# QuartzDesk Web Application Installation and Upgrade Guide for Apache Tomcat 6.x, 7.x, 8.x and 9.x

**QuartzDesk Version: 2.x**

April 24, 2017

# Table of Contents

# 1. Purpose

This document describes the installation and upgrade process for the QuartzDesk web application 2.x on Apache Tomcat 6.x, 7.x, 8.x and 9.x.

If you experience any problems installing or upgrading the 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 `.ear` extension. |
| **JAR** | Java Application Archive. A file with `.jar` extension. |
| **JVM** | Java Virtual Machine. |
| **WAR** | Web Application Archive. A file with `.war` extension. |

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

| Location / Property | Example | Description |
|---|---|---|
| **DB_HOST** | mydb.foo.com | QuartzDesk database server host. |
| **DB_PORT** | 5432 | QuartzDesk database server port. |
| **DB_NAME** | quartzdesk | QuartzDesk database name. |
| **DB_SCHEMA** | quartzdesk | QuartzDesk database schema. |
| **DB_USER** | quartzdesk | QuartzDesk database user. |
| **DB_PASSWORD** | quartzdesk | QuartzDesk 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 | QuartzDesk work directory. |

# 3. Requirements

## 3.1 Software Requirements

### 3.1.1 Browser

The QuartzDesk web application requires a modern JavaScript-enabled browser. Please make sure JavaScript is enabled and not blocked by third party anti-virus/anti-malware software.

The QuartzDesk web application has been tested with the following browser versions. These are also the minimum browsers versions required.

| Browser | Minimum Version |
|---|---|
| **Chrome** | 10 |
| **FireFox** | 3.6 |
| **Internet Explorer** | 8 |
| **Opera** | 11 |
| **Safari** | 6 |

### 3.1.2 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.

### 3.1.3 Java

Sun/Oracle Java (JDK) 6, 7, 8.
IBM Java (JDK) 6, 7, 8.
OpenJDK 6, 7, 8.

### 3.1.4 Application Server

Apache Tomcat 6.x.
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.<br><br>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 the 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.<br><br>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, you need to obtain the QuartzDesk WAR. 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 runs on any physical or virtualized hardware that supports the above software requirements.

# 4. Installation

This chapter describes the standard QuartzDesk installation. If you are only evaluating QuartzDesk, you may be interested in the **one-step installation mode** to dramatically reduce the number of required installation steps. For details, please refer to our FAQs (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 database named `quartzdesk`[1] ( DB_NAME) owned by the DB_USER.

In the QuartzDesk database create a new schema named `quartzdesk` (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 refer to the database engine documentation for details on how to perform the above database operations as they are all database-specific.

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

## 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 is running under (typically `tomcat` on Unix / Linux systems).

Alternatively you can create a directory for 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" classloader classpath. 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,${ca
talina.home}/lib,${catalina.home}/lib/*.jar
```

[1] DB2 restricts the database name length 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,${ca
talina.home}/lib,${catalina.home}/lib/*.jar,${catalina.base}/lib/e
xt,${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 6.x and 7.x (Apache DBCP 1):

```
<!--
  DB2 JDBC data-source used by the 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 the 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

We recommend using H2 for evaluation and/or experimental purposes only. We strongly discourage using H2 in production environments.

For Tomcat 6.x and 7.x (Apache DBCP 1):

```
<!--
  H2 JDBC data-source used by the 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 the 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 6.x and 7.x (Apache DBCP 1):

```
<!--
  MS SQL Server JDBC data-source used by the QuartzDesk
  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;applic
ationName=QuartzDesk"/>
```

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

```
<!--
  MS SQL Server JDBC data-source used by the QuartzDesk
  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;applic
ationName=QuartzDesk"/>
```

### 4.3.4 MySQL

For Tomcat 6.x and 7.x (Apache DBCP 1):

```
<!--
  MySQL JDBC data-source used by the 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 the 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 6.x and 7.x (Apache DBCP 1):

```
<!--
  Oracle JDBC data-source used by the 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 the 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 6.x and 7.x (Apache DBCP 1):

```
<!--
  PostgreSQL JDBC data-source used by the QuartzDesk
  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 the QuartzDesk
  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 work directory (WORK_DIR) anywhere on the local file system. The directory should be readable and writeable by the user the Tomcat is running under (typically `tomcat` on Unix/Linux systems).

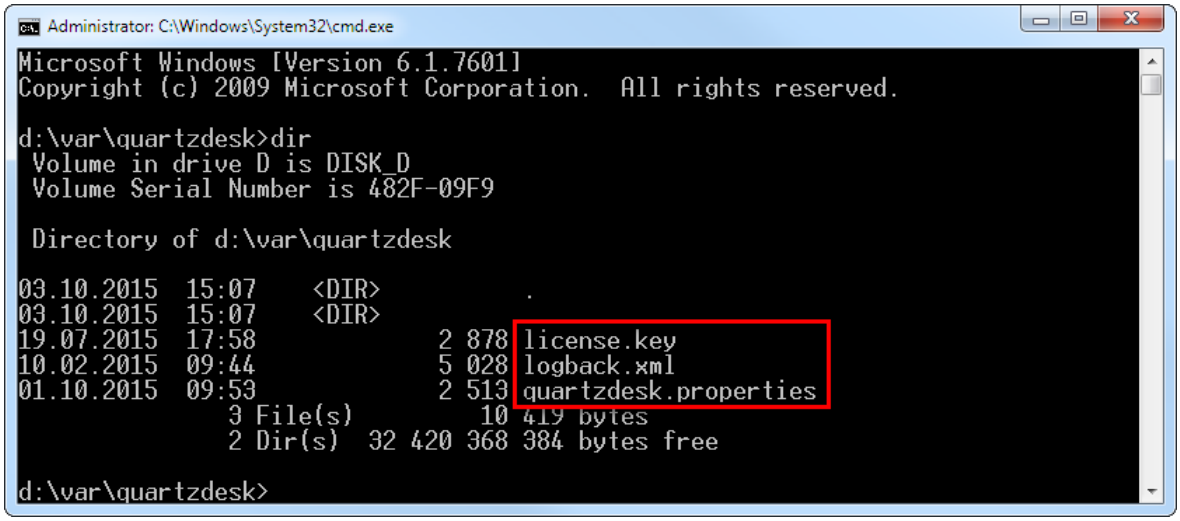
Copy your QuartzDesk license key file (`license.key`) to WORK_DIR.

