QuartzDesk Web Application Installation and Upgrade Guide for WildFly AS 8.x, 9.x, 10.x, 11.x, 12.x, 13.x and 14.x

QuartzDesk Version: 3.x

January 21, 2019

Table of Contents

1. Purpose

This document describes the installation and upgrade process for the QuartzDesk web application 3.x on WildFly Application Server 8.x, 9.x and 10.x 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.
WFAC	WildFly Administrative Console.
WFAS	WildFly 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 web application database server host.
DB_PORT	5432	QuartzDesk web application database server port.
DB_NAME	quartzdesk	QuartzDesk web application database name.
DB_SCHEMA	quartzdesk	QuartzDesk web application database schema.
DB_USER	quartzdesk	QuartzDesk web application database user.
DB_PASSWORD	quartzdesk	QuartzDesk web application database user password.
WFAS_INSTALL_ROOT	/usr/local/wildfly	WildFly Application Server installation directory.
WFAS_CONFIG	standalone	WildFly Application Server configuration.
WFAS_HTTP_HOST	localhost	WildFly HTTP listener host.
WFAS_HTTP_PORT	8080	WildFly HTTP listener port.
WORK_DIR	/var/quartzdesk-web.work	QuartzDesk web application work directory.

3. Requirements

3.1 Software Requirements

3.1.1 Browser

The QuartzDesk web application 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	17
FireFox	10
Internet Explorer	8
Opera	12
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) 7, 8, 9. 10.
IBM Java (JDK) 7, 8, 9.
OpenJDK 7, 8, 9, 10.

3.1.4 Application Server

WildFly Application Server 8.x.
WildFly Application Server 9.x.
WildFly Application Server 10.x.
WildFly Application Server 11.x.
WildFly Application Server 12.x.
WildFly Application Server 13.x.
WildFly Application Server 14.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	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	Microsoft JDBC driver 4.0 for SQL Server available at http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx. We strongly advise against using the alternative JTDS JDBC driver because it does not support the datetime2 data type at this time. As a result, all datetime values written by 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. For a comprehensive overview of JDBC driver versions vs. supported database versions, please refer to http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_02.
PostgreSQL	JDBC4 PostgreSQL driver available at http://jdbc.postgresql.org/.

3.1.7 QuartzDesk Web Application Archive

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

3.2 Hardware Requirements

QuartzDesk 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 web application database named `quartzdesk`[1] (DB_NAME) owned by DB_USER.

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

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

ⓘ Please note that you do not have to create any database objects (tables, keys, indices etc.) in the `quartzdesk` database / schema. These objects will be automatically created by the QuartzDesk web application during its first start.

4.2 JDBC Driver

Download and install the JDBC driver for the created database. For a list of supported JDBC drivers please refer to chapter 3.1.6.

Copy the JDBC driver JAR file(s) to WFAS`_INSTALL_ROOT/WFAS_CONFIG/deployments` directory and restart the application server.

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

4.3 JDBC Datasource

WFAS 8.x, 9.x:

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

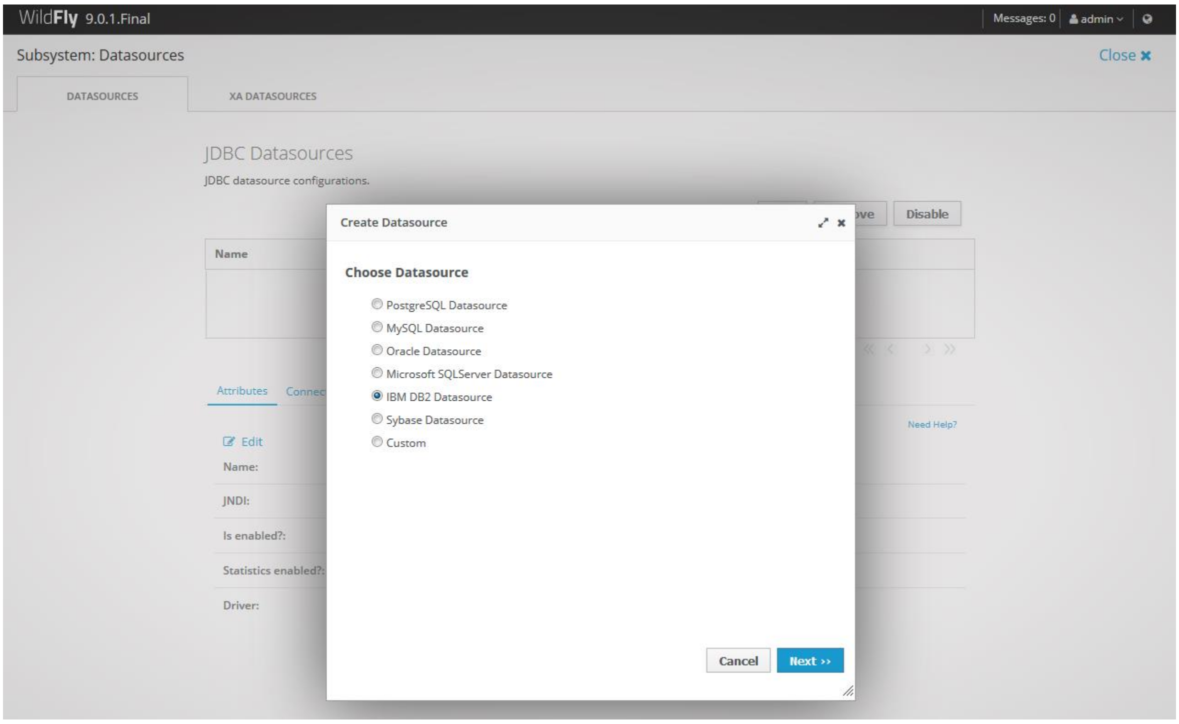
In WFAC select Configuration → Subsystems → Datasources → View and click the Add button to create a new JDBC datasource. The next steps depend on the QuartzDesk database type and are described in the following sub-chapters.

