cockpit
IT Service Manager

Monitoring - JAVA application access

FAQ document

Table of contents

Introduction

The Cockpit IT Service Manager Engine uses JMX console to monitor JAVA applications. The JMX console is available on the JAVA Virtual Machine.

The purpose of this document is to help technicians start the JMX console on JAVA applications and set up remote access.

Starting the JMX console

Modify the JAVA Virtual Machine startup options to start the JMX console. The startup mode is specific ti each application, you will probably need to refer to the administration documents for each application to change the startup options. The next sections will provide a starting procedure for the JMX console for some known applications.

However, the properties to be included in the startup options remain substantially the same and are described in the table below.

Property	Description	Values
com.sun.management.jmxremote	Enables the JMX remote agent and local monitoring via JMX connector published on a private interface used byjconsole. The jconsole tool can use this connector if it is executed by the same user ID as the user ID that started the agent. No password or access files are checked for requests coming via this connector.	true / false. Default is true.
com.sun.management.jmxremote. port	Enables the JMX remote agent and creates a remote JMX connector to listen through the specified port. By default, SSL, password, and access files properties are used for this connector. Also enables local monitoring as described for thecom.sun.management.jmxremote property.	Port number. No default.
com.sun.management.jmxremote.ssl	Enables secure monitoring via SSL. If false, then SSL is not used.	true / false. Default is true.
com.sun.management.jmxremote.ssl	Comma-delimited list of SSL/TLS protocol versions to enable. Used in conjunction withcom.sun.management.jmxremote.ssl	Default SSL/TLS protocol version.
com.sun.management.jmxremote.ssl.enabled.cipher.suites	A comma-delimited list of SSL/TLS cipher suites to enable. Used in conjunction withcom.sun.management.jmxremote.ssl.	Default SSL/TLS cipher suites.
com.sun.management.jmxremote.ssl.need.client.auth	If this property is true and the property com.sun.management.jmxremote.ssl is true, then client authentication will be performed.	true / false. Default is false
com.sun.management.jmxremote.authenticate	If this property is false then JMX does not use passwords or access files: all users are allowed all access.	true / false. Default is true.
com.sun.management.jmxremote.password.file	Specifies location for password file. If com.sun.management.jmxremote.password is false, then this property and the password and access files are ignored. Otherwise, the password file must exist and be in valid format. If the password file is empty or non-existent, then no access is allowed.	*JRE_HOME*/lib/management/jmxremote.password
com.sun.management.jmxremote.access.file	Specifies location for the access file. If com.sun.management.jmxremote.password is false, then this property and the password and access files are ignored. Otherwise, the access file must exist and be in the valid format. If the access file is empty or	*JRE_HOME*/lib/management/jmxremote.access

	non-existent, then no access is allowed.	
com.sun.management.jmxremote.login.config	Specifies the name of a JAAS login configuration entry to use when authenticating users of RMI monitoring. When using this property to override the default login configuration, the named configuration entry must be in a file that gets loaded by JAAS. In addition, the login modules specified in the configuration should use the name and password callbacks to acquire the user's credentials. If com.sun.management.jmxremote.authenticate is false, then this property and the password and access files are ignored.	Default login configuration is a file-based password authentication.

I. WebLogic

To start the JMX console on a WebLogic application, follow the next steps:

1. Go to the "bin" directory of the domain.
2. Edit the "setDomainEnv.cmd" file, add the following lines above the "CLASSPATH" section.

```
set JAVA_OPTIONS= %JAVA_OPTIONS%
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=XXXX
-Dcom.sun.management.jmxremote.ssl=XXXX
-Dcom.sun.management.jmxremote.authenticate=XXXX
```

3. Restart the WebLogic server.
4. Test the connection.

II. WebSphere

To start the JMX console on a WebSphere application, follow the next steps:

1. Log in to the Administration Console and check if the "PerfServletApp.ear" application is already deployed (context menu => WebSphere Enterpise Applications). If it is not deployed, select "New Application" then "WebSphere directory => AppServer => InstallableApps" to deploy.
2. Enable Performance Monitoring Infrastructure (PMI) data and all associated statistics. To do this, go to "context menu => Monitoring and Tuning => Performance Monitoring Infrastructure". Enable the "PMI" option and statistics at the "Configuration" tab. Also activate the statistics on the "Runtime" tab.
3. Select the server you want to enable the JMX console in "Servers => Server Types => WebSphere Application Servers". In the right frame, select "Server Infrastructure => Java and Process Management", then "Process definition". In the "Additional Properties of Configuration" tab, select "Java Virtual Machine". Add the argument "-Djavax.management.builder.initial = -Dcom.sun.management.jmxremote" in the Generic Jvm Argument field. Save.

4. Edit the "\ AppServer \ java \ jre \ lib \ management \ management.properties" file. Add the lines below.

```
set JAVA_OPTIONS= %JAVA_OPTIONS%
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=XXXX
-Dcom.sun.management.jmxremote.ssl=XXXX
-Dcom.sun.management.jmxremote.authenticate=XXXX
```

5. Restart the WebSphere server.
6. Test the connection.

III. JBoss

To start the JMX console on a JBoss application, follow the next steps:

1. Edit the "run.bat" file located in the "bin" directory of the JBoss tree.
2. Look for JAVA options (JAVA_OPTS) in the file.
3. Add the following options to the existing options.

```
-Djavax.management.builder.initial=org.jboss.system.server.jmx.MBeanServerBuilderImpl
-Djboss.platform.mbeanserver
-Dcom.sun.management.jmxremote.port=XXXX
-Dcom.sun.management.jmxremote.authenticate=XXXX
-Dcom.sun.management.jmxremote.ssl=XXXX
```

4. Restart the JBoss server.
5. Test the connection.

IV. Tomcat

To start the JMX console on a Tomcat application, follow the next steps:

1. Edit the "catalina.bat" file located in the "bin" directory of the Tomcat tree.
2. Look for JAVA options (JAVA_OPTS) in the file.
3. Add the following lines at the beginning of the file.

```
set JAVA_OPTS= %JAVA_OPTS% -Dcom.sun.management.jmxremote.port=XXXX
-Dcom.sun.management.jmxremote.ssl=XXXX
-Dcom.sun.management.jmxremote.authenticate=XXXX
```

4. Restart the Tomcat server.
5. Test the connection.

Ports

The JMX console port must be open to allow the connection between the Cockpit IT Service Manager Engine and the monitored application.
This port is configurable with the start option "com.sun.management.jmxremote.port".

Portal Configuration

Connect the Cockpit IT Service Manager Portal.

To enter connection information, follow the procedure below.

1. Go to the "Infrastructure / Other items / JAVA applications" menu
2. Open the target application (editing mode)
3. Fill in the following fields

Field	Notes
Connection port	
User	
Password	
SSL encryption	

4. Save

Document end