



# QuartzDesk Web Application Installation and Upgrade Guide for Oracle GlassFish AS 4.x

QuartzDesk Version: 2.x

April 24, 2017



## 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>Browser</i> .....	5
3.1.2	<i>Operating System</i> .....	5
3.1.3	<i>Java</i> .....	5
3.1.4	<i>Application Server</i> .....	5
3.1.5	<i>Database</i> .....	5
3.1.6	<i>Database JDBC Driver</i> .....	5
3.1.7	<i>QuartzDesk Web Application Archive</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	JDBC CONNECTION POOL .....	7
4.3.1	<i>DB2</i> .....	7
4.3.2	<i>H2</i> .....	8
4.3.3	<i>Microsoft SQL Server</i> .....	9
4.3.4	<i>MySQL</i> .....	10
4.3.5	<i>Oracle</i> .....	11
4.3.6	<i>PostgreSQL</i> .....	12
4.4	TEST JDBC CONNECTION POOL .....	13
4.5	JDBC RESOURCE .....	13
4.6	APPLICATION WORK DIRECTORY .....	14
4.7	APPLICATION CONFIGURATION .....	15
4.8	SECURITY .....	16
4.8.1	<i>Enable Default Principal to Role Mapping</i> .....	16
4.8.2	<i>Add Users</i> .....	18
4.9	DEPLOY APPLICATION .....	18
4.10	START APPLICATION .....	20
<b>5.</b>	<b>UPGRADING</b> .....	<b>21</b>
5.1	STOP EXISTING APPLICATION .....	21
5.2	BACKUP .....	21
5.3	REMOVE EXISTING APPLICATION .....	21
5.4	DEPLOY NEW APPLICATION .....	21
5.5	START NEW APPLICATION .....	22
<b>6.</b>	<b>CLUSTER DEPLOYMENT NOTES</b> .....	<b>23</b>
6.1	HTTP SESSION REPLICATION AND AFFINITY .....	23
6.2	SHARED WORK DIRECTORY .....	23
6.3	LOGGING CONFIGURATION .....	23
6.3.1	<i>Using Shared Log Files</i> .....	24
6.3.2	<i>Using Separate Log Files</i> .....	25
6.4	INTERNAL QUARTZ SCHEDULER .....	26

## 1. Purpose

This document describes the installation and upgrade process for the QuartzDesk web application 2.x on Oracle GlassFish Application Server 4.x.

If you experience any problems installing or upgrading the QuartzDesk web application, 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
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.
GAC	GlassFish Administrative Console.
GAS	GlassFish Application Server.
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	localhost	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.
GAS_INSTALL_ROOT	<code>/usr/local/glassfish4</code>	GlassFish Application Server installation directory.
GAS_CONFIG	server-config	GlassFish Application Server configuration.
GAS_DOMAIN_NAME	domain1	GlassFish Application Server domain name.
GAS_DOMAIN_DIR	<code>/usr/local/glassfish4/domains/domain1</code>	GlassFish Application Server domain directory.
GAS_HTTP_HOST	localhost	GlassFish HTTP listener host.
GAS_HTTP_PORT	9080	GlassFish HTTP listener port.
WORK_DIR	<code>/var/quartzdesk</code>	QuartzDesk work directory.

## 3. Requirements

### 3.1 Software Requirements

#### 3.1.1 Browser

The QuartzDesk web application GUI 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

Oracle GlassFish Application Server 4.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 <a href="http://www-01.ibm.com/support/docview.wss?uid=swg21363866">http://www-01.ibm.com/support/docview.wss?uid=swg21363866</a> .

<b>H2</b>	Database engine including the JDBC driver is available at <a href="http://www.h2database.com">http://www.h2database.com</a> .
<b>Microsoft SQL Server</b>	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 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 <a href="http://msdn.microsoft.com/en-us/library/ms187819.aspx">http://msdn.microsoft.com/en-us/library/ms187819.aspx</a> .
<b>MySQL</b>	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> .

### 3.1.7 QuartzDesk Web Application Archive

To install QuartzDesk, you need to obtain the QuartzDesk web application archive (WAR). The latest version can be downloaded at [www.quartzdesk.com](http://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 `quartzdesk1` (`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) to `GAS_DOMAIN_DIR/lib/ext` directory and restart the application server.

