



QuartzDesk Web Application Installation and Upgrade Guide for Apache Tomcat 7.x, 8.x and 9.x

QuartzDesk Version: 4.x

March 3, 2020



Table of Contents

1.	PURPOSE	4
2.	DEFINITIONS	5
3.	REQUIREMENTS.....	6
3.1	SOFTWARE REQUIREMENTS	6
3.1.1	<i>Browser</i>	6
3.1.2	<i>Operating System</i>	6
3.1.3	<i>JVM</i>	6
3.1.4	<i>Application Server</i>	6
3.1.5	<i>Database</i>	6
3.1.6	<i>Database JDBC Driver</i>	7
3.1.7	<i>QuartzDesk Web Application Archive</i>	7
3.2	HARDWARE REQUIREMENTS.....	7
4.	INSTALLATION.....	8
4.1	DATABASE.....	8
4.2	JDBC DRIVER	8
4.3	DATA SOURCE.....	9
4.3.1	<i>DB2</i>	9
4.3.2	<i>H2</i>	10
4.3.3	<i>Microsoft SQL Server</i>	11
4.3.4	<i>MySQL</i>	12
4.3.5	<i>Oracle</i>	13
4.3.6	<i>PostgreSQL</i>	14
4.4	APPLICATION WORK DIRECTORY	15
4.5	APPLICATION CONFIGURATION.....	16
4.6	STOP TOMCAT	16
4.7	DEPLOY APPLICATION.....	17
4.8	START TOMCAT.....	18
5.	UPGRADING	20
5.1	STOP TOMCAT	20
5.2	BACKUP.....	20
5.3	REMOVE EXISTING APPLICATION	20
5.4	DEPLOY NEW APPLICATION.....	20
5.5	START TOMCAT.....	20
6.	QUARTZDESK 2.X TO 3.X MIGRATION NOTES	21
6.1	MINIMUM REQUIRED JAVA VERSION.....	21
6.2	RENAME CONFIGURATION FILE	21
6.3	RENAME LOG FILES	21
6.4	ACCESS TO MONITORING URLS (REST API)	22
6.5	ACCESS TO JAX-WS ENDPOINTS	23
6.6	REMOVE UNUSED USERS	24
7.	QUARTZDESK 3.X TO 4.X MIGRATION NOTES	25
8.	CLUSTER DEPLOYMENT NOTES.....	26
8.1	HTTP SESSION REPLICATION AND AFFINITY	26
8.2	SHARED WORK DIRECTORY.....	26
8.3	LOGGING CONFIGURATION	26
8.3.1	<i>Using Shared Log Files</i>	27
8.3.2	<i>Using Separate Log Files</i>	28
8.4	INTERNAL QUARTZ SCHEDULER	29



1. Purpose

This document describes the installation and upgrade process for QuartzDesk Web Application 4.x on Apache Tomcat 7.x, 8.x and 9.x.

If you experience any problems installing or upgrading QuartzDesk Web Application, please let us know at support@quartzdesk.com.



2. Definitions

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

Acronym / Shortcut	Definition
AS	Application Server.
EAR	Enterprise Application Archive. A file with <code>.ear</code> extension.
JAR	Java Application Archive. A file with <code>.jar</code> extension.
JVM	Java Virtual Machine.
WAR	Web Application Archive. A file with <code>.war</code> extension.

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

Location / Property	Example	Description
DB_HOST	mydb.foo.com	QuartzDesk web application database server host.
DB_PORT	5432	QuartzDesk Web Application database server port.
DB_NAME	quartzdesk	QuartzDesk Web Application database name.
DB_SCHEMA	quartzdesk	QuartzDesk Web Application database schema.
DB_USER	quartzdesk	QuartzDesk Web Application database user.
DB_PASSWORD	quartzdesk	QuartzDesk Web Application database user password.
TOMCAT_HOME	/usr/share/tomcat7	Tomcat installation directory.
TOMCAT_HTTP_HOST	mytomcat.foo.com	Tomcat HTTP connector host.
TOMCAT_HTTP_PORT	8080	Tomcat HTTP connector port.
WORK_DIR	/var/quartzdesk-web.work	QuartzDesk Web Application work directory.

3. Requirements

3.1 Software Requirements

3.1.1 Browser

The QuartzDesk Web Application's GUI requires a modern JavaScript-enabled browser. Please make sure JavaScript is enabled and not blocked by third party anti-virus/anti-malware software.

QuartzDesk Web Application has been tested with the following browser versions. These are also the minimum browsers versions required.

Browser	Minimum Version
Chrome	64
Firefox	45
Internet Explorer	8
Microsoft Edge	12
Opera	43
Safari	10

3.1.2 Operating System

Windows 7, Windows 8, Windows 10.

Linux (any distribution) with kernel 2.6.x and above.

Solaris 11.x and above.

3.1.3 JVM

Oracle JDK 8–13.

IBM JDK 8.

OpenJDK 8–13.

3.1.4 Application Server

Apache Tomcat 7.x.

Apache Tomcat 8.x.

Apache Tomcat 9.x.

