

How to enable SSL (for HTTPS) on JBoss 3.2.5

A definitive step by step guide

dnickull@adobe.com

Preface:

In order to get Adobe liveCycle Policy Server working, Secure Socket Layers (SSL) must be enabled on the application server. The default installation for testing is often JBoss 3.2.5 but no current documentation exists to guide developers through the SSL enablement process. This guide was written to enable anyone to quickly get it done.

Environment:

This "How to" was done using the following configuration:

Macintosh MacBook Pro, Mac OS/X build 10.4.7*

JBoss 3.2.5 downloaded from jboss.org

Java version "1.5.0_06"

Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_06-112)

Java HotSpot(TM) Client VM (build 1.5.0_06-64, mixed mode, sharing)

Instructions

1. Install the latest JBoss application server.
2. To test if it is working, you must navigate to the ~/install_directory/bin and execute the run shell script by typing in

```
sh run.sh
```

The terminal window will tell you its progress as it starts up. Navigate to <http://localhost:8080> to make sure it installed correctly.

3. Open a 2nd terminal and generate a key. To make this go quickly, simply cut and paste the following text into your terminal window:

```
keytool -genkey -alias jboss -keypass rmi+ssl -keyalg RSA -keystore duane.keystore -  
validity 3650
```

You will be asked a few questions. Just answer them then hit enter.

```

duane-nickulls-computer:/Applications/jboss-3.2.5/bin nickull$ keytool -genkey -
alias:4jboss -keypass:rmi+ssl -keyalg:RSA -keystore:duane:keystore -validity 365
00:51:32,750 INFO [UILServerILService] JBossMQ UIL service available at : /0.0.
Enter keystore password: rmi+ssl
What is your first and last name? JNDI name: queue/DLQ
10[Unknown]: IEDBU [TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=file:/A
What is the name of your organizational unit? JmsXA to JNDI name 'java:/JmsXA'
10[Unknown]: IEDBU [TomcatDeployer] deploy, ctxPath=/web-console, warUrl=file:/A
What is the name of your organization? /deploy/jmx-console.war/
10[Unknown]: IAdobe Systems deploy, ctxPath=/web-console, warUrl=file:/A
What is the name of your City or Locality? /deploy/tmp753web-console.war/
10[Unknown]: IVancoverver JBoss (MX MicroKernel) [3.2.5 (build: CVSTag=JBoss_3
What is the name of your State or Province? s
10[Unknown]: IBC0 [Tomcat5] Saw org.jboss.system.server.started notification, s
What is the two-letter country code for this unit? CA
10[Unknown]: ICA0 [Http11Protocol] Starting Coyote HTTP/1.1 on http-0.0.0.0-808
Is CN=Duan Nickull, OU=EDBU, O=Adobe Systems, L=Vancouver, ST=BC, C=CA correct?
10[no]:3.955 INFO [ChannelSocket] JK2: ajp13 listening on /0.0.0.0:8009
10:51:33,958 INFO [JKMain] JK running ID=0 time=0/25 config=null
duane-nickulls-computer:/Applications/jboss-3.2.5/bin nickull$ _

```

4. To test to make sure your key got generated, type in the following:

Prompt> keytool -list -v

The v flag is for Verbose to give you the details. You will be prompted for the password you gave earlier. In this example it was “rmi+ssl”

NOTE: It is important to use the “-keystore <keystore>” flag to specify which keystore you wish to use. By default, the Keytool will use the default keystore.

```
duane-nickulls-computer:/ nickull$ keytool -list -v -keystore duane
Enter keystore password: rmi+ssl

Keystore type: jks
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: duane-key
Creation date: Sep 13, 2006
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Duan Nickull, OU=EDBU, O=Adobe, L=Vancouver, ST=BC, C=CA
Issuer: CN=Duan Nickull, OU=EDBU, O=Adobe, L=Vancouver, ST=BC, C=CA
Serial number: 45088135
Valid from: Wed Sep 13 15:07:49 PDT 2006 until: Tue Dec 12 14:07:49 PST 2006
Certificate fingerprints:
    MD5: F6:B3:D1:1A:8A:C5:CA:32:35:F2:EB:1E:F6:96:0B
    SHA1: 2E:F0:4D:41:FD:B0:3A:D7:4A:C2:AF:A8:1B:20:7D:74:D1:53:3F:97

*****
*****
To test to make sure your key got generated, type in the f
Prompt> keytool -list -v

duane-nickulls-computer:/ nickull$
```

You now have a key to use.