WFAS 10.x:

In WFAC select Configuration → Subsystems → Non-XA → and click the Add button in the Datasource column to create a new JDBC datasource. The next steps depend on the QuartzDesk database type and are described in the following sub-chapters.

4.3.1 DB2

In the Create Datasource dialog, select the IBM DB2 Datasource option.



Create Datasource

Choose Datasource

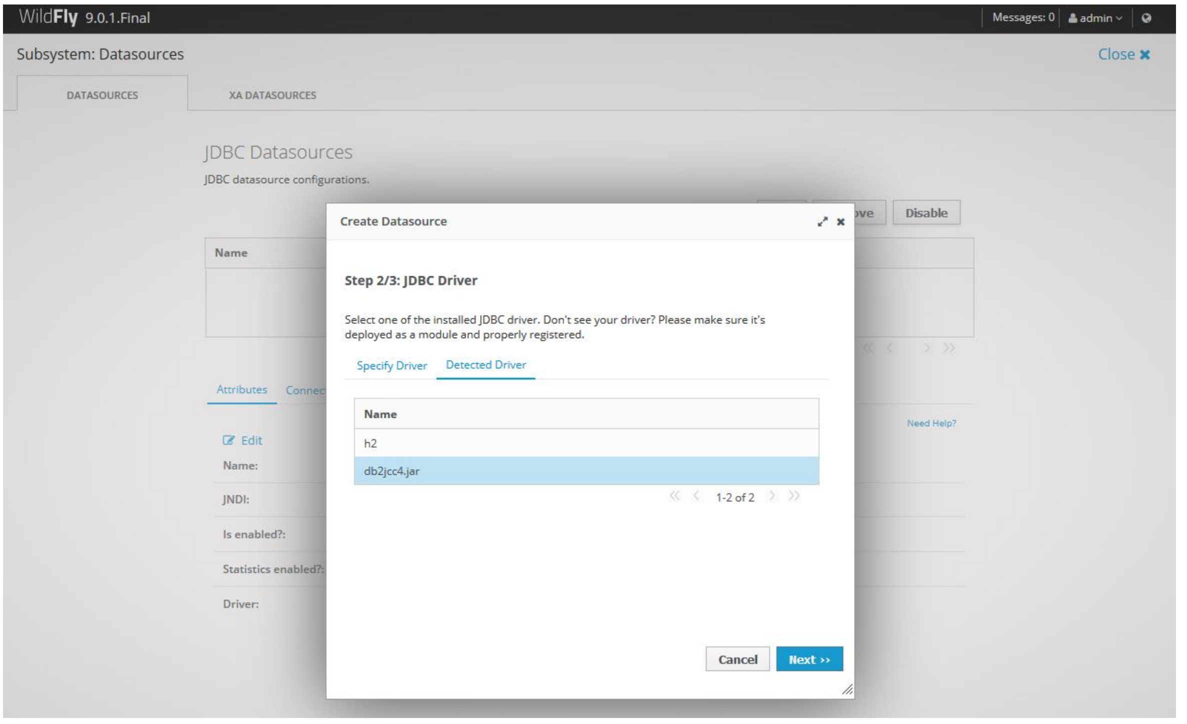
○ PostgreSQL Datasource
○ MySQL Datasource
○ Oracle Datasource
○ Microsoft SQLServer Datasource
● IBM DB2 Datasource
○ Sybase Datasource
○ Custom

Click Next.

In Step 1/3, enter the following datasource attributes:

Name: QuartzDeskDS
JNDI Name: java:/jdbc/QuartzDeskDS

In Step 2/3, select the Detected Driver tab and if the DB2 JDBC driver has been properly installed, it should appear in the list of detected drivers.