3.1.5 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

3.1.6 Database JDBC Driver

Database	JDBC Driver
DB2	IBM DB2 JDBC 4.0 driver available at http://www-01.ibm.com/support/docview.wss?uid=swg21363866 .
H2	Database engine including the JDBC driver is available at http://www.h2database.com .
Microsoft SQL Server	Microsoft JDBC driver 4.0 for SQL Server available at http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx . We strongly advise against using the alternative JTDS JDBC driver because it does not support the datetime2 data type at this time. As a result, all datetime values written by QuartzDesk Web Application would end up rounded up, or down. For datetime data type rounding details, please refer to http://msdn.microsoft.com/en-us/library/ms187819.aspx .
MySQL	Connector/J JDBC driver available at http://dev.mysql.com/downloads/connector/j/ .
Oracle	Oracle JDBC driver available at http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html . For a comprehensive overview of JDBC driver versions vs. supported database versions, please refer to http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_02 .
PostgreSQL	JDBC4 PostgreSQL driver available at http://jdbc.postgresql.org/ .

3.1.7 QuartzDesk Web Application Archive

To install QuartzDesk Web Application, you need to obtain the quartzdesk-web-x.y.z.war file. The latest version can be downloaded at www.quartzdesk.com (click Downloads → Latest Release → View files → quartzdesk-web-x.y.z.war).

3.2 Hardware Requirements

QuartzDesk Web Application runs on any physical or virtualized hardware that supports the above software requirements.

4. Installation

This chapter describes the standard QuartzDesk Web Application installation. If you are only evaluating, you can run QuartzDesk Web Application in the **one-step installation mode** to dramatically reduce the number of required installation steps. For details, please see our [FAQs](#) and search for “one-step installation”.

4.1 Database

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

Create a new QuartzDesk Web Application database named `quartzdesk`¹ (`DB_NAME`) owned by `DB_USER`.

In the `quartzdesk` database create a new schema named `quartzdesk` (`DB_SCHEMA`). The schema must be owned by `DB_USER`. Make the created `DB_SCHEMA` the default schema of `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 database objects (tables, keys, indices etc.) in the `quartzdesk` database / schema. These objects will be automatically created by QuartzDesk Web Application during its first start.

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 3.1.6.

Copy the JDBC driver JAR file(s) into `TOMCAT_HOME/lib` directory. Make sure the copied JAR files in `TOMCAT_HOME/lib` are readable by the user the Tomcat process runs under (typically `tomcat` on Unix / Linux systems).

Alternatively, you can create a directory for your Tomcat extensions, for example, `TOMCAT_HOME/lib/ext` and copy the JDBC driver file(s) to this directory. In this case, you will need to include this directory in the Tomcat's “common” class loader class path. You can do it by changing the Tomcat's `common.loader` property in `TOMCAT_HOME/conf/catalina.properties`.

Original

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,  
${catalina.home}/lib,${catalina.home}/lib/*.jar
```

¹ If you use DB2, the database name length is restricted to the maximum of 8 characters. Please adjust the database name accordingly (e.g. `qdesk`).

New

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,  
${catalina.home}/lib,${catalina.home}/lib/*.jar,${catalina.bas  
e}/lib/ext,${catalina.base}/lib/ext/*.jar
```

4.3 Data Source

Register a global data-source in the Tomcat server configuration file TOMCAT_HOME/conf/server.xml by adding the following XML snippet under the GlobalNamingResources element.

Tomcat data-source is built on top of the Apache DBCP library. For a comprehensive description of all available data-source configuration attributes, please refer to the [Apache DBCP configuration reference](#).

4.3.1 DB2

For Tomcat 7.x (Apache DBCP 1):

```
<!--  
    DB2 JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
    auth="Container"  
    type="javax.sql.DataSource"  
    removeAbandoned="true"  
    removeAbandonedTimeout="30"  
    maxActive="10"  
    maxIdle="1"  
    maxWait="2000"  
    validationQuery="select 1 from sysibm.sysdummy1"  
    poolPreparedStatements="true"  
    username="DB_USER"  
    password="DB_PASSWORD"  
    driverClassName="com.ibm.db2.jcc.DB2Driver"  
    url="jdbc:db2://DB_HOST:DB_PORT/quartzde"/>
```

For Tomcat 8.x and 9.x (Apache DBCP 2):

```
<!--  
  DB2 JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandonedOnBorrow="true"  
  removeAbandonedTimeout="30"  
  maxTotal="10"  
  maxIdle="1"  
  maxWaitMillis="2000"  
  validationQuery="select 1 from sysibm.sysdummy1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="com.ibm.db2.jcc.DB2Driver"  
  url="jdbc:db2://DB_HOST:DB_PORT/quartzde"/>
```

4.3.2 H2



H2 is a light-weight Java database with limited fault tolerance and recovery functionality. We recommend using H2 for evaluation and experimental purposes only.

