1. **Apache ErrorLog/Common log**

Edit /etc/httpd/httpd.conf

ErrorLog /data/apache\_log/error\_log

LogLevel warn

logFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %{X-Forwarded-For}i %{True-Client-IP}i" combined

LogFormat "%h %l %u %t \"%r\" %>s %b" common

LogFormat "%{Referer}i -> %U" referer

LogFormat "%{User-agent}i" agent

CustomLog /data/apache\_log/access\_log combined

You may wish to store all error-related information in one error log. If so, the above is fine, even for multiple domains. However, you can specify an error log file for each individual domain you have. This is done in the <VirtualHost> container with an entry like this:

<VirtualHost \*:80>

DocumentRoot /data/www/finra

ServerName finra-markets.morningstar.com

ErrorLog "/data/apache\_log/finra-error\_log"

CustomLog "/data/apache\_log/finra-access\_log" combined

……

</VirtualHost>

1. **Tomcat:**

**Tomcat access log:**

Set ‘rotatable’ to false. The default value is ‘true’

<Valve className="org.apache.catalina.valves.AccessLogValve" **rotatable="false"** prefix="access\_log"

pattern="%t|%s|%D|%{Referer}i|%h|%{X-Forwarded-For}i|%{True-Client-IP}i|%v/%p|%m|%{User-Agent}i|%I|%b|%U|%q|%{authlog}r"

resolveHosts="false" fileDateFormat="yyyy-MM-dd"/>

**Logj4 configuration file (log4j.properties under the directory ${tomcat-base}/lib. This is the global configuration for log4j in Tomcat.)**

log4j.rootLogger=INFO, CONSOLE

log4j.appender.CONSOLE=org.apache.log4j.RollingFileAppender

# Set the name of the file corresponding appender defined above

log4j.appender.CONSOLE.File=${catalina.base}/logs/catalina.txt

# Set the threshold

log4j.appender.CONSOLE.Threshold=INFO

#Set append to true, don't overwrite

log4j.appender.CONSOLE.Append=true

#set the maximum file size before rollover

log4j.appender.CONSOLE.MaxFileSize=500MB

#set the backup index

log4j.appender.CONSOLE.MaxBackupIndex=2

# Define all the appenders

log4j.appender.CATALINA=org.apache.log4j.RollingFileAppender

# Set the name of the file corresponding appender defined above

log4j.appender.CATALINA.File=${catalina.base}/logs/catalina.txt

# Set the threshold

log4j.appender.CATALINA.Threshold=INFO

#Set append to true, don't overwrite

log4j.appender.CATALINA.Append=true

#set the maximum file size before rollover

log4j.appender.CATALINA.MaxFileSize=500MB

#set the backup index

log4j.appender.CATALINA.MaxBackupIndex=2

log4j.appender.LOCALHOST=org.apache.log4j.RollingFileAppender

# Set the name of the file corresponding appender defined above

log4j.appender.LOCALHOST.File=${catalina.base}/logs/localhost.txt

# Set the threshold

log4j.appender.LOCALHOST.Threshold=INFO

#Set append to true, don't overwrite

log4j.appender.LOCALHOST.Append=true

#set the maximum file size before rollover

log4j.appender.LOCALHOST.MaxFileSize=500MB

#set the backup index

log4j.appender.LOCALHOST.MaxBackupIndex=2

log4j.appender.MANAGER=org.apache.log4j.RollingFileAppender

# Set the name of the file corresponding appender defined above

log4j.appender.MANAGER.File=${catalina.base}/logs/manager.txt

# Set the threshold

log4j.appender.MANAGER.Threshold=INFO

#Set append to true, don't overwrite

log4j.appender.MANAGER.Append=true

#set the maximum file size before rollover

log4j.appender.MANAGER.MaxFileSize=500MB

#set the backup index

log4j.appender.MANAGER.MaxBackupIndex=2

log4j.appender.HOST-MANAGER=org.apache.log4j.RollingFileAppender

# Set the name of the file corresponding appender defined above

log4j.appender.HOST-MANAGER.File=${catalina.base}/logs/host-manager.txt

# Set the threshold

log4j.appender.HOST-MANAGER.Threshold=INFO

#Set append to true, don't overwrite

log4j.appender.HOST-MANAGER.Append=true

#set the maximum file size before rollover

log4j.appender.HOST-MANAGER.MaxFileSize=500MB

#set the backup index

log4j.appender.HOST-MANAGER.MaxBackupIndex=2

**Change log4j configuration files using in applications in the similar way as above.**

For example:

log4j.rootLogger=warn,console

log4j.appender.console =org.apache.log4j.RollingFileAppender

log4j.appender.console.File=${catalina.base}/logs/webapp.txt

log4j.appender.console.Threshold=INFO

log4j.appender.console.Append=true

log4j.appender.console.MaxFileSize=500MB

log4j.appender.console.MaxBackupIndex=2

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

log4j.logger.error=debug,error

log4j.appender.error=org.apache.log4j.RollingFileAppender

log4j.appender.error.File=${catalina.base}/logs/error.txt

log4j.appender.error.Append=true

log4j.appender.error.MaxFileSize=500MB

log4j.appender.error.MaxBackupIndex=2

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

log4j.logger.authLog=info,authLog

log4j.appender.authLog = org.apache.log4j.RollingFileAppender

log4j.appender.authLog.File =${catalina.base}/logs/authLog.txt

log4j.appender.authLog.Append=true

log4j.appender.authLog.MaxFileSize=500MB

log4j.appender.authLog.MaxBackupIndex=2

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

1. **Rotate all log files on specific size and rotate hourly**
2. Copy the Logrotate script from /etc/cron.daily folder to /etc/cron.hourly

[mstar@msquote85 ~]$ sudo cp /etc/cron.daily/logrotate /etc/cron.hourly/

1. Update the /etc/cron.hourly/logrotate file; comment and replace the last line as follows

[mstar@msquote85 cron.hourly]$ sudo vim /etc/cron.hourly/logrotate

#/usr/sbin/logrotate /etc/logrotate.conf

/usr/sbin/logrotate /etc/logrotate.d/httpd\_hourly

/usr/sbin/logrotate /etc/logrotate.d/tomcat\_hourly

1. Create the following two files under the directory /etc/logrotate.d/httpd\_hourly and /etc/logrotate.d/tomcat\_hourly

[mstar@msquote62 ~]$ sudo vim /etc/logrotate.d/httpd\_hourly

/data/apache\_log/\*log {

hourly

size 200M

rotate 4

copytruncate

#compressvim

#delaycompress

missingok

notifempty

}

[mstar@msquote62 ~]$ sudo vim /etc/logrotate.d/tomcat\_hourly

/data/tomcat/logs/\*out /data/tomcat/logs/\*\_txt {

hourly

size 200M

rotate 5

copytruncate

#compressvim

#delaycompress

missingok

notifempty

}

1. The following screenshot show the file rotating while the cron job executing.