5. The next step is to copy the keystore (duane.keystore) to the correct directory where it will be called during runtime. You can do this with Finder or via the terminal. If you do it via the terminal, simply 'cp' it to:

`$JBOSS_HOME/server/default/ssl/`

Note that the "ssl" directory will not be there by default so you must create it. You should see the following structure in your Finder:

▼ jboss-3.2.5	Today, 2:21 PM
▶ bin	Today, 2:10 PM
▶ client	Today, 10:50 AM
▶ docs	Today, 10:50 AM
▶ jar-versions.xml	25-Jun-04, 10:28 PM
▶ lib	Today, 10:50 AM
▼ server	Today, 10:51 AM
▶ all	25-Jun-04, 10:22 PM
▼ default	Today, 2:21 PM
▶ conf	Today, 10:51 AM
▶ data	Today, 10:51 AM
▶ deploy	Today, 11:16 AM
▶ lib	Today, 10:50 AM
▶ log	Today, 10:51 AM
▼ ssl	Today, 2:10 PM
duane.keystore	Today, 2:10 PM
▶ tmp	Today, 2:22 PM
▶ work	Today, 11:40 AM
▶ minimal	25-Jun-04, 10:22 PM

6. The next step is to modify the server.xml file within the \$JBOSS_HOME/server/default/deploy/jbossweb-tomcat50.sar folder. This is the server.xml file that must tell the application server where to find the keystore and how to access it during startup.

▼ jboss-3.2.5	Today, 2:21 PM
▶ bin	Today, 2:10 PM
▶ client	Today, 10:50 AM
▶ docs	Today, 10:50 AM
▶ jar-versions.xml	25-Jun-04, 10:28 PM
▶ lib	Today, 10:50 AM
▼ server	Today, 10:51 AM
▶ all	25-Jun-04, 10:22 PM
▼ default	Today, 2:21 PM
▶ conf	Today, 10:51 AM
▶ data	Today, 10:51 AM
▼ deploy	Today, 11:16 AM
▶ cache-invalidation-service.xml	25-Jun-04, 10:22 PM
▶ client-deployer-service.xml	Today, 11:21 AM
▶ hsqldb-ds.xml	25-Jun-04, 10:22 PM
▶ http-invoker.sar	Today, 11:22 AM
▶ jboss-jca.sar	25-Jun-04, 10:27 PM
▶ jboss-local-jdbc.rar	25-Jun-04, 10:27 PM
▶ jboss-xa-jdbc.rar	25-Jun-04, 10:27 PM
▼ jbossweb-tomcat50.sar	Today, 11:16 AM
▶ ant.jar	25-Jun-04, 10:27 PM
▶ catalina-manager.jar	25-Jun-04, 10:27 PM
▶ catalina-optional.jar	25-Jun-04, 10:27 PM
▶ catalina.jar	25-Jun-04, 10:27 PM
▶ commons-beanutils.jar	25-Jun-04, 10:27 PM
▶ commons-collections.jar	25-Jun-04, 10:27 PM
▶ commons-digester.jar	25-Jun-04, 10:27 PM
▶ commons-el.jar	25-Jun-04, 10:27 PM
▶ commons-logging.jar	25-Jun-04, 10:27 PM
▶ commons-modeler.jar	25-Jun-04, 10:27 PM
▶ jakarta-regexp-1.3.jar	25-Jun-04, 10:27 PM
▶ jasper-compiler.jar	25-Jun-04, 10:27 PM
▶ jasper-runtime.jar	25-Jun-04, 10:27 PM
▶ jsp-api.jar	25-Jun-04, 10:27 PM
▶ META-INF	Today, 2:13 PM
▶ naming-common.jar	25-Jun-04, 10:27 PM
▶ naming-resources.jar	25-Jun-04, 10:27 PM
▶ ROOT.war	Today, 11:24 AM
▶ server.xml	Today, 2:19 PM
▶ servlet-api.jar	25-Jun-04, 10:27 PM

This is the right file!

