



How to Enable Remote JMX Access to Quartz Schedulers

May 12, 2015



Table of Contents

1.	PURPOSE	3
2.	DEFINITIONS	4
3.	ENABLING REMOTE JMX ACCESS	5
3.1	JMX/RMI	6
3.1.1	Apache Tomcat	6
3.1.2	IBM WebSphere AS	7
3.1.3	Oracle GlassFish AS	8
3.1.4	Oracle WebLogic 11g (10.23.x) and 12c (12.1.x)	9
3.1.5	Standalone Java Application	10
3.2	JMXMP	12
3.3	REMOTING-JMX (JBOSS ONLY)	13
3.3.1	JBoss 6.1.0 Community	13
3.3.2	JBoss 7.1.1 Community	14
3.3.3	JBoss 6.1 EAP	15
3.3.4	JBoss 6.2 EAP	15
3.3.5	JBoss 6.3 EAP	16
4.	EXPORTING QUARTZ SCHEDULER MBEAN	18
5.	VERIFYING REMOTE JMX ACCESS	19
5.1	TOMCAT, WEBSHERE AS, GLASSFISH AS, WEBLOGIC AS, STANDALONE JAVA APPLICATION	19
5.2	JBOSS AS	20



1. Purpose

QuartzDesk relies on the [Java Management Extensions](#) (JMX) access to manage and monitor Quartz schedulers. Enabling JMX access to a Quartz scheduler involves two configuration steps:

1. Enabling remote JMX access to the Platform MBean Server of the JVM the Quartz scheduler is running on (Platform MBean Server is a standard component of all modern JVMs).
2. Exporting the Quartz scheduler management interface (MBean) to the JVM's Platform MBean Server.

This document describes these configuration steps for all currently supported application servers. For the list of all currently supported application servers, please refer to <https://www.quartzdesk.com/documentation/supported-platforms>.

2. Definitions

The following table lists all acronyms and shortcuts used throughout this document.

Acronym / Shortcut	Definition
AS	Application Server.
GAS	GlassFish Application Server.
JVM	Java Virtual Machine.
WLAS	WebLogic Application Server.
WAS	WebSphere Application Server.

The following table lists all locations and properties used throughout this document.

Location / Property	Example	Description
GAS_CONFIG	server-config	GlassFish Application Server configuration.
JAVA_HOME	/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.9.x86_64	Java installation directory.
JBOSS_INSTALL_ROOT	/usr/share/jboss-eap-6.1	JBoss installation directory.
JMX_HOST	myhost.foo.com	JMX IP address / hostname.
JMX_PORT	1099	JMX port number.
TOMCAT_HOME	/usr/share/tomcat7	Tomcat installation directory.
WAS_SERVER_NAME	server1	WebSphere Application Server name.
WL_DOMAIN_HOME	/opt/Oracle/user_projects/domain1	WebLogic domain directory.

3. Enabling Remote JMX Access

QuartzDesk Web Application supports three JMX protocols that can be used to establish a JMX connection with the JVM Platform MBean Server through which it can manage and monitor Quartz schedulers running on that JVM.

From the table below, please choose the protocol that best suits your requirements and configuration and refer to the chapter that describes the protocol-specific configuration details.

Protocol	Description
JMX/RMI	<p>Standard JMX protocol.</p> <p>Works with all application servers but JBoss.</p> <p>Firewall-unfriendly protocol that uses a static and a dynamically allocated communication port (just like the standard Java RMI protocol).</p> <p>Requires multiple JVM system properties to enable.</p> <p>Supports security (SSL + password authentication).</p>
JMXMP	<p>Standard JMX protocol.</p> <p>Works with all application servers.</p> <p>Firewall-friendly protocol that uses a single communication port.</p> <p>No JVM system properties to enable. Enabled in the QuartzDesk JVM Agent configuration.</p> <p>Requires the QuartzDesk JVM Agent (only available for QuartzDesk Standard or Enterprise editions).</p> <p>Security is not currently supported for this protocol by QuartzDesk.</p> <p>Available only in QuartzDesk 1.5.0 and newer.</p>
REMOTING-JMX	<p>JBoss proprietary protocol.</p> <p>Works only on JBoss application servers.</p> <p>Configuration is very similar to the JMX/RMI protocol configuration.</p> <p>Must be used for Quartz schedulers running on JBoss.</p>

	Supports security (SSL + password authentication).
--	--

3.1 JMX/RMI

To enable remote JMX/RMI access to the JVM Platform MBean Server, the JVM must be typically started with the following system properties:

System Property	Example	Description
java.rmi.server.hostname	myserver.foo.com	IP address, or hostname of the server the JVM is running on. This hostname is passed to the JMX client (QuartzDesk) that will use it to communicate with the JVM. Please make sure the hostname is resolvable and reachable from the host the QuartzDesk Web Application is running on. The IP/hostname is referred to as <code>JMX_HOST</code> in this document.
javax.management.builder.initial		Class name of the <code>javax.management.MBeanServerBuilder</code> subclass that is used to create the JMX server instance. If empty, then the standard JMX server is created.
com.sun.management.jmxremote	true	Enables the JMX remote agent. Must be set to 'true'.
com.sun.management.jmxremote.port	1099	Port number used by the JMX client (QuartzDesk) to establish a connection with the JVM. The port number is referred to as <code>JMX_PORT</code> in this document.
com.sun.management.jmxremote.ssl	false	Flag indicating whether the JMX communication should be secured with SSL.
com.sun.management.jmxremote.authenticate	false	Flag indicating whether the client must authenticate to establish a JMX connection.

The following chapters describe the configuration steps to add these JVM properties in all supported application servers.

3.1.1 Apache Tomcat

Add the following JVM system properties to the Tomcat startup script.

