



| IBM Software Group

# How to diagnose starvation issue using IBM Thread and Monitor Dump Analyzer for Java Technology

Jinwoo Hwang ([jinwoo@us.ibm.com](mailto:jinwoo@us.ibm.com))  
IBM WebSphere Application Server support

Creator/Architect/Developer of the following tools:

- IBM HeapAnalyzer ( Base component of MDD4J )
- IBM Thread and Monitor Dump Analyzer
- IBM Pattern Modeling and Analysis Tool for Java Garbage Collector
- IBM Trace and Request Analyzer for WebSphere Application Server / MQ JMS
- IBM Web Server Plug-in Analyzer for WebSphere Application Server



ON DEMAND BUSINESS™

WebSphere® Support Technical Exchange

# Introduction

- What is a Java™ thread dump?
- Supports IBM, Solaris and HP-UX JVM
- One of the top download technologies at the alphaWorks (<http://www.alphaworks.ibm.com>)
- Patent pending technology



# Top download on alphaWorks

The screenshot shows the alphaWorks website interface. The main content area displays the 'My IBM Redbooks' page, which features a logo and a brief description of the application. To the right, a sidebar lists several links under the heading 'alphaWorks'. Blue arrows originate from the sidebar and point to specific sections of the main content, such as the 'RSS Feeds' link and the 'Top downloads' section. The 'Top downloads' section includes links to 'HeapAnalyzer', 'IBM Thread and Monitor Dump Analyzer for Java Technology', and 'IBM Toolkit for'. The overall layout is clean with a blue header and sidebar.



# Prerequisite

- Java 2 SDK/JRE 5.0 or higher for runtime of IBM Thread and Monitor Dump Analyzer
- Java thread dump generated from Java Virtual Machine 1.3.1, 1.4.x, 5.0 or 6.0

# Features

- Summary
- Thread detail view
- Monitor detail view
- List of hang suspects
- Thread comparison view
- Thread comparison summary
- Java Monitor lock comparison view



# How does it work?

- Parsing
- Directional graph
- Depth First Search
- Comparison



## Definition of a hang

- The process is still present, but is not responding in some sense.



## What may look like a hang?

- Deadlock (halt)
- Infinite loop
- Resource contention



# What do we need? Java thread dump!

- Javacore on IBM JVM™
- Generated by signal
- Summarizes the state of the JVM



# IBM Java thread dump example

```
NULL -----
0SECTION    TITLE subcomponent dump routine
NULL =====
1TISIGINFO  SIGQUIT received
1TIDATETIME Date:      2007/09/25 at 17:31:17
1TIFILENAME Javacore filename: C:\WebSphere\AppServer\profiles\javacore.20060925.173117.8308.txt
NULL -----
0SECTION    XHPI subcomponent dump routine
NULL =====
1XHERROR2   XHPI dump section only produced for SIGSEGV, SIGILL or SIGFPE.
NULL
NULL -----
0SECTION    CI subcomponent dump routine
NULL =====
1CIJAVAVERSION J2RE 1.4.2 IBM Windows 32 build cn142-20050609
1CIRUNNINGAS Running as a standalone JVM
1CICMDLINE   C:\WebSphere\AppServer/java/bin/java -Xbootclasspath/
```



# IBM Java Monitor Pool Dump

1LKMONPOOLDUMP Monitor Pool Dump (flat & inflated object-monitors):

```
2LKMONINUSE    sys_mon_t:0x08078EA8 infl_mon_t: 0x080820D8:  
3LKMONOBJECT   java.lang.ref.Reference$Lock@40500C78/40500C80: <unowned>  
3LKNOTIFYQ     Waiting to be notified:  
3LKWAITNOTIFY  "Reference Handler" (8160D48)  
2LKMONINUSE    sys_mon_t:0x08078F08 infl_mon_t: 0x08082100:  
3LKMONOBJECT   java.lang.ref.ReferenceQueue$Lock@405009B0/405009B8:  
<unowned>  
3LKNOTIFYQ     Waiting to be notified:  
3LKWAITNOTIFY  "Finalizer" (8162ED0)
```



# IBM Java Thread Details

2XMFULLTHDDUMP Full thread dump Classic VM (J2RE 1.4.2 IBM build cxia321420-20040626, native threads):

3XMTHREADINFO "Servlet.Engine.Transports : 1097" (TID:40D09810, sys\_thread\_t:B882E420, state:CW, native ID:3274046) prio=5

4XESTACKTRACE at java.lang.Object.wait(Native Method)

4XESTACKTRACE at com.ibm.wps.util.Semaphore.acquire(Semaphore.java(Compiled Code))

4XESTACKTRACE at com.ibm.wps.engine.Servlet doGet(Servlet.java(Compiled Code))

4XESTACKTRACE at com.ibm.wps.engine.Servlet doPost(Servlet.java(Compiled Code))



# Sun Java Thread Dump

Full thread dump Java HotSpot(TM) Client VM (1.4.2\_08-b03 mixed mode):

```
"Servlet.Engine.Transports : 3529" daemon prio=5 tid=0x0182c4c0 nid=0x1088 in
Object.wait() [1e9ff000..1e9ffc30]
at java.lang.Object.wait(Native Method)
at java.lang.Object.wait(Object.java:429)
at com.ibm.ws.util.BoundedBuffer.take(BoundedBuffer.java:307)
- locked <0x3f3a7880> (a com.ibm.ws.util.BoundedBuffer)
at com.ibm.ws.util.ThreadPool.getTask(ThreadPool.java:524)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:945)
```

```
"Servlet.Engine.Transports : 3526" daemon prio=5 tid=0x02b37580 nid=0x1085 runnable
[1dd7f000..1dd7fc30]
at java.net.SocketInputStream.socketRead0(Native Method)
at java.net.SocketInputStream.read(SocketInputStream.java:129)
at com.ibm.ws.io.Stream.read(Stream.java:17)
at com.ibm.ws.io.ReadStream.readBuffer(ReadStream.java:418)
at com.ibm.ws.io.ReadStream.read(ReadStream.java:110)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:454)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:912)
```



# HP Java Thread Dump

Full thread dump Java HotSpot(TM) Server VM (1.4.2 1.4.2.03-040401-18:59-PA\_RISC2.0 PA2.0 (aCC\_AP) mixed mode):

```
"NotificationService dispatcher : 360" daemon prio=10 tid=00097b68 nid=2538 lwp_id=39149
waiting for monitor entry [0x2e758000..0x2e7584f0]
at com.ibm.ws.util.ThreadPool.getTask(ThreadPool.java:528)
- waiting to lock <38adb3c0> (a com.ibm.ws.util.ThreadPool)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:945)
"LocalNotificationServiceDispatcher : 352" daemon prio=10 tid=0066d5e8 nid=2537
lwp_id=39147 in Object.wait() [0x2b575000..0x2b5754f0]
at java.lang.Object.wait(Native Method)
- waiting on <391eab60> (a com.ibm.ws.util.BoundedBuffer)
at java.lang.Object.wait(Object.java:429)
at com.ibm.ws.util.BoundedBuffer.take(BoundedBuffer.java:264)
- locked <391eab60> (a com.ibm.ws.util.BoundedBuffer)
at com.ibm.ws.util.ThreadPool.getTask(ThreadPool.java:524)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:945)
```



# Location of IBM Java thread dump

- **IBM\_JAVACOREDIR**
- **\_CEE\_DMPTARG** on z/OS.
- Current working directory
- **TMPDIR**
- /tmp directory (\Temp on Windows)
- STDERR



# IBM thread dump on various platforms

Operating System	Thread dump file name	Format Meaning
Windows and Linux	javacore.YYYYMMDD.HHMMSS.PID.txt	YYYY=year, MM=month, DD=day, SS=second, PID=processID
AIX	javacorePID.TIME.txt	PID=processID, TIME=seconds since 1/1/1970
Solaris and HP-UX	native_stdout.log	Standard output

