Silencing Agentcmd Junk in the uptime.log

Contents:

- Linux Monitoring Station:
- Windows Monitoring Station:

In 7.2 the agentcmd has extra logging enabled that often interferes with the regular output from a custom monitor. It is possible to omit this extra logging output with the following steps.
If you find your custom monitor behaves differently after upgrading to Uptime Infrastructure Monitor 7.2, you can verify if the agentcmd logging output is the culprit by running the following command.

```
<uptime_dir>/scripts/agentcmd -s -p 9998 localhost ver
```

If the output returned contains "Initialized AgentCommunicatorApp", then continue with the steps below to resolve.
This also shows up in the uptime.log file like so:

```
2014-03-06 10:37:29,250 INFO [] - Initialized AgentCommunicatorApp
2014-03-06 10:37:29,250 INFO [] - Database connection was not explicitly initialized; performing check...
```

Linux Monitoring Station:

1. Edit the agentcmd.jcnf file in the scripts directory of the monitoring station:
In the following section:

```
[jvm.additional.args]
-d64
```

Add:

```
-Dlog4j.configuration=file:///usr/local/uptime/log4j.properties
```

So should now see:

```
[jvm.additional.args]
-d64
-Dlog4j.configuration=file:///usr/local/uptime/log4j.properties
```

2. Create a log4j.properties file in the uptime directory and populate it with the following lines.

```
log4j.rootLogger=WARN, THIRDPARTY
#console logger
log4j.appender.CONSOLE=org.apache.log4j.RollingFileAppender
log4j.appender.CONSOLE.MaxFileSize=10MB
log4j.appender.CONSOLE.MaxBackupIndex=0
log4j.appender.CONSOLE.File=logs/uptime_scripts.log
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%d | %p | %X{context} | %t | %c | %m%n
#third party code logger - thirdparty.log
log4j.appender.THIRDPARTY=org.apache.log4j.RollingFileAppender
log4j.appender.THIRDPARTY.MaxFileSize=10MB
log4j.appender.THIRDPARTY.MaxBackupIndex=0
log4j.appender.THIRDPARTY.File=logs/thirdparty.log
log4j.appender.THIRDPARTY.layout=org.apache.log4j.PatternLayout
log4j.appender.THIRDPARTY.layout.ConversionPattern=%d | %p | %X{context} | %t | %c | %m%n
```

3. Now you should be able to run the agentcmd and see only the expected output. The new extra logging lines from agentcmd are now redirected to logs /uptime_scripts.log which will rollover whenever it reaches 10mb, so it will never exceed 10mb.

Windows Monitoring Station:

1. Create a agentcmd.l4j.ini file in the scripts directory of the Uptime Infrastructure Monitor Monitoring Station with the following contents.

```
# Launch4j runtime config
 -Dlog4j.configuration="file:/C:/Program Files/uptime software/uptime/log4j.properties"

 ## -Dlog4j.debug=true
```

2. Create a log4j.properties file in the uptime directory and populate it with the following lines.

```
log4j.rootLogger=WARN, THIRDPARTY
#console logger
log4j.appender.CONSOLE=org.apache.log4j.RollingFileAppender
log4j.appender.CONSOLE.MaxFileSize=10MB
log4j.appender.CONSOLE.MaxBackupIndex=0
log4j.appender.CONSOLE.File=logs/uptime_scripts.log
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%d | %p | %X{context} | %t | %c | %m%n
#third party code logger - thirdparty.log
log4j.appender.THIRDPARTY=org.apache.log4j.RollingFileAppender
log4j.appender.THIRDPARTY.MaxFileSize=10MB
log4j.appender.THIRDPARTY.MaxBackupIndex=0
log4j.appender.THIRDPARTY.File=logs/thirdparty.log
log4j.appender.THIRDPARTY.layout=org.apache.log4j.PatternLayout
log4j.appender.THIRDPARTY.layout.ConversionPattern=%d | %p | %X{context} | %t | %c | %m%n
```

3. Wherever the agentcmd command is being called from (ie. your custom monitor script), update the agentcmd line to include the --l4j-debug option. For example, if the line looked like this:

```
$NETCAT -p $port $hostname rexec $password $customscript
```

Then you will want to change it to look like this:

```
$NETCAT --l4j-debug -p $port $hostname rexec $password $customscript
```