Create Datasource

Step 2/3: JDBC Driver

Select one of the installed JDBC driver. Don't see your driver? Please make sure it's deployed as a module and properly registered.

Name
h2
db2jcc4.jar

Click Next.

In Step 3/3, enter the following values:

Connection URL: jdbc:db2://DB_HOST:DB_PORT/DB_NAME
Username: DB_USER
Password: DB_PASSWORD
Security Domain: leave empty

Click Done.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

Pool
Click the Edit button and enter the following values:

Min Pool Size: 2
Initial Pool Size: 2
Max Pool Size: 10
Prefil: check

Click Save.

Properties
Add the following properties:

Key: clientApplicationInformation
Value: QuartzDesk

Validation

Click the Edit button 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

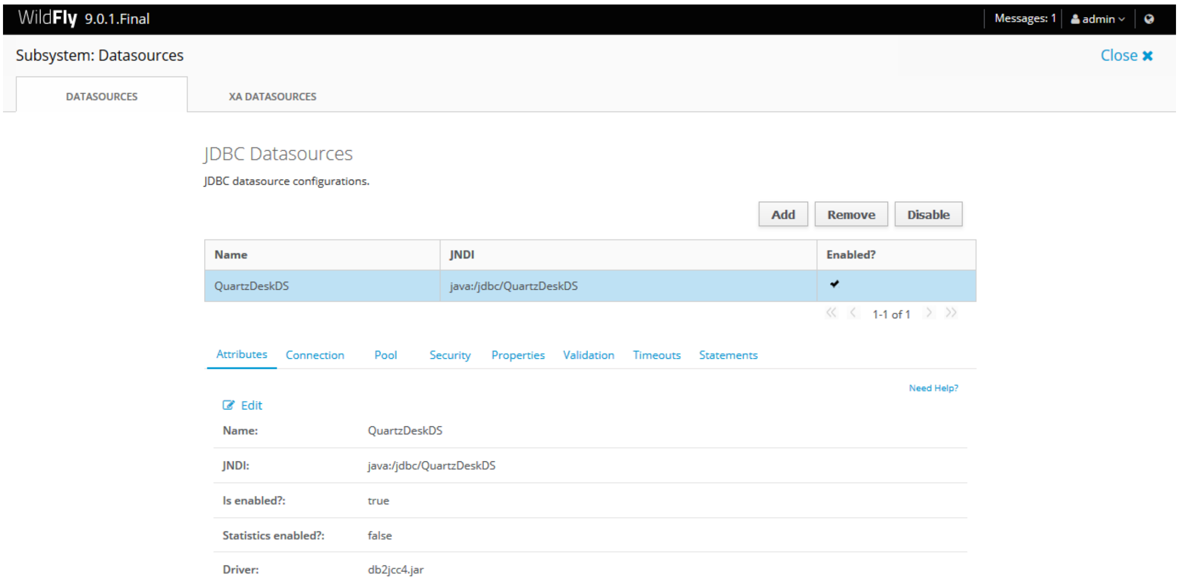
Click Save.

Statements
Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



JDBC Datasources

JDBC datasource configurations.