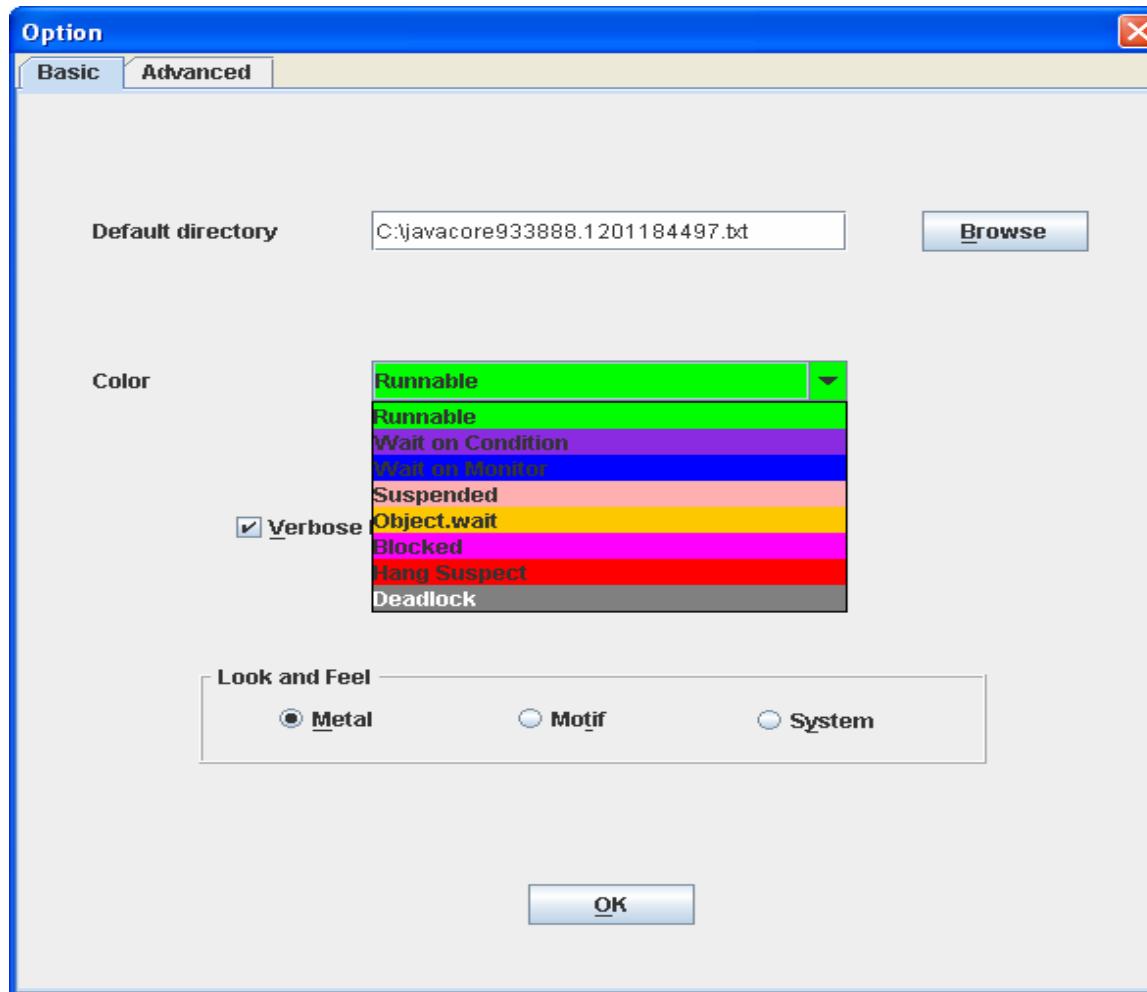


# Thread States

State	Name	Description
R	Runnable	Thread that has the ability to run or is running.
CW	Conditional Wait	Thread waiting on a condition variable.
MW	Monitor Wait	Thread waiting on a monitor lock
S	Suspended	Thread suspended.



# Color Coded Thread States



# Deadlock Detection

The screenshot shows the IBM Thread and Monitor Dump Analyzer for Java Technology interface. The main window title is "IBM Thread and Monitor Dump Analyzer for Java Technology". The menu bar includes File, Analysis, View, and Help. The toolbar contains various icons for file operations and analysis. A checkmark next to "Floatable" is visible in the top right corner.

The "Thread Dump List" table displays the following information:

Name	Runnable/Total Threads	Free/Allocated Heap(Free%)	AF(SC)/GC C...	Monitor	Deadlock
javacore1634460.11...	3/84	116,047,416/194,312,704(59%)	615/617		

A red warning message is displayed in the center of the window:

\*\*\*WARNING\*\*\* Deadlock detected in  
[Servlet.Engine.Transports : 14240] [Servlet.Engine.Transports : 14191]  
[Servlet.Engine.Transports : 13032] [Servlet.Engine.Transports : 13031]  
[Servlet.Engine.Transports : 5546] [Servlet.Engine.Transports : 5544]

Below the warning, details about the thread dump are listed:

- File name: javacore1634460.1133786553.txt
- Cause of thread dump: signal 3 received
- Date: 2005/12/05 at 13:42:33
- Process ID: 1634460

The status bar at the bottom shows several small icons and the text "Status".

# Deadlock in Monitor Detail View

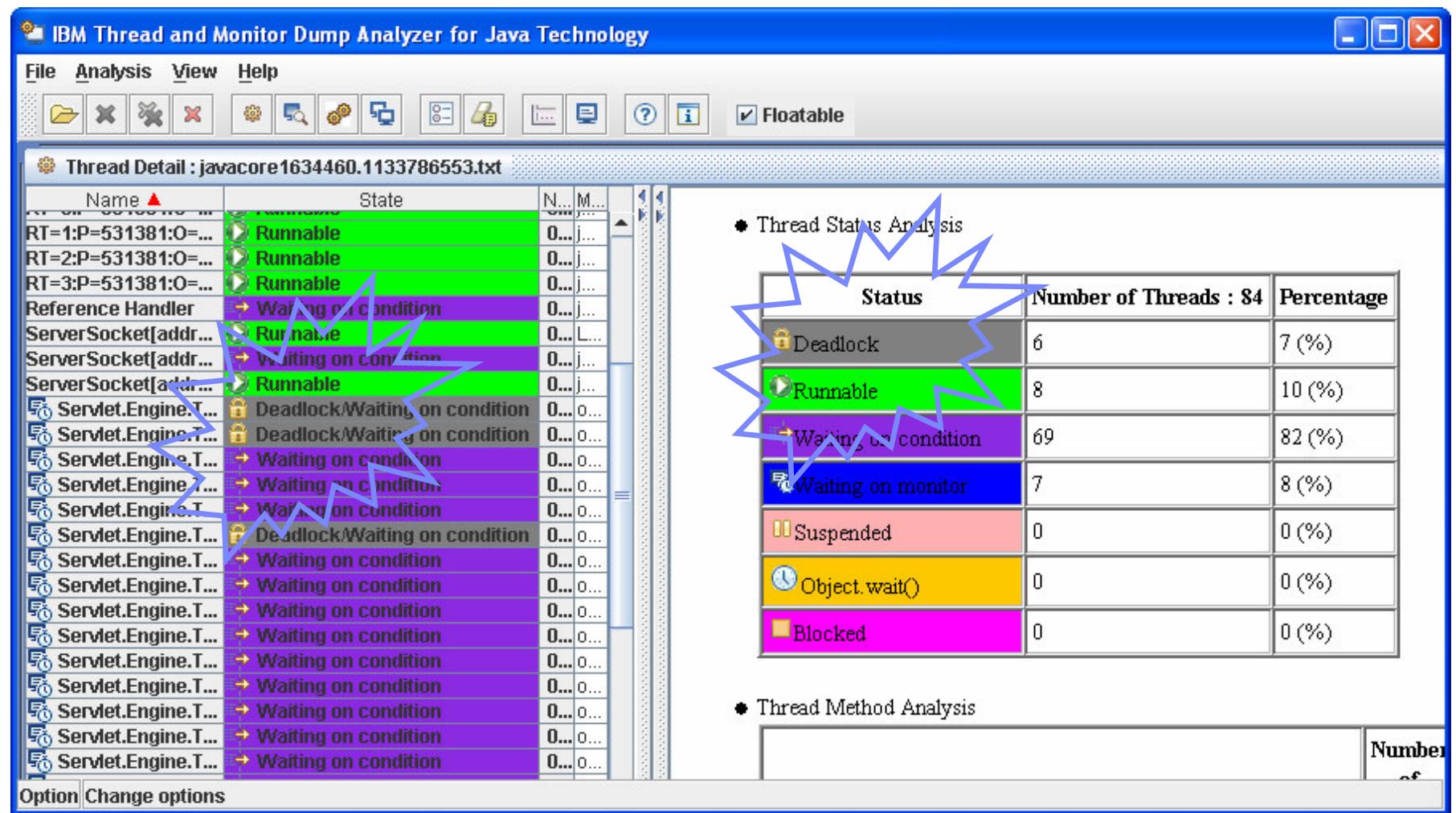
The screenshot shows the "Monitor Detail" view of the IBM Thread and Monitor Dump Analyzer. The title bar reads "Monitor Detail : jwacore103968.1147375649.txt". The main window displays a table of threads and their monitor locks.

Thread Name	Servlet.Engine.Transports : 16
State	Deadlock/Waiting on condition
Monitor	Waiting for Monitor Lock on com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher@A47660A8/A47 Owns Monitor Lock on com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher@A476F4D0/A47 8
	at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.include(WebAppRequestDispatcher.java(Compiled Code)) at com.ibm.wps.services.dispatcher.DispatcherServiceImpl.handleRequest(DispatcherServiceImpl.java(Compiled Code)) at com.ibm.wps.services.dispatcher.DispatcherServiceImpl.include(DispatcherServiceImpl.java(Compiled Code))

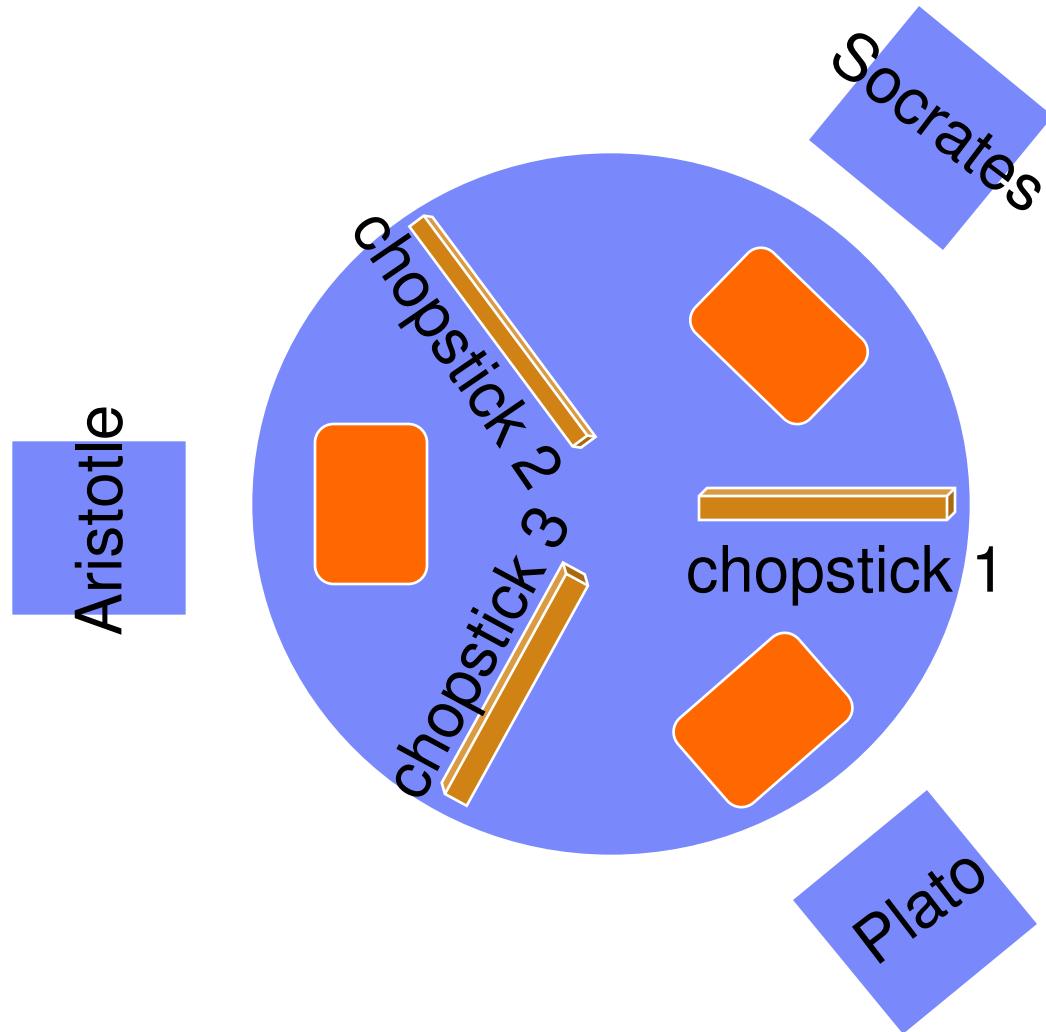
The left pane shows a tree view of thread names and their monitor locks, with many entries for "Servlet.Engine.Transports" threads. A blue starburst diagram is overlaid on the left pane, pointing towards the tree view. Another blue starburst diagram is overlaid on the right pane, pointing towards the table rows.