3.1.1.1 Windows

Edit TOMCAT_HOME/bin/catalina.bat and add the following lines below the first line:

```
set CATALINA_OPTS=%CATALINA_OPTS% -Djava.rmi.server.hostname=JMX_HOST
set CATALINA_OPTS=%CATALINA_OPTS% -Djavax.management.builder.initial=
set CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote=true
set CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote.port=JMX_PORT
set CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote.ssl=false
set CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false
```

Restart Tomcat for the changes to take effect.

3.1.1.2 Unix / Linux

Edit TOMCAT_HOME/bin/catalina.sh and add the following lines below the first line:

```
CATALINA_OPTS="${CATALINA_OPTS} -Djava.rmi.server.hostname=JMX_HOST"
CATALINA_OPTS="${CATALINA_OPTS} -Djavax.management.builder.initial="
CATALINA_OPTS="${CATALINA_OPTS} -Dcom.sun.management.jmxremote=true"
CATALINA_OPTS="${CATALINA_OPTS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
CATALINA_OPTS="${CATALINA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"
CATALINA_OPTS="${CATALINA_OPTS} -Dcom.sun.management.jmxremote.authenticate=false"
```

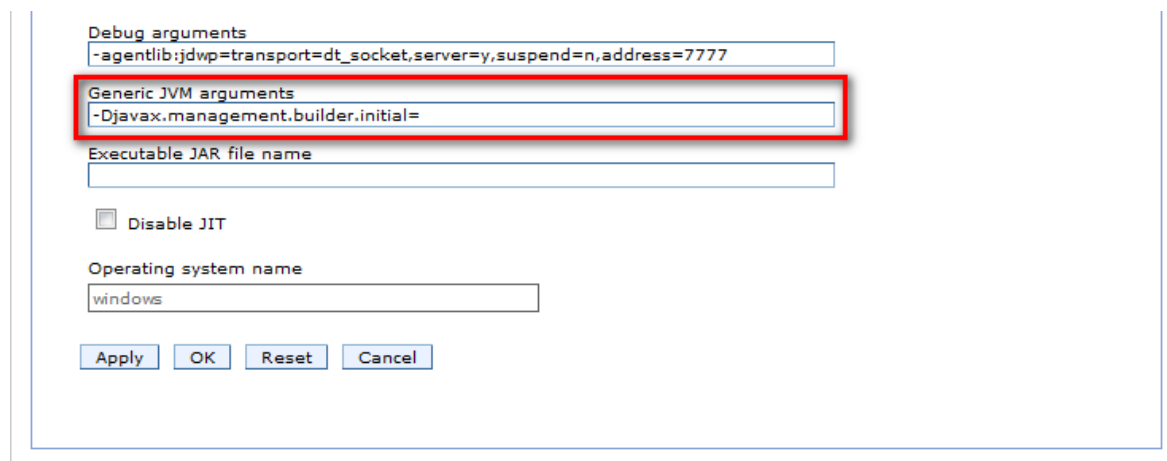
Restart Tomcat for the changes to take effect.

3.1.2 IBM WebSphere AS

In the WebSphere Administrative Console open Servers → Server Types → WebSphere application servers → WAS_SERVER_NAME → Java and Process Management → Process definition → Java Virtual Machine.

Add the following JVM property to the “Generic JVM arguments” .

-Djavax.management.builder.initial=



The screenshot shows the 'Generic JVM arguments' field in the WebSphere Administrative Console. The field contains the text '-Djavax.management.builder.initial=' and is highlighted with a red border. Other fields visible include 'Debug arguments' with '-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777', 'Executable JAR file name', 'Disable JIT' (unchecked), and 'Operating system name' with 'windows' entered. Buttons for 'Apply', 'OK', 'Reset', and 'Cancel' are at the bottom.

In the WebSphere Administrative Console, open Servers → Server Types → WebSphere application servers → WAS_SERVER_NAME → Java and Process Management → Process definition → Java Virtual Machine → Custom properties.

Add the following properties:

Name: java.rmi.server.hostname

Value: JMX_HOST

Name: javax.management.builder.initial

Value: none

Name: com.sun.management.jmxremote

Value: true

Name: com.sun.management.jmxremote.port

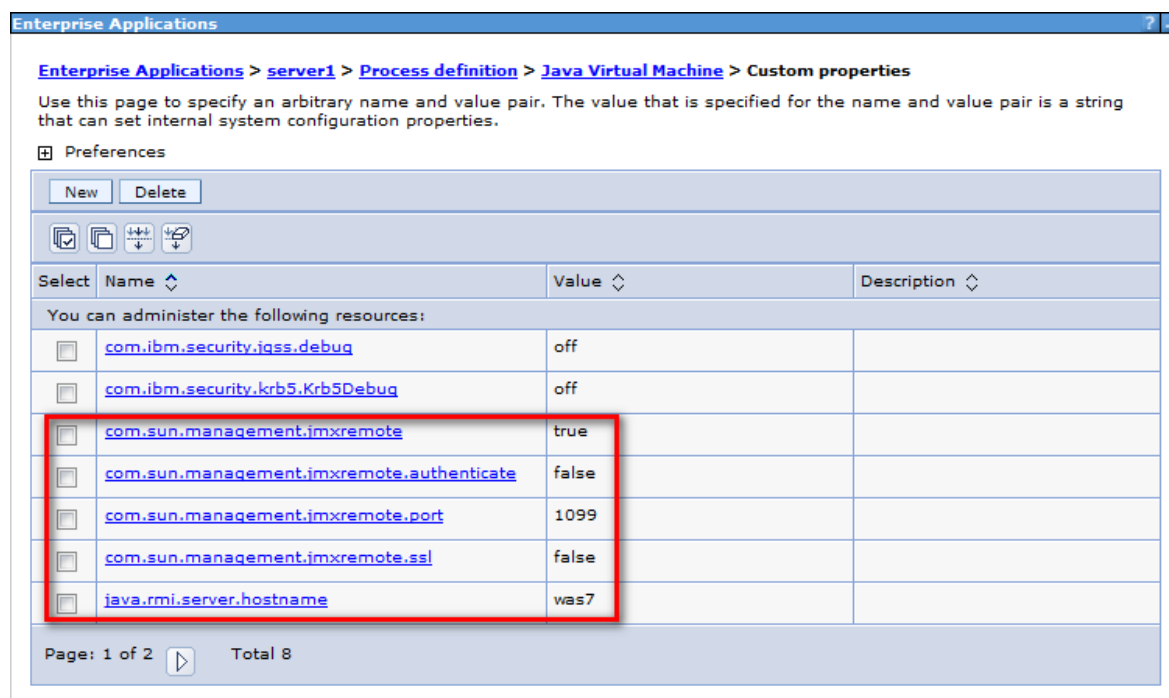
Value: JMX_PORT

Name: com.sun.management.jmxremote.ssl

Value: false

Name: com.sun.management.jmxremote.authenticate

Value: false



Select	Name	Value	Description
<input type="checkbox"/>	com.ibm.security.iqss.debug	off	
<input type="checkbox"/>	com.ibm.security.krb5.Krb5Debug	off	
<input type="checkbox"/>	com.sun.management.jmxremote	true	
<input type="checkbox"/>	com.sun.management.jmxremote.authenticate	false	
<input type="checkbox"/>	com.sun.management.jmxremote.port	1099	
<input type="checkbox"/>	com.sun.management.jmxremote.ssl	false	
<input type="checkbox"/>	java.rmi.server.hostname	was7	

Apply and Save changes.

Restart WAS for the changes to take effect.

3.1.3 Oracle GlassFish AS

In the GlassFish Console, open GAS_CONFIG → JVM Settings → JVM Options.

Add the following JVM options:

Value: -Djava.rmi.server.hostname=JMX_HOST

Value: -Dcom.sun.management.jmxremote=true

Value: `-Dcom.sun.management.jmxremote.port=JMX_PORT`

Value: `-Dcom.sun.management.jmxremote.ssl=false`

Value: `-Dcom.sun.management.jmxremote.authenticate=false`



Do not change the value of the `-Djavax.management.builder.initial` JVM option.

General Path Settings **JVM Options** Profiler

JVM Options

Save

Manage JVM options for the server. Values containing one or more spaces must be enclosed in double quotes ("*value string*").

Configuration Name: server-config

Options (37)

| Add JVM Option Delete

Select	Value
<input type="checkbox"/>	-client
<input type="checkbox"/>	-DANTLR_USE_DIRECT_CLASS_LOADING=true
<input type="checkbox"/>	-Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.config.serverbeans.Appse
<input type="checkbox"/>	-Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as
<input type="checkbox"/>	-Dcom.sun.management.jmxremote.authenticate=false
<input type="checkbox"/>	-Dcom.sun.management.jmxremote.port=1099
<input type="checkbox"/>	-Dcom.sun.management.jmxremote.ssl=false
<input type="checkbox"/>	-Dcom.sun.management.jmxremote=true
<input type="checkbox"/>	-Dfelix.fileinstall.bundles.new.start=true
<input type="checkbox"/>	-Dfelix.fileinstall.bundles.startTransient=true
<input type="checkbox"/>	-Dfelix.fileinstall.dir=\${com.sun.aas.installRoot}/modules/autostart/
<input type="checkbox"/>	-Dfelix.fileinstall.disableConfigSave=false
<input type="checkbox"/>	-Dfelix.fileinstall.log.level=2
<input type="checkbox"/>	-Dfelix.fileinstall.poll=5000
<input type="checkbox"/>	-Dgosh.args=--nointeractive
<input type="checkbox"/>	-Djava.awt.headless=true
<input type="checkbox"/>	-Djava.endorsed.dirs=\${com.sun.aas.installRoot}/modules/endorsed\${path.separator}\${com.sun.aas.installRo
<input type="checkbox"/>	-Djava.ext.dirs=\${com.sun.aas.javaRoot}/lib/ext\${path.separator}\${com.sun.aas.javaRoot}/jre/lib/ext\${path.sep
<input type="checkbox"/>	-Djava.rmi.server.hostname=glassfish7
<input type="checkbox"/>	-Djava.security.auth.login.config=\${com.sun.aas.instanceRoot}/config/login.conf
<input type="checkbox"/>	-Djava.security.policy=\${com.sun.aas.instanceRoot}/config/server.policy
<input type="checkbox"/>	-Djavax.management.builder.initial=com.sun.enterprise.v3.admin.AppServerMBeanServerBuilder
<input type="checkbox"/>	-Djavax.net.ssl.keyStore=\${com.sun.aas.instanceRoot}/config/keystore.jks

Restart GAS for the changes to take effect.

3.1.4 Oracle WebLogic 11g (10.23.x) and 12c (12.1.x)

Add the following JVM system properties to the WebLogic domain environment initialization script.

3.1.4.1 Windows

Edit `WL_DOMAIN_HOME/bin/setDomainEnv.cmd` and add the following lines at the end of the file:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Djava.rmi.server.hostname=JMX_HOST
set JAVA_OPTIONS=%JAVA_OPTIONS% -
Djavax.management.builder.initial=weblogic.management.jmx.mbeanserver.WLSMBeanServerBuilder
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.port=JMX_PORT
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.ssl=false
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.authenticate=false
```

Restart the WLAS for the changes to take effect.

3.1.4.2 Unix / Linux

Edit `WL_DOMAIN_HOME/bin/setDomainEnv.cmd` and add the following lines at the end of the file:

```
JAVA_OPTIONS="{JAVA_OPTIONS} -Djava.rmi.server.hostname=JMX_HOST"
JAVA_OPTIONS="{JAVA_OPTIONS} -
Djavax.management.builder.initial=weblogic.management.jmx.mbeanserver.WLSMBeanServerBuilder"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.authenticate=false"
export JAVA_OPTIONS
```

Restart the WLAS for the changes to take effect.

3.1.5 Standalone Java Application

Edit the standalone Java application's startup script and add the following JVM system properties to the list of command line options passed to the JVM the application is started with:

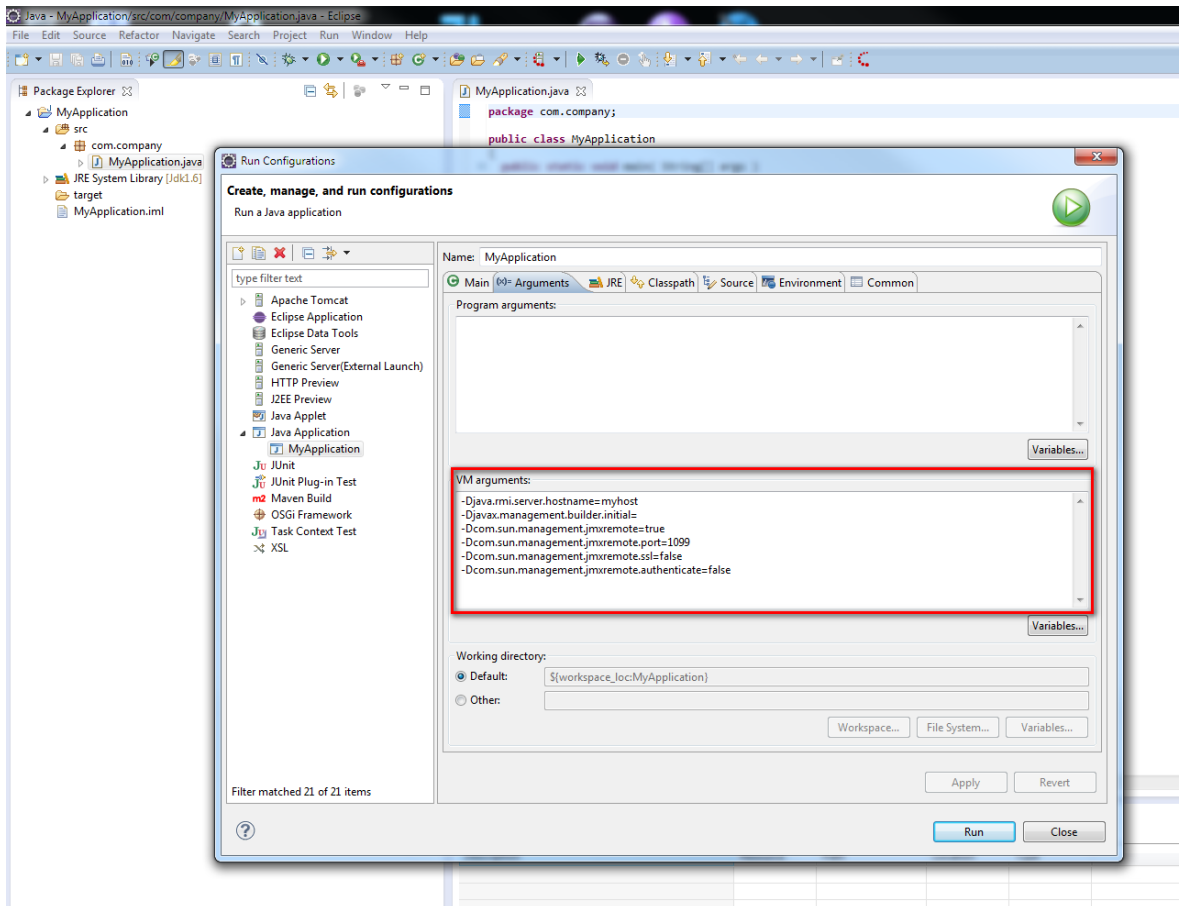
```
-Djava.rmi.server.hostname=JMX_HOST
-Djavax.management.builder.initial=
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=JMX_PORT
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

3.1.5.1 Eclipse IDE

If you start the standalone Java application from Eclipse IDE, please go to:

Run > Run Configurations > Java Application > [Your Application Name] > Arguments

Add the above JVM system properties to the VM Arguments text area highlighted in the following figure.