### 4.3 JDBC Connection Pool

In GAC (Resources → JDBC → JDBC Connection Pools) create a new JDBC connection pool for the QuartzDesk database.

The following steps depend on the QuartzDesk database type and are described in the following sub-chapters.

#### 4.3.1 DB2

In Step 1, enter the following values:

##### General Settings

---

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

Pool Name: QuartzDeskDS  
Resource Type: javax.sql.ConnectionPoolDataSource  
Database Driver Vendor: DB2  
Introspect: uncheck

Click Next.

In Step 2, enter the following values:

#### General Settings

Datasource Clasname: COM.ibm.db2.jcc.DB2ConnectionPoolDataSource  
Ping: check

#### Pool Settings

Initial and Minimum Pool Size: 2  
Maximum Pool Size: 10  
Pool Resize Quality: 2  
Idle Timeout: 300  
Max Wait Time: 5000

#### Additional Properties

portNumber: DB\_PORT  
databaseName: DB\_NAME  
serverName: DB\_HOST  
user: DB\_USER  
password: DB\_PASSWORD  
driverType: 4 (add this property if missing)

Click Finish.

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS) and click on the Advanced tab and enter the following values:

Statement Cache Size: 100

#### Connection Validation

Connection Validation: check Required  
Validation Method: table  
Table Name: sysibm.sysdummy1

Click Save.

### 4.3.2 H2



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

In Step 1, enter the following values:

#### General Settings

Pool Name: QuartzDeskDS  
Resource Type: javax.sql.DataSource  
Database Driver Vendor: leave empty



Introspect: uncheck

Click Next.

In Step 2, enter the following values:

#### General Settings

Datasource Clasname: org.h2.jdbcx.JdbcDataSource

Ping: check

#### Pool Settings

Initial and Minimum Pool Size: 2

Maximum Pool Size: 10

Pool Resize Quality: 2

Idle Timeout: 300

Max Wait Time: 5000

#### Additional Properties

User: DB\_USER

Password: DB\_PASSWORD

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](http://www.h2database.com/html/features.html#database_url).

Click Finish.

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS) and click on the Advanced tab and enter the following values:

Statement Cache Size: 100

#### Connection Validation

Connection Validation: check Required

Validation Method: table

Table Name: dual

Click Save.

### 4.3.3 Microsoft SQL Server

In Step 1, enter the following values:

#### General Settings

Pool Name: QuartzDeskDS

Resource Type: javax.sql.ConnectionPoolDataSource

Database Driver Vendor: MicrosoftSqlServer

Introspect: uncheck

Click Next.

In Step 2, enter the following values:

### General Settings

Datasource Clasname: com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource

Ping: check

### Pool Settings

Initial and Minimum Pool Size: 2

Maximum Pool Size: 10

Pool Resize Quality: 2

Idle Timeout: 300

Max Wait Time: 5000

### Additional Properties

User: DB\_USER

DatabaseName: DB\_NAME

ApplicationName: QuartzDesk

Password: DB\_PASSWORD

ServerName: DB\_HOST

PortNumber: DB\_PORT

Depending on your Microsoft SQL Server configuration, you may need to set the value of the InstanceName property.

Click Finish.

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS) and click on the Advanced tab and enter the following values:

Statement Cache Size: 100

### Connection Validation

Connection Validation: check Required

Validation Method: table

Table Name: qd\_schema\_update

Click Save.

## 4.3.4 MySQL

In Step 1, enter the following values:

### General Settings

Pool Name: QuartzDeskDS

Resource Type: javax.sql.ConnectionPoolDataSource

Database Driver Vendor: MySql

Introspect: uncheck

Click Next.

In Step 2, enter the following values:

### General Settings

Datasource Clasname: com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource

Ping: check

### **Pool Settings**

Initial and Minimum Pool Size: 2  
Maximum Pool Size: 10  
Pool Resize Quality: 2  
Idle Timeout: 300  
Max Wait Time: 5000

### **Additional Properties**

User: DB\_USER  
ServerName: DB\_HOST  
Port: DB\_PORT  
DatabaseName: DB\_NAME  
Password: DB\_PASSWORD

Click Finish.

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS) and click on the Advanced tab and enter the following values:

Statement Cache Size: 100

### **Connection Validation**

Connection Validation: check Required  
Validation Method: table  
Table Name: qd\_schema\_update

Click Save.

