**Logging Framework – log4J:**

What is Log4j and what are its uses?

Log4J is a reliable, fast and flexible logging framework (APIs) written in Java which is distributed under the Apache Software License. It is used by programmers to output log statements to a variety of targets. These are very useful in case of errors as the root cause of the error can be located.

Log4J is developed by apache to log any errors, output etc.

Why Log4J:

Log4j is fully configurable using .properties or .xml.

Log4j enables you to find out easily which are information messages, which are debug messages, which are errors and which are fatal errors.

Log4j records information upto class level. For example, you can output only error messages for one class and for another info. Messages only. Similarly at package level also.

If any error occurs, it will show you the class name also, if configured.

**Installing log4j:**

Go to google and type apache log4j download.

Download the binary zip file.

**Important note: Now once you unzip this you will see different jar files and you need only log4jcore and log4japi. If you include anymore files in build path you will get classnotfound exception.**

Example:

**package** Log4JTutorials;

**import** org.apache.logging.log4j.LogManager;

**import** org.apache.logging.log4j.Logger;

**public** **class** TextBox {

**private** **static** Logger *log* = LogManager.*getLogger*(TextBox.**class**.getName());

**public** **static** **void** main(String[] args) {

*log*.debug("This is debug message");

*log*.info("This is info message");

*log*.error("This is error message");

*log*.fatal("This is fatal message");

}

}

Import the following package manually.

**import** org.apache.logging.log4j.\*;

Now create a logger object inside the class by passing the class name.

**private** **static** Logger *log* = LogManager.*getLogger*(TextBox.**class**.getName());

The different methods inside this object are debug(), error(), fatal(), info() etc.

Result:

ERROR StatusLogger No Log4j 2 configuration file found. Using default configuration (logging only errors to the console), or user programmatically provided configurations. Set system property 'log4j2.debug' to show Log4j 2 internal initialization logging. See https://logging.apache.org/log4j/2.x/manual/configuration.html for instructions on how to configure Log4j 2

00:53:01.107 [main] ERROR package1.First - This is error message

00:53:01.107 [main] FATAL package1.First - This is fatal message

The above prints only error and fatal messages and not debug and info messages. This is because it needs a configuration file called Log4j 2. If we don’t use the configuration file, it uses a default configuration file and prints only Error and Fatal message.

If you don’t use the log 4j 2 configuration file, it will use default configuration file and debug and info messages won’t be printed.