# Deadlock in Thread Detail View



## Example : Starving Philosophers



# Sample Class Chopstick

// Creator: Jinwoo Hwang © Copyright IBM Corporation 2008. All rights reserved.

```
public class Chopstick {  
    Philosopher owner = null;  
    String name = null;  
    public Chopstick(String name){  
        this.name = name;  
    }  
    public String getName(){  
        return name;  
    }  
    public synchronized void pickUp(Philosopher user) throws InterruptedException{  
        while (owner != null){  
            wait(1000);  
        }  
        if (owner == null) {  
            owner = user;  
        }  
    }  
  
    public synchronized void putDown(Philosopher user) {  
        if (user == owner){  
            owner = null;  
            notify();  
        }  
    }  
}
```



# Sample Class Philosopher

// Creator: Jinwoo Hwang © Copyright IBM Corporation 2008. All rights reserved.

```
public class Philosopher extends Thread {  
    Chopstick leftChopstick,rightChopstick;  
    String name=null;  
    int howLong=0,t=0;  
    public Philosopher(String name,Chopstick left, Chopstick right, int priority){  
        this.name = name;  
        this.leftChopstick=left;  
        this.rightChopstick=right;  
        setPriority(priority);  
        setName(name);  
    }  
    public String getPhilosopherName(){  
        return name;  
    }  
    public void run(){  
        while(true){  
            try{  
                synchronized(leftChopstick) {  
                    synchronized(rightChopstick) {  
                        System.out.format("%s is eating with %s and %s.\n",  
                            getPhilosopherName(),leftChopstick.getName(),rightChopstick.getName());  
                        sleep(2000);  
                    }  
                }  
            }catch( InterruptedException ie) {  
            }  
        }  
    }  
}
```



# Sample Class DiningPhilosophers

```
// Creator: Jinwoo Hwang © Copyright IBM Corporation 2008. All rights reserved.
```

```
public class DiningPhilosophers {  
    public static void main(String[] args) {  
  
        Chopstick c1 = new Chopstick("chopstick 1");  
        Chopstick c2 = new Chopstick("chopstick 2");  
        Chopstick c3 = new Chopstick("chopstick 3");  
  
        Philosopher p1 = new Philosopher("Socrates priority5",c1,c2,5);  
        Philosopher p2 = new Philosopher("Plato priority9",c3,c1,9);  
        Philosopher p3 = new Philosopher("Aristotle priority5",c2,c3,5);  
        p1.start();  
        p2.start();  
        p3.start();  
    } }
```



# Running the example

```
C:\java DiningPhilosophers  
Plato priority9 is eating with chopstick 3 and chopstick 1.
```

```
JVMDUMP006I Processing Dump Event "user", detail "" - Please Wait.  
JVMDUMP007I JVM Requesting Java Dump using  
'C:\javacore.20080229.104813.808.txt'  
JVMDUMP010I Java Dump written to C:\javacore.20080229.104813.808.txt  
JVMDUMP013I Processed Dump Event "user", detail "".
```

Plato priority9 is eating with chopstick 3 and chopstick 1.

```
JVMDUMP006I Processing Dump Event "user", detail "" - Please Wait.  
JVMDUMP007I JVM Requesting Java Dump using  
'C:\javacore.20080229.104819.808.txt'  
JVMDUMP010I Java Dump written to C:\javacore.2008022  
9.104819.808.txt  
JVMDUMP013I Processed Dump Event "user", detail "".
```

Plato priority9 is eating with chopstick 3 and chopstick 1.



# Starvation Threads

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore.20080229.104819.808.txt

Name State

Aristotle priority5	Blocked
DestroyJavaVM helper...	Waiting on co...
JIT Compilation Thread	Waiting on co...
Plato priority9	Waiting on co...
Signal Dispatcher	Runnable
Socrates priority5	Blocked

Waiting Threads

Blocked by

● Thread Status Analysis

Status	Number of Threads :	Percentage
Deadlock	0	0 (%)
Runnable	1	17 (%)
Waiting on condition	3	50 (%)
Waiting on monitor	0	0 (%)
Suspended	0	0 (%)
Object.wait()	0	0 (%)
Blocked	2	33 (%)

● Thread Method Analysis

Option Change options

Status	Number of Threads :	Percentage
Deadlock	0	0 (%)
Runnable	1	17 (%)
Waiting on condition	3	50 (%)
Waiting on monitor	0	0 (%)
Suspended	0	0 (%)
Object.wait()	0	0 (%)
Blocked	2	33 (%)

# Blocked Thread Information

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore.20080229.104819.808.txt

Name	State
Aristotle priority5	Blocked
DestroyJavaVM helper...	Waiting on co...
JIT Compilation Thread	Waiting on co...
Plato priority9	Waiting on co...
Signal Dispatcher	Runnable
Socrates priority5	Blocked

Waiting Threads

Blocked by  
Plato priority9

Thread Name	Socrates priority5
State	Blocked
Monitor	Waiting for Monitor Lock on Chopstick@004BEA68/004BEA74
Java Stack	at Philosopher.run(Philosopher.java:30(Compiled Code))
Native Stack	No Native stack trace available

Option Change options



# Blocked Thread

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore.20080229.104819.808.txt

Name	State
Aristotle priority5	Blocked
DestroyJavaVM helper...	Waiting on co...
JIT Compilation Thread	Waiting on co...
Plato priority9	Waiting on co...
Signal Dispatcher	Runnable
Socrates priority5	Blocked

Waiting Threads

Blocked by  
Plato priority9

Thread Name	Plato priority9
State	Waiting on condition
Monitor	Owns Monitor Lock on Chopstick@004BEA68/004BEA74 , Chopstick@004BEA98/004BEAA4
Java Stack	at java/lang/Thread.sleep(Native Method) at java/lang/Thread.sleep(Thread.java:929(Compiled Code)) at Philosopher.run(Philosopher.java:30(Compiled Code))
Native Stack	No Native stack trace available

Option Change options



# Thread Comparison

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Compare Threads : javacore.20080229.104813.808.txt javacore.20080229.104819.808.txt

Thread	Priority	State	Method
Aristotle priority5	5	Blocked	Philosopher.run(Philosopher.java:32(Compiled Code))
DestroyJavaVM	9	Waiting on condition	java/lang/Thread.sleep(Native Method)
JIT Compiler	9	Runnable	com/ibm/misc/SignalDispatcher.waitForSignal(Native Method)
Plato priority9	9	Blocked	Philosopher.run(Philosopher.java:30(Compiled Code))
Signal Dispatcher	9	Runnable	com/ibm/misc/SignalDispatcher.waitForSignal(Native Method)
Socrates priority5	5	Blocked	Philosopher.run(Philosopher.java:32(Compiled Code))

● Number of hang suspects : 4

● List of hang suspects

Thread Name	State	Method
Aristotle priority5	Blocked	Philosopher.run(Philosopher.java:32(Compiled Code))
Plato priority9	Waiting on condition	java/lang/Thread.sleep(Native Method)
Signal Dispatcher	Runnable	com/ibm/misc/SignalDispatcher.waitForSignal(Native Method)
Socrates priority5	Blocked	Philosopher.run(Philosopher.java:30(Compiled Code))

Status



# Monitor Detail of Blocking Thread

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Monitor Detail : javacore.20080229.104813.808.txt

[TotalSize/Size] ThreadName (ObjectName) 1	Plato priority9
↳ [2/2]	Socrates priority5 (Chopstick@004BEA68/004BEAA4)
	Aristotle priority5 (Chopstick@004BEA98/004BEA74)
<b>Thread Name</b>	<b>Plato priority9</b>
<b>State</b>	Waiting on condition
<b>Monitor</b>	Owes Monitor Lock on Chopstick@004BEA98/004BEAA4 , Chopstick@004BEA68/004BEA74
<b>Java Stack</b>	at java/lang/Thread.sleep(Native Method) at java/lang/Thread.sleep(Thread.java:929(Compiled Code)) at Philosopher.run(Philosopher.java:30(Compiled Code))
<b>Native Stack</b>	No Native stack trace available

