



QuartzDesk Web Application Installation and Upgrade Guide for RedHat JBoss AS 6.1.0 and 7.x Community, 6.x EAP and 7.x EAP

QuartzDesk Version: 2.x

April 24, 2017



Table of Contents

1.	PURPOSE	3
2.	DEFINITIONS	4
3.	REQUIREMENTS	5
3.1	SOFTWARE REQUIREMENTS	5
3.1.1	Browser	5
3.1.2	Operating System	5
3.1.3	Java	5
3.1.4	Application Server.....	5
3.1.5	Database	5
3.1.6	Database JDBC Driver	6
3.1.7	QuartzDesk Web Application Archive.....	6
3.2	HARDWARE REQUIREMENTS	6
4.	INSTALLATION	7
4.1	DATABASE.....	7
4.2	JDBC DRIVER	7
4.3	JDBC DATASOURCE.....	7
4.3.1	DB2	9
4.3.2	H2.....	11
4.3.3	Microsoft SQL Server.....	13
4.3.4	MySQL.....	15
4.3.5	Oracle.....	17
4.3.6	PostgreSQL	19
4.4	TEST JDBC DATASOURCE.....	21
4.5	APPLICATION WORK DIRECTORY	22
4.6	APPLICATION CONFIGURATION.....	23
4.7	SECURITY	23
4.7.1	Add Users.....	24
4.7.2	Create Security Domain.....	26
4.8	DEPLOY APPLICATION.....	30
4.9	START APPLICATION	31
5.	UPGRADING	32
5.1	STOP EXISTING APPLICATION	32
5.2	BACKUP	33
5.3	REMOVE EXISTING APPLICATION	34
5.4	DEPLOY NEW APPLICATION.....	34
5.5	START NEW APPLICATION	34
6.	CLUSTER DEPLOYMENT NOTES.....	35
6.1	HTTP SESSION REPLICATION AND AFFINITY	35
6.2	SHARED WORK DIRECTORY.....	35
6.3	LOGGING CONFIGURATION	35
6.3.1	Using Shared Log Files.....	36
6.3.2	Using Separate Log Files.....	37
6.4	INTERNAL QUARTZ SCHEDULER	39

1. Purpose

This document describes the installation and upgrade process for the QuartzDesk web application 2.x on RedHat JBoss Application Server 6.1.0 Community, 7.x Community, 6.x EAP and 7.x EAP running in the **standalone** mode.

The installation and upgrade process in the **domain** mode is similar and is not described herein.

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.
JAC	JBoss Administrative Console.
JAS	JBoss Application Server.
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	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.
JAS_INSTALL_ROOT	/usr/local/jboss	JBoss Application Server installation directory.
JAS_CONFIG	standalone	JBoss Application Server configuration.
JAS_HTTP_HOST	localhost	JBoss HTTP listener host.
JAS_HTTP_PORT	8080	JBoss HTTP listener port.
WORK_DIR	/var/quartzdesk	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

JBoss Application Server 6.1.0 Community.

JBoss Application Server 7.x Community.

JBoss Application Server 6.x EAP.

JBoss Application Server 7.x EAP.

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	8.1

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	<p>Microsoft JDBC driver 4.0 for SQL Server available at http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx.</p> <p>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.</p>
MySQL	Connector/J JDBC driver available at http://dev.mysql.com/downloads/connector/j/ .
Oracle	<p>Oracle JDBC driver available at http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.</p> <p>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#02_02.</p>
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 web application archive (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) to `JAS_INSTALL_ROOT/JAS_CONFIG/deployments` directory and restart the application server.



To install the H2 JDBC driver, do not copy it to `JAS_INSTALL_ROOT/JAS_CONFIG/deployments` directory because the H2 driver is already installed as a module (`JAS_INSTALL_ROOT/modules/.../com/h2database/h2`). To update the existing driver module JAR, copy the new H2 driver JAR to the module's main directory and edit the module descriptor file `module.xml` to update the JAR name.

4.3 JDBC Datasource

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

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

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Click the Add button to create a new JDBC datasource.