- Open the server.xml file with a text editor of your choice and uncomment the block of XML to declare the SSL. This is easy and you should only have to modify two pieces. The first one is the actual path to the keystore file relative to the jboss home directory; the second is the password required for the application server to access the key.

```

<!-- SSL/TLS Connector configuration using the admin devl guide keystore-->
<Connector port="8443" address="${jboss.bind.address}"
    maxThreads="100" minSpareThreads="5" maxSpareThreads="15"
    scheme="https" secure="true" clientAuth="false"
    keystoreFile="${jboss.server.home.dir}/ssl/duane.keystore"
    keystorePass="rmi+ssl" sslProtocol = "TLS" />
<!--end ssl-->

```

Note that the KeystoreURL and the KeyStorePass must be the same as the values you used when you generated the key.

8. Save this file then navigate to your \$JBOSS_HOME/bin directory and start the server. On Mac the syntax is “sh run.sh”. You should see jBoss Start up.

NOTE: By default, I have not entered any specific configuration of JBoss and apply this only to the default jboss instance. If you want to enable ssl on other instances of JBoss, you will have to modify the files under those server directories.

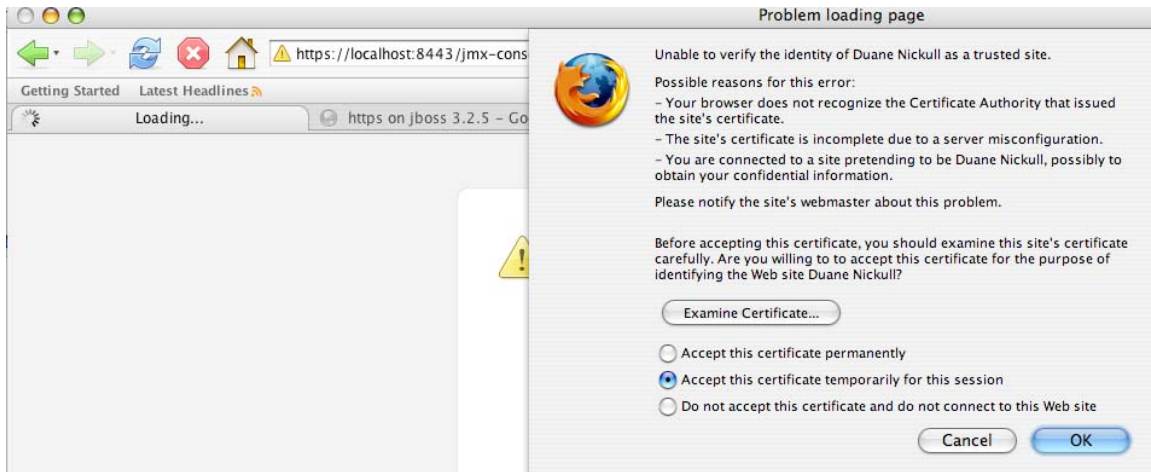
9. One error that may occur due to different class loaders is that the jboss instance will be unable to find the keystore. Examine the resulting trace to the shell to determine the problem. For example, the output below clearly shows that the JBoss server could not find the keystore because it wasn't in the directory where it was looking. Modify the server.xml files as required to help JBoss find the keystore.

```

14:19:52,530 WARN [Tomcat5] Failed to startConnectors
LifecycleException: Protocol handler start failed: java.io.FileNotFoundException: /Applications/jboss-3.2.5/server/default/ssl/duane.keystore (No such file or directory)
    at org.apache.coyote.tomcat5.CoyoteConnector.start(CoyoteConnector.java:1529)

```

Grab your browser and hit <https://localhost:8443>

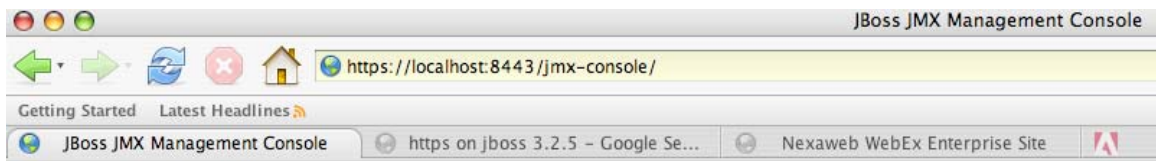


You should see the following message warning that the certificate is a bit dodgy. Since it is your certificate you just made, you can accept it permanently and you shouldn't see this warning any more.

A second warning will appear like the one below. Click "OK".



Voila! JBoss using SSL



JMX Agent View

duane-nickulls-computer.local

ObjectName Filter (e.g. "jboss:*", "*:service=invoker,*"):

Catalina

- [type=Server](#)