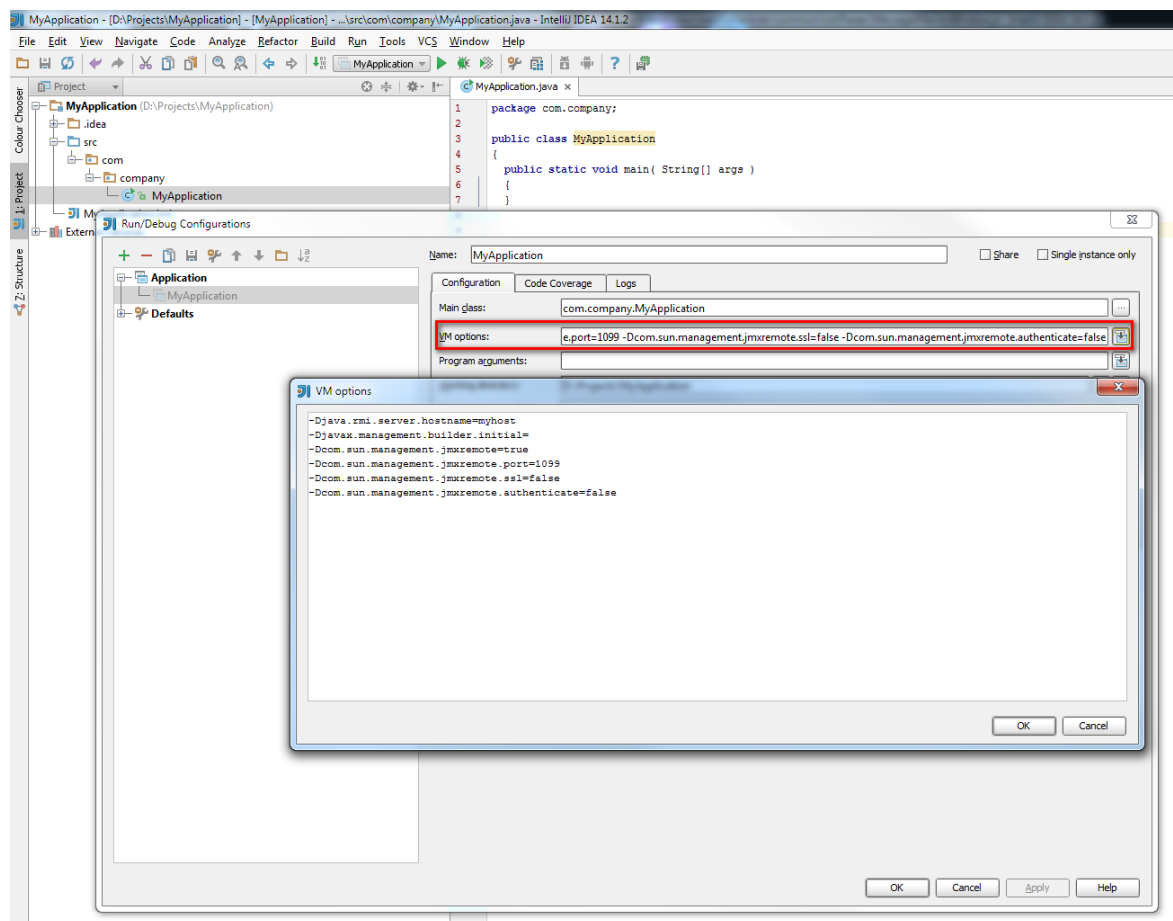
3.1.5.2 IntelliJ IDE

If you start the standalone Java application from IntelliJ IDE, please go to:

Run > Edit Configurations > Application > [Your Application Name] > Configuration

Add the above JVM system properties to the VM Options text field highlighted in the following figure.





3.2 JMXMP

To enable access to the Java Platform MBean Server through the JMXMP protocol, the JVM must be started with the QuartzDesk JVM Agent. In the QuartzDesk JVM Agent configuration (typically `AGENT_WORK_DIR/quartzdesk-agent.properties`), it is necessary to adjust values of the following configuration properties:

Configuration Property	Default Value	Description
jmxConnector.jmxmp.enabled	false	Flag indicating if the JMXMP connector is enabled.
jmxConnector.jmxmp.bindAddress	n/a	The hostname/IP the JMXMP connector binds to. If not set, the JMXMP connector binds to all hostnames/IPs.
jmxConnector.jmxmp.port	11099	The port number the JMXMP connector listens on for inbound connections.
jmxConnector.jmxmp.sasl.enabled	false	Flag indicating if the JMXMP connector is secured with SASL. Please refer to TODO for details on how to configure JMXMP security.

<code>jmxConnector.jmxmp.sasl.username</code>	n/a	Username used by SASL. It must be specified in SASL is enabled.
<code>jmxConnector.jmxmp.sasl.password</code>	n/a	Password used by SASL. It must be specified in SASL is enabled.

3.3 REMOTING-JMX (JBoss Only)

This chapter describes changes to the JBoss startup scripts and assumes that JBoss is run in standalone mode. If you are running JBoss in domain mode, the required changes are similar, but you need to apply them to a different JBoss startup script (e.g. `domain.conf.bat` / `domain.conf` etc.).

3.3.1 JBoss 6.1.0 Community

Windows

Open the `JBOSS_INSTALL_ROOT\bin\run.conf.bat` file in a text editor and add the following lines at the end of the file.

```
rem Enable remote JMX access
set "JAVA_OPTS=%JAVA_OPTS% -Djava.rmi.server.hostname=JMX_HOST"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=JMX_PORT"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false"

set "JAVA_OPTS=%JAVA_OPTS% -Djboss.platform.mbeanserver"
set "JAVA_OPTS=%JAVA_OPTS% -
Djavax.management.builder.initial=org.jboss.system.server.jmx.MBeanServerBuild
erImpl"

set "JAVA_OPTS=%JAVA_OPTS% -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
set "JAVA_OPTS=%JAVA_OPTS% -
Dorg.jboss.logging.Logger.pluginClass=org.jboss.logging.logmanager.LoggerPlug
inImpl"

set "JBOSS_CLASSPATH=../lib/jboss-logmanager.jar"
```

Unix/Linux

Open the `JBOSS_INSTALL_ROOT/bin/run.conf` file in a text editor and add the following lines at the end of the file.

```
# Enable remote JMX access
JAVA_OPTS="${JAVA_OPTS} -Djava.rmi.server.hostname=JMX_HOST"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"

JAVA_OPTS="${JAVA_OPTS} -Djboss.platform.mbeanserver"
JAVA_OPTS="${JAVA_OPTS} -
Djavax.management.builder.initial=org.jboss.system.server.jmx.MBeanServerBuild
erImpl"

JAVA_OPTS="${JAVA_OPTS} -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
JAVA_OPTS="${JAVA_OPTS} -
Dorg.jboss.logging.Logger.pluginClass=org.jboss.logging.logmanager.LoggerPlug
inImpl"

set "JBOSS_CLASSPATH=../lib/jboss-logmanager.jar"
```

3.3.2 JBoss 7.1.1 Community

Windows

Open the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file in a text editor and add the following lines at the end of the file.

```
rem Enable remote JMX access
set "JAVA_OPTS=%JAVA_OPTS% -Djava.rmi.server.hostname=JMX_HOST"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=JMX_PORT"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false"

rem Required to fix "JBAS014670: Failed initializing module
org.jboss.as.logging" JBoss startup error
set "JBOSS_HOME=JBOSS_INSTALL_ROOT"
set "JAVA_OPTS=%JAVA_OPTS% -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\org\jboss\logmanager\main\jboss-
logmanager-1.2.2.GA.jar"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\org\jboss\logmanager\log4j\main\jboss-
logmanager-log4j-1.0.0.GA.jar"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\org\apache\log4j\main\log4j-1.2.16.jar"
```

In the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file locate a line that sets the `org.jboss.modules.system.pkgs` JVM system property and add the `org.jboss.logmanager` package to the specified list of comma-separated packages.

