Now, if you check the configurations written in sample6, the error messages which needs to printed in logfile2, is printed in logfile1 as well. The reason behind this, is that log4j loggers follow hierarchies. So, RootLogger prints the message in logFile1. And obviously the logger defined as ErrorMessageReceivedLogger previously, is a child of RootLogger.

**<Logger name="ErrorMessageRecievedLogger">**

**<AppenderRef ref="RollingFile2" level="ERRORMESSASERECIEVED"/>**

**</Logger>**

And whole logger configuration is like:

**<Loggers>**

**<Root level="DEBUG">**

**<AppenderRef ref="Console"/>**

**<AppenderRef ref="RollingFile1"/>**

**</Root>**

**<Logger name="ErrorMessageRecievedLogger" additivity="false">**

**<AppenderRef ref="RollingFile2" level="ERRORMESSASERECIEVED"/>**

**</Logger>**

**</Loggers>**

As a result, the message printed in RollingFile2, will also be printed in RollingFile1 and console. Now, suppose, according to your porgram requirements, this is a redundancy. TO avoid this redundancy log4j additivity can be used. Now, The message printed by logger **ErrorMessageRecievedLogger** wll no longer be printed by **RootLogger.**

However, if you want to print the error message recieved in console and rollingfile2, just not in RollingFile1, change the logger configuration to the following:

**<Loggers>**

**<Root level="DEBUG">**

**<AppenderRef ref="Console"/>**

**<AppenderRef ref="RollingFile1"/>**

**</Root>**

**<Logger name="ErrorMessageRecievedLogger" additivity="false">**

**<AppenderRef ref="RollingFile2" level="ERRORMESSASERECIEVED"/>**

**<AppenderRef ref=”Console”/>**

**</Logger>**

**</Loggers>**