

Remotely monitoring a sipXecs Java application

Remotely monitoring a sipXecs Java application

This page documents the steps required to remotely monitor a sipXecs Java application.

This HowTo assumes that you have a JDK installed on the client machine.

Step 1: Enable remote monitoring on the Java application

On the target sipXecs server where the Java application resides one needs to instruct the associated Java Virtual Machine (JVM) to allow remote monitoring. In this example the sipXpage application is being monitored.

Edit file `/usr/bin/sipxpage.sh`.

```
=> At the beginning of the file add those lines:

# Parameters required to enable jconsole access
JConsoleParams="\
-Dcom.sun.management.jmxremote.port=23458 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false"

=> Right after line:
JavaCmd=`/usr/bin/sipx-config --java`

=> Add this line:
JavaCmd="$JavaCmd $JConsoleParams"
```

Then restart the application to start using the new JVM parameters.

```
sipxproc -r PageServer
```

If the remote server has different internal and external IPs, RMI will bind to the local one and remote connections will not work. This is the case, for example, when running in AWS. In these cases, an extra parameter is needed in the startup script:

```
-Djava.rmi.server.hostname=<hostname_not_ip>-Djava.rmi.server.hostname=<hostname_not_ip>
```

Step 2: Use JConsole to monitor

Before we start JConsole one can download additional plugins. There is a good one that can be used to monitor CPU usage per thread. You can download it from [\[here\]](http://blog.luminis.nl/luminis/resource/peter/topthreads.jar)<http://blog.luminis.nl/luminis/resource/peter/topthreads.jar>.

Given the target system IP address and port number assigned to the JVM one can start JConsole as follows:

```
jconsole -pluginpath topthreads.jar <IP address or hostname>:23458 &
```