Status



# Monitor Detail of Blocked Thread

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

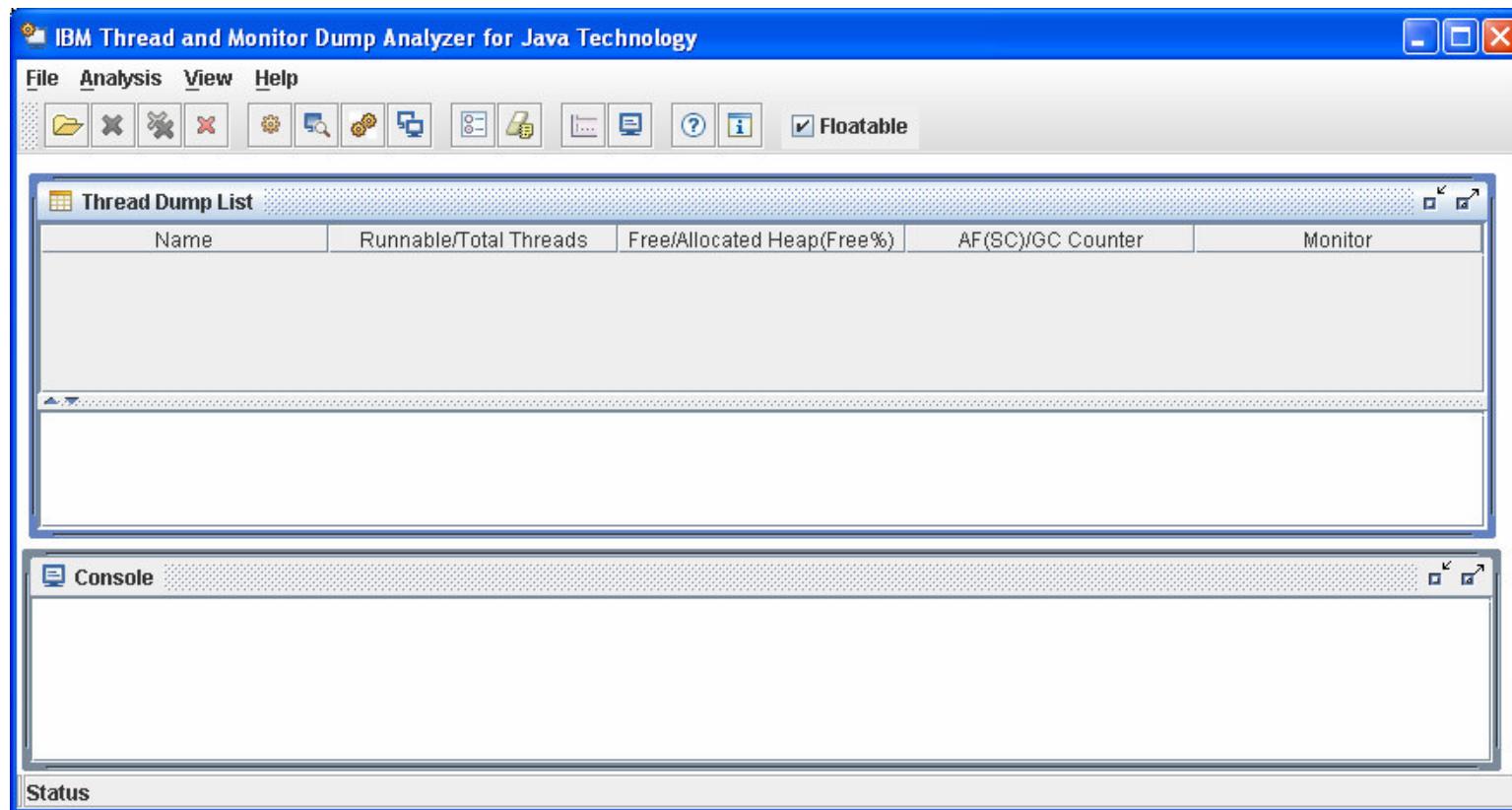
Monitor Detail : javacore.20080229.104813.808.txt

[TotalSize/Size] ThreadName (ObjectName) 1	Thread Name	Socrates priority5
↳ [2/2] Plato priority9	State	Blocked
↳ Socrates priority5 (Chopstick@004BEA68/004BEA74)	Monitor	Waiting for Monitor Lock on Chopstick@004BEA68/004BEA74
↳ Aristotle priority5 (Chopstick@004BEA98/004BEA9C)	Java Stack	at Philosopher.run(Philosopher.java:30(Compiled Code))
	Native Stack	No Native stack trace available

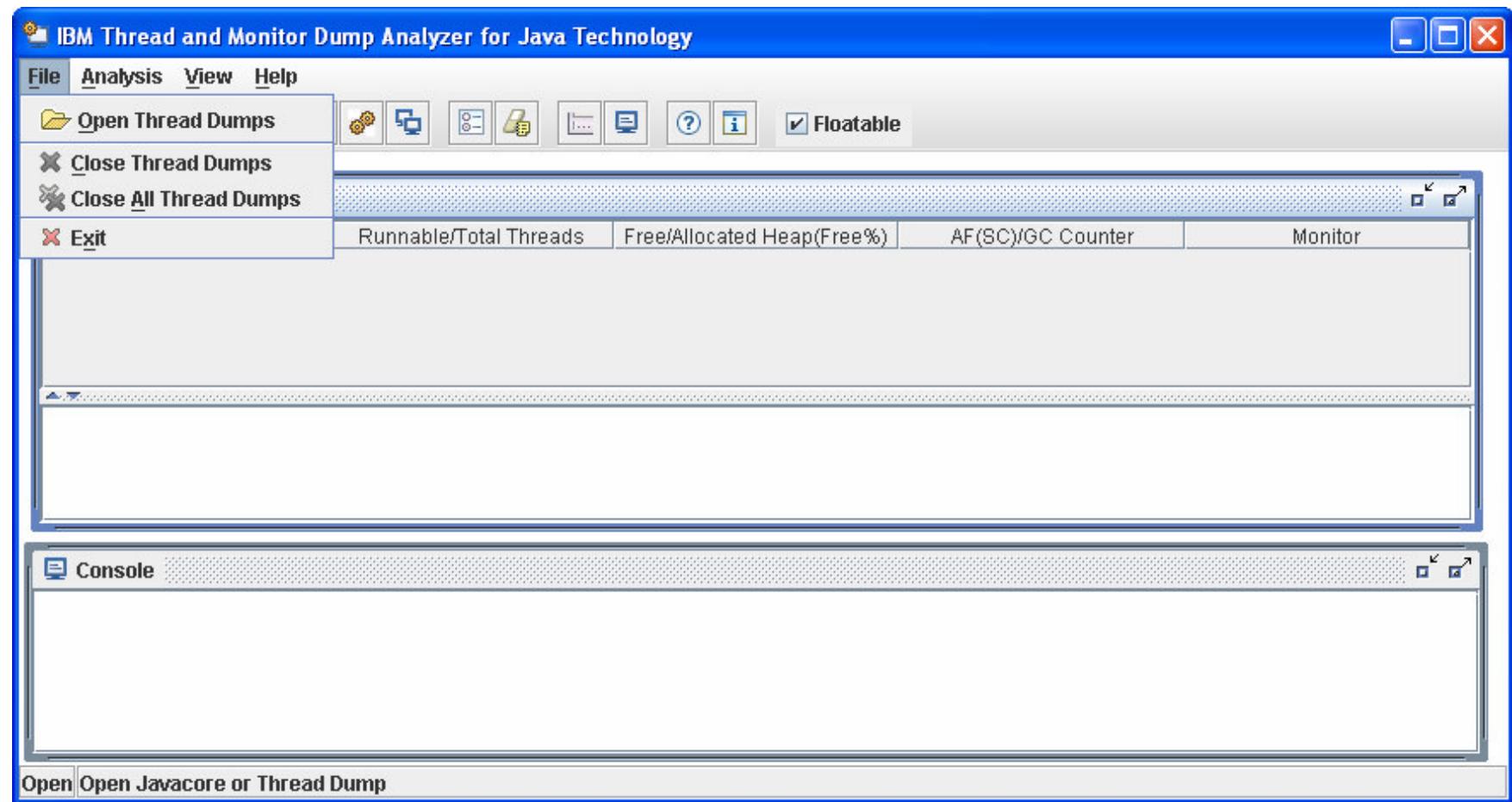
Status



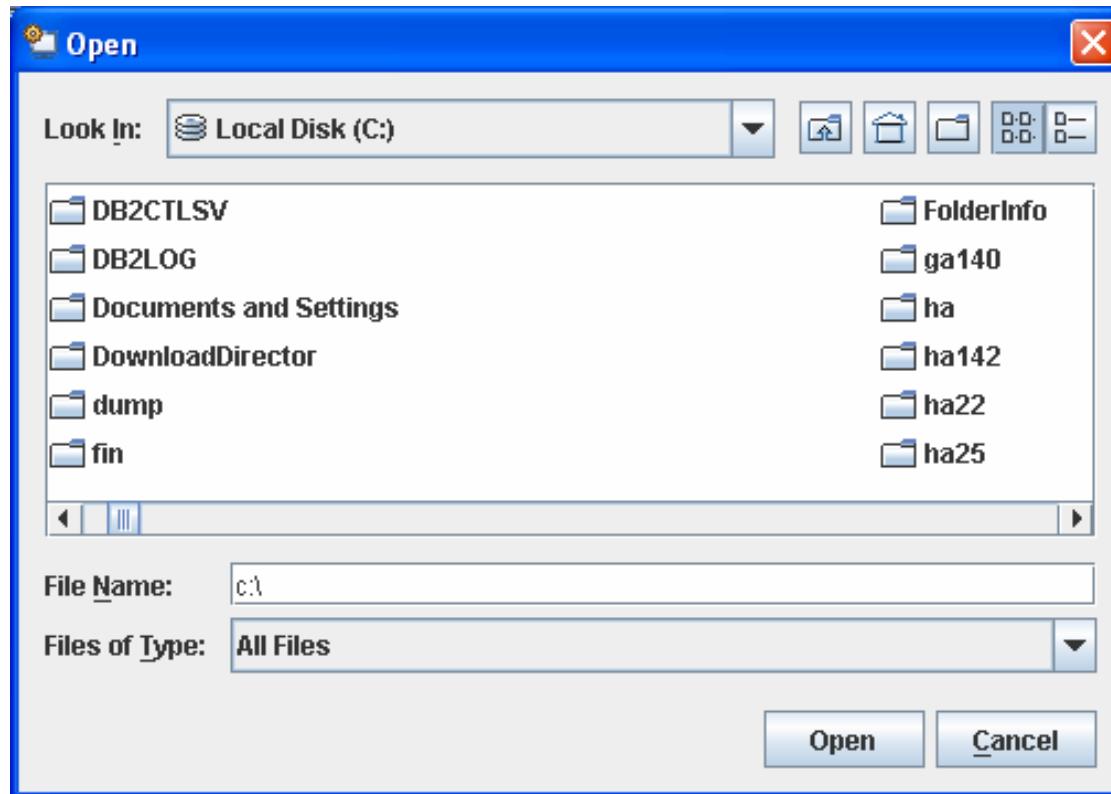
# How do I run IBM Thread and Monitor Dump Analyzer?



