

[Home](#) / [Log4j](#) / log4j.properties example – Log4j properties file example

log4j.properties example – Log4j properties file example

Log4j is a simple and flexible logging framework. Logging equips the developer with detailed context for application failures. With log4j it is possible to enable logging at runtime without modifying the application binary. The log4j package is designed so that these statements can remain in shipped code without incurring a heavy performance cost.

This this **log4j properties file tutorial**, I am showing the example code for configuring log4j using **log4j.properties** file.

1. Log4j maven dependencies

[Create a maven java project](#) and update log4j maven dependencies.

pom.xml

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. log4j.properties file

This is the main properties file having all runtime configuration used by log4j. This file will have log4j appenders information, log level information and output file names for file appenders.

log4j.properties

```
log4j.rootLogger=DEBUG, consoleAppender, fileAppender

log4j.appender.consoleAppender=org.apache.log4j.ConsoleAppender
log4j.appender.consoleAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.consoleAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n

log4j.appender.fileAppender=org.apache.log4j.RollingFileAppender
log4j.appender.fileAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.fileAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n
log4j.appender.fileAppender.File=demoApplication.log
```

3. log4j.properties example

Log4jPropertiesConfigurationExample.java

```
package com.howtodoinjava;

import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;

public class Log4jPropertiesConfigurationExample
{
```

```

static Logger logger = Logger.getLogger(Log4jPropertiesConfigurationExample.class);
public static void main(String[] args)
{
    //PropertiesConfigurator is used to configure logger from properties file
    PropertyConfigurator.configure("log4j.properties");

    //Log in console in and log file
    logger.debug("Log4j appender configuration is successful !!");
}
}

```

Output in console and **demoApplication.log** in project root folder:

Console

```
[main] DEBUG com.howtodoinjava.Log4jPropertiesConfigurationExample - Log4j appender configuration is successful !!
```

Now let's see some **log4j.properties examples** to output log messages to specific location.

4. Log4j ConsoleAppender – Logging to console

Java program to output logs to console.

```

# Root logger
log4j.rootLogger=INFO, console

log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.Target=System.out
log4j.appender.console.layout=org.apache.log4j.PatternLayout

```

```
log4j.appender.consoleAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n
```

Log4j RollingFileAppender – Logging to file

Java program to output logs to file.

```
# Root logger
log4j.rootLogger=INFO, file

# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=C:\\temp\\info.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=[%t] %-5p %c %x - %m%n
```

Let me know if any question.

Happy Learning !!

Was this post helpful?

YES

NO

ADVERTISEMENTS



About Lokesh Gupta

A family guy with fun loving nature. Love computers, programming and solving everyday problems. Find me on [Facebook](#) and [Twitter](#).

Feedback, Discussion and Comments

Pratik

July 2, 2020

Hi,

How do I choose between two appenders based on a condition and define in log4j.properties?

Thanks in advance.

NarsingRao Manda

January 9, 2020

Hi,

Actually i am using log4j in my python(pyspark) code but i am unable to log .

please observe below log4j.properties

log4j.rootCategory=INFO,console,file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=D:\\Logs\\Error.log

log4j.appender.file.MaxFileSize=10MB

log4j.appender.file.MaxBackupIndex=10

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=[%t] %-5p %c %x - %m%n

log4j.appender.console=org.apache.log4j.ConsoleAppender

```
log4j.appender.console.target=System.err
log4j.appender.console.layout=org.apache.log4j.PatternLayout
log4j.appender.console.layout.ConversionPattern=%d{yy/MM/dd HH:mm:ss} %p %c{1}: %m%n

please observe my python code

spark_session = SparkSession.builder.appName("Transform Raw to Refined").getOrCreate()
log4jLogger = spark_session._jvm.org.apache.log4j
LOGGER = log4jLogger.LogManager.getLogger('MYLOGGER')
LOGGER.setLevel(log4jLogger.Level.INFO)
LOGGER.info('This message should go to the log file')

i cant get what did i make mistake

please help me.
```

[peng.w](#)

June 20, 2019

4. Log4j ConsoleAppender – Logging to console

Java program to output logs to console.

Root logger

```
log4j.rootLogger=INFO, console
```

```
log4j.appender.console=org.apache.log4j.ConsoleAppender
```

```
log4j.appender.console.Target=System.out
```

```
log4j.appender.console.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.consoleAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n
```

Make sure the last line is the word consoleAppender ?