## **4.3.5 Oracle**

In Step 1, enter the following values:

### **General Settings**

Pool Name: QuartzDeskDS  
Resource Type: javax.sql.ConnectionPoolDataSource  
Database Driver Vendor: Oracle  
Introspect: uncheck

Click Next.

In Step 2, enter the following values:

### **General Settings**

Datasource Clasname: oracle.jdbc.pool.OracleConnectionPoolDataSource  
Ping: check

### **Pool Settings**

Initial and Minimum Pool Size: 2  
Maximum Pool Size: 10  
Pool Resize Quality: 2  
Idle Timeout: 300  
Max Wait Time: 5000

#### Additional Properties

User: DB\_USER  
DatabaseName: DB\_NAME  
Password: DB\_PASSWORD  
ServerName: DB\_HOST  
DriverType: thin  
PortNumber: DB\_PORT  
URL: jdbc:oracle:thin:@DB\_HOST:DB\_PORT:ORACLE\_SERVICE\_NAME

Click Finish.

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS) and click on the Advanced tab and enter the following values:

Statement Cache Size: 100

#### Connection Validation

Connection Validation: check Required  
Validation Method: table  
Table Name: dual

Click Save.

### 4.3.6 PostgreSQL

In Step 1, enter the following values:

#### General Settings

Pool Name: QuartzDeskDS  
Resource Type: javax.sql.ConnectionPoolDataSource  
Database Driver Vendor: Postgresql  
Introspect: uncheck

Click Next.

In Step 2, enter the following values:

#### General Settings

Datasource Clasname: org.postgresql.ds.PGConnectionPoolDataSource  
Ping: check

#### Pool Settings

Initial and Minumum Pool Size: 2  
Maximum Pool Size: 10  
Pool Resize Quality: 2  
Idle Timeout: 300  
Max Wait Time: 5000

#### Additional Properties

User: DB\_USER  
ApplicationName: QuartzDesk  
DatabaseName: DB\_NAME  
Password: DB\_PASSWORD

ServerName: DB\_HOST  
PortNumber: DB\_PORT

Click Finish.

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS) and click on the Advanced tab and enter the following values:

Statement Cache Size: 100

#### Connection Validation

Connection Validation: check Required

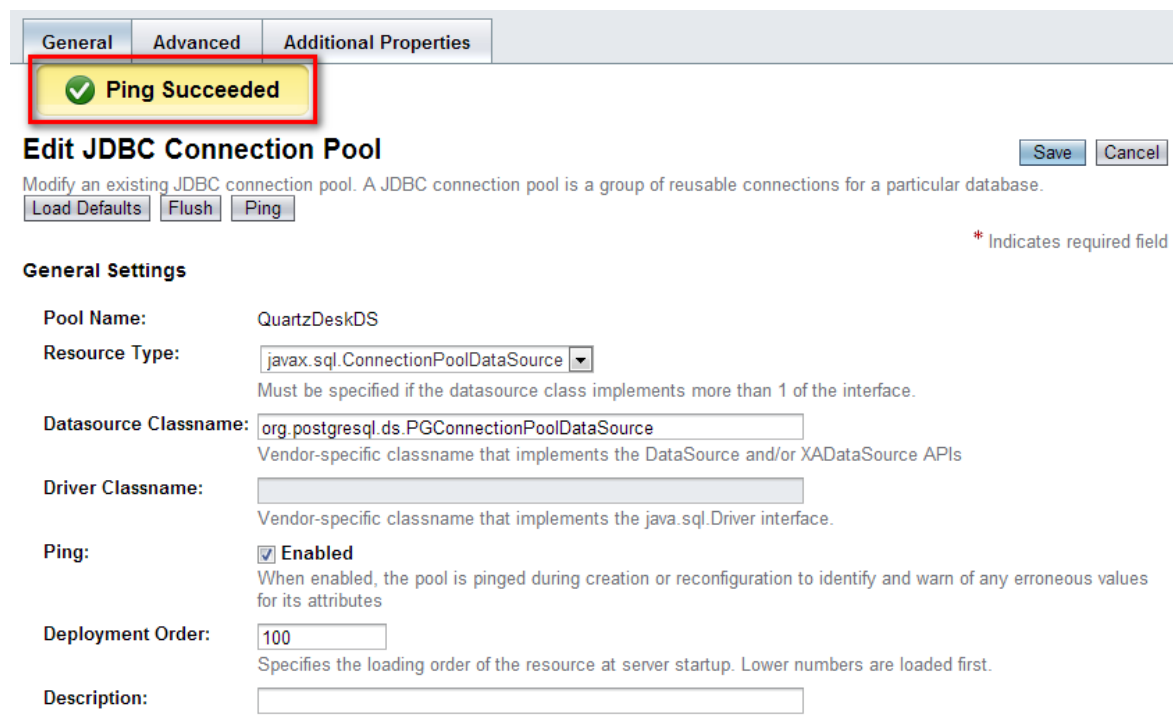
Validation Method: table

Table Name: qd\_schema\_update

Click Save.

## 4.4 Test JDBC Connection Pool

In GAC click on the created JDBC connection pool (Resources → JDBC → JDBC Connection Pools → QuartzDeskDS). In the General tab click on the Ping button to test the JDBC connection pool.



General Advanced Additional Properties

✓ Ping Succeeded

### Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

Save Cancel

Load Defaults Flush Ping

\* Indicates required field

#### General Settings

Pool Name: QuartzDeskDS

Resource Type:

Datasource Classname:

Driver Classname:

Ping:  Enabled

Deployment Order:

Description:

If the JDBC connection pool test fails, an error message is displayed and an exception is logged in the application server log (GAS\_DOMAIN\_DIR/logs/server.log).

## 4.5 JDBC Resource

In GAC (Resources → JDBC → JDBC Resources) create a new JDBC resource for the QuartzDeskDS JDBC connection pool create in the previous steps.

JNDI Name: jdbc/QuartzDeskDS

Pool Name: QuartzDeskDS

## New JDBC Resource

OK Cancel

Specify a unique JNDI name that identifies the JDBC resource you want to create. The name must contain only alphanumeric, underscore, dash, or dot characters.

JNDI Name: \*

Pool Name:

Use the JDBC Connection Pools page to create new pools

Description:

Status:  Enabled

Additional Properties (0)			
Add Property		Delete Properties	
Select	Name	Value	Description
No items found.			

Click OK.

## 4.6 Application Work Directory

Create QuartzDesk work directory (WORK\_DIR) anywhere on the local file system. The directory must be readable and writeable by the user the GAS process is running under.

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](http://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.

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

d:\var\quartzdesk>dir
Volume in drive D is DISK_D
Volume Serial Number is 482F-09F9