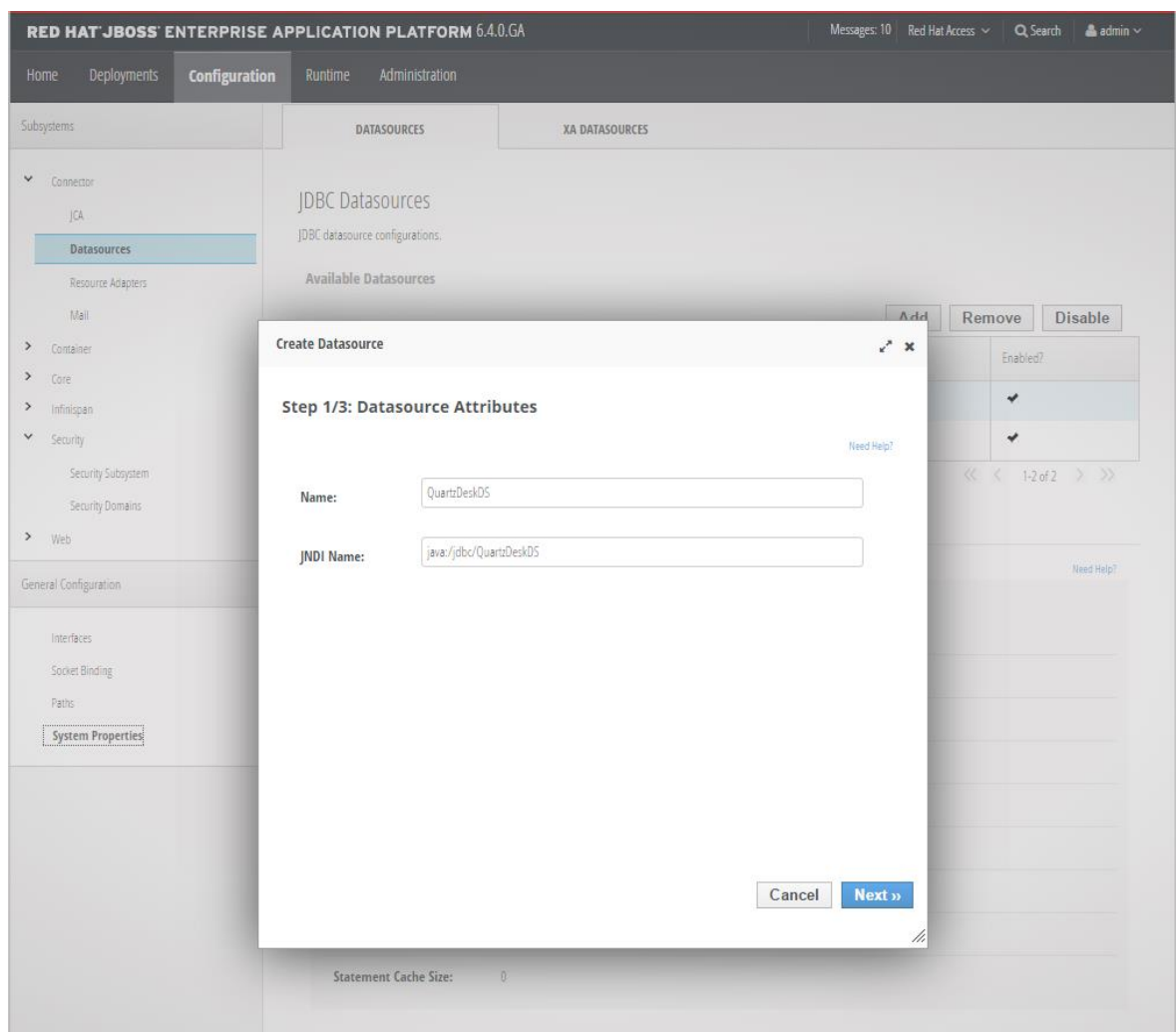
JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the new JDBC datasource type. Click Next.

Enter these datasource names:

Name: QuartzDeskDS

JNDI Name: java:/jdbc/QuartzDeskDS



Click Next.

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

4.3.1 DB2

In Step 2, select the DB2 JDBC driver. Click Next.

In Step 3, enter the following values:

Connection URL: jdbc:db2://DB_HOST:DB_PORT/DB_NAME

Username: DB_USER

Password: DB_PASSWORD

Security Domain: leave empty

JBoss 7.x Community, JBoss 6.x EAP: Click Done.

JBoss 7.x EAP: Click Next and Finish.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Click View.

Modify the datasource configuration under the following tabs:

Attributes

Click the Edit button/link and enter the following values:

Statement Cache Size: 100

Click Save.

Properties

Add the following properties:

Key: clientApplicationInformation

Value: QuartzDesk

Pool

Click the Edit button/link and enter the following values:

Min Pool Size: 2

Max Pool Size: 10

Prefil enabled: check

Click Save.

Validation

Click the Edit button/link and enter the following values:

Valid Connection Checker:

`org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker`

State Connection Checker:

`org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker`

Exception Sorter:

`org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter`

Click Save.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Click the Enable button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the datasource menu and click Enable.

Click the Confirm button in the confirmation dialog.

The QuartzDeskDS datasource should now be enabled.

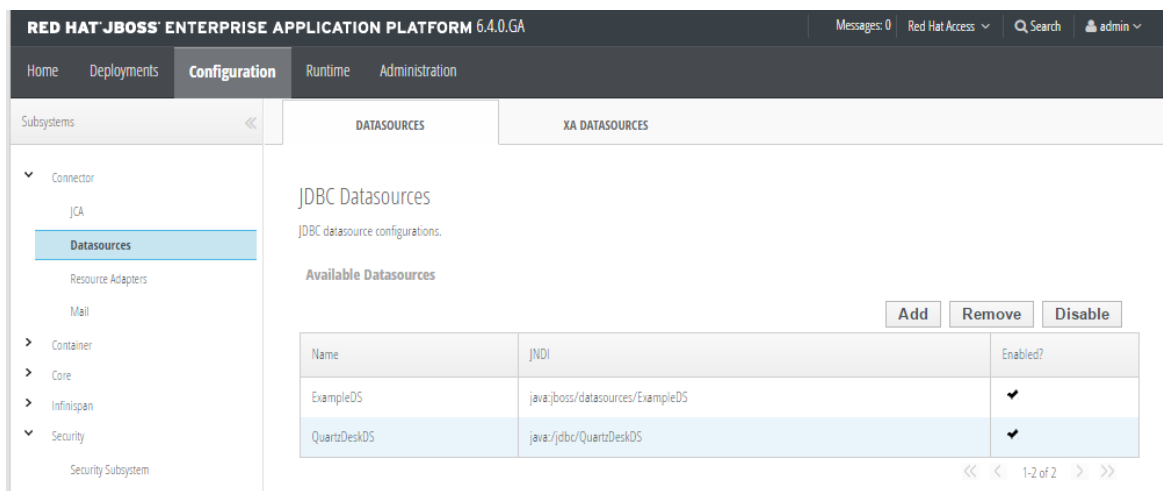
JDBC Datasources

JDBC datasource configurations.