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

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 work directory correctly set up on a Microsoft Windows machine.




## 4.5 Application Configuration

Open the QuartzDesk configuration file `WORK_DIR/quartzdesk.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 (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 logging parameters by editing the `WORK_DIR/logback.xml` configuration file. The default sample `logback.xml` configuration file makes QuartzDesk log under the `WORK_DIR/logs` directory that is automatically created when QuartzDesk 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.work.dir` parameter to the actual QuartzDesk work directory location. Save the changed file and close the text editor.

Alternatively you can remove the quartzdesk.work.dir parameter from the quartzdesk.xml file and instead set the `quartzdesk.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.work.dir=WORK_DIR
```

**Unix / Linux**
```
CATALINA_OPTS=${CATALINA_OPTS} -Dquartzdesk.work.dir=WORK_DIR
```

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

## 4.8 Security

QuartzDesk supports the HTTP/S Basic authentication scheme to authenticate users who access the application. To configure application security, map the following QuartzDesk security roles to users and/or groups defined in the configured Tomcat realm(s).

| Security Role | Description |
|---|---|
| **QuartzDeskUser** | Role required to access the QuartzDesk web application UI (QuartzDesk GUI). |

| | |
|---|---|
| **QuartzDeskMonitor** | Role required to access the scheduler, job and trigger monitoring URLs (REST API). |
| **QuartzDeskService** | Role required to access QuartzDesk web-services (e.g. the QuartzAnywhere web-service). |

This document assumes that Tomcat is configured with the default UserDatabase realm backed by the `conf/tomcat-users.xml` file that stores individual users and user-role mappings.

Open the `TOMCAT_HOME/conf/tomcat-users.xml` file in a text editor (e.g. Notepad) and add users who will access the QuartzDesk web application and associate these users with the above QuartzDesk security roles.

The following snippet is an example of the `tomcat-users.xml` file defining 3 users (user, monitor and service), each associated with one of the QuartzDesk security roles:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  ...
  <!-- 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>
```

Save the changes and exit the text editor.

## 4.9 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:

```
Sep 30, 2015 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 the 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 works.

# 5. Upgrading

## 5.1 Stop Tomcat

Stop Tomcat by following the steps outlined in 4.6.

## 5.2 Backup

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

Backup the contents of the QuartzDesk 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 QuartzDesk web application by following steps outlined in 4.7.

## 5.5 Start Tomcat

Start Tomcat by following the steps outlined in 4.9.

When the QuartzDesk web application starts, it automatically attempts to apply all necessary database alter (upgrade) scripts to the QuartzDesk database. Please check the Tomcat logs and 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 is running.

Check the version number of the deployed QuartzDesk web application to make sure the application has been successfully upgraded. For details on how to find out the version number of a deployed QuartzDesk web application, please refer to our FAQs at www.quartzdesk.com (click Support → FAQs and search for “find out version”).

# 6. Cluster Deployment Notes

When deploying the 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.

## 6.1 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.

## 6.2 Shared Work Directory

We recommend that you put the QuartzDesk work directory, described in chapter 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 requited by all “Save” (for example, Save Log, Save Chart etc.) actions provided by the QuartzDesk web application 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.

## 6.3 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.log and quartzdesk-trace.log that are stored in `WORK_DIR/logs` directory. The following chapters discuss these two options.

### 6.3.1 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.log</file>
  <append>true</append>
  <prudent>true</prudent>
  ...
</appender>


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

### 6.3.2 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>${logs.dir}/quartzdesk-${cluster.member.instanceId}.log</file>
  <append>true</append>

  ...

  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- daily rollover -->
    <fileNamePattern>${logs.dir}/quartzdesk-
${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>${logs.dir}/quartzdesk-${cluster.member.instanceId}-trace.log</file>
  <append>true</append>

  ...

  <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>${logs.dir}/quartzdesk-${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"/>

...
```

## 6.4 Internal Quartz Scheduler

QuartzDesk web application ships with an embedded Quartz scheduler to periodically execute its internal jobs. When deploying the 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.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 |
|---|---|
| **`org.quartz.simpl.HostnameInstanceIdGenerator`** | 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.<br><br>This is the default implementation used by QuartzDesk. No QuartzDesk configuration changes are necessary to use this instance ID generator. |
| **`org.quartz.simpl.SystemPropertyInstanceIdGenerator`** | This implementation is suitable for QuartzDesk web application deployments where some of the clustered QuartzDesk web application instances run on the same host.<br><br>This implementation extracts the Quartz scheduler instance ID from the `org.quartz.scheduler.instanceId` JVM system property that must be explicitly set.<br><br>In Tomcat it is typically necessary to add this JVM system property to `TOMCAT_HOME/catalina.bat` (`catalina.sh` on Unix/Linux), or `TOMCAT_HOME/setenv.bat` (`setenv.sh` on 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.