# File Menu



# Open a Java thread dump



# Processing completed

IBM Thread and Monitor Dump Analyzer for Java Technology

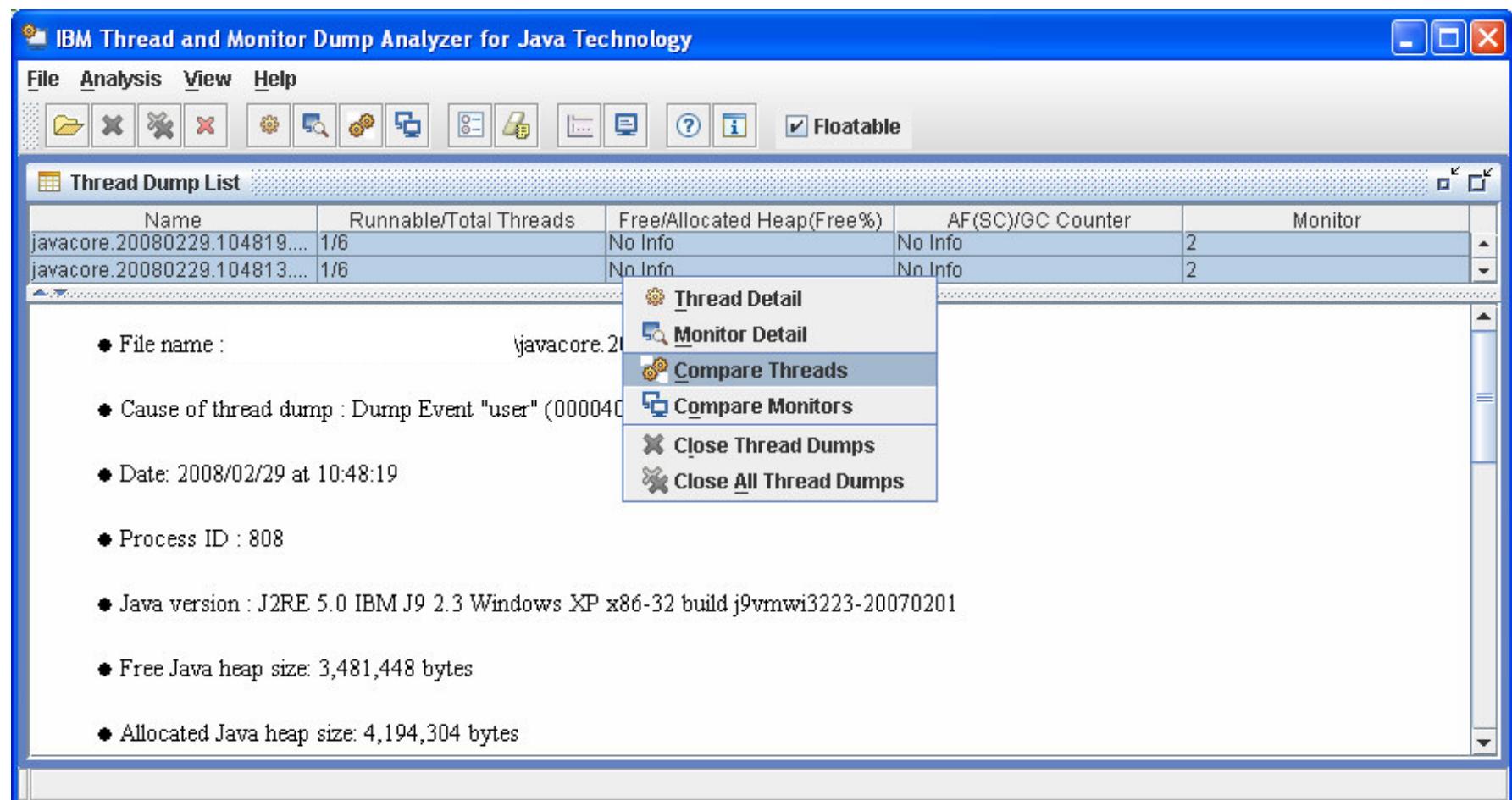
**Thread Dump List**

Name	Runnable/Total Threads	Free/Allocated Heap(Free%)	AF(SC)/GC Counter	Monitor
javacore.20080229.104819.8...	1/6	No Info	No Info	2

● File name : \javacore.20080229.104819.808.txt  
● Cause of thread dump : Dump Event "user" (00004000) received  
● Date: 2008/02/29 at 10:48:19  
● Process ID : 808  
● Java version : J2RE 5.0 IBM J9 2.3 Windows XP x86-32 build j9vmwi3223-20070201  
● Free Java heap size: 3,481,448 bytes  
● Allocated Java heap size: 4,194,304 bytes



# Thread Comparison



# Thread comparison summary

IBM Thread and Monitor Dump Analyzer for Java Technology

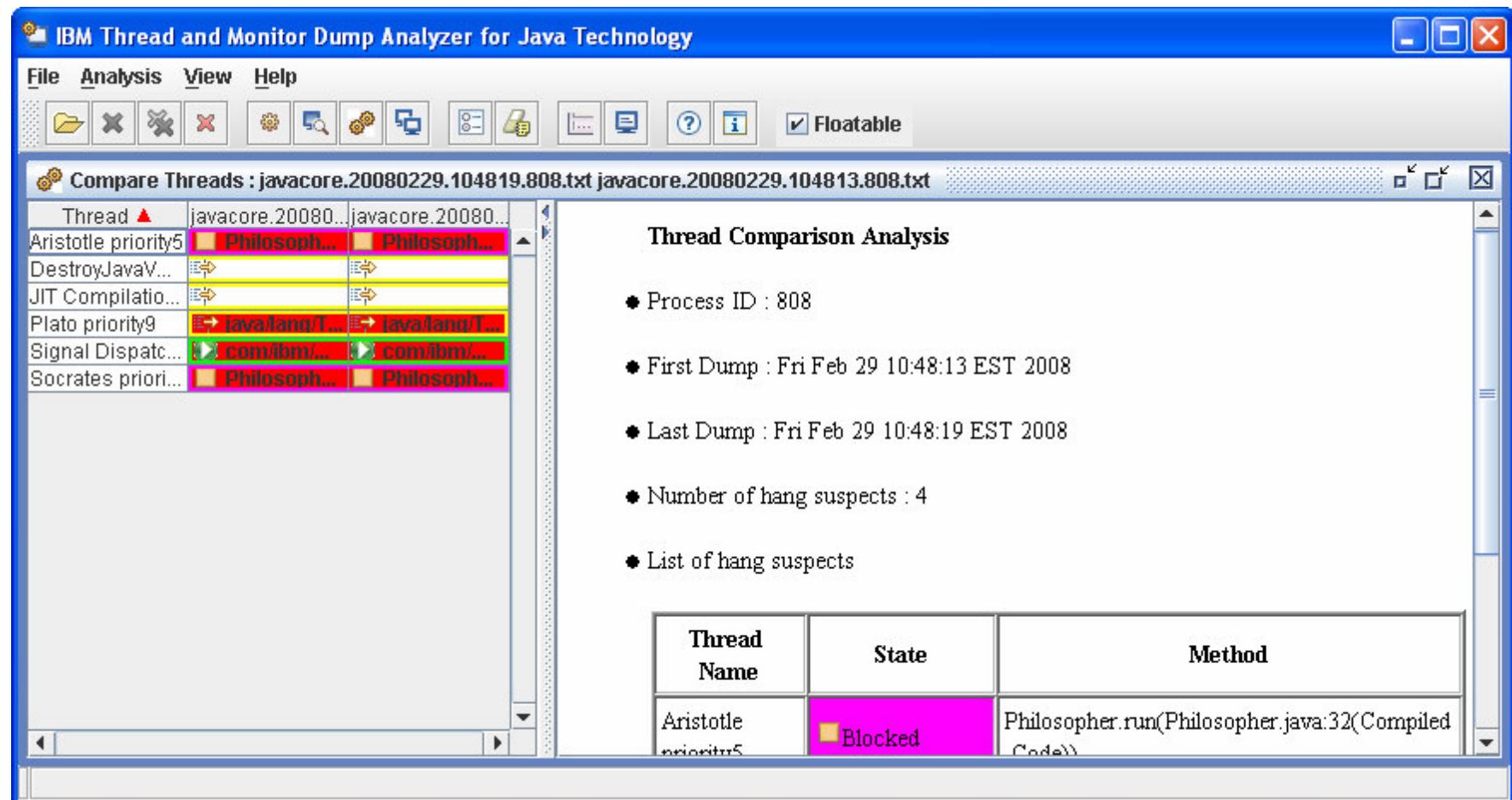
File Analysis View Help

Compare Threads : javacore.20080229.104819.808.txt javacore.20080229.104813.808.txt

