



# QuartzDesk JVM Agent Installation and Upgrade Guide for Standalone Java Applications

QuartzDesk Version: 3.x

January 21, 2019



## Table of Contents

<b>1.</b>	<b>PURPOSE</b> .....	<b>3</b>
<b>2.</b>	<b>DEFINITIONS</b> .....	<b>4</b>
<b>3.</b>	<b>REQUIREMENTS</b> .....	<b>5</b>
3.1	SOFTWARE REQUIREMENTS .....	5
3.1.1	<i>Operating System</i> .....	5
3.1.2	<i>Java</i> .....	5
3.1.3	<i>Database</i> .....	5
3.1.4	<i>Database JDBC Driver</i> .....	5
3.1.5	<i>QuartzDesk JVM Agent Library</i> .....	6
3.1.6	<i>QuartzDesk Public API Library</i> .....	6
3.2	HARDWARE REQUIREMENTS.....	6
<b>4.</b>	<b>INSTALLATION</b> .....	<b>7</b>
4.1	DATABASE.....	7
4.2	JDBC DRIVER .....	7
4.3	JVM AGENT WORK DIRECTORY .....	8
4.4	JVM AGENT CONFIGURATION .....	8
4.5	INSTALL JVM AGENT.....	9
4.5.1	<i>Windows</i> .....	9
4.5.2	<i>Unix / Linux</i> .....	9
4.6	INSTALL PUBLIC API LIBRARY .....	10
4.7	STOP APPLICATION.....	10
4.8	START APPLICATION .....	10
<b>5.</b>	<b>UPGRADING</b> .....	<b>12</b>
5.1	STOP APPLICATION.....	12
5.2	BACKUP .....	12
5.3	UPGRADE JVM AGENT.....	12
5.4	UPGRADE PUBLIC API LIBRARY .....	12
5.5	START APPLICATION .....	12
<b>6.</b>	<b>QUARTZDESK 2.X TO 3.X MIGRATION NOTES</b> .....	<b>13</b>
6.1	MINIMUM REQUIRED JAVA VERSION.....	13
6.2	CHANGED NAMES OF CONFIGURATION PROPERTIES .....	13
6.3	UPGRADE STEPS .....	13



## 1. Purpose

This document describes the installation and upgrade process for the QuartzDesk JVM Agent 3.x in standalone Java applications.

QuartzDesk JVM Agent is a Java Virtual Machine (JVM) plugin that must be installed in all JVMs powering applications with embedded Quartz schedulers managed by QuartzDesk. The QuartzDesk JVM Agent enables the following QuartzDesk features:

- Adding and editing of triggers
- Job and trigger execution history
- Execution notification rules
- Job chaining
- Job and trigger health indicators
- Viewing of misfired triggers
- Viewing of planned job execution
- Job and trigger execution statistics

Please note that the installation of the QuartzDesk JVM Agent is required only by the QuartzDesk Standard and Enterprise editions. The QuartzDesk Lite edition does not contain any of the above features and therefore it does not require the installation of the QuartzDesk JVM Agent.



If the QuartzDesk GUI detects the QuartzDesk JVM Agent is not installed / enabled in a remote JVM it connects to, it displays a warning message and the above listed features are disabled in the QuartzDesk GUI.

**If you experience any problems installing or upgrading the QuartzDesk JVM Agent, please let us know at [support@quartzdesk.com](mailto:support@quartzdesk.com).**

## 2. Definitions

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

Acronym / Shortcut	Definition
<b>EAR</b>	Enterprise Application Archive. A file with <code>.ear</code> extension.
<b>JAR</b>	Java Application Archive. A file with <code>.jar</code> extension.
<b>JVM</b>	Java Virtual Machine.