Unix/Linux

Open the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file in a text editor and add the following lines at the end of the file.

```
# Enable remote JMX access
JAVA_OPTS="{JAVA_OPTS} -Djava.rmi.server.hostname=JMX_HOST"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"

# Required to fix "JBAS014670: Failed initializing module
org.jboss.as.logging" JBoss startup error
JBOSS_HOME="JBOSS_INSTALL_ROOT"
JAVA_OPTS="{JAVA_OPTS} -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
JAVA_OPTS="{JAVA_OPTS} -
Xbootclasspath/p:{$JBOSS_HOME}/modules/org/jboss/logmanager/main/jboss-
logmanager-1.2.2.GA.jar"
JAVA_OPTS="{JAVA_OPTS} -
Xbootclasspath/p:{$JBOSS_HOME}/modules/org/jboss/logmanager/log4j/main/jboss-
logmanager-log4j-1.0.0.GA.jar"
JAVA_OPTS="{JAVA_OPTS} -
Xbootclasspath/p:{$JBOSS_HOME}/modules/org/apache/log4j/main/log4j-1.2.16.jar"
```

In the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file locate a line that sets the `org.jboss.modules.system.pkgs` JVM system property and add the `org.jboss.logmanager` package to the specified list of comma-separated packages.

3.3.3 JBoss 6.1 EAP

Windows

Open the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file in a text editor and add the following lines at the end of the file.

```
rem Enable remote JMX access
set "JAVA_OPTS=%JAVA_OPTS% -Djava.rmi.server.hostname=JMX_HOST"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=JMX_PORT"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false"
```

Unix/Linux

Open the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file in a text editor and add the following lines at the end of the file.

```
# Enable remote JMX access
JAVA_OPTS="{JAVA_OPTS} -Djava.rmi.server.hostname=JMX_HOST"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"
```

3.3.4 JBoss 6.2 EAP

Windows

Open the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file in a text editor and add the following lines at the end of the file.

```
rem Enable remote JMX access
set "JAVA_OPTS=%JAVA_OPTS% -Djava.rmi.server.hostname=JMX_HOST"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=JMX_PORT"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false"

rem Required to fix "JBAS014670: Failed initializing module
org.jboss.as.logging" JBoss startup error
set "JBOSS_HOME=JBOSS_INSTALL_ROOT"
set "JAVA_OPTS=%JAVA_OPTS% -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\system\layers\base\org\jboss\logmanager
\main\jboss-logmanager-1.5.1.Final-redhat-1.jar"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\system\layers\base\org\jboss\log4j\logm
anager\main\log4j-jboss-logmanager-1.0.2.Final-redhat-1.jar"
```

In the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file locate a line that sets the `org.jboss.modules.system.pkgs` JVM system property and add the `org.jboss.logmanager` package to the specified list of comma-separated packages.

Unix/Linux

Open the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file in a text editor and add the following lines at the end of the file.

```
# Enable remote JMX access
JAVA_OPTS="${JAVA_OPTS} -Djava.rmi.server.hostname=JMX_HOST"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"

# Required to fix "JBAS014670: Failed initializing module
org.jboss.as.logging" JBoss startup error
JBOSS_HOME="JBOSS_INSTALL_ROOT"
JAVA_OPTS="${JAVA_OPTS} -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
JAVA_OPTS="${JAVA_OPTS} -
Xbootclasspath/p:${JBOSS_HOME}/modules/system/layers/base/org/jboss/logmanager
/main/jboss-logmanager-1.5.1.Final-redhat-1.jar"
JAVA_OPTS="${JAVA_OPTS} -
Xbootclasspath/p:${JBOSS_HOME}/modules/system/layers/base/org/jboss/log4j/logm
anager/main/log4j-jboss-logmanager-1.0.2.Final-redhat-1.jar"
```

In the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file locate a line that sets the `org.jboss.modules.system.pkgs` JVM system property and add the `org.jboss.logmanager` package to the specified list of comma-separated packages.

3.3.5 JBoss 6.3 EAP

Windows

Open the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file in a text editor and add the following lines at the end of the file.

```
rem Enable remote JMX access
set "JAVA_OPTS=%JAVA_OPTS% -Djava.rmi.server.hostname=JMX_HOST"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=JMX_PORT"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false"

rem Required to fix "JBAS014670: Failed initializing module
org.jboss.as.logging" JBoss startup error
set "JBOSS_HOME=JBOSS_INSTALL_ROOT"
set "JAVA_OPTS=%JAVA_OPTS% -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\system\layers\base\org\jboss\logmanager
\main\jboss-logmanager-1.5.2.Final-redhat-1.jar"
set "JAVA_OPTS=%JAVA_OPTS% -
Xbootclasspath/p:%JBOSS_HOME%\modules\system\layers\base\org\jboss\log4j\logm
anager\main\log4j-jboss-logmanager-1.1.0.Final-redhat-2.jar"
```

In the `JBOSS_INSTALL_ROOT\bin\standalone.conf.bat` file locate a line that sets the `org.jboss.modules.system.pkgs` JVM system property and add the `org.jboss.logmanager` package to the specified list of comma-separated packages.