Name	JNDI	Enabled?
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

Name: QuartzDeskDS

JNDI: java:/jdbc/QuartzDeskDS

Is enabled?: true

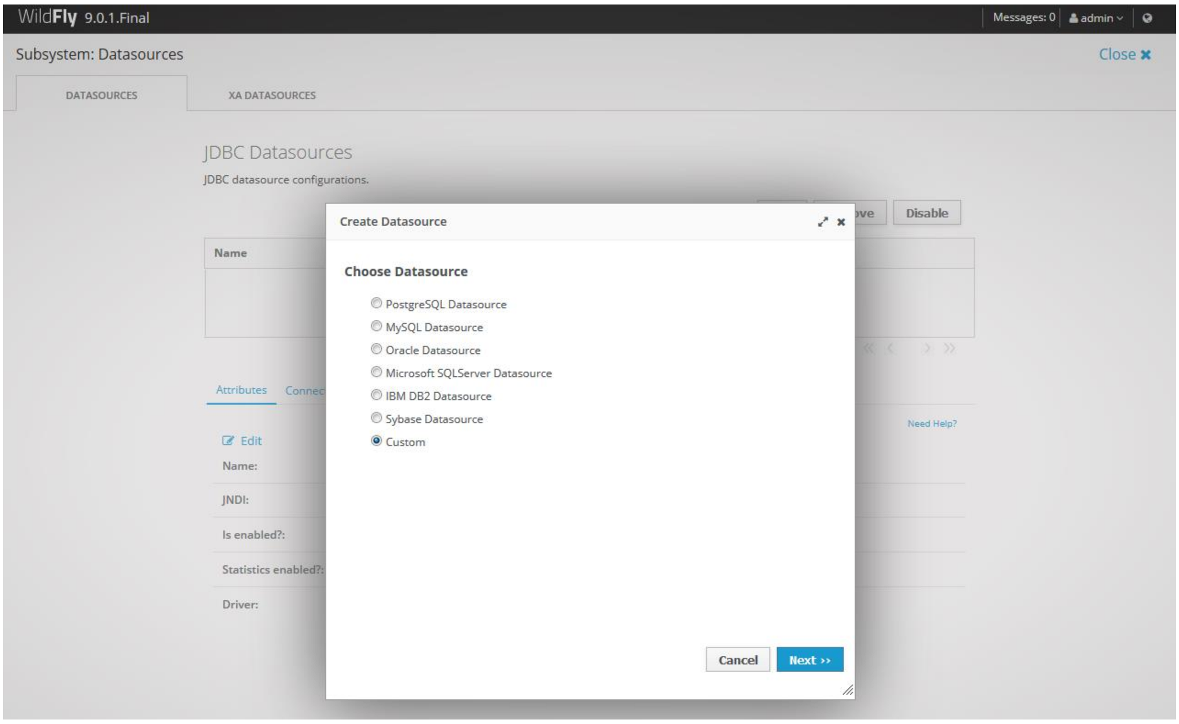
Statistics enabled?: false

Driver: db2jcc4.jar

4.3.2 **H2**

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

In the Create Datasource dialog, select the Custom option.



Create Datasource

Choose Datasource

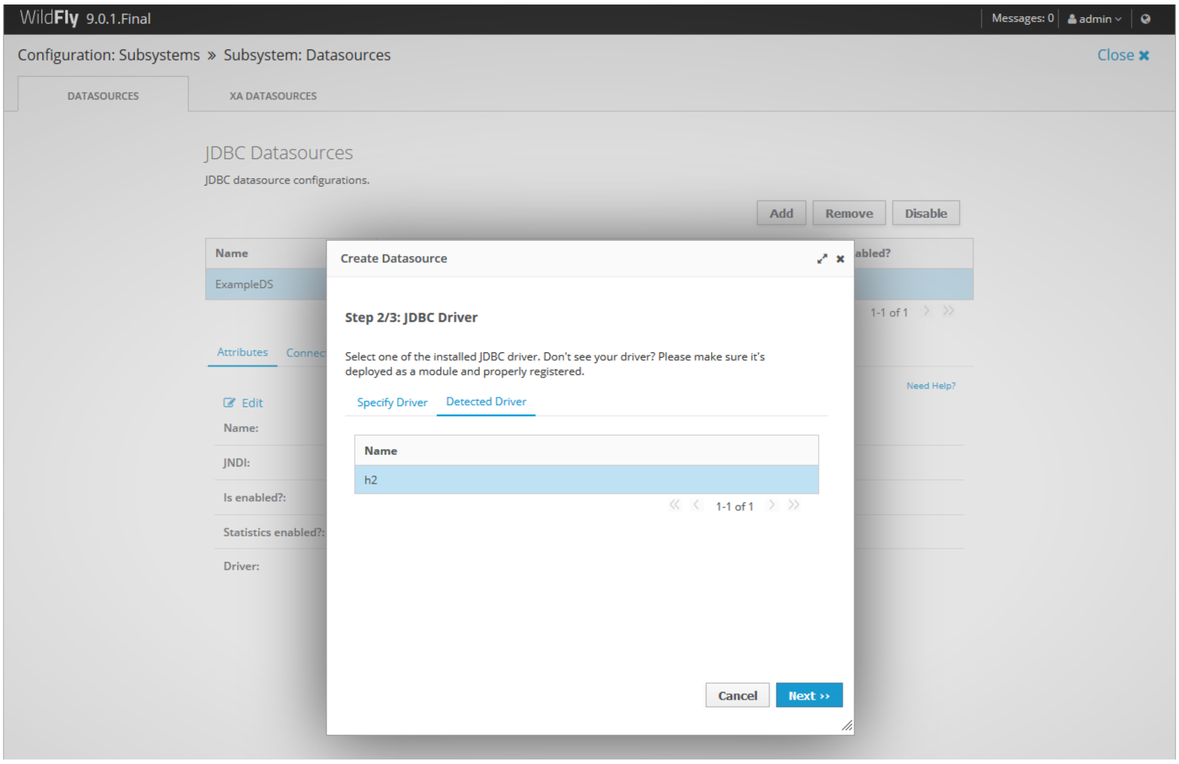
○ PostgreSQL Datasource
○ MySQL Datasource
○ Oracle Datasource
○ Microsoft SQLServer Datasource
○ IBM DB2 Datasource
○ Sybase Datasource
● Custom

Click Next.

In Step 1/3, enter the following datasource attributes:

Name: QuartzDeskDS
JNDI Name: java:/jdbc/QuartzDeskDS

In Step 2/3, select the Detected Driver tab and if the H2 JDBC driver has been properly installed, it should appear in the list of detected drivers.