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

Location / Property	Example	Description
<b>AGENT_WORK_DIR</b>	<code>/var/quartzdesk-agent.work</code>	QuartzDesk JVM Agent work directory.
<b>DB_HOST</b>	<code>localhost</code>	QuartzDesk JVM Agent database server host.
<b>DB_PORT</b>	<code>5432</code>	QuartzDesk JVM Agent database server port.
<b>DB_NAME</b>	<code>quartzdesk_agent</code>	QuartzDesk JVM Agent database name.
<b>DB_SCHEMA</b>	<code>quartzdesk_agent</code>	QuartzDesk JVM Agent database schema.
<b>DB_USER</b>	<code>quartzdesk_agent</code>	QuartzDesk JVM Agent database user.
<b>DB_PASSWORD</b>	<code>quartzdesk_agent</code>	QuartzDesk JVM Agent database user password.



## 3. Requirements

### 3.1 Software Requirements

#### 3.1.1 Operating System

Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10.  
Linux (any distribution) with kernel 2.6.x and above.  
Solaris 11.x and above.

#### 3.1.2 Java

Sun/Oracle Java (JDK) 7, 8, 9, 10.  
IBM Java (JDK) 7, 8, 9.  
OpenJDK 7, 8, 9, 10.

#### 3.1.3 Database

Database	Minimum Version
DB2	10.1
H2	1.3.174
Microsoft SQL Server	2008 R2 SP1
MySQL	5.6.4
Oracle	10.2 (10g R2)
PostgreSQL	9.1

#### 3.1.4 Database JDBC Driver

Database	JDBC Driver
DB2	IBM DB2 JDBC 4.0 driver available at <a href="http://www-01.ibm.com/support/docview.wss?uid=swg21363866">http://www-01.ibm.com/support/docview.wss?uid=swg21363866</a> .
H2	Database engine including the JDBC driver is available at <a href="http://www.h2database.com">http://www.h2database.com</a> .
Microsoft SQL Server	Microsoft JDBC driver 4.0 for SQL Server available at <a href="http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx">http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx</a> .  We strongly advise against using the alternative JTDS JDBC driver, because it currently does not support the datetime2 data type. As a result, all datetime values written by the QuartzDesk application would end up rounded up, or down. For datetime data type rounding details, please refer to <a href="http://msdn.microsoft.com/en-us/library/ms187819.aspx">http://msdn.microsoft.com/en-us/library/ms187819.aspx</a> .
MySQL	Connector/J JDBC driver available at <a href="http://dev.mysql.com/downloads/connector/j/">http://dev.mysql.com/downloads/connector/j/</a> .

<b>Oracle</b>	Oracle JDBC driver available at <a href="http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html">http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html</a> .  For a comprehensive overview of JDBC driver versions vs. supported database versions, please refer to <a href="http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_02">http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_02</a> .
<b>PostgreSQL</b>	JDBC4 PostgreSQL driver available at <a href="http://jdbc.postgresql.org/">http://jdbc.postgresql.org/</a> .

### 3.1.5 QuartzDesk JVM Agent Library

To install the QuartzDesk JVM Agent, you need to obtain the QuartzDesk JVM Agent JAR. The latest version can be downloaded at [www.quartzdesk.com](http://www.quartzdesk.com) (click Downloads → Latest Release → View files → quartzdesk-agent-x.y.z.jar).

### 3.1.6 QuartzDesk Public API Library

QuartzDesk JVM Agent requires all applications with embedded Quartz schedulers deployed on the given JVM to have the QuartzDesk Public API Library on their classpath. The library must be loaded by the same classloader that loads the Quartz API.

The latest version of the QuartzDesk Public API Library can be downloaded at [www.quartzdesk.com](http://www.quartzdesk.com) (click Downloads → Latest Release → View files → quartzdesk-api-x.y.z.jar). It is also available in the Maven Central repository – see <http://search.maven.org/#search|ga|1|quartzdesk-api>.

## 3.2 Hardware Requirements

QuartzDesk JVM Agent runs on any physical or virtualized hardware that supports the above software requirements.



## 4. Installation

### 4.1 Database

Create a new database user named `quartzdesk_agent` (`DB_USER`) with an arbitrary password (`DB_PASSWORD`).

Create a new QuartzDesk JVM Agent database named `quartzdesk_agent1` (`DB_NAME`) owned by the `DB_USER`.