Thread Comparison Analysis

- Process ID : 808
- First Dump : Fri Feb 29 10:48:13 EST 2008
- Last Dump : Fri Feb 29 10:48:19 EST 2008
- Number of hang suspects : 4
- List of hang suspects

Thread Name	State	Method
Aristotle priority5	Blocked	Philosopher.run(Philosopher.java:32(Compiled Code))



# Browse threads

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore.20080229.104819.808.txt

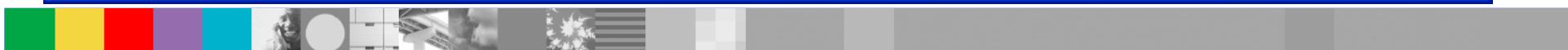
Name	State
Aristotle priority5	Blocked
DestroyJavaVM helper...	Waiting on co...
JIT Compilation Thread	Waiting on co...
Plato priority9	Waiting on co...
Signal Dispatcher	Runnable
Socrates priority5	Blocked

Waiting Threads

Blocked by  
Plato priority9

Thread Name	Socrates priority5
State	Blocked
Monitor	Waiting for Monitor Lock on Chopstick@004BEA68/004BEA74
Java Stack	at Philosopher.run(Philosopher.java:30(Compiled Code))
Native Stack	No Native stack trace available

Option Change options



# Thread Analysis

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Dump List

Name	Runnable/Total Threads	Free/Allocated Heap(Free%)	AF(SC)/GC Counter	Monitor
javacore.20080229.104819....	1/6	No Info	No Info	
javacore.20080229.104813....	1/6	No Info	No Info	

● File name : javacore.20080229.104819.808.txt  
● Cause of thread dump : Dump Event "user" (00004000) received  
● Date: 2008/02/29 at 10:48:19  
● Process ID : 808  
● Java version : J2RE 5.0 IBM J9 2.3 Windows XP x86-32 build j9vmwi3223-20070201  
● Free Java heap size: 3,481,448 bytes  
● Allocated Java heap size: 4,194,304 bytes

Thread Detail  
Monitor Detail  
Compare Threads  
Compare Monitors  
Close Thread Dumps  
Close All Thread Dumps



# Thread Status Analysis

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Floatable

Thread Detail : javacore1634460.1133786553.txt

Name	State	NativeID	Method
PU01stave...	Waiting	0x131d	java.lang....
ProcessDi...	Waiting	0x232c	IDLE
RT=0:P=53...	Runnable	0xa0b	java.net.S...
RT=1:P=53...	Runnable	0x1821	java.net.S...
RT=2:P=53...	Runnable	0x1922	java.net.S...
RT=3:P=53...	Runnable	0x1b24	java.net.S...
Reference...	Waiting	0x203	java.lang....
ServerSoc...	Runnable	0x1216	LISTEN
ServerSoc...	Waiting	0xc1d	java.lang....
ServerSoc...	Runnable	0x151e	java.net.Pl...
Servlet...	Deadlock	0x337c	oracle.jdbc...
Servlet...	Deadlock	0x347d	oracle.jdbc...
Servlet...	Waiting	0x3cad	oracle.jdbc...
Servlet...	Waiting	0x350f	oracle.jdbc...
Servlet...	Waiting	0x2a14	oracle.jdbc...
Servlet...	Deadlock	0x2e1c	oracle.jdbc...
Servlet...	Waiting	0x321f	oracle.jdbc...
Servlet...	Waiting	0x3e28	oracle.jdbc...
Servlet...	Waiting	0xd2b	oracle.jdbc...
Servlet...	Waiting	0x142c	oracle.jdbc...
Servlet...	Waiting	0x262d	oracle.jdbc...
Servlet	Waiting	0x2c2e	oracle.jdbc...

Thread Status Analysis

Status	Number of Threads : 84	Percentage
Deadlock	6	7 (%)
Runnable	8	10 (%)
Waiting on condition	69	82 (%)
Waiting on monitor	7	8 (%)
Suspended	0	0 (%)
Object.wait()	0	0 (%)
Blocked	0	0 (%)

Thread Method Analysis

Option Change options



# Thread Method Analysis

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Floatable

Thread Detail : javacore103968.1147375649.txt

Name	State	NativeID	Method
MessageF...	Waitin...	0x596a	java.lang....
Notificatio...	Waitin...	0x619a	IDLE
ORB.threa...	Waitin...	0x3e44	IDLE
ORB.threa...	Waitin...	0x164a	IDLE
ORB.threa...	Waitin...	0x697e	IDLE
ORB.threa...	Waitin...	0x6c7f	IDLE
ORB.threa...	Waitin...	0x6d80	IDLE
PoolScave...	Waitin...	0x3f48	java.lang....
ProcessDi...	Waitin...	0x6677	IDLE
RT=0:P=98...	Runna...	0x2021	SOCKET ...
RT=10:P=9...	Runna...	0xc1fd	SOCKET ...
RT=11:P=9...	Runna...	0xc3ff	java.net.S...
RT=12:P=9...	Runna...	0xc400	SOCKET ...
RT=13:P=9...	Runna...	0xc502	java.net.S...
RT=14:P=9...	Runna...	0xc603	java.net.S...
RT=1:P=98...	Runna...	0x2122	java.net.S...
RT=2:P=98...	Runna...	0x2224	SOCKET ...
RT=3:P=98...	Runna...	0x3b41	SOCKET ...
RT=4:P=98...	Runna...	0x3c42	java.net.S...
RT=5:P=98...	Runna...	0x3d43	java.net.S...
RT=6:P=98...	Runna...	0x5667	SOCKET ...
RT=7:P=98...	Runna...	0x5768	java.net.S...

● Thread Method Analysis

Method Name	Number of Threads
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.include (WebAppRequestDispatcher.java(Compiled Code))	49
java.lang.Object.wait(Native Method)	38
IDLE	19
java.net.SocketInputStream.socketRead0(Native Method)	16
java.lang.Thread.sleep(Native Method)	12
java.net.PlainSocketImpl.socketAccept(Native Method)	8
SOCKET READ	7
NO JAVA STACK	3



# Thread Aggregation Analysis

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore103968.1147375649.txt

Name	State	NativeID	Method
MessageF...	Waitin...	0x596a	java.lang...
Notificatio...	Waitin...	0x619a	IDLE
ORB.threa...	Waitin...	0x3e44	IDLE
ORB.threa...	Waitin...	0x164a	IDLE
ORB.threa...	Waitin...	0x697e	IDLE
ORB.threa...	Waitin...	0x6c7f	IDLE
ORB.threa...	Waitin...	0x6d80	IDLE
PoolScave...	Waitin...	0x3f48	java.lang...
ProcessDi...	Waitin...	0x6677	IDLE
RT=0:P=98...	Runna...	0x2021	SOCKET ...
RT=10:P=9...	Runna...	0xc1fd	SOCKET ...
RT=11:P=9...	Runna...	0xc3ff	java.net.S...
RT=12:P=9...	Runna...	0xc400	SOCKET ...
RT=13:P=9...	Runna...	0xc502	java.net.S...
RT=14:P=9...	Runna...	0xc603	java.net.S...
RT=1:P=98...	Runna...	0x2122	java.net.S...
RT=2:P=98...	Runna...	0x2224	SOCKET ...
RT=3:P=98...	Runna...	0x3b41	SOCKET ...
RT=4:P=98...	Runna...	0x3c42	java.net.S...
RT=5:P=98...	Runna...	0x3d43	java.net.S...
RT=6:P=98...	Runna...	0x5667	SOCKET ...

● Thread Aggregation Analysis

Thread Type	Number of Threads : 168	Percentage
Servlet.Engine.Transports :	50	30 (%)
Thread	45	27 (%)
Alarm	18	11 (%)
ORB.thread.pool	5	3 (%)
ServerSocket	3	2 (%)
SoapConnectorThreadPool	3	2 (%)

● \*\*\*WARNING\*\*\* Deadlock detected in

[Servlet.Engine.Transports : 49] [Servlet.Engine.Transports : 30]  
[Servlet.Engine.Transports : 48] [Servlet.Engine.Transports : 14]  
[Servlet.Engine.Transports : 47] [Servlet.Engine.Transports : 45]  
[Servlet.Engine.Transports : 46] [Servlet.Engine.Transports : 43]

Status

# Thread Dump Summary

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore1634460.1133786553.txt

Name	State	NativeID	Method
PoolScavenge	Waiting	0x1f1d	java.lang...
ProcessDi...	Waiting	0x232c	IDLE
RT=0:P=53...	Running	0xa0b	java.net.S...
RT=1:P=53...	Running	0x1821	java.net.S...
RT=2:P=53...	Running	0x1922	java.net.S...
RT=3:P=53...	Running	0x1b24	java.net.S...
Reference...	Waiting	0x203	java.lang...
ServerSoc...	Running	0x1216	LISTEN
ServerSoc...	Waiting	0xc1d	java.lang...
ServerSoc...	Running	0x151e	java.net.PI...
Servlet...	Deadlock	0x337c	oracle.jdbc...
Servlet...	Deadlock	0x347d	oracle.jdbc...
Servlet...	Waiting	0x3cad	oracle.jdbc...
Servlet...	Waiting	0x350f	oracle.jdbc...
Servlet...	Waiting	0x2a14	oracle.jdbc...
Servlet...	Deadlock	0xe1c	oracle.jdbc...
Servlet...	Waiting	0x321f	oracle.jdbc...
Servlet...	Waiting	0x3e28	oracle.jdbc...
Servlet...	Waiting	0xd2b	oracle.jdbc...
Servlet...	Waiting	0x142c	oracle.jdbc...
Servlet...	Waiting	0x262d	oracle.jdbc...
Servlet	Waiting	0x2c2e	oracle.jdbc...