For Tomcat 7.x (Apache DBCP 1):

```
<!--  
  H2 JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandoned="true"  
  removeAbandonedTimeout="30"  
  maxActive="10"  
  maxIdle="1"  
  maxWait="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="org.h2.Driver"  
  url="jdbc:h2:file:<H2_DB_FILE_PATH>"/>
```

For Tomcat 8.x and 9.x (Apache DBCP 2):

```
<!--  
  H2 JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandonedOnBorrow="true"  
  removeAbandonedTimeout="30"  
  maxTotal="10"  
  maxIdle="1"  
  maxWaitMillis="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="org.h2.Driver"  
  url="jdbc:h2:file:<H2_DB_FILE_PATH>"/>
```

Please note that H2 can be configured to run in various operating modes by adjusting the database URL value. For details, please refer to the H2 documentation at http://www.h2database.com/html/features.html#database_url.

4.3.3 Microsoft SQL Server

For Tomcat 7.x (Apache DBCP 1):

```
<!--  
  MS SQL Server JDBC data-source used by QuartzDesk Web  
  Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandoned="true"  
  removeAbandonedTimeout="30"  
  maxActive="10"  
  maxIdle="1"  
  maxWait="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"  
  url="jdbc:sqlserver://DB_HOST:DB_PORT;databaseName=quartzdesk"/>
```

For Tomcat 8.x and 9.x (Apache DBCP 2):

```
<!--  
  MS SQL Server JDBC data-source used by QuartzDesk Web  
  Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandonedOnBorrow="true"  
  removeAbandonedTimeout="30"  
  maxTotal="10"  
  maxIdle="1"  
  maxWaitMillis="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"  
  url="jdbc:sqlserver://DB_HOST:DB_PORT;databaseName=quartzdesk"/>
```

4.3.4 MySQL

For Tomcat 7.x (Apache DBCP 1):

```
<!--  
  MySQL JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandoned="true"  
  removeAbandonedTimeout="30"  
  maxActive="10"  
  maxIdle="1"  
  maxWait="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="com.mysql.jdbc.Driver"  
  url="jdbc:mysql://DB_HOST:DB_PORT/quartzdesk  
?cachePrepStmts=true"/>
```

For Tomcat 8.x and 9.x (Apache DBCP 2):

```
<!--  
  MySQL JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandonedOnBorrow="true"  
  removeAbandonedTimeout="30"  
  maxTotal="10"  
  maxIdle="1"  
  maxWaitMillis="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="com.mysql.jdbc.Driver"  
  url="jdbc:mysql://DB_HOST:DB_PORT/quartzdesk  
?cachePrepStmts=true"/>
```

4.3.5 Oracle

For Tomcat 7.x (Apache DBCP 1):

```
<!--  
  Oracle JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandoned="true"  
  removeAbandonedTimeout="30"  
  maxActive="10"  
  maxIdle="1"  
  maxWait="2000"  
  validationQuery="select 1 from dual"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="oracle.jdbc.OracleDriver"  
  url="jdbc:oracle:thin:@DB_HOST:DB_PORT"/>
```

For Tomcat 8.x and 9.x (Apache DBCP 2):

```
<!--  
  Oracle JDBC data-source used by QuartzDesk Web Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandonedOnBorrow="true"  
  removeAbandonedTimeout="30"  
  maxTotal="10"  
  maxIdle="1"  
  maxWaitMillis="2000"  
  validationQuery="select 1 from dual"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="oracle.jdbc.OracleDriver"  
  url="jdbc:oracle:thin:@DB_HOST:DB_PORT"/>
```

4.3.6 PostgreSQL

For Tomcat 7.x (Apache DBCP 1):

```
<!--  
  PostgreSQL JDBC data-source used by QuartzDesk Web  
  Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandoned="true"  
  removeAbandonedTimeout="30"  
  maxActive="10"  
  maxIdle="1"  
  maxWait="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="org.postgresql.Driver"  
  url="jdbc:postgresql://DB_HOST:DB_PORT/quartzdesk"/>
```

For Tomcat 8.x and 9.x (Apache DBCP 2):

```
<!--  
  PostgreSQL JDBC data-source used by QuartzDesk Web  
  Application.  
-->  
<Resource name="jdbc/QuartzDeskDS"  
  auth="Container"  
  type="javax.sql.DataSource"  
  removeAbandonedOnBorrow="true"  
  removeAbandonedTimeout="30"  
  maxTotal="10"  
  maxIdle="1"  
  maxWaitMillis="2000"  
  validationQuery="select 1"  
  poolPreparedStatements="true"  
  username="DB_USER"  
  password="DB_PASSWORD"  
  driverClassName="org.postgresql.Driver"  
  url="jdbc:postgresql://DB_HOST:DB_PORT/quartzdesk"/>
```

4.4 Application Work Directory

Create a QuartzDesk Web Application work directory (`WORK_DIR`) anywhere on the local file system. The directory should be readable and writable by the user the Tomcat process runs under (typically `tomcat` on Unix/Linux systems).

Copy your QuartzDesk license key file (`license.key`) to `WORK_DIR`.



You can obtain a free 30-day trial license key at www.quartzdesk.com (go to Try / Purchase > Get Trial License Key).

Open the QuartzDesk Web Application archive (`quartzdesk-web-x.y.z.war`) and copy all files from the `extras/work` directory into `WORK_DIR`.