If the database supports database schemas, create a new schema named `quartzdesk_agent` (`DB_SCHEMA`). The schema must be owned by the `DB_USER`. Make the created `DB_SCHEMA` the default schema of the `DB_USER` and/or add the schema to the `DB_USER`'s schema search path.

Please contact your DBA, or refer to the database engine documentation for instructions on how to complete the above database-specific tasks.



Please note that you do not have to create any other database objects (tables, keys, indices etc.) in the QuartzDesk JVM Agent database. These objects will be automatically created by the QuartzDesk JVM Agent during its first run.

### 4.2 JDBC Driver

Download and install the JDBC driver for the created database. For a list of supported JDBC drivers please refer to chapter 3.1.4.

In the standalone Java application's startup script add the JDBC driver JAR file(s) to the JVM's boot classpath. The JDBC driver JAR file(s) must be added to the boot classpath so that QuartzDesk JVM Agent, that is loaded by the JVM's system classloader, can access JDBC driver classes. Please use the following JVM command line option:

```
-Xbootclasspath/p:JDBC DRIVER JAR FILE(S)

For example:

Windows:
-Xbootclasspath/p:c:\jdbc\db2\db2jcc_license_cu.jar;c:\jdbc\db2\db2jcc4.jar

Unix/Linux:
-Xbootclasspath/p:/opt/db2/db2jcc_license_cu.jar:/opt/jdbc/db2/db2jcc4.jar
```

Make sure the JDBC driver JAR file(s) are readable by the user the standalone Java application process runs under.

---

<sup>1</sup> DB2 restricts the database name length to the maximum of 8 characters. Please adjust the database name accordingly (e.g. qdagent).

### 4.3 JVM Agent Work Directory

Create the QuartzDesk JVM Agent work directory (AGENT\_WORK\_DIR) anywhere on the local file system. The directory should be readable and writeable by the user the standalone Java application process runs under.

Copy your QuartzDesk license key file (license.key) to AGENT\_WORK\_DIR.



You can obtain a free 30-day trial license key at [www.quartzdesk.com](http://www.quartzdesk.com) (open the Try / Purchase menu).

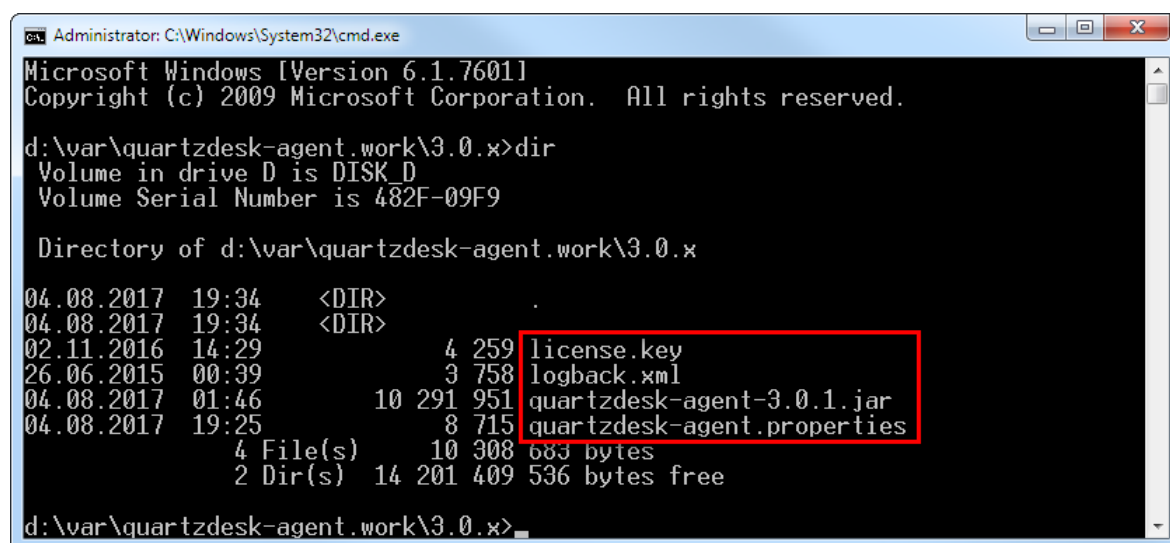
Copy the QuartzDesk JVM Agent JAR file (quartzdesk-agent-x.y.z.jar) to AGENT\_WORK\_DIR.