Unix/Linux

Open the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file in a text editor and add the following lines at the end of the file.

```
# Enable remote JMX access
JAVA_OPTS="${JAVA_OPTS} -Djava.rmi.server.hostname=JMX_HOST"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.port=JMX_PORT"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"

# Required to fix "JBAS014670: Failed initializing module
org.jboss.as.logging" JBoss startup error
JBOSS_HOME="JBOSS_INSTALL_ROOT"
JAVA_OPTS="${JAVA_OPTS} -
Djava.util.logging.manager=org.jboss.logmanager.LogManager"
JAVA_OPTS="${JAVA_OPTS} -
Xbootclasspath/p:${JBOSS_HOME}/modules/system/layers/base/org/jboss/logmanager
/main/jboss-logmanager-1.5.2.Final-redhat-1.jar"
JAVA_OPTS="${JAVA_OPTS} -Xbootclasspath/p:
"${JBOSS_HOME}/modules/system/layers/base/org/jboss/log4j/logmanager/main/log4
j-jboss-logmanager-1.1.0.Final-redhat-2.jar"
```

In the `JBOSS_INSTALL_ROOT/bin/standalone.conf` file locate a line that sets the `org.jboss.modules.system.pkgs` JVM system property and add the `org.jboss.logmanager` package to the specified list of comma-separated packages.

4. Exporting Quartz Scheduler MBean

In order to manage a Quartz scheduler remotely, its management interface (MBean) must be registered in the JVM MBean Platform Server.

Quartz scheduler can automatically register its MBean in the JVM MBean Platform Server upon its initialization by setting the `org.quartz.scheduler.jmx.export` Quartz configuration property to true.

This property can either be specified in the `quartz.properties` configuration file, or as a JVM system property (`-Dorg.quartz.scheduler.jmx.export=true`).

In Spring-based applications you can add the `org.quartz.scheduler.jmx.export` property to the list of properties specified in the `quartzProperties` attribute of the `org.springframework.scheduling.quartz.SchedulerFactoryBean`.

```
<!-- Quartz scheduler bean -->
<bean id="quartzScheduler"
      class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
  ...
  <property name="quartzProperties">
    <util:properties>
      ...
      <!--
        Must be set to true, otherwise the Quartz scheduler is not registered in the
        JMX server.
      -->
      <prop key="org.quartz.scheduler.jmx.export">true</prop>
      ...
    </util:properties>
  </property>
  ...
</bean>
```

5. Verifying Remote JMX Access

5.1 Tomcat, WebSphere AS, GlassFish AS, WebLogic AS, Standalone Java Application

To verify remote JMX access to a Quartz scheduler, you can use the jconsole tool that is distributed with the Java Development Kit (JDK). Start jconsole like so:

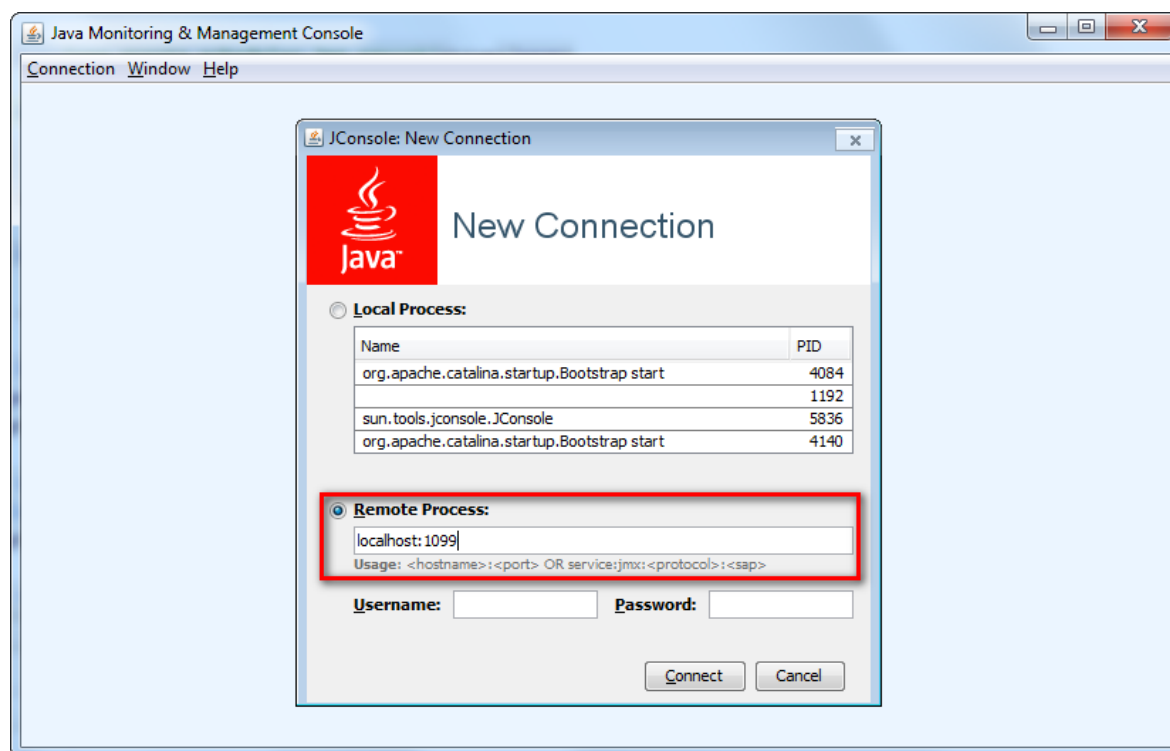
Windows

```
JAVA_HOME\bin\jconsole.exe
```

