printed.

02-04-2017 05:44:02:247	WARN	Student	1	FILE SIZE EXCEEDED!!! The log file has exceeded the maximum size.
----------------------------	------	---------	---	---