Create Datasource

Step 2/3: JDBC Driver

Select one of the installed JDBC driver. Don't see your driver? Please make sure it's deployed as a module and properly registered.

Name
h2

Click Next.

In Step 3/3, 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.

Click Done.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

Pool
Click the Edit button and enter the following values:

Min Pool Size: 2
Initial Pool Size: 2
Max Pool Size: 10
Prefil: check

Click Save.

Properties
Add the following properties:

Key: applicationName
Value: QuartzDesk

Validation
Click the Edit button 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

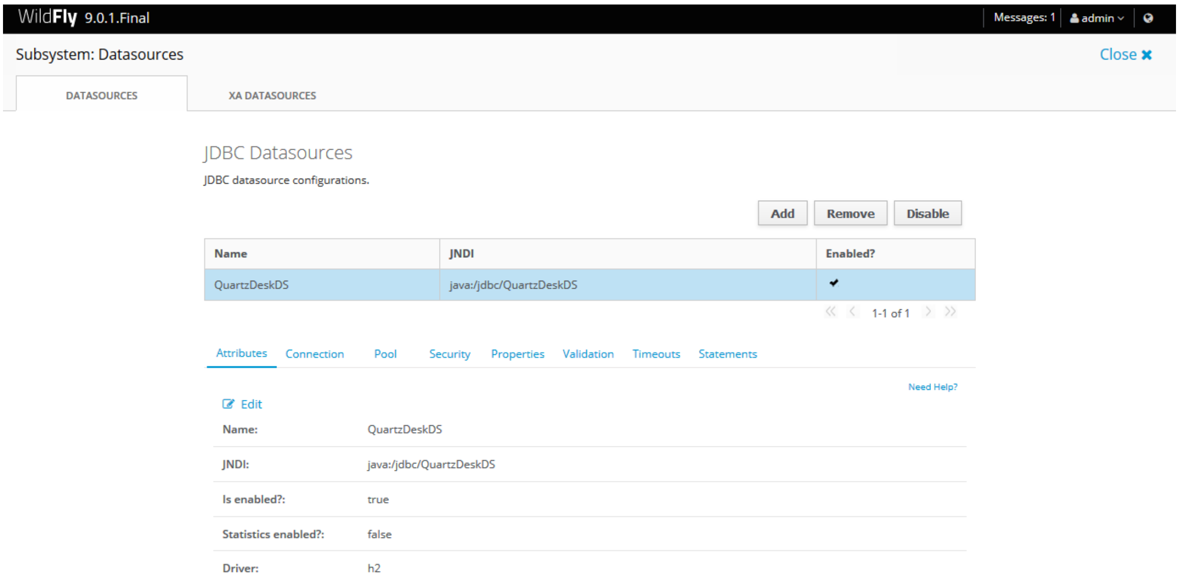
Click Save.

Statements
Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



JDBC Datasources

JDBC datasource configurations.

