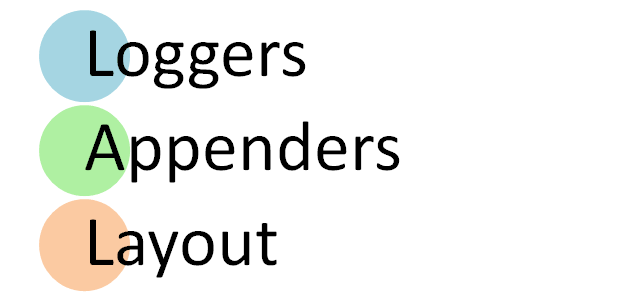
**What is Log4j?**

Log4j is a fast, flexible and reliable logging framework (APIS) written in[Java](https://www.guru99.com/java-tutorial.html)developed in early 1996. It is distributed under the[Apache](https://www.guru99.com/apache.html)Software License. Log4J has been ported to the C, C++, C#, [Perl](https://www.guru99.com/perl-tutorials.html), [Python](https://www.guru99.com/python-tutorials.html), Ruby and Eiffel Languages. It is a tool used for small to large scale Selenium Automation projects.

## Why use Log4j?

* It is an open source
* With Log4j, it is possible to store the flow details of our Selenium Automation in a file or databases
* Log4j is used for large as well as small projects
* In Log4j, we use log statements rather than SOPL statements in the code to know the status of a project while it is executing

## Log4j has three principal components

[](https://www.guru99.com/images/5-2015/050115_0810_TutorialonL1.png)

1. **Loggers**: It is responsible for logging information. To implement loggers into a project following steps need to be performed -

* **Create an instance for logger class**: Logger class is a Java-based utility that has got all the generic methods already implemented to use log4j
* **Define the Log4j level**: Primarily there are five kinds of log levels
  1. All - This level of logging will log everything ( it turns all the logs on )
  2. DEBUG – print the debugging information and is helpful in development stage
  3. INFO – print informational message that highlights the progress of the application
  4. WARN – print information regarding faulty and unexpected system behavior.
  5. ERROR – print error message that might allow system to continue
  6. FATAL – print system critical information which are causing the application to crash
  7. OFF – No logging

1. **Appenders**: It is used to deliver LogEvents to their destination. It decides what will happen with log information. In simple words, it is used to write the logs in file. Following are few types of Appenders
   1. ConsoleAppender logs to standard output
   2. File appender prints logs to some file
   3. Rolling file appender to a file with maximum size

Note: In log4j properties we can call appender with any name. There are other appenders as well but we will restrict to these few.

1. **Layouts**: It is responsible for formatting logging information in different styles.

The Logger class provides different methods to handle logging activities. It provides two static methods for obtaining a Logger Object.

Public static Logger getRootLogger()

Public static Logger getLogger(String name)

**How log4j is configured?**

To configure log4j we have to decide which appender to implement. Accordingly, parameters of appender will be set.

* We will use DEBUG level and RollingFileAppender
* We will do two configurations or logs,
  + First: root logger, that will write all system generated logs in file name i.e. Selenium.logs
  + Second: Will write the information generated by manual commands in code into the file name- Manual.logs
* Layout will be PatternLayout

#Root logger

log4j.rootLogger=DEBUG,file

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

log4j.appender.file.File=D:\\Guru99\\src\\Selenium.logs

log4j.appender.file.maxFileSize=900KB

log4j.appender.file.maxBackupIndex=5

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

log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c<strong>**{1}**</strong>:%L - %m%n

log4j.appender.file.Append=false

#Application Logs

log4j.logger.devpinoyLogger=DEBUG, dest1

log4j.appender.dest1=org.apache.log4j.RollingFileAppender

log4j.appender.dest1.maxFileSize=900KB

log4j.appender.dest1.maxBackupIndex=6

log4j.appender.dest1.layout=org.apache.log4j.PatternLayout

log4j.appender.dest1.layout.ConversionPattern=%d{dd/MM/yyyy HH:mm:ss} %c %m%n

log4j.appender.dest1.File=D:\\ Guru99\\src\\Manual.logs

log4j.appender.dest1.Append=false

In the above Example, we have configured log4j to log in two different files named as Selenium.log and Manual.log.