Available Datasources

<div>Add Remove Disable</div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
QuartzDeskDS	java:/jdbc/QuartzDeskDS	<input checked="" type="checkbox"/>

<< < 1-2 of 2 > >>



RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6.4.0.GA Messages: 0 Red Hat Access Search admin

Home Deployments **Configuration** Runtime Administration

Subsystems << DATASOURCES XA DATASOURCES

JDBC Datasources
JDBC datasource configurations.

Available Datasources

<div>Add Remove Disable</div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
QuartzDeskDS	java:/jdbc/QuartzDeskDS	<input checked="" type="checkbox"/>

<< < 1-2 of 2 > >>

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 2, select the H2 JDBC driver. Click Next.

In Step 3, enter the following values:

Connection URL: jdbc:h2:file:<H2_DB_FILE_PATH>

Username: DB_USER

Password: DB_PASSWORD

Security Domain: leave empty

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

JBoss 7.x Community, JBoss 6.x EAP: Click Done.

JBoss 7.x EAP: Click Next and Finish.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Click View.

Modify the datasource configuration under the following tabs:

Attributes

Click the Edit button/link and enter the following values:

Statement Cache Size: 100

Click Save.

Properties

Add the following properties:

Key: applicationName

Value: QuartzDesk

Pool

Click the Edit button/link and enter the following values:

Min Pool Size: 2

Max Pool Size: 10

Prefill enabled: check

Click Save.

Validation

Click the Edit button/link and enter the following values:

Valid Connection Checker:

org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker

Exception Sorter:

org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter

Click Save.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Click the Enable button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the datasource menu and click Enable.

Click the Confirm button in the confirmation dialog.

The QuartzDeskDS datasource should now be enabled.

JDBC Datasources

JDBC datasource configurations.

Available Datasources

<div> Add Remove Disable </div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓
QuartzDeskDS	java:/jdbc/QuartzDeskDS	<input checked="" type="checkbox"/>

<< < 1-2 of 2 > >>

4.3.3 Microsoft SQL Server

In Step 2, select the Microsoft SQL Server JDBC driver. Click Next.

In Step 3, enter the following values:

Connection URL: jdbc:sqlserver://DB_HOST:DB_PORT;databaseName=DB_NAME

Username: DB_USER

Password: DB_PASSWORD

Security Domain: leave empty

JBoss 7.x Community, JBoss 6.x EAP: Click Done.

JBoss 7.x EAP: Click Next and Finish.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Click View.

Modify the datasource configuration under the following tabs:

Attributes

Click the Edit button/link and enter the following values:

Statement Cache Size: 100

Click Save.

Properties

Add the following properties:

Key: applicationName

Value: QuartzDesk

Pool

Click the Edit button/link and enter the following values:

Min Pool Size: 2

Max Pool Size: 10

Prefill enabled: check

Click Save.

Validation

Click the Edit button/link and enter the following values:

Valid Connection Checker:

org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker

Click Save.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Click the Enable button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the datasource menu and click Enable.

Click the Confirm button in the confirmation dialog.

The QuartzDeskDS datasource should now be enabled.

JDBC Datasources

JDBC datasource configurations.

Available Datasources

<div> Add Remove Disable </div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

<< < 1-2 of 2 > >>

4.3.4 MySQL

In Step 2, select the PostgreSQL JDBC driver. Click Next.

In Step 3, enter the following values:

Connection URL: jdbc:mysql://DB_HOST:DB_PORT/DB_NAME

Username: DB_USER

Password: DB_PASSWORD

Security Domain: leave empty

JBoss 7.x Community, JBoss 6.x EAP: Click Done.

JBoss 7.x EAP: Click Next and Finish.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Click View.

Modify the datasource configuration under the following tabs:

Attributes

Click the Edit button/link and enter the following values:

Statement Cache Size: 100

Click Save.

Properties

Add the following properties:

Key: cachePrepStmts

Value: true

Pool

Click the Edit button/link and enter the following values:

Min Pool Size: 2

Max Pool Size: 10

Prefill enabled: check

Click Save.

Validation

Click the Edit button/link and enter the following values:

Valid Connection Checker:

org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker

Exception Sorter:

org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter

Click Save.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Click the Enable button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the datasource menu and click Enable.

Click the Confirm button in the confirmation dialog.

The QuartzDeskDS datasource should now be enabled.

JDBC Datasources

JDBC datasource configurations.

Available Datasources

<div> Add Remove Disable </div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

<< < 1-2 of 2 > >>

4.3.5 Oracle

In Step 2, select the Oracle JDBC driver. Click Next.

In Step 3, enter the following values:

Connection URL: jdbc:oracle:thin:@DB_HOST:DB_PORT: ORACLE_SERVICE_NAME

Username: DB_USER

Password: DB_PASSWORD

Security Domain: leave empty

JBoss 7.x Community, JBoss 6.x EAP: Click Done.

JBoss 7.x EAP: Click Next and Finish.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Click View.

Modify the datasource configuration under the following tabs:

Attributes

Click the Edit button/link and enter the following values:

Statement Cache Size: 100

Click Save.

Properties

Add the following properties:

Key: driverType

Value: thin

Key: databaseName

Value: DB_NAME

Pool

Click the Edit button/link and enter the following values:

Min Pool Size: 2

Max Pool Size: 10

Prefill enabled: check

Click Save.