File name : javacore1634460.1133786553.txt  
Cause of thread dump : signal 3 received  
Date: 2005/12/05 at 13:42:33  
Process ID : 1634460  
Java version : J2RE 1.3.1 IBM AIX build ca131-20041210  
Java Heap Information  
Maximum Java heap size : 768m  
Initial Java heap size : 128m  
GC Counter: 617  
AF Counter: 615

Option Change options



# Memory Segment Analysis

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore.20080229.104819.808.txt

Memory Segment Analysis

Memory Type	# of Segments	Used Memory( bytes)	Used Memory( %)	Free Memory( bytes)	Free Memory( %)	Total Memory( bytes)
Internal	6	263,804	67.09	129,412	32.91	393,216
Object	1	4,194,304	100	0	0	4,194,304
Class	22	1,354,496	82.55	286,232	17.45	1,640,728
JIT Code Cache	1	0	0	524,288	100	524,288
JIT Data Cache	1	12,500	2.38	511,788	97.62	524,288
Overall	31	5,825,104	80.05	1,451,720	19.95	7,276,824

Option Change options



# Thread Monitors

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Floating

Thread Detail : javacore1634460.1133786553.txt

Name	State	NativeID	Method
Servlet...	Waitin...	0x43f9	oracle.jdbc...
Servlet...	Waitin...	0x44fa	oracle.jdbc...
Servlet...	Waitin...	0x45fb	oracle.jdbc...
Servlet...	Waitin...	0x47fc	oracle.jdbc...
Servlet...	Waitin...	0x48fd	oracle.jdbc...
Servlet...	Waitin...	0x4afe	oracle.jdbc...
Servlet...	Waitin...	0x4bff	oracle.jdbc...
Servlet...	Waitin...	0x4c00	oracle.jdbc...
Servlet...	Waitin...	0x4d01	oracle.jdbc...
Servlet...	Waitin...	0x4e02	oracle.jdbc...
Servlet...	Waitin...	0x4f03	oracle.jdbc...
Servlet...	Waitin...	0x5004	oracle.jdbc...
Servlet...	Waitin...	0x5105	oracle.jdbc...
Servlet...	Waitin...	0x5206	oracle.jdbc...
Servlet...	Waitin...	0x5307	oracle.jdbc...
Servlet...	Waitin...	0x5408	oracle.jdbc...
Servlet...	Waitin...	0x5509	oracle.jdbc...
Servlet...	Deadlo...	0x2dde	oracle.jdbc...
Servlet...	Deadlo...	0x2fe0	oracle.jdbc...
Signal ...	Runna...	0x102	NO JAVA ...
SoapConn...	Waitin...	0x2029	IDLE
SoapConn...	Waitin...	0x212a	IDLE
SoapConn...	Waitin...	0x222b	IDLE

Waiting Threads

- Alarm : 1
- Alarm : 4
- Alarm : 5
- LT=0:P=531381:O=0:port=35575
- LT=1:P=531381:O=0:port=9812
- Thread-12
- Thread-2

Blocked by

Thread Name	Signal dispatcher
State	Runnable
Monitor	Owns Monitor Lock on Heap lock
Java Stack	No Java stack trace available
Native Stack	No Native stack trace available

Option Change options



# Thread stack trace

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

Thread Detail : javacore1634460.1133786553.txt

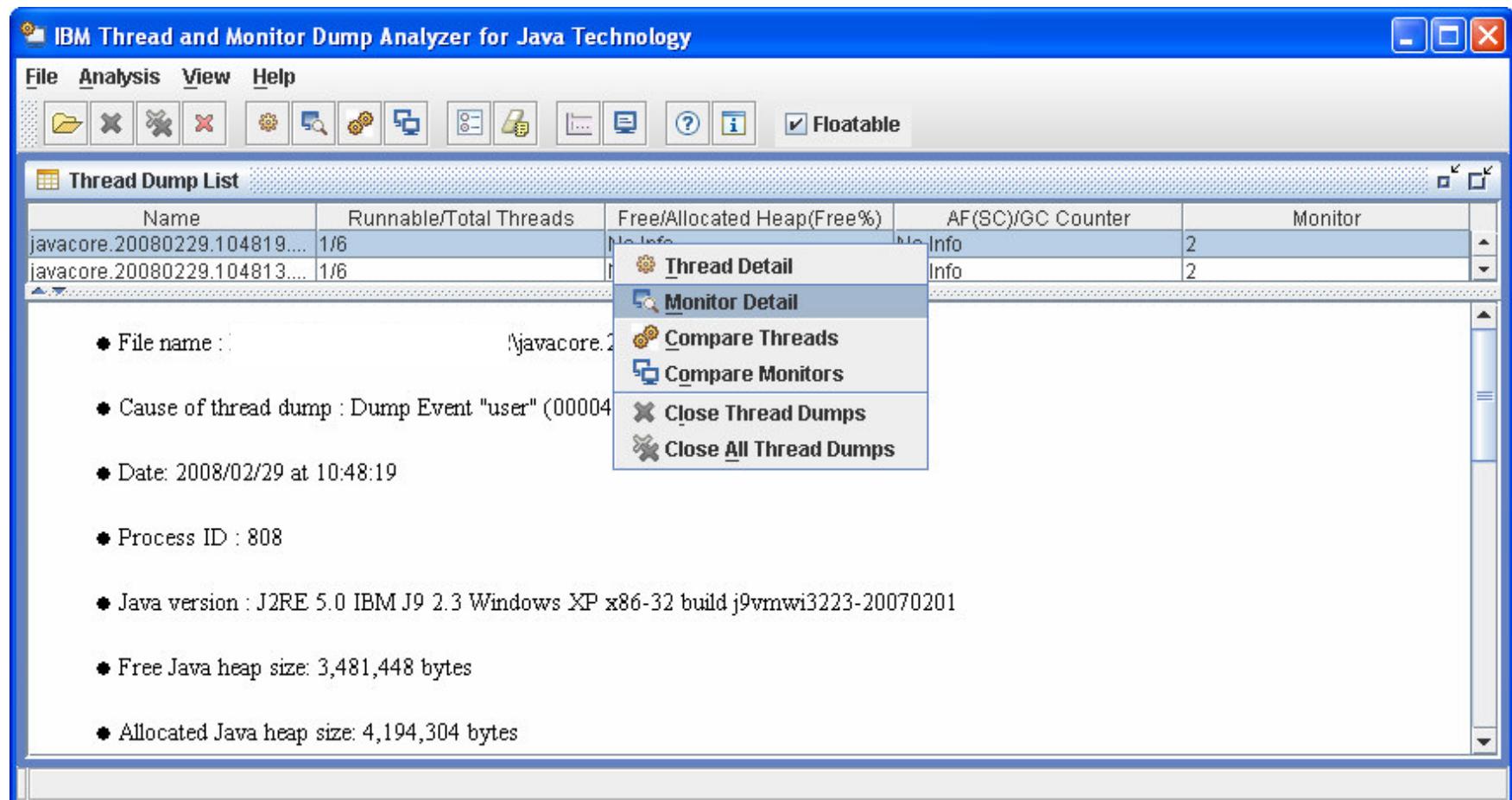
Name	State	NativeID	Method
Servlet...	Waitin...	0x48fd	oracle.jdbc...
Servlet...	Waitin...	0x4afe	oracle.jdbc...
Servlet...	Waitin...	0x4bff	oracle.jdbc...
Servlet...	Waitin...	0x4c00	oracle.jdbc...
Servlet...	Waitin...	0x4d01	oracle.jdbc...
Servlet...	Waitin...	0x4e02	oracle.jdbc...
Servlet...	Waitin...	0x4f03	oracle.jdbc...
Servlet...	Waitin...	0x5004	oracle.jdbc...
Servlet...	Waitin...	0x5105	oracle.jdbc...
Servlet...	Waitin...	0x5206	oracle.jdbc...
Servlet...	Waitin...	0x5307	oracle.jdbc...
Servlet...	Waitin...	0x5408	oracle.jdbc...
Servlet...	Waitin...	0x5509	oracle.jdbc...
Servlet...	Deadlo...	0x2dde	oracle.jdbc...
Servlet...	Deadlo...	0x2fe0	oracle.jdbc...
Signal ...	Runna...	0x102	NO JAVA ...
SoapConn...	Waitin...	0x2029	IDLE
SoapConn...	Waitin...	0x212a	IDLE
SoapConn...	Waitin...	0x222b	IDLE
Thread-11	Waitin...	0x2733	java.lang....
Thread...	Waitin...	0x293a	com.ibm.e...

