

Runtime issues

{scrollbar}

Most of the issues you may find when running Geronimo will be at start up time; and most likely driven by some conflicting resources from the environment where Geronimo is set up.

JVM arguments

Apache Geronimo v2.2 is Java EE certified. With that said, it will likely run on different JVM versions however the results may be unpredictable. Whenever possible use jdk 1.5 or above.

Another common problem related to Java is that, sometimes, certain environment variables are not defined at Java installation time. For instance, Geronimo requires **JAVA_HOME** and **JRE_HOME** to be defined before running the server. As a convenience, make sure you also add <JAVA_HOME>/bin directory to the system **PATH**.

You can choose one of following options to use JVM arguments depending on how you're starting Geronimo

1. startup.sh or geronimo.sh run
 - set JAVA_OPTS environment variable: solid jeffchi@Local:~/geronimo-tomcat6-javaee5-2.2-SNAPSHOT\$ export JAVA_OPTS="-Xmx256m -XX:MaxPermSize=128m -XX:+HeapDumpOnOutOfMemoryError" jeffchi@Local:~/geronimo-tomcat6-javaee5-2.2-SNAPSHOT\$./bin/geronimo.sh run
 - and/or append the following code to <Geronimo_Home>/bin/setjavaenv.sh(bat) file: if [-z "\$JAVA_OPTS"]; then JAVA_OPTS="-Xmx256m -XX:MaxPermSize=128m -XX:+HeapDumpOnOutOfMemoryError" fi
2. start-server or gsh geronimo/start-server
 - edit <Geronimo_Home>/etc/rc.d/start-server, default.groovy file: // Append some reasonable java flags if none were configured already if (command.javaFlags.empty) { command.javaFlags << '-Xmx256m' command.javaFlags << '-XX:MaxPermSize=128m' command.javaFlags << '-XX:+HeapDumpOnOutOfMemoryError' }
 - or use -J flag: Solid jeffchi@Local:~/geronimo-tomcat6-javaee5-2.2-SNAPSHOT\$./bin/gsh geronimo/start-server -J "-Xmx256m -XX:MaxPermSize=128m -XX:+HeapDumpOnOutOfMemoryError"

Port conflicts

The second most common startup issue is associated to port conflicts, check no other application is using or blocking Geronimo's default ports:

Service	Port
SMTPPort	25
COSNamingPort	1050
NamingPort	1099
ORBSSLPort	2001
OpenEJBPort	4201
ORBPort	6882
AJPPort	8009
HTTPPort	8080
HTTPSPort	8443
JMXPort	9999
ActiveMQStompPort	61613
ActiveMQPort	61616

If you identify port conflicts you can use the <GERONIMO_HOME>/var/config/config-substitutions.properties to change any of these ports. From this configuration file you can also set a port offset and have all these increased by that amount.

Keep also in mind that personal firewalls, anti virus and spyware protection products may block some of these ports as well, even if you turn off such software sometimes those "rules" are still in effect.

Refer to the [Initial configuration](#) section for additional details on prerequisites and different configurations.

Spring version conflicts

If your application contains its own version of Spring you might see some problems deploying or running the application on the Jetty assembly. The Jetty assembly is by default configured with Apache CXF as the JAX-WS provider. Apache CXF uses Spring to configure itself. Sometimes, the Spring version used by CXF conflicts with the Spring version supplied with your application. To prevent these conflicts add the following <hidden-classes> entry to the Geronimo deployment descriptor:

```
xml <hidden-classes> <filter>org.springframework</filter> <filter>META-INF/spring</filter> </hidden-classes>
```

java.lang.UnsatisfiedLinkError: lic (Library is already loaded in another ClassLoader)

This can be caused by the same jar (such as an Oracle Driver using OCI) being loaded in two separate class loaders, each trying to load a system library.

To avoid the problem, you may try [Adding JARs to the Geronimo repository](#) and define the dependency in your deployment plan.

See Also : <https://issues.apache.org/jira/browse/GERONIMO-4629>

Enable multicasting for IPv4 network

To use multicasting properly in a clustered environment, you will need to set the system property `java.net.preferIPv4Stack` to `true` when starting Geronimo.

Administration console unaccessbile by HTTPs in Internet Explorer 6

If you navigate your browser Internet Explorer (IE) 6 to the console via `https://_hostname_:8443/console` where *hostname* is replaced with the host name where the Geronimo server is running, you will get an error "The page cannot be displayed". To remedy this problem, click **Tools->Internet Options**, and select the **Advanced** tab. In the **Security** section, select the checkbox **TLS 1.0**, and you will be prompted with a Security Alert. Click **Yes** to view the administrative console. If you want to see information about the certificate, click **View Certificate**. If you want IE to accept WASCE CA permanently, you can install the certification in IE.

java.io.IOException of remote EJB on Windows

`javax.naming.NamingException: Cannot lookup '/MyBeanRemote'. [Root exception is java.rmi.RemoteException: Cannot connect to server 'ejbd://localhost:4201'; nested exception is: java.io.IOException: Cannot connect to server: 'ejbd://localhost:4201'. Exception: java.net.BindException : Address already in use: connect] at org.apache.openejb.client.JNDIContext.lookup(JNDIContext.java:214) at javax.naming.InitialContext.lookup(Unknown Source) at MyClientScalabilityTest$MyClientTest.call(MyClientScalabilityTest.java:86) at MyClientScalabilityTest$MyClientTest.call(MyClientScalabilityTest.java:1) at java.util.concurrent.FutureTask$Sync.innerRun(Unknown Source) at java.util.concurrent.FutureTask.run(Unknown Source) at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(Unknown Source) at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source) at java.lang.Thread.run(Unknown Source) Caused by: java.rmi.RemoteException: Cannot connect to server 'ejbd://localhost:4201'; nested exception is: java.io.IOException: Cannot connect to server: 'ejbd://localhost:4201'. Exception: java.net.BindException : Address already in use: connect`

One possibility is that the available user port numbers are being exhausted. On Windows, when a socket is closed, it goes into a `TIME_WAIT` state and isn't actually closed until some delay time. By default, the max user port address is 5000 and the `TIME_WAIT` delay is 4 minutes. So, it's not too difficult to exhaust all possible user port addresses.

You have to update the Windows Registry to change these values. Here's a Windows 2000 doc on the registry settings – <http://technet.microsoft.com/en-us/library/bb726981.aspx>

- `MaxUserPorts` controls the upper range for user ports.
- `TcpTimedWaitDelay` controls the `TIME_WAIT` delay.

Increasing `MaxUserPorts` (e.g. 65534) and decreasing `TcpTimedWaitDelay` (e.g. 30) may fix the problem.

Parse XML error when deploying an EJB security jar

When you deploy an EJB application with security constraint on Geronimo server, A parsing exception will show if you use a default namespace in the deployment plan. The problem is caused by a known [JAXB issue](#). To workaround this problem, you have to add explicit named namespaces to every element or redefine the default namespace in each element in `open-ejb.xml` file.

Server failed to start due to ActiveMQ module failure

Most likely Geronimo server was not shut down cleanly, which results in ActiveMQ log files to have an unexpected entry. You can avoid this by editing `var/activemq/conf/activemq.xml` and adding `schedulerSupport="false"` if you don't require JobScheduler support:

```
xml:activemq.xml <broker xmlns="http://activemq.apache.org/schema/core" brokerName="${activemq.brokerName}" tmpDataDirectory="${activemq.data}"/>
/tmp_storage" useShutdownHook="false" start="false" schedulerSupport="false">
```