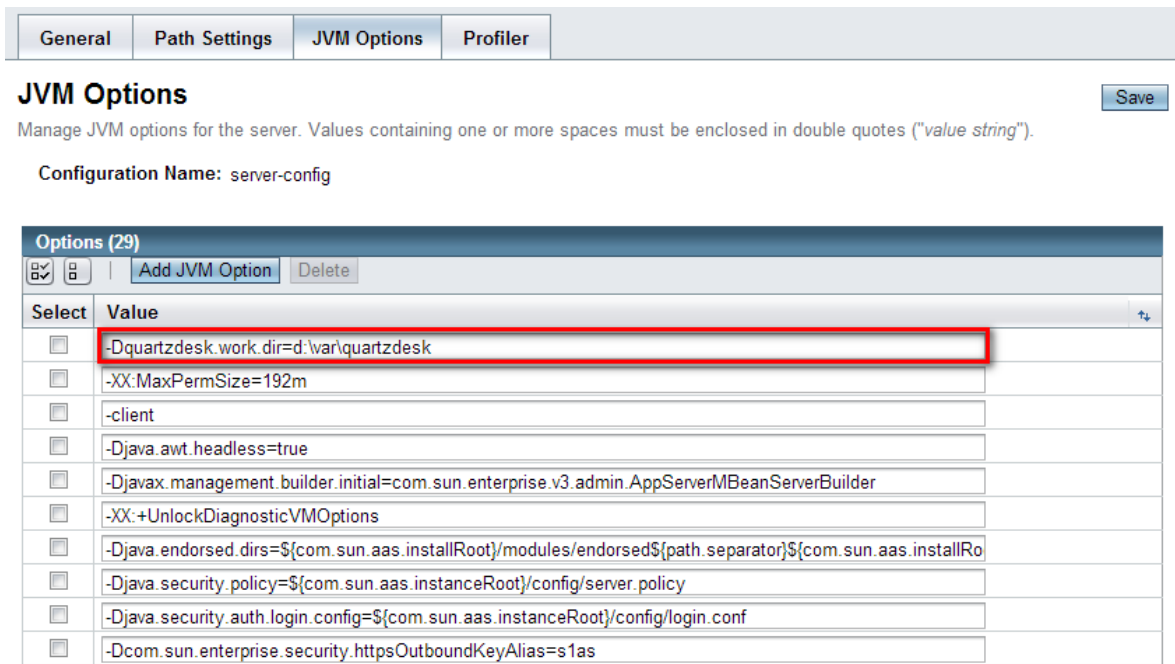
Directory of d:\var\quartzdesk

03.10.2015  15:07    <DIR>          .
03.10.2015  15:07    <DIR>          ..
19.07.2015  17:58                2 878 license.key
10.02.2015  09:44                5 028 logback.xml
01.10.2015  09:53                2 513 quartzdesk.properties
               3 File(s)              10 419 bytes
               2 Dir(s)      32 420 368 384 bytes free