Arun

January 17, 2019

How to write log4j without trace

uday

March 26, 2018

Hi, can log4j support deleting of old log files ?

this is the configuration of my log file:


```
# log4j.rootLogger=debug, stdout, R
log4j.rootLogger=trace, R

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

# Pattern to output the caller's file name and line number.
log4j.appender.stdout.layout.ConversionPattern=%d{%Y-%m-%d %H:%M:%S} %r %-5p [%t] (%F:%l) - %m%n

log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=/home/blueserver/symphony/config/securityservice.log

#size of the File
log4j.appender.R.MaxFileSize=10MB

# Keep one backup file
log4j.appender.R.MaxBackupIndex=10

log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{%Y-%m-%d %H:%M:%S} %r %-5p [%t] - %m%n
```

[Lokesh Gupta](#)

March 26, 2018

Is MaxBackupIndex not working as expected? Your max log archive files count should be 10. After that they should be deleted.

Ravi

June 3, 2017

Thanks for the example

sunil

July 21, 2016

how we can create a separate log file for each and every test case in test class

Suppose I have two test cases with names X and Y, when I execute all the test cases (X and Y), then I need to create folder (in parallel to execution) with current time stamp and inside that folder I need to create two log files with name same as the test case names (i.e. X and Y)

please help if any body having idea

Note : I am using Log4j framework with java language.

automated-testing

sandeep

May 18, 2015

how to create two different log file and how to copy the content of console to file

Antonio

November 20, 2014

it is not named log4j.properties instead of log4g.properties? It is called in the method as log4j.properties

tarun

August 5, 2014

Hi in my console logs are not getting displayed.

I have added the dependency.

The log4j.property file is mentioned below:

Set root logger level to DEBUG and its only appender to A1.

log4j.rootLogger=DEBUG, A1

A1 is set to be a ConsoleAppender.

```
log4j.appender.A1=org.apache.log4j.ConsoleAppender
```

***** Set root logger level to WARN and its two appenders to stdout and R.

```
log4j.rootLogger=INFO, R
```

```
log4j.appender.R=org.apache.log4j.RollingFileAppender
```

```
log4j.appender.R.File=auto-test.log
```

```
log4j.appender.R.MaxFileSize=10MB
```

```
log4j.appender.R.MaxBackupIndex=10
```

```
log4j.appender.R.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.R.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %p - %m%n
```

I am using maven with TestNG in IntelliJ IDEA

[Lokesh Gupta](#)

August 5, 2014

Why “log4j.rootLogger” is written twice? Is it permitted? I believe it should be written only once in top of log file. Set different logging level using threshold property.

```
log4j.appender.R.Threshold = INFO
```

```
log4j.appender.A1.Threshold = DEBUG
```

Neelam Joshi

February 6, 2014

Hi, Could you please help me to create log file on parallel run because the log outputs to the console will be messed up with each other in parallel run.

Lokesh Gupta

February 6, 2014

Can you please elaborate more what exactly you want to achieve?

Neelam Joshi

February 6, 2014

Sorry for misunderstanding. Actually i am using log4j for my automation script(Using testng) and i am creating log files in html format. So earlier i was executing multiple scripts one by one and log files get generated properly correspondence to their tests(each log file for each method). Now i am running my scripts parallelly(lets say 10browsers/ methods running simultaneously). So i am facing issue because console for each tests are started writing on other test's log file. So if i am running 10 tests parallelly, at the the end of the script execution all the print command appearing on last log file and other 9 logs file are partially filled.

I hope you got my point 😊

Thanks

[Lokesh Gupta](#)

February 6, 2014

If possible, can you please share your log4j properties/configuration file here. That will help in determining the correct solution for you.

[Neelam Joshi](#)

February 6, 2014

Okay. My log4j.properties file is as follows:

log = /usr/home/log4j

log4j.rootLogger = INFO, file

log4j.appender.file = com.logica.part.test.util.NewLogForEachRunFileAppender

log4j.appender.file.File=C:/workspace/Evolve/log/logs.html

log4j.appender.file.encoding=UTF-8

log4j.appender.file.layout=com.ceb.shl.test.util.CustomLogHtmlLayout

log4j.appender.file.layout.Title=Output

```
log4j.appender.file.layout.LocationInfo=true
```

```
#log4j.appender.myappender.encoding=UTF-8
```

NewLogForEachRunFileAppender is my java file where i am renaming the file name for each test.

