

# Dump java active threads

There were some cases when users reported sipXivr or sipXconfig eating lots of CPU. Dumping active threads could really help developers to figure out where the problem lies. For getting a dump with active threads you have to:

1. Identify the pid for the java process eating CPU cycles (e.g. 3788)

2. Open 2 sessions as root:

- in the first session issue:  
`cat /proc/3788/fd/1`
- in the 2nd session issue (mind that this will terminate the process)  
`kill -3 3788`

3. Provide the thread dump within the first console, e.g.:

```
cat /proc/3788/fd/1
2011-03-10 10:25:56
Full thread dump OpenJDK Client VM (14.0-b16 mixed mode):

"P2-19" prio=10 tid=0x09480400 nid=0x14ea in Object.wait() 0x6dac1000
java.lang.Thread.State: TIMED_WAITING (on object monitor)
at java.lang.Object.wait(Native Method)
- waiting on <0x76e16df8> (a org.mortbay.util.ThreadPool$PoolThread)
at org.mortbay.util.ThreadPool$PoolThread.run(ThreadPool.java:522)
- locked <0x76e16df8> (a org.mortbay.util.ThreadPool$PoolThread)

"P2-18" prio=10 tid=0x0947ec00 nid=0x14e9 in Object.wait() 0x6db12000
java.lang.Thread.State: TIMED_WAITING (on object monitor)
at java.lang.Object.wait(Native Method)
- waiting on <0x76e16e80> (a org.mortbay.util.ThreadPool$PoolThread)
at org.mortbay.util.ThreadPool$PoolThread.run(ThreadPool.java:522)
- locked <0x76e16e80> (a org.mortbay.util.ThreadPool$PoolThread)
```

Note:

In latest versions you will need to:

1. Edit /etc/init.d/sipxivr and change line 47 to:

```
runuser -s /bin/bash sipx -c "${Command}" >/var/log/sipxpbx/ivr-dump.log 2>&1 &
```

instead of

```
runuser -s /bin/bash sipx -c "${Command}" >/dev/null 2>&1 &
```

2. Issue service sipxivr restart

3. Get process ID:

```
service sipxivr status -----> find the PID
```

4. kill -3 {PID}

5. Check the output in /var/log/sipxpbx/ivr-dump.log