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

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

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.657]
(c) 2019 Microsoft Corporation. All rights reserved.

d:\var\quartzdesk-web.work\4.0.x>dir
Volume in drive D is DISK_D
Volume Serial Number is 7A4F-989B

Directory of d:\var\quartzdesk-web.work\4.0.x

2020-02-19  15:08    <DIR>          .
2020-02-19  15:08    <DIR>          ..
2020-02-19  15:10    <DIR>          customization
2020-02-19  15:10             4,259 license.key
2020-02-19  15:10             4,831 logback.xml
2020-02-19  15:10             6,511 quartzdesk-web.properties
            3 File(s)              15,601 bytes
            3 Dir(s)  2,853,464,084,480 bytes free

d:\var\quartzdesk-web.work\4.0.x>
    
```

4.5 Application Configuration

Open the QuartzDesk Web Application configuration file `WORK_DIR/quartzdesk-web.properties`.

Based on the type and version of the database created in 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 (MyISAM)	>= 5.6	mysql
MySQL (InnoDB)	>= 5.6	mysql_innodb
Oracle	== 8i	oracle8
Oracle	>= 9i	oracle9
PostgreSQL	== 8.1	postgres81
PostgreSQL	>= 8.2	postgres82

Optionally, you can adjust the QuartzDesk Web Application logging parameters by editing the `WORK_DIR/logback.xml` configuration file. The default sample `logback.xml` configuration file makes QuartzDesk Web Application log under the `WORK_DIR/logs` directory that is automatically created when the web application starts. Please refer to the [Logback Manual](#) for Logback configuration details.

4.6 Stop Tomcat

Stop Tomcat by executing the following command:

Windows

```
TOMCAT_HOME\bin\shutdown.bat
```

If Tomcat is configured as a Windows service, open the Services management console and stop the Tomcat service from the console.

Unix / Linux

```
TOMCAT_HOME/bin/shutdown.sh
```

If Tomcat is configured as a System V service with an init script, use the following command:

```
service <tomcat_service_name> stop
```

Wait for the action to complete.

Make sure the Tomcat process has been successfully stopped.

4.7 Deploy Application

Open the QuartzDesk Web Application archive (`quartzdesk-web-x.y.z.war`) and copy the `extras/tomcat/quartzdesk.xml` file into the `TOMCAT_HOME/conf/[enginename]/[hostname]` directory (typically `TOMCAT_HOME/conf/Catalina/localhost`).

Open the `TOMCAT_HOME/conf/[enginename]/[hostname]/quartzdesk.xml` file in a text editor (e.g. Notepad) and change the default value of the `quartzdesk-web.work.dir` parameter to the actual QuartzDesk Web Application work directory location. Save the changed file and close the text editor.



Alternatively, you can remove the `quartzdesk-web.work.dir` parameter from the `quartzdesk.xml` file and instead set the `quartzdesk-web.work.dir` JVM system property in one of the Tomcat startup scripts, typically in `TOMCAT_HOME\bin\setenv.bat` (Windows), or `TOMCAT_HOME/bin/setenv.sh` (Unix/Linux).

Windows

```
set CATALINA_OPTS=%CATALINA_OPTS% -Dquartzdesk-  
web.work.dir=WORK_DIR
```

Unix / Linux

```
CATALINA_OPTS=${CATALINA_OPTS} -Dquartzdesk-  
web.work.dir=WORK_DIR
```

Rename the QuartzDesk Web Application WAR file (`quartzdesk-web-x.y.z.war`) to `quartzdesk.war` and copy it to `TOMCAT_HOME/webapps`.

4.8 Start Tomcat

Start Tomcat by executing the following command:

Windows

```
TOMCAT_HOME\bin\startup.bat
```

If Tomcat is configured as a Windows service, open the Services management console and start the Tomcat service from the console.

Unix / Linux

```
TOMCAT_HOME/bin/startup.sh
```

If Tomcat is configured as a System V service with an init script, use the following command:

```
service <tomcat_service_name> stop
```

Wait for the action to complete.

Make sure the Tomcat process has been successfully started by checking the catalina log file in `TOMCAT_HOME/logs`. If Tomcat has been successfully started, the following lines are present in the log file:

```
Feb 20, 2020 04:36:11 PM org.apache.catalina.startup.Catalina start  
INFO: Server startup in 30053 ms
```

Check all Tomcat logs under `TOMCAT_HOME/logs` for errors.

Check QuartzDesk Web Application logs for errors.

If there are no errors, point your browser to http://TOMCAT_HTTP_HOST:TOMCAT_HTTP_PORT/quartzdesk/ and verify that the QuartzDesk Web Application's GUI is accessible.

Check the version number of deployed QuartzDesk Web Application.



To log in, use the default administrator login credentials:

Username: admin

Password: admin123

Once logged in, you can go to Settings > Users to manage users with access to the QuartzDesk Web Application's GUI. Users can be assigned different access permissions based on their intended roles.

In Settings > Groups, you can manage groups and assign access permissions to these groups. A group can contain users (members) who inherit access permissions of the group. A user can be a member of any number of groups.

Effective access permissions of a user are permissions associated directly with the user plus access permissions of all groups the user is a member of.



5. Upgrading

5.1 Stop Tomcat

Stop Tomcat by following the steps outlined in 4.6.

5.2 Backup

Backup your QuartzDesk Web Application database. We recommend performing a **full database backup**.

Backup the contents of the QuartzDesk Web Application work directory.

Backup the existing QuartzDesk Web Application WAR file –
`TOMCAT_HOME/webapps/quartzdesk.war`.

Store the backup files in a safe place so you can restore the original QuartzDesk Web Application version if the need arises.

5.3 Remove Existing Application

Delete the existing QuartzDesk Web Application WAR file –
`TOMCAT_HOME/webapps/quartzdesk.war`.

Delete the `TOMCAT_HOME/webapps/quartzdesk` directory if it exists.

5.4 Deploy New Application

Deploy the new version of QuartzDesk Web Application by following steps outlined in 4.7.

5.5 Start Tomcat

Start Tomcat by following the steps outlined in 4.8.

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

To upgrade QuartzDesk Web Application 2.x to 3.x, follow the upgrade steps outlined in 5.

Before deploying the new QuartzDesk Web Application WAR file (`quartzdesk-web-x.y.z.war`), as outlined in 5.4, make sure you have implemented changes described in this chapter.

6.1 Minimum Required Java Version

QuartzDesk Web Application 3.x requires Java 7 or higher. Make sure Tomcat is configured to use Java 7 or higher.

6.2 Rename Configuration File

The name of the QuartzDesk Web Application 3.x configuration file has changed from `quartzdesk.properties` to `quartzdesk-web.properties`.

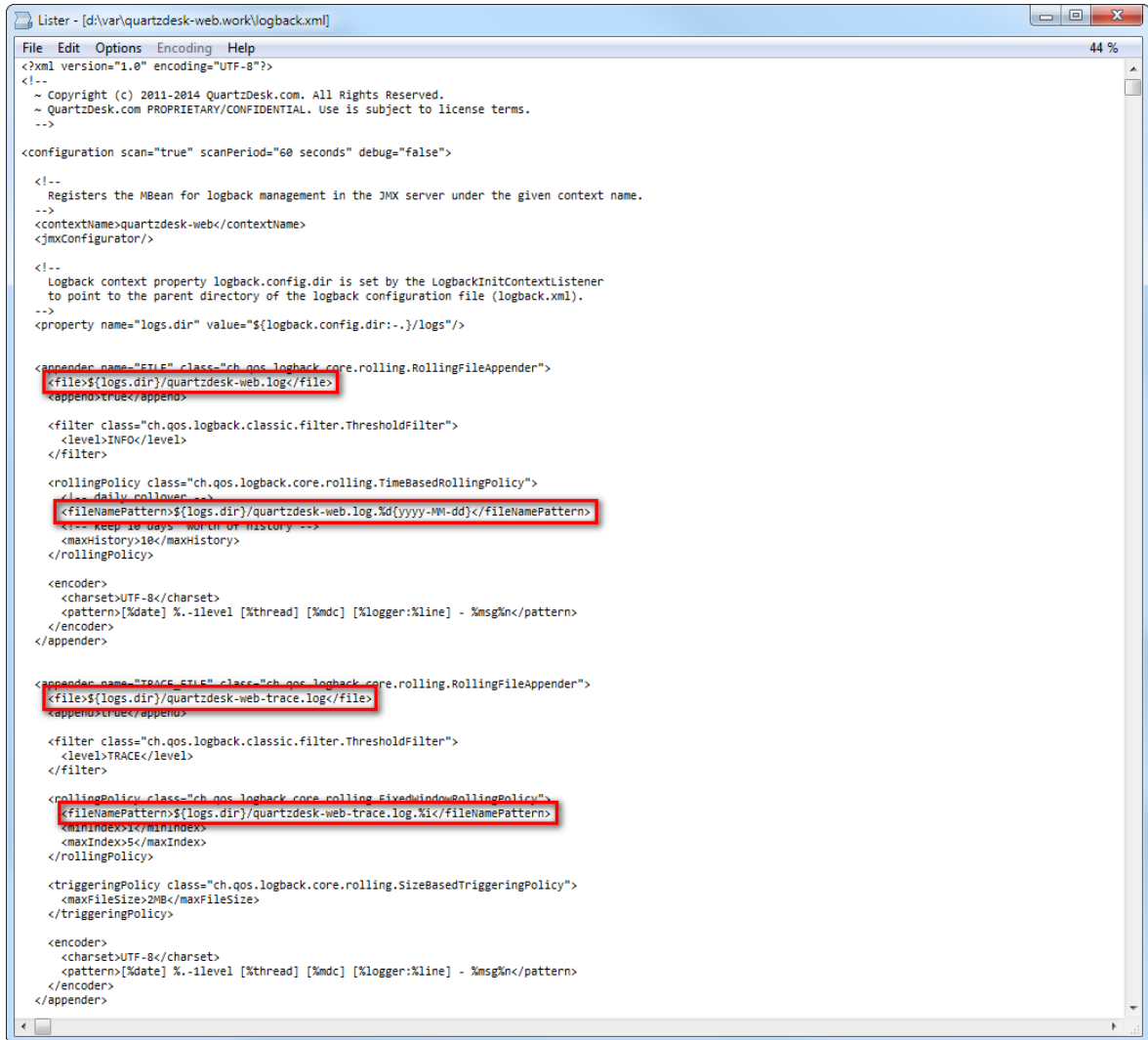
Rename the existing configuration file `quartzdesk.properties` located in the QuartzDesk Web Application work directory.

6.3 Rename Log Files

The names of QuartzDesk Web Application 3.x log files have changed.

Original Log File Name (2.x)	New Log File Name (3.x)
<code>quartzdesk.log</code>	<code>quartzdesk-web.log</code>
<code>quartzdesk-trace.log</code>	<code>quartzdesk-web-trace.log</code>

To use these new log file names, edit the QuartzDesk Web Application logging configuration file (`WORK_DIR/logback.xml`) and change the following lines:



```
File Edit Options Encoding Help
<?xml version="1.0" encoding="UTF-8"?>
<!--
~ Copyright (c) 2011-2014 QuartzDesk.com. All Rights Reserved.
~ QuartzDesk.com PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.
-->

<configuration scan="true" scanPeriod="60 seconds" debug="false">

  <!--
  Registers the MBean for logback management in the JMX server under the given context name.
  -->
  <contextName>quartzdesk-web</contextName>
  <JMXConfigurator/>

  <!--
  Logback context property logback.config.dir is set by the LogbackInitContextListener
  to point to the parent directory of the logback configuration file (logback.xml).
  -->
  <property name="logs.dir" value="${logback.config.dir:./}/logs"/>

  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${logs.dir}/quartzdesk-web.log</file>
    <append>true</append>

    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
      <level>INFO</level>
    </filter>

    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <!-- daily rollover -->
      <fileNamePattern>${logs.dir}/quartzdesk-web.log.%d{yyyy-MM-dd}</fileNamePattern>
      <!-- keep 10 days' worth of history -->
      <maxHistory>10</maxHistory>
    </rollingPolicy>

    <encoder>
      <charset>UTF-8</charset>
      <pattern>[%date] %.-1level [%thread] [%mdc] [%logger:%line] - %msg%n</pattern>
    </encoder>
  </appender>

  <appender name="TRACE_FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${logs.dir}/quartzdesk-web-trace.log</file>
    <append>true</append>

    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
      <level>TRACE</level>
    </filter>

    <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
      <fileNamePattern>${logs.dir}/quartzdesk-web-trace.log.%i</fileNamePattern>
      <minIndex>1</minIndex>
      <maxIndex>5</maxIndex>
    </rollingPolicy>

    <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
      <maxFileSize>2MB</maxFileSize>
    </triggeringPolicy>

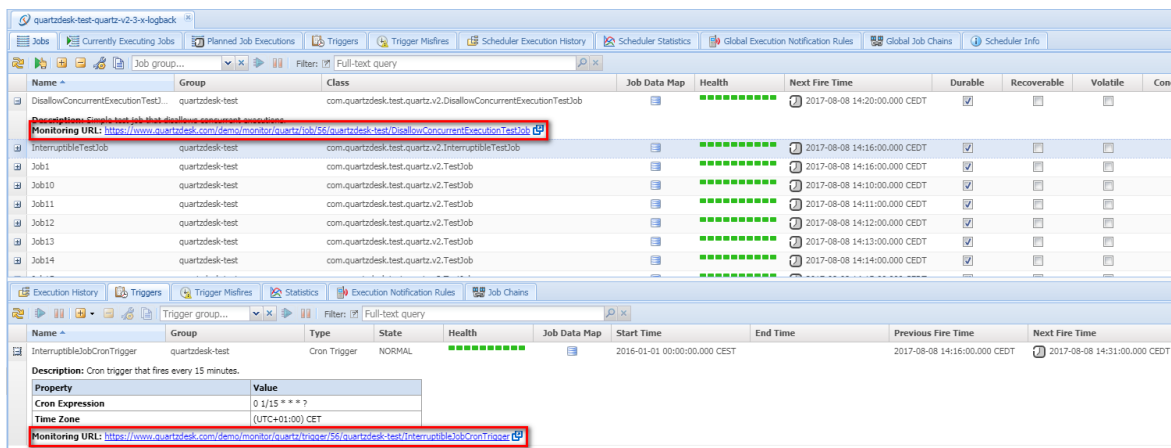
    <encoder>
      <charset>UTF-8</charset>
      <pattern>[%date] %.-1level [%thread] [%mdc] [%logger:%line] - %msg%n</pattern>
    </encoder>
  </appender>
```

Alternatively, extract the default `logback.xml` configuration file from the QuartzDesk Web Application 3.x WAR (`quartzdesk-web-x.y.z.war/extras/work/logback.xml`) and copy it to `WORK_DIR`.

6.4 Access to Monitoring URLs (REST API)

In QuartzDesk Web Application 2.x, the monitoring REST API URLs could be accessed by users with the QuartzDeskMonitor J2EE security role. In QuartzDesk Web Application 3.x, these monitoring URLs can be accessed by all authenticated users.





We recommend that you create a dedicated user account to access these monitoring URLs. The user account can be created in Settings → Users in the QuartzDesk Web Application’s GUI.



All monitoring URLs in QuartzDesk Web Application 3.x support the HTTP Basic authentication scheme where the user’s authentication credentials are passed in the `Authorization` HTTP header. Please note that the same authentication scheme was used by monitoring URLs in QuartzDesk Web Application 2.x.

6.5 Access to JAX-WS Endpoints

In QuartzDesk Web Application 2.x, all JAX-WS web service endpoints could be accessed by users with the QuartzDeskService J2EE security role. In QuartzDesk Web Application 3.x, these web service end points can only be accessed by authenticated users with particular access permissions.

The following table lists all JAX-WS web services and the security permissions that are required to access these web services.

JAX-WS Service	Required Permission
Connection Service	WS_CONNECTION
Security Service	WS_SECURITY
Quartz Service	WS_QUARTZ
Quartz Execution History Service	WS_QUARTZ_EXEC_HISTORY
Quartz Execution Notification Rule Service	WS_QUARTZ_EXEC_NOTIF_RULE
Quartz Job Chain Service	WS_QUARTZ_JOB_CHAIN

We recommend that you create a dedicated user account to access these JAX-WS endpoints. The user account can be created in Settings → Users in the QuartzDesk Web Application’s GUI. Do not forget to assign the user the relevant permission(s).



All JAX-WS web service endpoints in QuartzDesk Web Application 3.x support the HTTP Basic authentication scheme where the user’s authentication credentials are passed in the `Authorization` HTTP header. Please note that the same authentication scheme was used by JAX-WS endpoints in QuartzDesk Web Application 2.x.

6.6 Remove Unused Users

QuartzDesk Web Application 3.x no longer uses the J2EE role-based security model. Therefore, you can remove all previously registered users, who were used to access QuartzDesk Web Application 2.x, from the `TOMCAT_HOME/conf/tomcat-users.xml` file.

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  ...
  No longer needed.
  <!-- QuartzDesk web application roles -->
  <user username="user" password="password" roles="QuartzDeskUser"/>
  <user username="monitor" password="password" roles="QuartzDeskMonitor"/>
  <user username="service" password="password" roles="QuartzDeskService"/>
</tomcat-users>
```