Name	JNDI	Enabled?
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

Name: QuartzDeskDS

JNDI: java:/jdbc/QuartzDeskDS

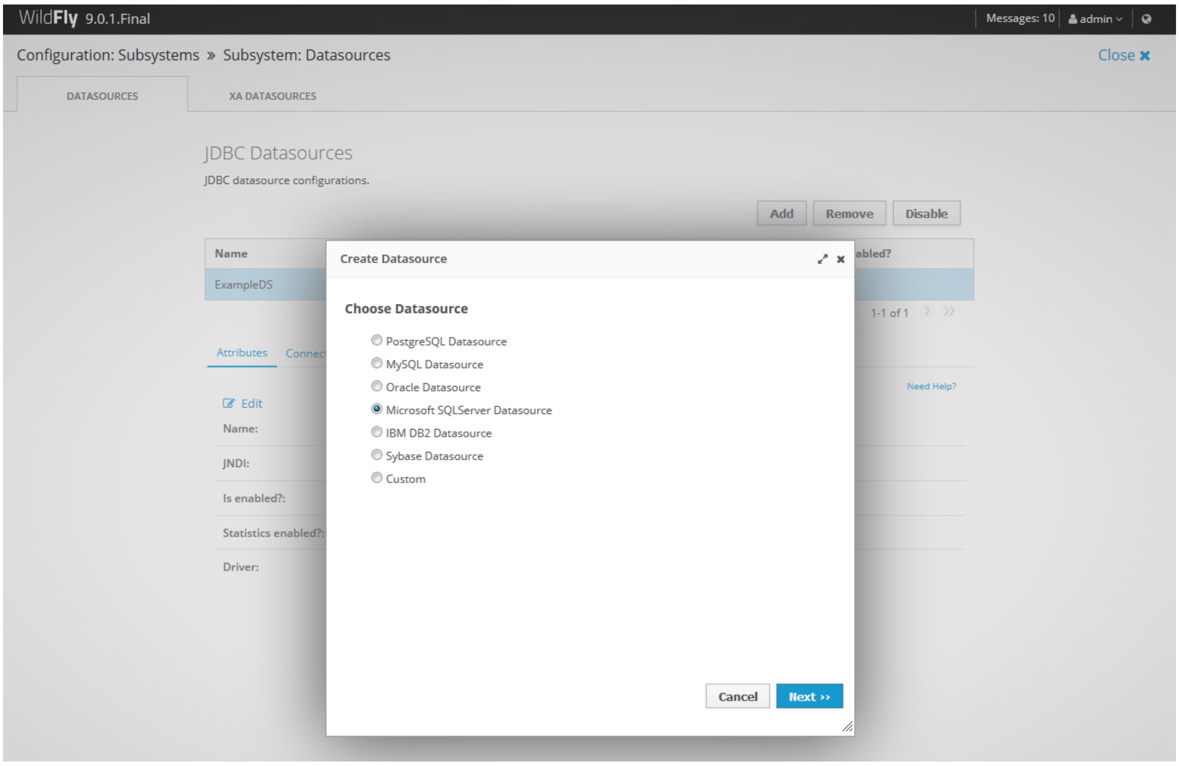
Is enabled?: true

Statistics enabled?: false

Driver: h2

4.3.3 Microsoft SQL Server

In the Create Datasource dialog, select the Microsoft SQLServer Datasource option.



Create Datasource

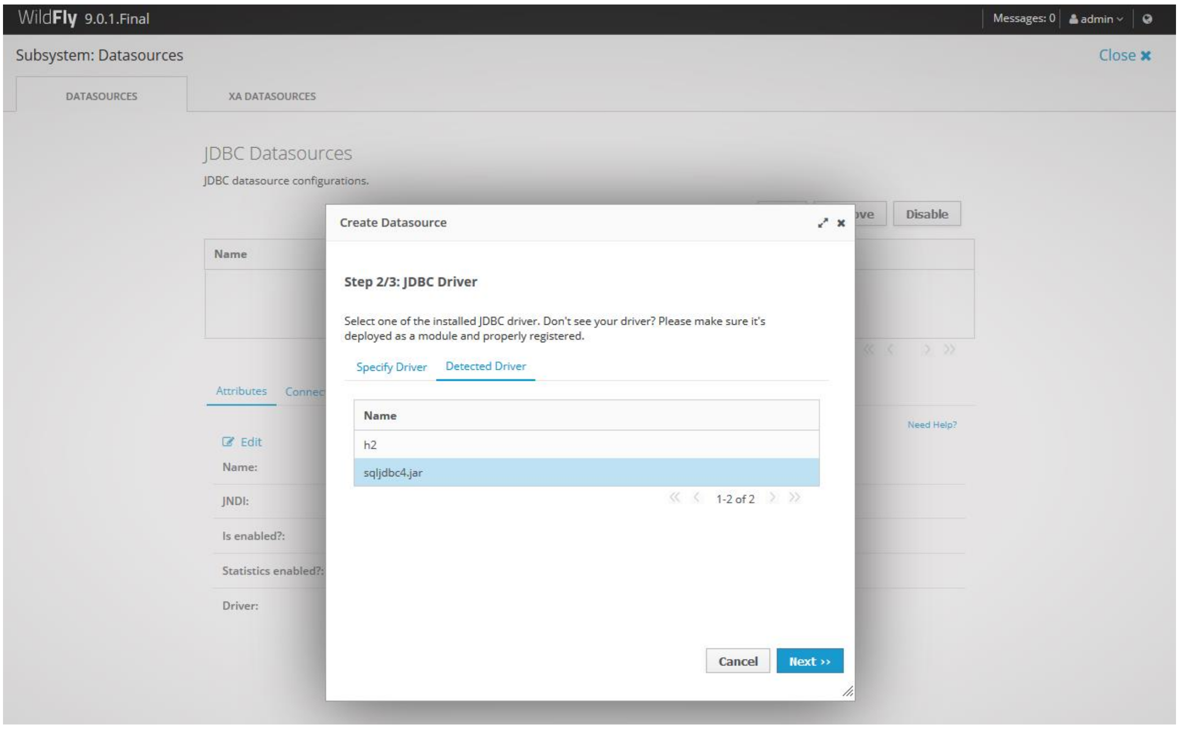
Choose Datasource

○ PostgreSQL Datasource
○ MySQL Datasource
○ Oracle Datasource
● Microsoft SQLServer Datasource
○ IBM DB2 Datasource
○ Sybase Datasource
○ Custom

Click Next.

In Step 1/3, enter the following datasource attributes:

Name: QuartzDeskDS
JNDI Name: java:/jdbc/QuartzDeskDS
In Step 2/3, select the Detected Driver tab and if the MS SQL Server JDBC driver has been properly installed, it should appear in the list of detected drivers.



Click Next.