Thread Name	SoapConnectorThreadPool : 0
State	Waiting on condition
Java Stack	at java.lang.Object.wait(Native Method) at java.lang.Object.wait(Object.java(Compiled Code)) at com.ibm.ws.util.BoundedBuffer.take(BoundedBuffer.java(Compiled Code)) at com.ibm.ws.util.ThreadPool.getTask(ThreadPool.java(Compiled Code)) at com.ibm.ws.util.ThreadPool\$Worker.run(ThreadPool.java:669)
Native Stack	at 0xD0069AB0 in _cond_wait at 0xD005D72C in _event_wait at 0xD0069658 in _cond_wait_local at 0xD0069AB0 in _cond_wait at 0xD006A560 in pthread_cond_wait at 0xD09EF90C in condvarWait at 0xD09EE5EC in sysMonitorWait at 0xD4EC6808 in lkMonitorWait

Option Change options



# Monitor Analysis



# Monitor Analysis

IBM Thread and Monitor Dump Analyzer for Java Technology

File Analysis View Help

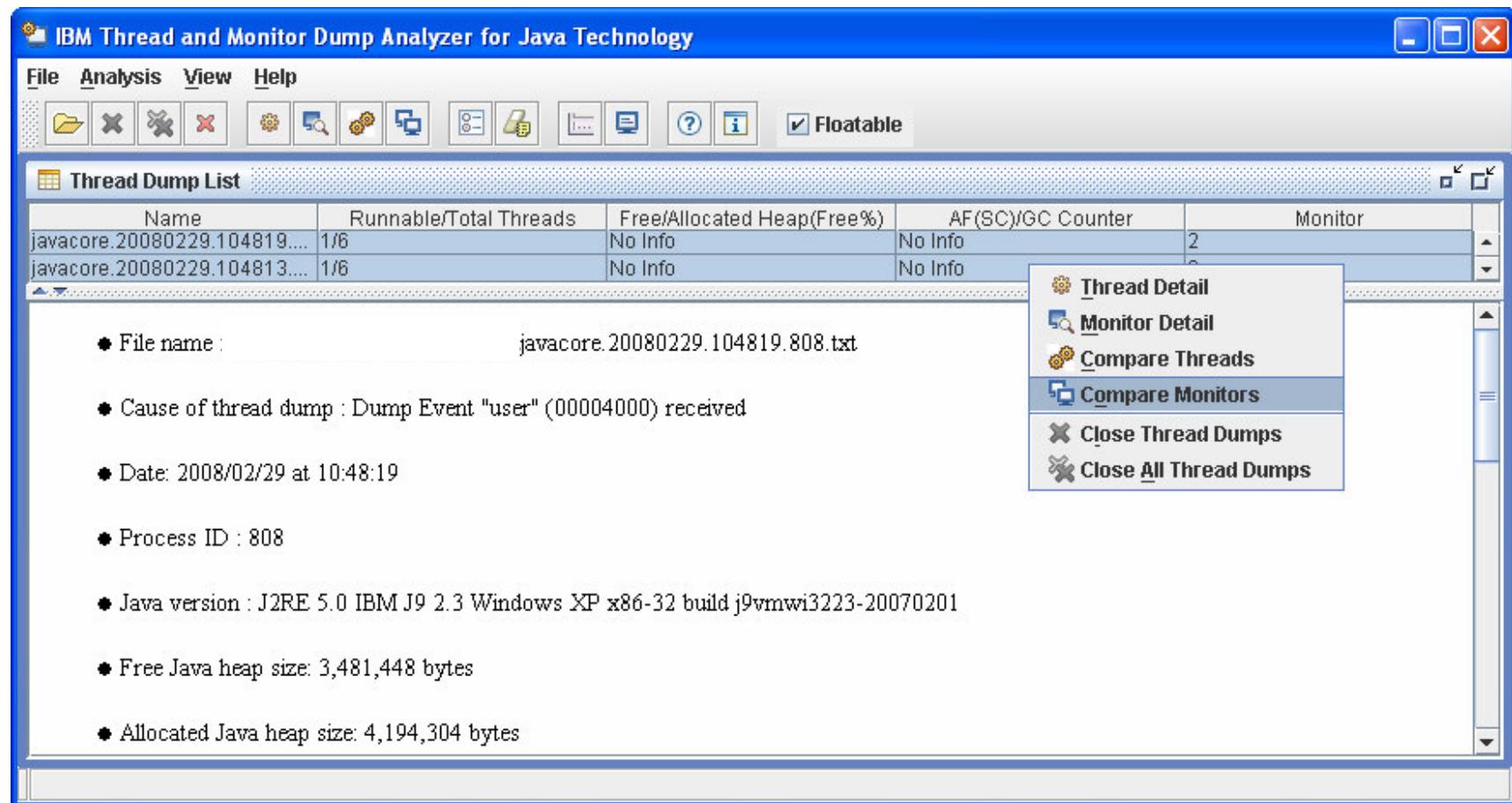
Monitor Detail : javacore.20080229.104819.808.txt

[TotalSize/Size] ThreadName (ObjectName) 1  
[2/2] Plato priority9  
  Socrates priority5 (Chopstick@004BEA68/004BEAA4)  
  Aristotle priority5 (Chopstick@004BEA98/004BEAA4)

Thread Name	Aristotle priority5
State	Blocked
Monitor	Waiting for Monitor Lock on Chopstick@004BEA98/004BEAA4
Java Stack	at Philosopher.run(Philosopher.java:32(Compiled Code))
Native Stack	No Native stack trace available



# Monitor Dump Comparison



# Monitor Dump Comparison

IBM Thread and Monitor Dump Analyzer for Java Technology

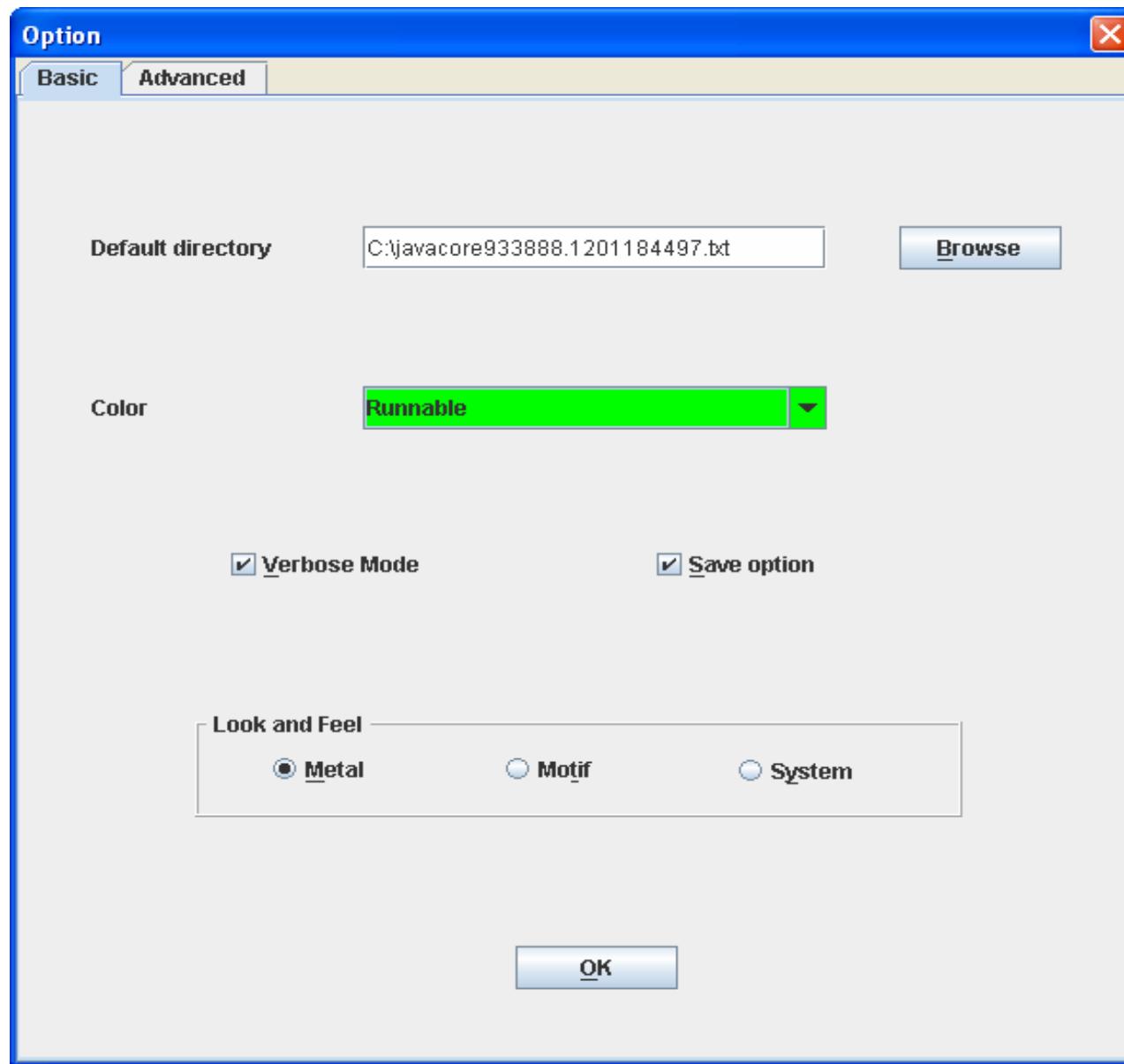
File Analysis View Help

Compare Monitors : javacore.20050822.103956.1996.txt javacore.20050822.104423.1996.txt

Thread Name	Servlet.Engine.Transpo
State	Waiting on condition
Monitor	Owns Monitor Lock on java.util.HashSet@1309F8
	at java.lang.Object.wait(Native Method) at java.lang.Object.wait(Object.java:443) at com.ibm.ws.util.BoundedBuffer.put(BoundedBu at com.ibm.ws.util.ThreadPool.execute(ThreadPo at com.ibm.ws.util.ThreadPool.execute(ThreadPo at com.ibm.ws.management.event.NotificationDisp AllListeners.dispatch(NotificationDispatcher.java:1 at com.ibm.ws.management.event.NotificationDisp



# Option Basic



# Option Advanced

**Option**

Basic Advanced

ID List	socketread	<u>Delete</u>
ID(No spaces)	socketread	
Stack Name	SOCKET READ	
Stack Trace	at java.net.SocketInputStream.read at com.ibm.rmi.iiop.Connection.readMoreData at com.ibm.rmi.iiop.Connection.createInputStream at com.ibm.rmi.iiop.Connection.doReaderWorkOnce at com.ibm.rmi.transport.ReaderThread.run	 
Description	Transport socket reader thread	

New    Add    Update

**NOTE : Thread dumps should be reopened after patterns are updated or added**



## For more information

- IBM® Thread and Monitor Dump Analyzer for Java™ Technology (<http://www.alphaworks.ibm.com/tech/jca>)
- IBM HeapAnalyzer  
(<http://www.alphaworks.ibm.com/tech/heapanalyzer>)
- IBM Pattern Modeling and Analysis Tool for Java Garbage Collector  
(<http://www.alphaworks.ibm.com/tech/pmat>)
- IBM SDK Diagnostics Guide (<http://www-128.ibm.com/developerworks/java/jdk/diagnosis/>)



# Questions and Answers