Open the QuartzDesk JVM Agent JAR file and copy all files from the extras/work directory into AGENT\_WORK\_DIR.



If you cannot open the JAR file directly, rename it to \*.zip and then open it. Do not forget to rename the file back to \*.jar once you have extracted the required files.

In the following figure you can see an example of a QuartzDesk JVM Agent work directory correctly set up on a Microsoft Windows machine.



```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

d:\var\quartzdesk-agent.work\3.0.x>dir
Volume in drive D is DISK D
Volume Serial Number is 482F-09F9

Directory of d:\var\quartzdesk-agent.work\3.0.x

04.08.2017  19:34    <DIR>          .
04.08.2017  19:34    <DIR>          ..
02.11.2016  14:29             4 259 license.key
26.06.2015   0:39             3 758 logback.xml
04.08.2017   1:46          10 291 951 quartzdesk-agent-3.0.1.jar
04.08.2017   19:25             8 715 quartzdesk-agent.properties
              4 File(s)      10 308 683 bytes
              2 Dir(s)   14 201 409 536 bytes free

d:\var\quartzdesk-agent.work\3.0.x>
    
```

### 4.4 JVM Agent Configuration

Open the QuartzDesk JVM Agent configuration file AGENT\_WORK\_DIR/quartzdesk-agent.properties.

Based on the type and version of the database created in step 4.1, change the value of the db.profile configuration property according to the following table.

Database	Database Version	db.profile Value
DB2	>= 10.0	db2
H2	>= 1.3.170	h2
Microsoft SQL Server	>= 2008	mssql



MySQL	>= 5.6	mysql
MySQL (Inno)	>= 5.6	mysql_inno
Oracle	== 8i	oracle8
Oracle	>= 9i	oracle9
PostgreSQL	== 8.1	postgres81
PostgreSQL	>= 8.2	postgres82

Uncomment the Agent JDBC pool configuration section based on the QuartzDesk JVM Agent database type. Make sure the JDBC pool configuration sections for other database types are commented out (prefixed with '#'). The default sample `quartzdesk-agent.properties` file assumes the use of a PostgreSQL database.

Adjust values of the JDBC pool configuration parameters to match your database configuration. You will typically need to change the default host value (localhost) in the `jdbc.url` parameter to point to DB\_HOST. Please refer to the JDBC driver manual for a description of the JDBC URL format and related details.

Set the value of the `jdbc.pool.maxTotal` JDBC pool configuration parameter to be 10-20% higher than the maximum number of **concurrently executing** Quartz jobs on the JVM the QuartzDesk JVM Agent will be installed on.

To adjust QuartzDesk JVM Agent logging parameters, edit the `AGENT_WORK_DIR/logback.xml` configuration file. The default sample `logback.xml` configuration file creates the QuartzDesk JVM Agent log under the `AGENT_WORK_DIR/logs` directory that is automatically created when the QuartzDesk JVM Agent starts. Please refer to the [Logback Manual](#) for Logback configuration details.

## 4.5 Install JVM Agent

To manage Quartz schedulers embedded in a standalone Java application, you must first enable remote JMX access to the application. Please refer to the **How to Enable Remote JMX Access to Quartz Schedulers** document available at [www.quartzdesk.com/documentation/how-tos](http://www.quartzdesk.com/documentation/how-tos). Once the remote JMX access has been enabled, continue with the steps below.