In Step 3/3, 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

Click Done.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

Pool
Click the Edit button and enter the following values:

Min Pool Size: 2
Initial Pool Size: 2
Max Pool Size: 10
Prefil: check

Click Save.

Properties

Add the following properties:

Key: applicationName
Value: QuartzDesk

Validation

Click the Edit button and enter the following values:

Valid Connection Checker:
org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker

Click Save.

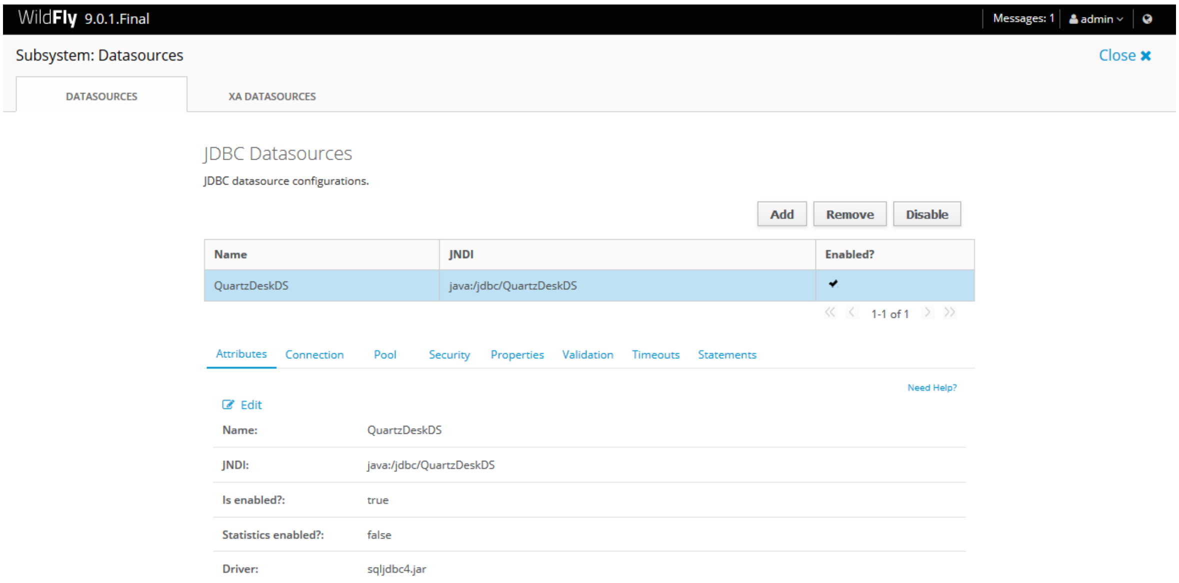
Statements

Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.