Unix/Linux

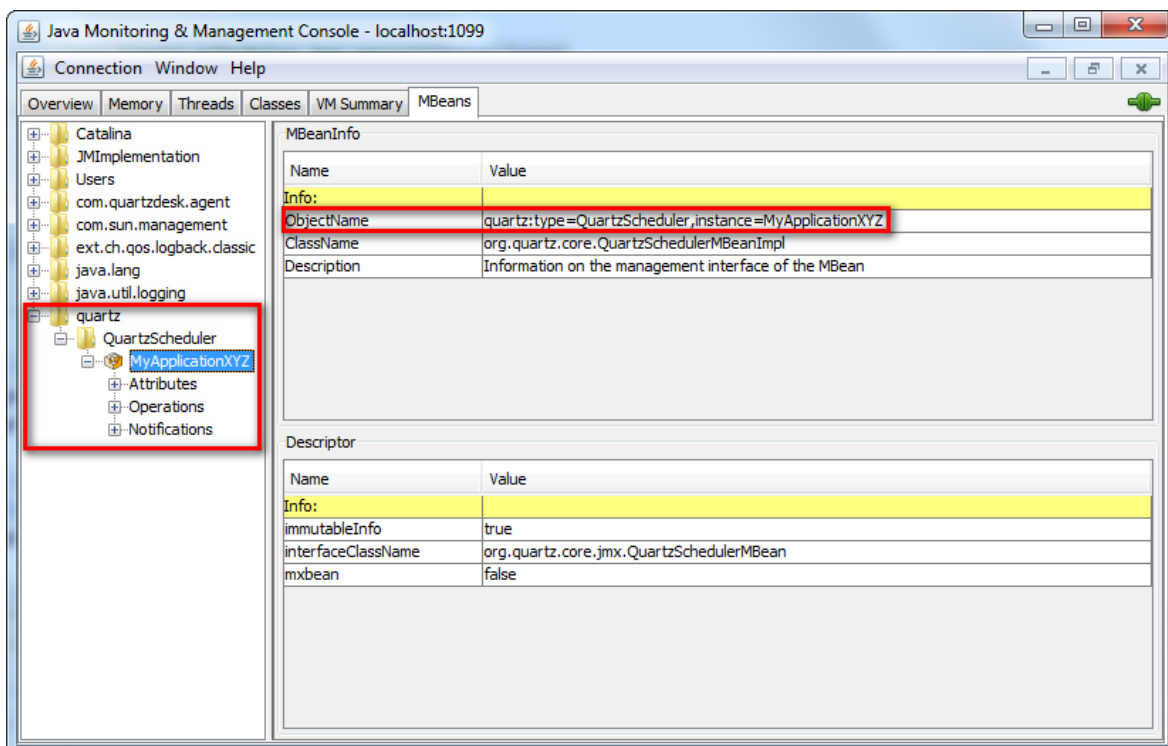
```
JAVA_HOME/bin/jconsole
```

Select the “Remote Process” radio button and provide the JMX_HOST and JMX_PORT values of the JVM the Quartz scheduler is running on, separated by a colon.



Click Connect.

Upon successful connection, select the MBeans tab and in the tree panel on the left look for the Quartz scheduler MBean. The configured Quartz scheduler object name components, separated by commas, determine the location of the MBean in the tree.



5.2 JBoss AS

To verify remote JMX access to a Quartz scheduler running on JBoss, you can use the jconsole tool started with the jconsole startup script from the JBoss distribution. Start jconsole like so:

Windows

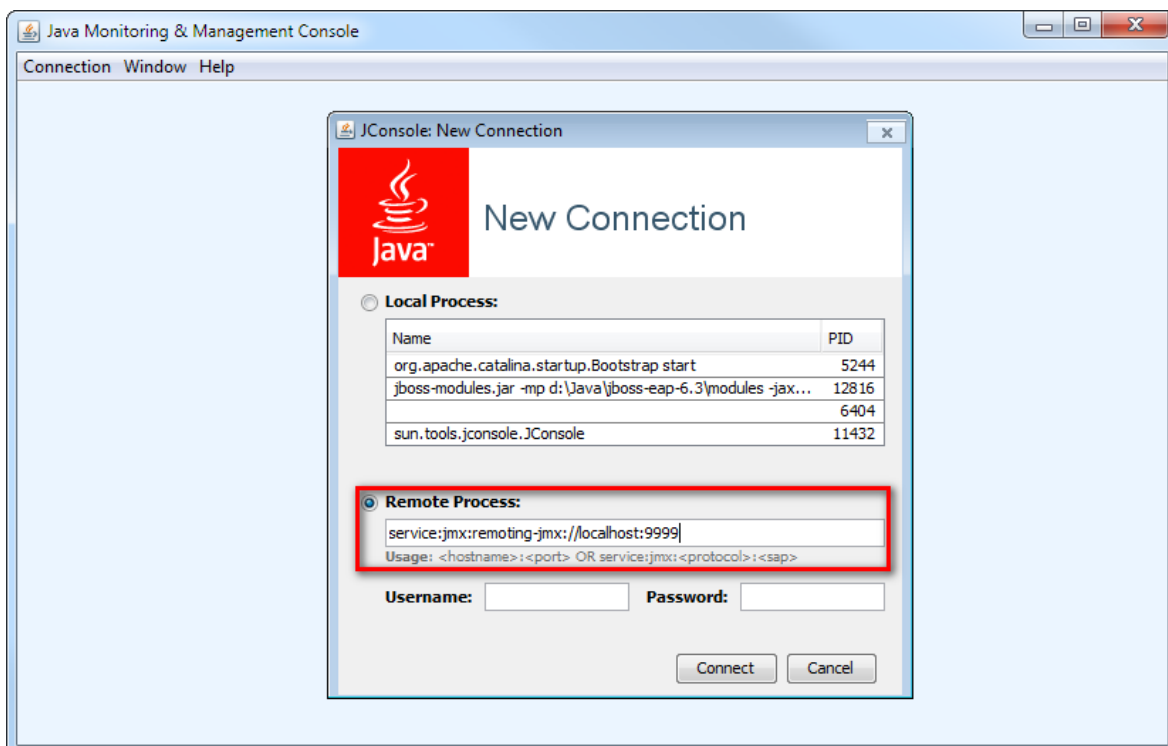
```
JBOSS_INSTALL_ROOT\bin\jconsole.bat
```

Unix/Linux

```
JBOSS_INSTALL_ROOT/bin/jconsole.sh
```

Select the “Remote Process” radio button and enter a JMX URL in the `service:jmx:remoting-jmx://JMX_HOST:JMX_PORT` format for the JVM the Quartz scheduler is running on.





Click Connect.

Upon successful connection, select the MBeans tab and in the tree panel on the left look for the Quartz scheduler MBean. The configured Quartz scheduler object name components, separated by commas, determine the location of the MBean in the tree.