7. QuartzDesk 3.x to 4.x Migration Notes

No configuration changes are required.



8. Cluster Deployment Notes

When deploying QuartzDesk Web Application to a Tomcat cluster you need to follow the configuration steps described in preceding chapters. In addition to these, there are several extra configuration steps that must be performed for a cluster deployment.

8.1 HTTP Session Replication and Affinity

QuartzDesk Web Application makes use of HTTP sessions and to store some short-lived and user-specific data. To achieve high-availability (HA), it is necessary to make the session data available on all application cluster members so that when one cluster member becomes unavailable, the remaining cluster members can take over and handle user requests without the user noticing any service interruption. To make the session data available on all application cluster members, the HTTP session replication process must be enabled on the cluster.



The amount of data stored by QuartzDesk Web Application in an HTTP session is kept at the absolute minimum to reduce the session replication overhead. The total size of data stored in the session does not exceed 1KB.

When configuring session replication, we recommend that you also enable session affinity (sticky-sessions) on the load-balancer so that all user requests are preferably passed to the Tomcat instance that handled the first user request that established the session.

Please refer to the Tomcat documentation for details on how to configure session replication and session affinity because the actual steps may vary depending on the Tomcat cluster topology and configuration.

8.2 Shared Work Directory

We recommend that you put the QuartzDesk Web Application work directory, described in 4.4, on a shared drive and make this work directory available to all cluster members. Not only does this make application and configuration upgrading easier, it is actually required by all “Save” (for example, Save Log, Save Chart etc.) actions provided by the QuartzDesk Web Application’s GUI. These actions trigger two subsequent HTTP requests where the first request prepares the data and stores it in the `WORK_DIR/tmp` directory and the second request downloads the data and makes the browser open the Save As dialog.

During a fail-over or if the session affinity is not enabled, it can easily happen that the first request is handled by cluster member A and the second request is handled by cluster member B. If A and B are not configured to use the same `WORK_DIR/tmp` directory, then B will fail to serve the data prepared by A during the preceding request because the data will not be found.

8.3 Logging Configuration

If you set up your cluster to use a shared QuartzDesk Web Application work directory, as described in the previous chapter, you will need to edit the QuartzDesk Web Application logging configuration file `WORK_DIR/logback.xml` and decide where QuartzDesk Web Application instances running on individual cluster members should log. There are two options:

- 1) Logging into the same (shared) log files.
- 2) Logging into separate log files.

QuartzDesk Web Application uses two log files – `quartzdesk-web.log` and `quartzdesk-web-trace.log` that are stored in `WORK_DIR/logs` directory. The following chapters discuss these two options.