### 4.5.1 Windows

Edit standalone Java application's startup script (typically with bat, or cmd extension) and add the following two parameters to the list of standalone application's JVM startup parameters.

```
-javaagent:AGENT_WORK_DIR/quartzdesk-agent-x.y.z.jar -Dquartzdesk-agent.work.dir=AGENT_WORK_DIR
```

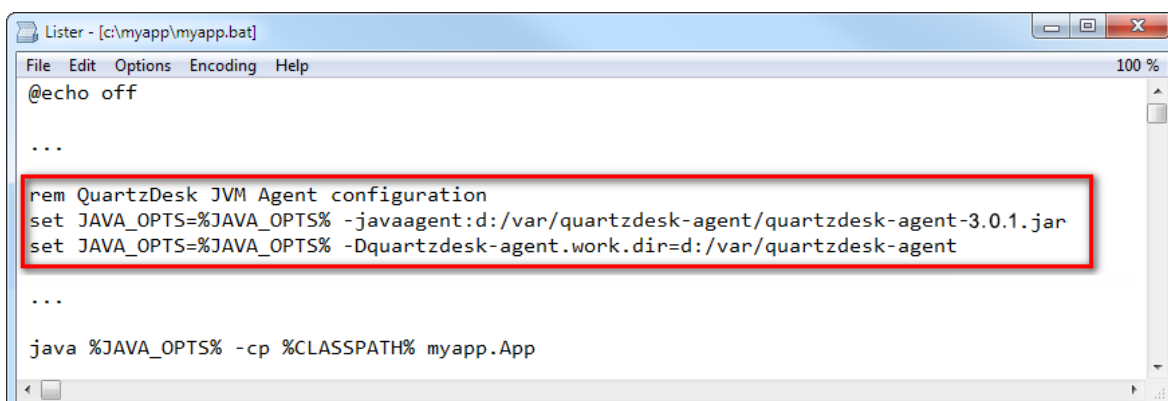
In the following figure you can see an example of a standalone application's startup script with all required JVM parameters.

### 4.5.2 Unix / Linux

Edit standalone Java application's startup script (typically with bat, or cmd extension) and add the following two parameters to the list of standalone application's JVM startup parameters.

```
-javaagent:AGENT_WORK_DIR/quartzdesk-agent-x.y.z.jar -Dquartzdesk-agent.work.dir=AGENT_WORK_DIR
```

In the following figure you can see an example of a standalone application's startup script with all required JVM parameters:



```
Listner - [c:\myapp\myapp.bat]
File Edit Options Encoding Help 100 %
@echo off
...
rem QuartzDesk JVM Agent configuration
set JAVA_OPTS=%JAVA_OPTS% -javaagent:d:/var/quartzdesk-agent/quartzdesk-agent-3.0.1.jar
set JAVA_OPTS=%JAVA_OPTS% -Dquartzdesk-agent.work.dir=d:/var/quartzdesk-agent
...
java %JAVA_OPTS% -cp %CLASSPATH% myapp.App
```

## 4.6 Install Public API Library

The QuartzDesk Public API Library (`quartzdesk-api-<version>.jar`) works as an interface between the Quartz library (typically distributed as `quartz-<version>.jar` or `quartz-all-<version>.jar`) used by an application and the QuartzDesk JVM Agent. **The QuartzDesk Public API Library must be loaded by the same Java classloader that loads the Quartz library.**

For standalone applications, the Quartz library is typically placed on the application's classpath in its startup script. To install the QuartzDesk Public API Library, it is only necessary to add this library on the classpath.

**No application code changes are required to install the QuartzDesk Public API Library.**

## 4.7 Stop Application

Stop the standalone Java application if it is running.

## 4.8 Start Application

Start the standalone Java application, e.g. by executing its startup script. Wait for the application to start and check the application log for errors.

Check the QuartzDesk JVM Agent logs (in `AGENT_WORK_DIR/logs` directory) for errors and verify the version number of the QuartzDesk JVM Agent.