Validation

Click the Edit button/link and enter the following values:

Valid Connection Checker:

org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker

State Connection Checker:

org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker

Exception Sorter:

org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter

Click Save.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Click the Enable button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the datasource menu and click Enable.

Click the Confirm button in the confirmation dialog.

The QuartzDeskDS datasource should now be enabled.

JDBC Datasources

JDBC datasource configurations.

Available Datasources

<div> Add Remove Disable </div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
QuartzDeskDS	java:/jdbc/QuartzDeskDS	<input checked="" type="checkbox"/>

<< < 1-2 of 2 > >>

4.3.6 PostgreSQL

In Step 2, select the PostgreSQL JDBC driver. Click Next.

In Step 3, enter the following values:

Connection URL: jdbc:postgresql://DB_HOST:DB_PORT/DB_NAME

Username: DB_USER

Password: DB_PASSWORD

Security Domain: leave empty

JBoss 7.x Community, JBoss 6.x EAP: Click Done.

JBoss 7.x EAP: Click Next and Finish.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Click View.

Modify the datasource configuration under the following tabs:

Attributes

Click the Edit button/link and enter the following values:

Statement Cache Size: 100

Click Save.

Properties

Add the following properties:

Key: applicationName

Value: QuartzDesk

Pool

Click the Edit button/link and enter the following values:

Min Pool Size: 2

Max Pool Size: 10

Prefill enabled: check

Click Save.

Validation

Click the Edit button/link and enter the following values:

Valid Connection Checker:

org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker

Exception Sorter:

org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter

Click Save.

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Click the Enable button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the datasource menu and click Enable.

Click the Confirm button in the confirmation dialog.

The QuartzDeskDS datasource should now be enabled.

JDBC Datasources

JDBC datasource configurations.

Available Datasources

<div> Add Remove Disable </div>		
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
QuartzDeskDS	java:/jdbc/QuartzDeskDS	<input checked="" type="checkbox"/>

<< < 1-2 of 2 > >>

4.4 Test JDBC Datasource

In JAC go to:

JBoss 7.x Community: Profile → Connector → Datasources

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Connector → Datasources

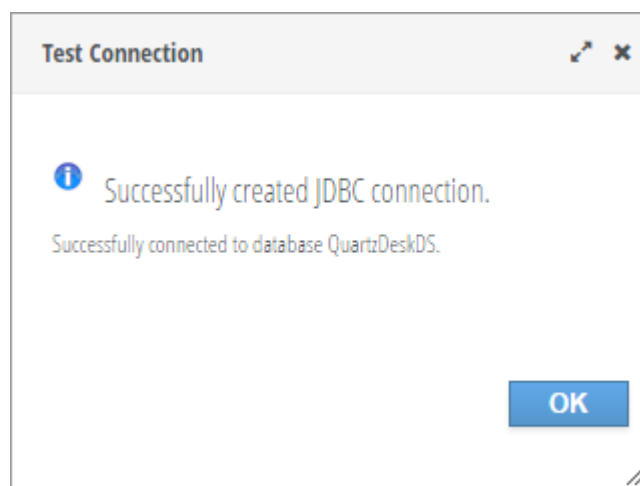
JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Connector → Datasources

Select the QuartzDeskDS datasource. Open the Connection tab and click the Test Connection button.

JBoss 7.x EAP: Configuration → Subsystems → Datasources → Non-XA

Select the QuartzDeskDS datasource. Open the local menu and click Test Connection.

If the JDBC connection pool test is successful, a dialog with “Successfully created JDBC connection” message is displayed.



If the JDBC connection pool test fails, no dialog is displayed and an error exception is logged in the JAS log (JAS_INSTALL_ROOT/JAS_CONFIG/log/server.log).

4.5 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 JAS process is running under.

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



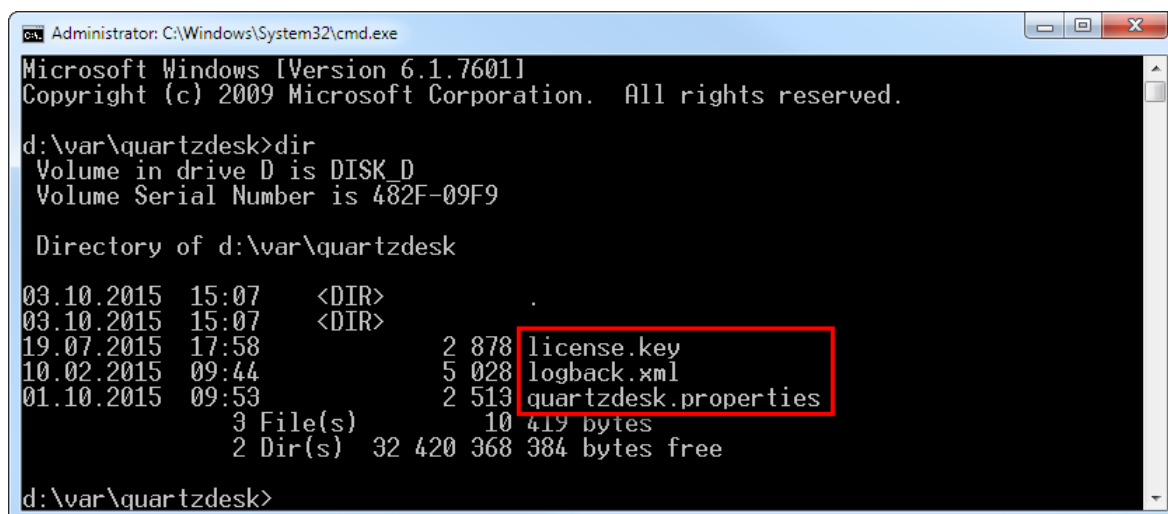
You can obtain a free 30-day trial license key at www.quartzdesk.com (open the Try / Purchase menu).