* file and dest1 are the two identifiers.
* "File" is used to give file name in which logs will be saved
* "maxFileSize" is used to configure the maximum size of the log file. When file reaches this size, a new file will be created with the same name and the old file name will be add as an Index to it.
* "maxBackupIndex" is used to configure maximum number of files to be backup.
* "layout" is used to set the format of the log file.
* "Append" is used to set append function. If it is set to false, than every time a new file will be created rather than old file will be used for logging

**How log4j is used within script?**

In code, we have used "log" as a reference variable referencing getLogger method of Logger Class

Logger log = Logger.getLogger("devpinoyLogger");

Use "log" referencing variable and debug method to log the information we want.

log.debug("--information--");

**What is a LogExpert tool?**

1. LogExpert tool is a tool for Windows developed to tail the logs
2. It is free and open source log viewer.
3. It is a log analysis tool with multiple features like searching, filtering, bookmarking and highlighting the logs
4. In this tool logs, files get automatically updated when opened
5. In this tool, we can open multiple log file in different tabs
6. We can also put comments on bookmarks, and there is the shortcut key to navigate in between different bookmarks. We can also see complete bookmark list and navigate from there
7. Shortcuts of the tool are given in help file so that they can be referred to the tool.

## Steps to use Log4j with Selenium

**Step 1)** In Eclipse create a new project with name log4j\_demo

**Step 2)**Right click on src -> Build Path -> Configure Build Path

**Step 2)**Click on Libraries and Add Log4J Library . You can download it from <https://logging.apache.org/log4j/1.2/download.html>

**Step 3)**Create a new file. This file will include all the log4j configuration

1. Right click on src -> New -> Other -> General -> File
2. Give the file name as "log4j.properties"
3. Click on Finish

Create two more files and give them names such as Selenium.logs and Manual.logs. These files will contain all the logs created by system and manually logged statements

**Step 4)**In log4j.properties copy the entire configuration.

**Step 5)**Create main class:

1. Right click on default package -> New -> Class
2. Give the class name and click on finish

**Step 6)**Copy the following code in to the main class

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.apache.log4j.Logger;

public class LoggingDemo {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

WebDriver driver = new FirefoxDriver();

Logger log = Logger.getLogger("devpinoyLogger");

driver.get("<http://healthunify.com/bmicalculator/>");

log.debug("opening webiste");

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

log.debug("entring weight");

driver.findElement(By.name("wg")).sendKeys("87");

log.debug("selecting kilograms");

driver.findElement(By.name("opt1")).sendKeys("kilograms");

log.debug("selecting height in feet");

driver.findElement(By.name("opt2")).sendKeys("5");

log.debug("selecting height in inchs");

driver.findElement(By.name("opt3")).sendKeys("10");

log.debug("Clicking on calculate");

driver.findElement(By.name("cc")).click();

log.debug("Getting SIUnit value");

String SIUnit = driver.findElement(By.name("si")).getAttribute("value");

log.debug("Getting USUnit value");

String USUnit = driver.findElement(By.name("us")).getAttribute("value");

log.debug("Getting UKUnit value");

String UKUnit = driver.findElement(By.name("uk")).getAttribute("value");

log.debug("Getting overall description");

String note = driver.findElement(By.name("desc")).getAttribute("value");

System.out.println("SIUnit = " + SIUnit);

System.out.println("USUnit = " + USUnit);

System.out.println("UKUnit = " + UKUnit);

System.out.println("note = " + note);

driver.quit();

}

}

In the above code, we visit <http://healthunify.com/bmicalculator/>and verify BMI calculator. The weight entered is 87KG and the height is 5 Feet 10 inches. The script checks output in SE, US and UK units.

Using Logger.getLogger("devpinoyLogger") we create system level logs

Using log.debug method we store data into Manual.log

**Step 7)**Run the script. Open the location of Manual and Selenium logs to check logging data.

**How LogExpert tool can be used to analyze logs**

1. Download the tool from <http://logexpert.codeplex.com/> . Go to LogExpert download folder
2. Open LogExpert.exe
3. Click on File -> Open and Browse to the path where Manual.log and Selenium.log files are stored. Select the file
4. Select the "Follow tail" option