# QuartzDesk JVM Agent Installation and Upgrade Guide for Standalone Java Applications



```
File Edit Options Encoding Help
Lister - [D:\var\quartzdesk-agent-work\3.0\logs\quartzdesk-agent.log]
Serial Number: 1
Issue Date: 2016-11-02
Type: REGULAR
Expiry Date: n/a
License: Text License
Issuer: CH\QuartzDesk.com CA2, O\QuartzDesk, sales@quartzdesk.com, www.quartzdesk.com
Licensed Product:
  ID=QuartzDesk, Name=QuartzDesk Enterprise Edition
  [2017-08-04 19:27:12,565] I [main] [com.quartzdesk.agent.Agent:472] - Licensed product features:
  Feature Set, id=general
  Features:
    Customization=true
    MConnectionService=true
    MSecurityService=true
    OS=true
    SoftwareMaintenanceExpiry=2020-12-31+01:00
  Feature Set, id=quartz
  Features:
    MaxJobPersScheduler=unlimited
    MaxTriggerPersJob=unlimited
    ExecutionHistory=true
    PlannedJobExecution=true
    MisfiredTriggers=true
    ExecutionNotifications=true
    PollMonitoring=true
    ExecutionStatistics=true
    JobChain=true
    MQuartzService=true
    MQuartzExecutionNotificationService=true
    MQuartzJobChainService=true
  [2017-08-04 19:27:12,605] I [main] [com.quartzdesk.agent.Agent:539] - Successfully created DB dialect: com.quartzdesk.agent.dialect.PostgreSQLDialect@2f93e448 for configured DB profile: postgres42
  [2017-08-04 19:27:12,646] I [main] [com.quartzdesk.agent.ob.a:129] - Detected QuartzDesk JVM Agent database schema version: 3.0.1, desired database schema version: 3.0.1
  [2017-08-04 19:27:12,693] I [main] [com.quartzdesk.agent.ob.a:143] - QuartzDesk JVM Agent database schema is up-to-date.
  [2017-08-04 19:27:12,886] I [main] [com.quartzdesk.agent.Agent:648] - Successfully registered MBean: com.quartzdesk.agent.Agent@beanImpl@64e137c0, object name: com.quartzdesk.agent:type=AgentMBean
  [2017-08-04 19:27:12,919] I [main] [com.quartzdesk.agent.Agent:662] - Successfully registered MBean: com.quartzdesk.agent.DataSynchronization@beanImpl@6e1c079, object name: com.quartzdesk.agent:type=DataSynchronizationMBean
  [2017-08-04 19:27:12,937] I [main] [com.quartzdesk.agent.Agent:676] - Successfully registered MBean: com.quartzdesk.agent.scheduler.common.message.Message@beanImpl@300b0a9f, object name: com.quartzdesk.agent:type=MessageMBean
  [2017-08-04 19:27:12,938] I [main] [com.quartzdesk.agent.Agent:705] - Successfully registered MBean: com.quartzdesk.agent.scheduler.quartz.history.QuartzExecutionHistory@beanImpl@2c040b09, object name: com.quartzdesk.agent:type=QuartzExecutionHistoryMBean
  [2017-08-04 19:27:12,939] I [main] [com.quartzdesk.agent.Agent:719] - Successfully registered MBean: com.quartzdesk.agent.scheduler.quartz.notify.QuartzExecutionNotification@beanImpl@6770a09, object name: com.quartzdesk.agent:type=QuartzExecutionNotificationMBean
  [2017-08-04 19:27:12,945] I [main] [com.quartzdesk.agent.Agent:733] - Successfully registered MBean: com.quartzdesk.agent.scheduler.quartz.stats.QuartzExecutionStatistics@beanImpl@7e5388f4, object name: com.quartzdesk.agent:type=QuartzExecutionStatisticsMBean
  [2017-08-04 19:27:12,954] I [main] [com.quartzdesk.agent.Agent:747] - Successfully registered MBean: com.quartzdesk.agent.scheduler.quartz.chain.QuartzJobChain@beanImpl@96cc200, object name: com.quartzdesk.agent:type=QuartzJobChainMBean
  [2017-08-04 19:27:12,963] I [main] [com.quartzdesk.agent.Agent:761] - Successfully registered MBean: com.quartzdesk.agent.scheduler.quartz.misfire.QuartzMisfireTrigger@beanImpl@151703b, object name: com.quartzdesk.agent:type=QuartzMisfireTriggerMBean
  [2017-08-04 19:27:12,971] I [main] [com.quartzdesk.agent.Agent:775] - Successfully registered MBean: com.quartzdesk.agent.scheduler.quartz.monitor.QuartzMonitoring@beanImpl@80e0120f, object name: com.quartzdesk.agent:type=QuartzMonitoringMBean
  [2017-08-04 19:27:12,986] I [main] [com.quartzdesk.agent.a.b:197] - Starting RMI connector: listening on registry port: 11171 and server port: 11171, SSL: false
  [2017-08-04 19:27:14,146] I [main] [com.quartzdesk.agent.a.b:191] - Successfully started RMI connector: java.management.remote.rmi.RMIConnectorServer@2cc83806 listening on registry port: 11171 and server port: 11171, SSL: false
  [2017-08-04 19:27:14,152] I [main] [com.quartzdesk.agent.a.b:191] - Starting JMX connector: listening on port: 11171, SSL: false
  [2017-08-04 19:27:14,162] I [main] [com.quartzdesk.agent.a.b:188] - Successfully started JMX connector: java.management.remote.jmx.JMXConnectorServer@400c32e0 listening on port: 11171 and server port: 11171, SSL: false
  [2017-08-04 19:27:14,217] I [main] [com.quartzdesk.agent.Agent:175] - Successfully initialized JVM Agent: com.quartzdesk.agent.Agent@3ab62081 [QuartzDesk JVM Agent v3.0.1], enabled: true
```