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   9:44             5 028 logback.xml
01.10.2015   9:53             2 513 quartzdesk.properties
               3 File(s)            10 419 bytes
               2 Dir(s)  32 420 368 384 bytes free

d:\var\quartzdesk>
```

In JAC go to:

JBoss 7.x Community: ???

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → General Configuration → System Properties

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → General Configuration → System Properties

Click the Add button.

JBoss 7.x EAP: Configuration → System Properties

Click the View button. Click the Add button.

Add a new boot-time system property.

Name: quartzdesk.work.dir

Value: WORK_DIR



JAC requires all backslashes in the WORK_DIR to be escaped as two consecutive backslash characters (\\). Alternatively, use forward slash characters (/).

SYSTEM PROPERTIES

System Properties

These properties are available throughout the configuration. The Boot-Time flag specifies if a property should be passed into the JVM start (-Dproperty=value)

Add
Remove

Key	Value
quartzdesk.work.dir	d:/var/quartzdesk.work

1-1 of 1

4.6 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.

4.7 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:

4.7.1 Add Users

The QuartzDesk web application defines the following security roles that are required to access various parts of the application.

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).

Before registering QuartzDesk web application users in JBoss, create a new `JAS_INSTALL_ROOT/JAS_CONFIG/configuration/quartzdesk-users.properties` file with the following contents:

```
#
# Properties declaration of users for the realm 'QuartzDeskRealm'.
#
# Users can be added to this properties file at any time, updates
# after the server has started will be automatically detected.
#
# The format of this realm is as follows: -
# username=HEX( MD5( username ':' realm ':' password))
#
# A utility script is provided which can be executed from the bin
# folder to add the users:
#
# - Linux
#   bin/add-user.sh
#
# - Windows
#   bin\add-user.bat
#
#$REALM_NAME=QuartzDeskRealm$
#
```

Now you can add new QuartzDesk users by running the following command:

Windows

```
JAS_INSTALL_ROOT\bin\add-user.bat -a -up
JAS_INSTALL_ROOT\JAS_CONFIG\configuration\quartzdesk-
users.properties -r QuartzDeskRealm
```

Unix / Linux

```
JAS_INSTALL_ROOT/bin/add-user.sh -a -up
JAS_INSTALL_ROOT/JAS_CONFIG/configuration/quartzdesk-
users.properties -r QuartzDeskRealm
```

When prompted, enter the following values:

Enter the details of the new user to add.

Using realm 'QuartzDeskRealm' as discovered from the existing property files.

Username: <username>

Password: <password>

Re-Enter Password: <password>

About to add user '<username>' for realm 'QuartzDeskRealm'

Is this correct yes/no?

Enter: yes

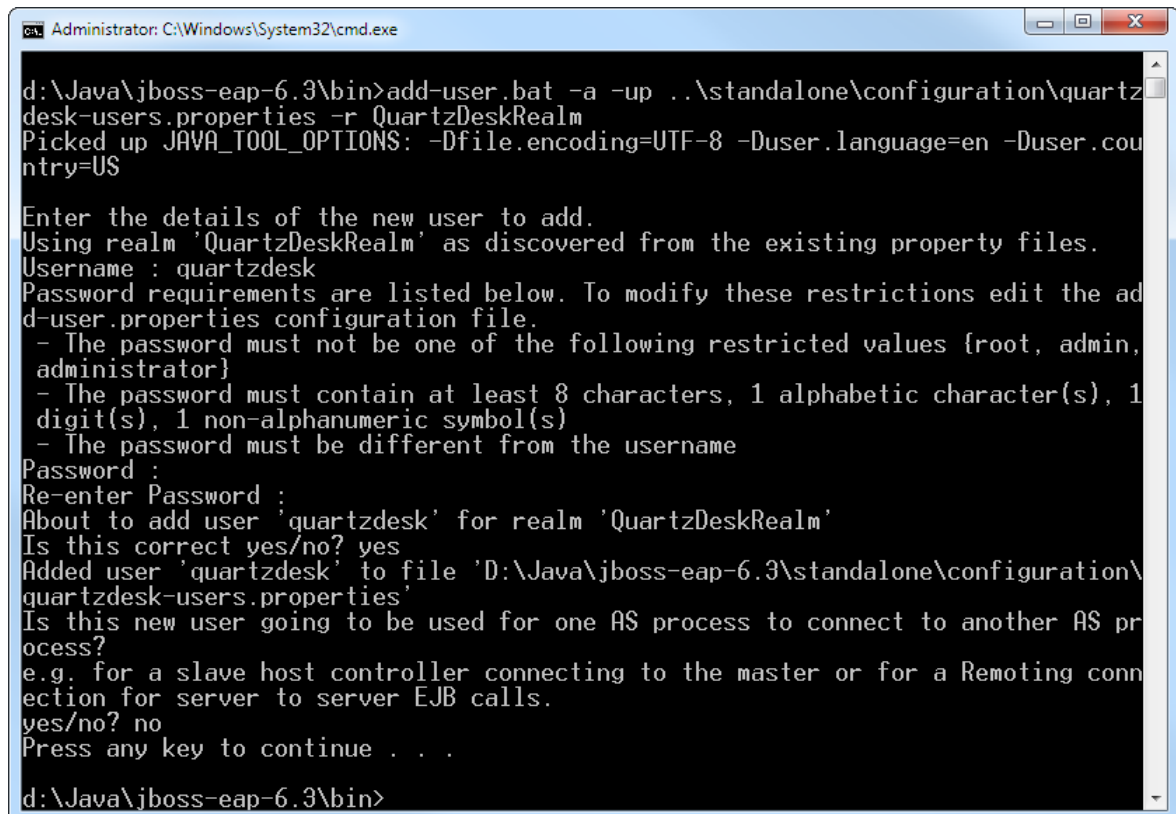
Is this new user going to be used for one AS process to connect to another AS process?

e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.

yes/no?

Enter: no

This is an example of adding a new user with the username of 'quartzdesk':