and the code i am using is:

```
package com.logica.part.test.util;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Calendar;
```

```
import java.util.UUID;
```

```
import org.apache.log4j.FileAppender;
```

```
import org.apache.log4j.Layout;
```

```
import org.apache.log4j.spi.ErrorCode;
```

```
public class NewLogForEachRunFileAppender extends FileAppender {
```

```
    public NewLogForEachRunFileAppender() {
```

```
    }
```

```
    public NewLogForEachRunFileAppender(Layout layout, String filename,
```

```
        boolean append, boolean bufferedIO, int bufferSize)
```

```
throws IOException
{
    super(layout, filename, append, bufferedIO, bufferSize);
}

public NewLogForEachRunFileAppender(Layout layout, String filename,
    boolean append) throws IOException {
    super(layout, filename, append);
}

public NewLogForEachRunFileAppender(Layout layout, String filename)
    throws IOException {
    super(layout, filename);
}

public void activateOptions() {
    if (fileName != null) {
        try {
            fileName = getNewLogFileName();
            setFile(fileName, fileAppend, bufferedIO, bufferSize);
        } catch (Exception e) {
            errorHandler.error("Error while activating log options", e,
```



```

        ErrorCode.FILE_OPEN_FAILURE);
    }
}

private String getNewLogFileName() {
    if (fileName != null) {
        System.out.println("filename-exist: "+fileName);
        final String DOT = ".";
        final String HIPHEN = "-";
        final File logFile = new File(fileName);
        final String fileName = logFile.getName();
        String newFileName = "";

        final int dotIndex = fileName.indexOf(DOT);
        if (dotIndex != -1) {
            System.out.println("logFile");
            UUID logId = UUID.randomUUID();
            newFileName = "log_"+logId+".html";
            // newFileName = fileName.substring(0, dotIndex)+System.currentTimeMillis()+fileName.substring(dotIndex);
        } else {

```

```
System.out.println("elseLogFile");  
  
// the file name has no extension. So, just append the timestamp  
  
// at the end.  
  
newFileName = fileName;  
  
}  
  
System.out.println(logFile.getParent()+File.separator+newFileName);  
  
return logFile.getParent()+File.separator+newFileName;  
  
}  
  
return null;  
  
}  
  
}
```

Saro

November 12, 2013

Hi can we configure to create a separate file for INFO, DEBUG, ERROR

Lokesh Gupta

November 12, 2013

I believe we cannot. It's related to appenders. Log level has nothing to do with it.

Asif C

November 10, 2013

Hi, Where can i place the log4j.properties file under tomcat folder?

mauroprogram

September 12, 2013

But for the file log4g.properties what it is the location into the project ? into the directory resource ?????

Lokesh Gupta

September 12, 2013

Yes, you are right

[mauroprogram](#)

September 14, 2013

your site it is very very interessant .

All java technologies are explained thoroughly on your site.

Comments are closed on this article!

Search Tutorials



ADVERTISEMENTS

ADVERTISEMENTS

ADVERTISEMENTS

ADVERTISEMENTS

Log4j2 Tutorial

Log4j2 – Introduction

Log4j2 – JSON Config

Log4j2 – Properties Config

Log4j2 – XML Config

Log4j2 – RollingFileAppender

Log4j2 – Multiple appenders

Log4j2 – LevelRangeFilter

Log4j2 – HTMLLayout

Log4j2 – Fish Tagging

Log4j2 – Conversion Patterns

Log4j2 – JUnit

Log4j Tutorial

Log4j – Introduction

[Log4j – Properties Config](#)

Log4j – XML Config

Log4j – Maven Config

Log4j – Logging Levels

Log4j – ConsoleAppender

Log4j – RollingFileAppender

Log4j – SocketAppender

Log4j – JDBCAppender

Log4j – XMLLayout

Log4j – HTMLLayout

Log4j – Runtime Reload

Log4j vs. SLF4j

Log4j – RESTEasy + Tomcat 7

Meta Links

[About Me](#)

[Contact Us](#)

[Privacy policy](#)

[Advertise](#)

[Guest and Sponsored Posts](#)

Recommended Reading

[10 Life Lessons](#)

[Secure Hash Algorithms](#)

[How Web Servers work?](#)

[How Java I/O Works Internally?](#)

[Best Way to Learn Java](#)

Java Best Practices Guide

Microservices Tutorial

REST API Tutorial

How to Start New Blog

Copyright © 2020 · [HowToDoInJava.com](#) · All Rights Reserved. | [Sitemap](#)