d:\var\quartzdesk>
```

In GAC go to Configurations → GAS\_CONFIG → JVM Settings → JVM Options. Add a new JVM option:

Value: -Dquartzdesk.work.dir=WORK\_DIR



**JVM Options** Save

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

Configuration Name: server-config

Select	Value
<input type="checkbox"/>	-Dquartzdesk.work.dir=d:\var\quartzdesk
<input type="checkbox"/>	-XX:MaxPermSize=192m
<input type="checkbox"/>	-client
<input type="checkbox"/>	-Djava.awt.headless=true
<input type="checkbox"/>	-Djavax.management.builder.initial=com.sun.enterprise.v3.admin.AppServerMBeanServerBuilder
<input type="checkbox"/>	-XX:+UnlockDiagnosticVMOptions
<input type="checkbox"/>	-Djava.endorsed.dirs=\${com.sun.aas.installRoot}/modules/endorsed\${path.separator}\${com.sun.aas.installRo
<input type="checkbox"/>	-Djava.security.policy=\${com.sun.aas.instanceRoot}/config/server.policy
<input type="checkbox"/>	-Djava.security.auth.login.config=\${com.sun.aas.instanceRoot}/config/login.conf
<input type="checkbox"/>	-Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as

Save changes.

## 4.7 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

<b>Microsoft SQL Server</b>	>= 2008	mssql
<b>MySQL (MyISAM)</b>	>= 5.6	mysql
<b>MySQL (InnoDB)</b>	>= 5.6	mysql_innodb
<b>Oracle</b>	== 8i	oracle8
<b>Oracle</b>	>= 9i	oracle9
<b>PostgreSQL</b>	== 8.1	postgres81
<b>PostgreSQL</b>	>= 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.

## 4.8 Security

QuartzDesk supports the HTTP/S Basic authentication scheme to authenticate users who access the application. To configure application security, perform the following two steps:

### 4.8.1 Enable Default Principal to Role Mapping

In GAC go to Configurations → GAS\_CONFIG → Security and check the Default Principal To Role Mapping option.





## Security

Save

Set security properties for the entire server.

Configuration Name: server-config

<b>Security Manager</b>	<input type="checkbox"/> <b>Enabled</b> Enable the security manager for the domain by adding an option in the JVM Settings
<b>Audit Logging</b>	<input type="checkbox"/> <b>Enabled</b> Enable server to load and run all audit modules specified in the Audit Modules setting
<b>Default Realm</b>	file <input type="text"/> Default realm used by all applications for authentication
<b>Default Principal</b>	<input type="text"/> User name used by the server when no principal is provided; must contain only alphanumeric, underscore, dash, or dot characters
<b>Default Principal Password</b>	<input type="text"/> Required if Default Principal contains a value
<b>JACC</b>	default <input type="text"/> Name of the <code>jacc-provider</code> element to use for configuring the JACC infrastructure
<b>Audit Modules</b>	<input type="text"/> List of audit provider modules used by the audit subsystem; Control-click to multiple-select
<b>Default Principal To Role Mapping</b>	<input checked="" type="checkbox"/> <b>Enabled</b> Apply default principal-to-role mapping at deployment when application-specific mapping is not defined; does not affect currently deployed applications
<b>Mapped Principal Class</b>	<input type="text"/> Customize the <code>java.security.Principal</code> implementation class used for default principal-to-role mapping

Click Save.

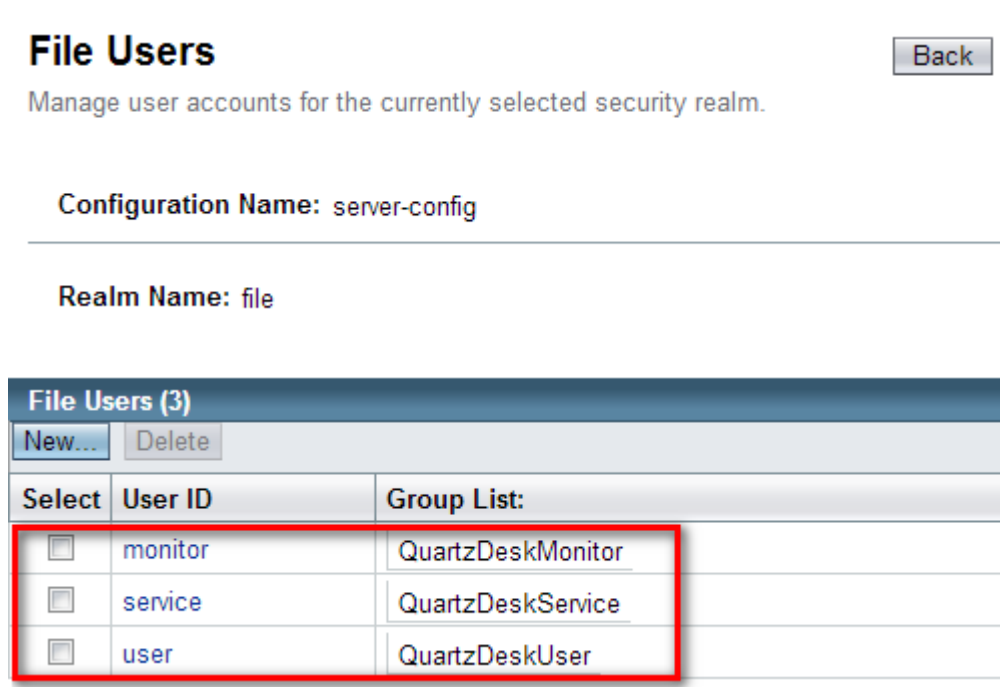
This allows for an automatic mapping of the following three security roles defined in the QuartzDesk web application to the GlassFish security groups with the same name that will be defined in the following step.

Security Role	Description
<b>QuartzDeskUser</b>	Role required to access the QuartzDesk web application UI (QuartzDesk GUI).
<b>QuartzDeskMonitor</b>	Role required to access the scheduler, job and trigger monitoring URLs (REST API).
<b>QuartzDeskService</b>	Role required to access QuartzDesk web-services (e.g. the QuartzAnywhere web-service).

## 4.8.2 Add Users

In GAC go to Configurations → GAS\_CONFIG → Security. Note the value of the configured Default Realm option. We will assume that this option is set to “file”.

In GAC go to Configurations → GAS\_CONFIG → Security → Realms → file. Click Manage Users button and define users that will be accessing the QuartzDesk web application. The users must be assigned to the security groups whose names correspond to the above defined QuartzDesk security roles. The following figure shows an example of three users, each assigned to one of such security groups.



**File Users** Back

Manage user accounts for the currently selected security realm.

Configuration Name: server-config

---

Realm Name: file

File Users (3)		
New...	Delete	
Select	User ID	Group List:
<input type="checkbox"/>	monitor	QuartzDeskMonitor
<input type="checkbox"/>	service	QuartzDeskService
<input type="checkbox"/>	user	QuartzDeskUser

## 4.9 Deploy Application

In GAC go to Applications.

Click the Deploy... button and select the quartzdesk-web-x.y.z.war file and provide the following values:

Type: Web Application

Context Root: Enter the web servlet context path for the QuartzDesk web application. We recommend using “/quartzdesk” (without quotes).

Virtual Servers: Select the virtual servers the QuartzDesk web application will be served by.

Status: Uncheck. This causes GlassFish not to start the application automatically upon its deployment. The application will be started manually in the following step.

## Deploy Applications or Modules

OK Cancel

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

\* Indicates required field

Location:  **Packaged File to Be Uploaded to the Server**

quartzdesk-...-2.1.0.war

**Local Packaged File or Directory That Is Accessible from GlassFish Server**

Type: \*

Context Root:   
Path relative to server's base URL.

Application Name: \*

Virtual Servers:   
Associates an Internet domain name with a physical server.

Status:  **Enabled**  
Allows users to access the application.

Precompile JSPs:   
Precompiles JSP pages during deployment.

Run Verifier:   
Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed.

Force Redeploy:   
Forces redeployment even if this application has already been deployed or already exists.

Keep State:   
Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments.

Deployment Order:   
A number that determines the loading order of the application at server startup. Lower numbers are loaded first. The default is 100.

Libraries:   
A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. Specify relative paths relative to *instance-root/lib/applibs*. The libraries are made available to the application in the order specified.

Description:

Click OK.

## 4.10 Start Application

In GAC go to Applications. Select the QuartzDesk web application deployed in the previous step. Click the Enable button to start the application and wait for the startup procedure to complete.

Check the GAS `server.log` log file under `GAS_DOMAIN_DIR/logs` for errors.

You can safely ignore the following warning messages:

```
[2015-09-30T13:59:46.644+0200] [glassfish 4.0] [WARNING] []  
[org.apache.jasper.runtime.TldScanner] [tid: _ThreadID=39  
_ThreadName=admin-listener(3)] [timeMillis: 1373975986644]  
[levelValue: 900] [[  
    PWC6351: In TLD scanning, the supplied resource  
file:/D:/Java/glassfish4/glassfish/domains/domain1/applications/li  
b/activation-1.1.jar does not exist  
java.io.FileNotFoundException:  
D:\Java\glassfish4\glassfish\domains\domain1\applications\lib\acti  
vation-1.1.jar (The system cannot find the path specified)
```

Check the QuartzDesk web application logs (by default in the `WORK_DIR/logs` directory) for errors.

If there are no errors, point your browser to [http://GAS\\_HTTP\\_HOST:GAS\\_HTTP\\_PORT/quartzdesk](http://GAS_HTTP_HOST:GAS_HTTP_PORT/quartzdesk) and verify that the QuartzDesk web application works.

## 5. Upgrading

### 5.1 Stop Existing Application

In GAC go to Applications. Select the checkbox next to the QuartzDesk web application in the Applications list. Click the Disable button at the top of the list. Confirm this action in a dialog window that opens and wait for the action to complete.

Upon successful stopping, the Enabled flag shown next the existing QuartzDesk web application in the Deployments list, indicates that the applications has been stopped.

#### Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Deployed Applications (8)						
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">Deploy...</a>	<a href="#">Undeploy</a>	<a href="#">Enable</a>	<a href="#">Disable</a>	Filter: <input type="text"/>
Select	Name	Deployment Order	Enabled	Engines	Action	
<input type="checkbox"/>	quartzdesk-web-2.1.0	100	<input checked="" type="checkbox"/>	webservices, web	<a href="#">Redeploy</a>   <a href="#">Reload</a>	

### 5.2 Backup

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

Backup the contents of the QuartzDesk work directory.

Make sure you still have the WAR file of the existing QuartzDesk web application.

Store the backup files in a safe place so that you can restore the original QuartzDesk web application version if the need arises.

### 5.3 Remove Existing Application

In GAC go to Applications. Select the checkbox next to the existing QuartzDesk web application in the Applications list. Click the Undeploy button at the top of the list. Confirm this action in a dialog window that opens and wait for the action to complete.

Upon successful removal, the QuartzDesk web application disappears from the Applications list.

### 5.4 Deploy New Application

Deploy the new version of the QuartzDesk web application by following the deployment steps outlined in 4.9.

## 5.5 Start New Application

Start the new QuartzDesk web application by following the steps outlined in 4.10.

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](http://www.quartzdesk.com) (click Support → FAQs and search for “find out version”).



## 6. Cluster Deployment Notes

When deploying the QuartzDesk web application to a GlassFish 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 GlassFish instance that handled the first user request that established the session.

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

### 6.2 Shared Work Directory

We recommend that you put the QuartzDesk work directory, described in chapter 4.6, 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 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
<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. No QuartzDesk 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>Please refer to the GlassFish documentation for details on how to add a new JVM system property.</p>

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.