```
Administrator: C:\Windows\System32\cmd.exe

d:\Java\jboss-eap-6.3\bin>add-user.bat -a -up ..\standalone\configuration\quartzdesk-users.properties -r QuartzDeskRealm
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Duser.language=en -Duser.country=US

Enter the details of the new user to add.
Using realm 'QuartzDeskRealm' as discovered from the existing property files.
Username : quartzdesk
Password requirements are listed below. To modify these restrictions edit the add-user.properties configuration file.
- The password must not be one of the following restricted values {root, admin, administrator}
- The password must contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
- The password must be different from the username
Password :
Re-enter Password :
About to add user 'quartzdesk' for realm 'QuartzDeskRealm'
Is this correct yes/no? yes
Added user 'quartzdesk' to file 'D:\Java\jboss-eap-6.3\standalone\configuration\quartzdesk-users.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? no
Press any key to continue . . .

d:\Java\jboss-eap-6.3\bin>
```

Repeat executing the add-user command to add other users that will be accessing QuartzDesk Web Application.

To associate added users with security roles defined in the QuartzDesk Web Application, create a new JAS_INSTALL_ROOT/JAS_CONFIG/configuration/quartzdesk-roles.properties file with the following contents:

```
#
# Properties declaration of users roles for the realm 'QuartzDeskRealm'.
#
# Users can be added to this properties file at any time, updates after
# the server has started will be automatically detected.
#
# The format of this file is as follows: -
# username=role1,role2,role3
#
# The following illustrates how quartzdesk user can be assigned all three
# QuartzDesk Web Application security roles:
#
quartzdesk=QuartzDeskUser,QuartzDeskService,QuartzDeskMonitor
```

Edit the created `JAS_INSTALL_ROOT/JAS_CONFIG/configuration/quartzdesk-roles.properties` file and use the documented syntax to associate individual QuartzDesk users with QuartzDesk Web Application security roles.

4.7.2 Create Security Domain

In JAC go to:

JBoss 7.x Community: Profile → Security → Security Domains

JBoss 6.1 EAP, JBoss 6.2 EAP: Profile → Subsystems → Security → Security Domains

JBoss 6.3 EAP, JBoss 6.4 EAP: Configuration → Subsystems → Security → Security Domains

JBoss 7.x EAP: Configuration → Subsystems → Security

Click the Add button.

Enter the following values:

Name: quartzdesk

Cache Type: default

Add Security Domains

Need Help?

Name:

Cache Type:

Click Save.

A new security domain should be created and present in the table listing the available security domains.

Security Domains

Registered security domains. Please select a domain to edit the security policies.

Name	Option
jboss-ejb-policy	View >
jboss-web-policy	View >
other	View >
quartzdesk	View >

<< < 1-4 of 4 > >>

Details

Attributes

Need Help?

Cache Type: default

Name: quartzdesk

Click on the View link/button next to the quartzdesk security domain name in the table to set up security domain's login modules.

Click on the Add button to add a new login module to the security domain. Enter the following values:

JBoss 7.x EAP:

Name: RealmUsersRoles

Code: RealmUsersRoles

Flag: required

Click Save.

The RealmUserRoles login module should now be present in the list of security domain's login modules.

Security Domain: quartzdesk

Authentication configuration for this domain. Can either be classic or jasp.

Login Modules

Code	Flag	
RealmUsersRoles	required	

<< < 1-1 of 1 > >>

Details

Attributes Module Options

▲ Key	Value
No Items!	

<< < > >>

Select the registered RealmUsersRoles login module in the table.

JBoss 7.x Community, JBoss 6.x EAP:

Click on the Module Options tab. Add the following module options using the Add button.

Key: usersProperties

Value: JAS_INSTALL_ROOT/JAS_CONFIG/quartzdesk-users.properties

Key: rolesProperties

Value: JAS_INSTALL_ROOT/JAS_CONFIG/quartzdesk-roles.properties

Key: realm

Value: QuartzDeskRealm

Key: password-stacking

Value: useFirstPass

The Module Options table should look like this:

Security Domain: quartzdesk

Authentication configuration for this domain. Can either be classic or jaspri.

Login Modules

Add Remove

Code	Flag	
RealmUsersRoles	required	

<< < 1-1 of 1 > >>

Details

Attributes Module Options

Add Remove

Key	Value
usersProperties	d:/java/jboss-eap-6.4/standalone/configuration/quartzdesk-users.properties
rolesProperties	d:/java/jboss-eap-6.4/standalone/configuration/quartzdesk-roles.properties
realm	QuartzDeskRealm
password-stacking	useFirstPass

<< < 1-4 of 4 > >>

JBoss 7.x EAP:

Click the Edit link in the Attributes tab. Add the following module options to the Module options text area:

realm=QuartzDeskRealm

usersProperties=JAS_INSTALL_ROOT/JAS_CONFIG/quartzdesk-users.properties

rolesProperties=JAS_INSTALL_ROOT/JAS_CONFIG/quartzdesk-roles.properties

Click Save.

The login module attributes should look like this:

Authentication Modules

List of authentication modules

Name
RealmUsersRoles

<< < 1-1 of 1 > >>

Attributes

[Edit](#)

Code: RealmUsersRoles

Flag: required

Module:

Module options:
realm = QuartzDeskRealm
usersProperties = d:/java/jboss-eap-7.0/standalone/configuration/quartzdesk-users.properties
rolesProperties = d:/java/jboss-eap-7.0/standalone/configuration/quartzdesk-roles.properties

[Need Help?](#)

Restart the JAS for the changes to take effect.

4.8 Deploy Application

In JAC go to:

JBoss 7.x Community: Runtime → Deployments → Manage Deployments. Click the Add Content button.

JBoss 6.1 EAP, JBoss 6.2 EAP, JBoss 6.3 EAP: Runtime → Server → Manage Deployments. Click the Add button.

JBoss 6.4 EAP: Deployments. Click the Add button.

In Step 1/2 (Deployment Selection) click on the Choose File button to select the `quartzdesk-web-x.y.z.war` file. Click Next.

In Step 2/2 (Verify Deployment Name(s)) make no changes and click Save.

JBoss 7.x EAP: Deployments. Click the Add button.

Select “Upload a new deployment” option and click Next.

Click on the Choose File button to select the `quartzdesk-web-x.y.z.war` file. Click Next.

In the next step (Verify Upload), uncheck the Enable option. Make no changes to the deployment names and click Finish.

4.9 Start Application

In JAC go to:

JBoss 7.x Community: Runtime → Deployments → Manage Deployments.

Click the Enable button next to the QuartzDesk web application in the Deployments list.

JBoss 6.1 EAP, JBoss 6.2 EAP, JBoss 6.3 EAP: Runtime → Server → Manage Deployments

JBoss 6.4 EAP: Deployments.

Select the QuartzDesk web application and click the Enable button. Confirm this action in the dialog window that gets opened.

JBoss 7.x EAP: Deployments.

In the Deployments list, open local menu for the QuartzDesk web application and choose the Enable option. Confirm this action in the dialog window that gets opened.

The application should be starting now.

Monitor the `JAS server.log` log file under `JAS_INSTALL_ROOT/JAS_CONFIG/logs` for errors and wait for the startup procedure to complete.

You can safely ignore the following warning messages:

```
16:35:37,714 WARN [org.jboss.as.server.deployment] (MSC service
thread 1-5) JBAS015960: Class Path entry lib/quartzdesk-api-2.6.0-
SNAPSHOT.jar in /d:/Java/jboss-eap-6.4/bin/content/quartzdesk-web-
<version>.war does not point to a valid jar for a Class-Path
reference.
```

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://JAS_HTTP_HOST:JAS_HTTP_PORT/quartzdesk and verify that the QuartzDesk web application works.

5. Upgrading

5.1 Stop Existing Application

In JAC go to:

JBoss 7.x Community: Runtime → Deployments → Manage Deployments

Click the Disable button next to the QuartzDesk web application in the Deployments list. Confirm this action in a dialog window that opens and wait for the action to complete.

Upon successful disabling, the Enabled flag, shown next the existing QuartzDesk web application in the Deployments list, indicates that the application is no longer enabled.

Deployments

Name	Runtime Name	Enabled	En/Disable	Remove
db2jcc4.jar	db2jcc4.jar	✓	Disable	Remove
db2jcc_license_cu.jar	db2jcc_license_cu.jar	✓	Disable	Remove
mysql-connector-java-5.1.23-bin.jar	mysql-connector-java-5.1.23-bin.jar	✓	Disable	Remove
ojdbc6.jar	ojdbc6.jar	✓	Disable	Remove
postgresql-9.2-1002.jdbc4.jar	postgresql-9.2-1002.jdbc4.jar	✓	Disable	Remove
quartzdesk-web-2.1.0.war	quartzdesk-web-2.1.0.war	⊘	Enable	Remove
sqljdbc4.jar	sqljdbc4.jar	✓	Disable	Remove

1 - 7 of 7

JBoss 6.1 EAP, JBoss 6.2 EAP, JBoss 6.3 EAP: Runtime → Server → Manage Deployments

JBoss 6.4 EAP: Deployments

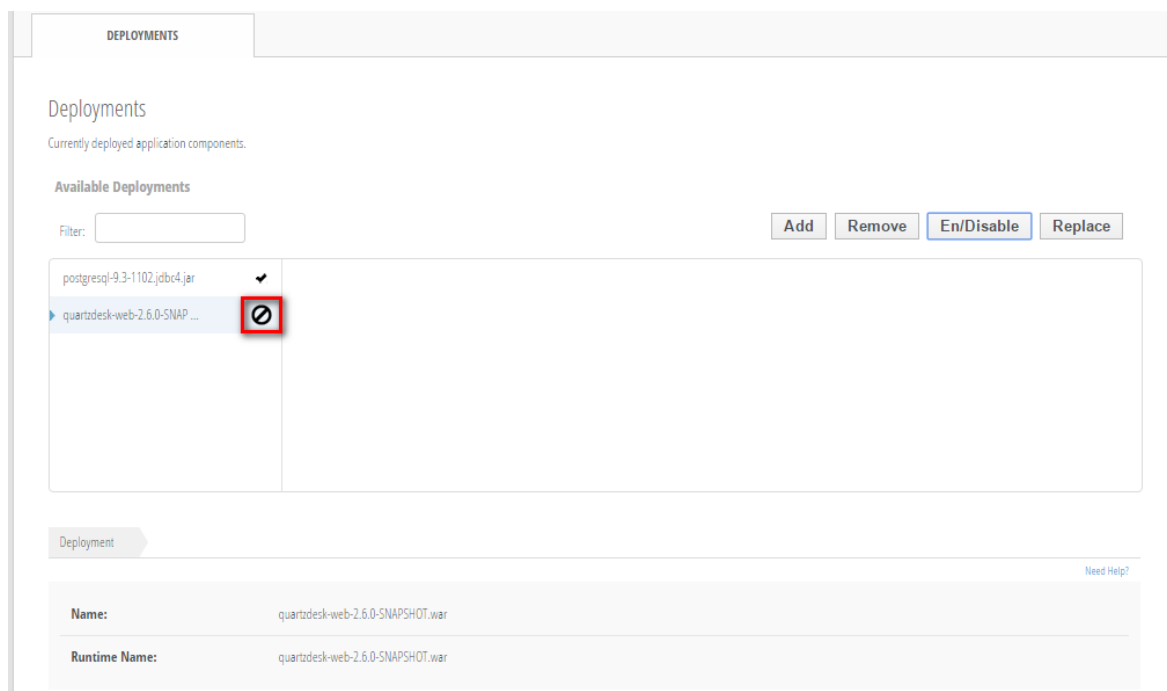
Click the En/Disable button next to the QuartzDesk web application in the Deployments list. Confirm this action in a dialog window that opens and wait for the action to complete.

JBoss 7.x EAP: Deployments

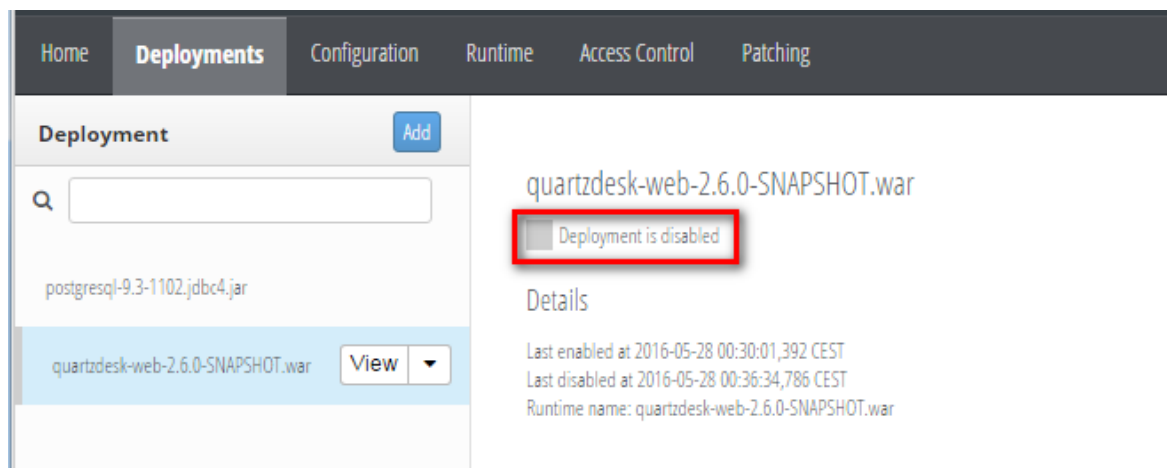
In the Deployments list, open local menu for the QuartzDesk web application and choose the Disable option. Confirm this action in a dialog window that opens and wait for the action to complete.

Upon successful disabling, the flag, shown next the existing QuartzDesk web application in the Deployments list, indicates that the application is no longer enabled.

