

Troubleshooting wotaskd and JavaMonitor on OS X

Performance



The logging settings outlined on this page are intended to help you troubleshoot a problematic install. They are **not** intended to be used in a running, production environment. They will cause substantial performance degradation.

Logging for JavaMonitor and wotaskd

It is very useful to capture the JavaMonitor and wotaskd logs by adding the following to the .plist files in `/Library/LaunchDaemons`. This is the preferred method of capturing log messages since it works even if JavaMonitor or wotaskd fail to launch.

Add this code to both of these files:

- **wotaskd:** `/Library/LaunchDaemons/com.apple.webobjects.wotaskd.plist`
- **JavaMonitor:** `/Library/LaunchDaemons/com.apple.webobjects.womonitor.plist`

womonitor.plist and wotaskd.plist Addition

```
<key>StandardOutPath</key>
<string>/Library/WebObjects/Logs/womonitor.log</string>

<key>StandardErrorPath</key>
<string>/Library/WebObjects/Logs/womonitor.log</string>
```

The `/Library/WebObjects/Logs` should already be owned by appserver so there shouldn't be any permissions troubles with writing the logs there. If the files fail to show up, double check the permissions on the `/Library/WebObjects/Logs` directory. `% ls -la /Library/WebObjects` will show you.

Now that log files are being written, turn on Deployment Debugging by adding these two `<string>` elements to the `<ProgramArguments>` element of womonitor.plist and/or wotaskd.plist:

womonitor.plist and/or wotaskd.plist Addition

```
<key>ProgramArguments</key>
<array>
  ...
  <string>_DeploymentDebugging</string>
  <string>true</string>
</array>
```

Or, if you manually launch JavaMonitor and/or wotaskd, you can simply add `_DeploymentDebugging true` to your launch command.

`_DeploymentDebugging` is the equivalent of passing all the following settings in either on the command-line or in a Properties file:

Deployment Debugging Properties

```
NSLog.debug.setIsVerbose(true);
NSLog.out.setIsVerbose(true);
NSLog.err.setIsVerbose(true);
NSLog.allowDebugLoggingForGroups(NSLog.DebugGroupDeployment);
NSLog.debug.setAllowedDebugLevel(NSLog.DebugLevelDetailed);
```

Which gives nice details like this :

Log Example

```
2009-08-09 21:45:14,938 DEBUG 7.67 MB/24.02 MB [WorkerThread0] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample2 leopards.macti.lan 2003
2009-08-09 21:45:15,027 DEBUG 7.71 MB/23.97 MB [WorkerThread2] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample2 leopards.macti.lan 2004
2009-08-09 21:45:25,324 DEBUG 7.76 MB/23.93 MB [WorkerThread15] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample leopards.macti.lan 2005
2009-08-09 21:45:25,497 DEBUG 7.8 MB/23.88 MB [WorkerThread3] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample leopards.macti.lan 2001
2009-08-09 21:45:25,542 DEBUG 7.85 MB/23.84 MB [WorkerThread6] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample leopards.macti.lan 2006
2009-08-09 21:45:25,602 DEBUG 7.89 MB/23.8 MB [WorkerThread10] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample leopards.macti.lan 2002
2009-08-09 21:45:25,617 DEBUG 7.94 MB/23.75 MB [WorkerThread9] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat AjaxExample leopards.macti.lan 2007
2009-08-09 21:45:35,144 DEBUG 7.98 MB/23.71 MB [WorkerThread1] logging.ERXNSLogLog4jBridge - @@@@ Received
Lifebeat: lifebeat JavaMonitor leopards.macti.lan 56789
```

Logging Application Startup

Modify **SpawnOfWotaskd.sh** in `/System/Library/WebObjects/JavaApplications/wotaskd.woa/Contents/Resources` to make it capture logs to the same folder. This is very useful if your application won't launch when you start it with JavaMonitor.


SpawnOfWotaskd.sh Logging Addition

```
#!/bin/sh
# Modified by Mark Ritchie in Mar 2008
# - We now keep a log of any troubles while launching an application.

# $@ 1>/dev/null 2>&1 &
LOG=/Library/WebObjects/Logs/SpawnOfWotaskd.log
echo "*****" >>${LOG}
echo "date: `date`" >>${LOG}
echo "args: $@" >>${LOG}
$@ 1>>${LOG} 2>&1 &
```

Standard WebObjects Installs will Overwrite This Change

 This script gets overwritten each time you install an updated copy of WebObjects or packages which include WebObjects so keep a backup of it handy. Don't Use the Standard Install Location

 Consider installing JavaMonitor and wotaskd in `/Library/WebObjects/JavaApplications` instead of `/System/Library/WebObjects/JavaApplications`. A standard install of WebObjects will not overwrite them in this new location, but it does require additional work to make sure everything knows about the new location.