Verify that the application works as expected.

## 5. Upgrading

### 5.1 Stop Application

Stop the standalone Java application if it is running.

### 5.2 Backup

Backup your QuartzDesk JVM Agent database. We recommend performing a **full database backup**.

Backup the contents of the QuartzDesk JVM Agent work directory.

Store the backups in a safe place so that you can restore the original QuartzDesk JVM Agent version if the need arises.

### 5.3 Upgrade JVM Agent

Delete the old QuartzDesk JVM Agent JAR file in `AGENT_WORK_DIR`. Copy the new `quartzdesk-agent-x.y.z.jar` to `AGENT_WORK_DIR`.

Rename the `AGENT_WORK_DIR/quartzdesk-agent.properties` configuration file to `quartzdesk-agent.properties.old`.

Open the QuartzDesk JVM Agent archive (`quartzdesk-agent-x.y.z.jar`) and copy the `extras/work/quartzdesk-agent.properties` configuration file to `AGENT_WORK_DIR`.



If you cannot open the JAR file directly, rename it to `*.zip` and then open it. Do not forget to rename the file back to `*.jar` once you have extracted the required files.

Adjust the values of the configuration properties in the new configuration file `AGENT_WORK_DIR/quartzdesk-agent.properties` to match your system setup. You can use the old configuration file as a reference.

Please refer to 4.4 for a description of the configuration parameters that you need to adjust.

### 5.4 Upgrade Public API Library

Upgrade the QuartzDesk Public API library on the standalone Java application's classpath. Make sure the new library is included on the application's classpath. Please refer to 4.6 for details.

### 5.5 Start Application

Start the standalone Java application by following steps outlined in 4.8.

## 6. QuartzDesk 2.x to 3.x Migration Notes

### 6.1 Minimum Required Java Version

QuartzDesk JVM Agent v3.x requires Java 7 or higher. Java 6 is no longer supported.

### 6.2 Changed Names of Configuration Properties

The following two `quartzdesk-agent.properties` configuration properties have been renamed in QuartzDesk JVM Agent 3.x. Make sure your `quartzdesk-agent.properties` file uses the new property names.

Original Configuration Property Name (2.x)	New Configuration Property Name (3.x)
<code>jdbc.pool.maxActive</code>	<code>jdbc.pool.maxTotal</code>
<code>jdbc.pool.maxWait</code>	<code>jdbc.pool.maxWaitMillis</code>

### 6.3 Upgrade Steps

To upgrade QuartzDesk JVM Agent 2.x to 3.x, apply upgrade steps outlined in 5. When completing steps outlined in 5.3, make sure to use the new configuration property names.