JBoss 7.x Community, JBoss 6.x EAP:



JBoss 7.x EAP:



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 have the WAR file of the existing QuartzDesk web application.

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

In JAC go to:

JBoss 7.x Community: Runtime → Deployments → Manage Deployments

Click the Remove button next to the existing QuartzDesk web application. 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 Deployments list.

JBoss 6.1 EAP, JBoss 6.2 EAP, JBoss 6.3 EAP: Runtime → Server → Manage Deployments

JBoss 6.4 EAP: Deployments

Select the deployed QuartzDesk web application and click the Remove button. Confirm this action in a dialog window that opens and wait for the action to complete.

JBoss 7.x EAP: Deployments

In the Deployments list, open the local menu for QuartzDesk web application and choose the Remove option. 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 Deployments list.

5.4 Deploy New Application

Deploy the new version of the QuartzDesk web application by following the deployment steps outlined in 4.8.



Some JAS releases contain a bug that prevents the application server from invalidating and recompiling the JSP cache of redeployed web applications. Therefore, we recommend that you to stop the WFAS instance and manually purge the QuartzDesk web application JSP cache located at `JAS_INSTALL_ROOT/JAS_CONFIG/tmp/work/jboss.web/[host]/quartzdesk`.

5.5 Start New Application

Start the new QuartzDesk web application by following the steps outlined in 4.9.

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 JBoss 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 JBoss instance that handled the first user request that established the session.

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

6.2 Shared Work Directory

We recommend that you put the QuartzDesk work directory, described in chapter 4.3, 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 JBoss 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.