8.3.1 Using Shared Log Files

In order to make individual QuartzDesk Web Application instances log into the same log files, you must enable the prudent mode on both file appenders used in the `WORK_DIR/logback.xml` configuration file:

```
...

<appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web.log</file>
  <append>true</append>
  <prudent>true</prudent>
  ...
</appender>

<appender name="TRACE_FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web-trace.log</file>
  <append>true</append>
  <prudent>true</prudent>
  ...

<!--
  We must use the TimeBasedRollingPolicy because the
  FixedWindowRollingPolicy is not supported in prudent mode!
-->
<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
  <!-- daily rollover -->
  <fileNamePattern>${logs.dir}/quartzdesk-web.log.%d{yyyy-MM-
dd}</fileNamePattern>
  <!-- keep 10 days' worth of history -->
  <maxHistory>10</maxHistory>
</rollingPolicy>

<!--
  The SizeBasedTriggeringPolicy removed because it is used only in
  conjunction with the FixedWindowRollingPolicy.
-->

<encoder>
  <charset>UTF-8</charset>
  <pattern>[%date] %.-1level [%thread] [%mdc] [%logger:%line] -
%msg%n</pattern>
</encoder>
</appender>

...
```

For details on the Logback prudent mode, please refer to <http://logback.qos.ch/manual/appenders.html#FileAppender>.



Because prudent mode relies on exclusive file locks to manage concurrent access to the log files and these locks can have negative impact on the QuartzDesk Web Application's performance, we generally discourage using the prudent mode and shared log files.

8.3.2 Using Separate Log Files

In order to make individual QuartzDesk Web Application instances log into separate log files, you can use a JVM system property set on all cluster member JVMs. The value of this property must be unique for all cluster members. The property can be referred to from the `WORK_DIR/logback.xml` logging configuration file.

The following examples assume the use of the `cluster.member.instanceId` JVM system property, but any JVM system property name can be used.

There are two common approaches as to where the separate log files produced by individual QuartzDesk Web Application instances are stored:

- 1) Log files created under a common log root directory.

```
...
<appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web--${cluster.member.instanceId}.log</file>
  <append>true</append>
...
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- daily rollover -->
    <fileNamePattern>${logs.dir}/quartzdesk-web-
-${cluster.member.instanceId}.log.%d{yyyy-MM-dd}</fileNamePattern>
    <!-- keep 10 days' worth of history -->
    <maxHistory>10</maxHistory>
  </rollingPolicy>
...
</appender>

<appender name="TRACE_FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web--${cluster.member.instanceId}-
trace.log</file>
  <append>true</append>
...
  <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
  <fileNamePattern>${logs.dir}/quartzdesk-web-
-${cluster.member.instanceId}-trace.log.%i</fileNamePattern>
  <minIndex>1</minIndex>
  <maxIndex>5</maxIndex>
  </rollingPolicy>
...
</appender>
...
```

2) Log files created in separate (cluster member specific) log root directories.

```

...
<!--
  Logback context property logback.config.dir is set by the
  LogbackInitContextListener to point to the parent directory of the Logback
  configuration file (logback.xml).
-->
<property name="logs.dir" value="${logback.config.dir:-
.}/${cluster.member.instanceId}/logs"/>
...

```

8.4 Internal Quartz Scheduler

QuartzDesk Web Application ships with an embedded Quartz scheduler to periodically execute its internal jobs. When deploying QuartzDesk Web Application to a cluster, it is necessary to **assign unique instance IDs to Quartz scheduler instances** running in the clustered QuartzDesk Web Application instances.

For these purposes the QuartzDesk Web Application configuration (`quartzdesk-web.properties` file) provides the `scheduler.org.quartz.scheduler.instanceIdGenerator.class` configuration property. The value of this property must be a fully-qualified class name of a Java class that implements the `org.quartz.spi.InstanceIdGenerator` Quartz API interface. Quartz API provides two out of the box implementations suitable for clustered QuartzDesk Web Application deployments:

Implementation	Description
<code>org.quartz.simpl.HostnameInstanceIdGenerator</code>	<p>This implementation is suitable for QuartzDesk Web Application deployments where individual clustered QuartzDesk Web Application instances run on distinct hosts and each of these hosts is assigned a unique hostname.</p> <p>This is the default implementation used by QuartzDesk Web Application. No configuration changes are necessary to use this instance ID generator.</p>
<code>org.quartz.simpl.SystemPropertyInstanceIdGenerator</code>	<p>This implementation is suitable for QuartzDesk Web Application deployments where some of the clustered QuartzDesk Web Application instances run on the same host.</p> <p>This implementation extracts the Quartz scheduler instance ID from the <code>org.quartz.scheduler.instanceId</code> JVM system property that must be explicitly set.</p> <p>In Tomcat it is typically necessary to add this JVM system property to <code>TOMCAT_HOME/catalina.bat</code> (<code>catalina.sh</code> on Unix/Linux), or <code>TOMCAT_HOME/setenv.bat</code> (<code>setenv.sh</code> on</p>

Unix/Linux) file. Please refer to the Tomcat documentation
for details on how to add a new JVM system property.

Please refer to the table above and optionally modify the value of the
`scheduler.org.quartz.scheduler.instanceIdGenerator.class` configuration
property according to the cluster configuration.