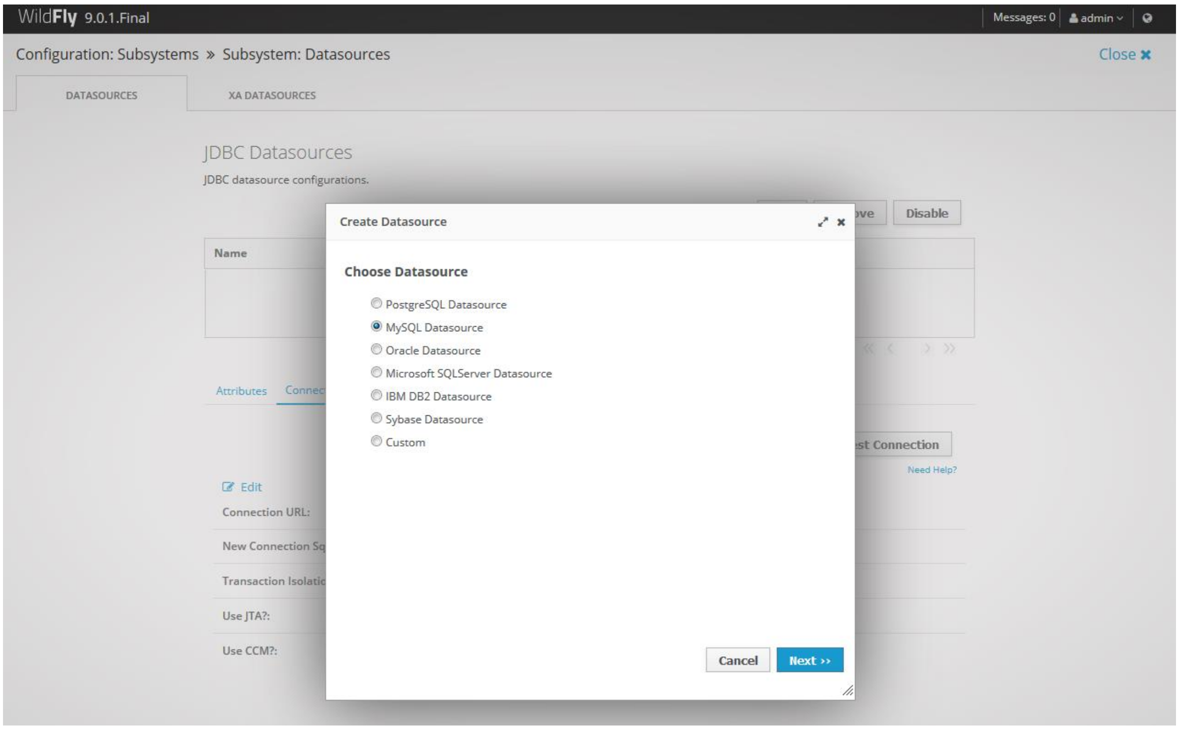


Name	JNDI	Enabled?
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

Name: QuartzDeskDS
JNDI: java:/jdbc/QuartzDeskDS
Is enabled?: true
Statistics enabled?: false
Driver: sqljdbc4.jar

4.3.4 MySQL

In the Create Datasource dialog, select the MySQL Datasource option.



Create Datasource

Choose Datasource

- ○ PostgreSQL Datasource
- ● MySQL Datasource
- ○ Oracle Datasource
- ○ Microsoft SQLServer Datasource
- ○ IBM DB2 Datasource
- ○ Sybase Datasource
- ○ Custom

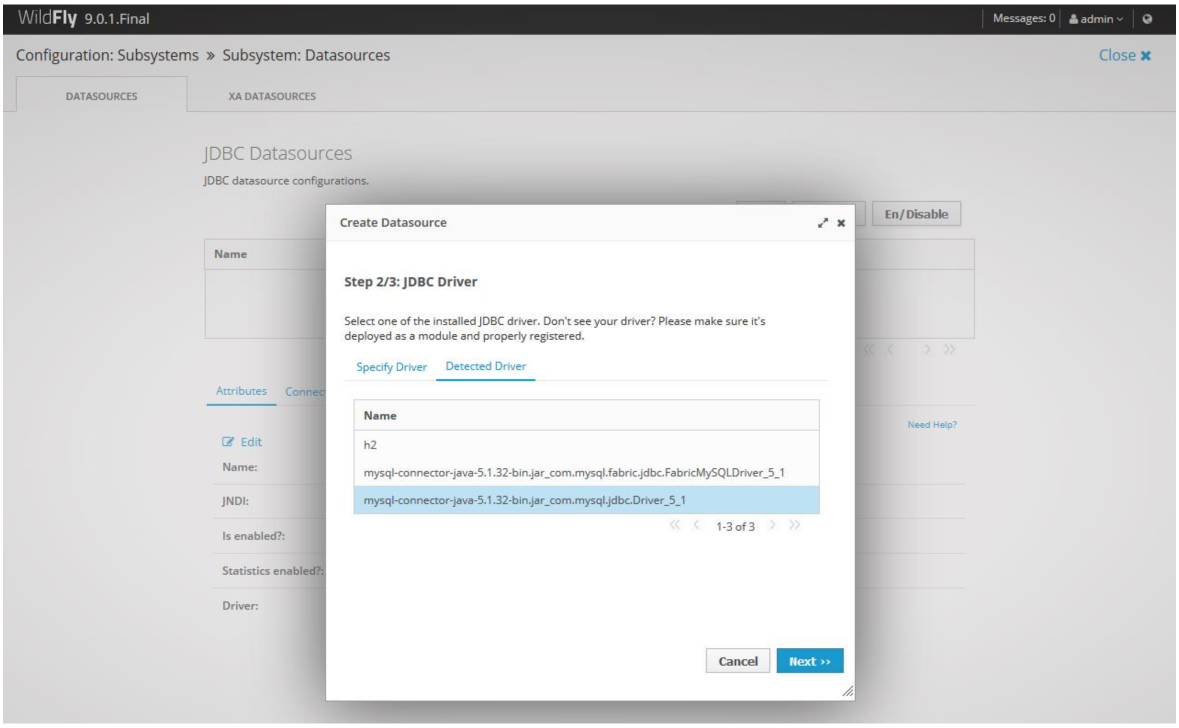
Cancel | Next ››

Click Next.

In Step 1/3, enter the following datasource attributes:

Name: QuartzDeskDS
JNDI Name: java:/jdbc/QuartzDeskDS

In Step 2/3, select the Detected Driver tab and if the MySQL Server JDBC driver has been properly installed, it should appear in the list of detected drivers.



Create Datasource

Step 2/3: JDBC Driver

Select one of the installed JDBC driver. Don't see your driver? Please make sure it's deployed as a module and properly registered.

Specify Driver | Detected Driver

Name
h2
mysql-connector-java-5.1.32-bin.jar_com.mysql.fabric.jdbc.FabricMySQLDriver_5_1
mysql-connector-java-5.1.32-bin.jar_com.mysql.jdbc.Driver_5_1

1-3 of 3

Cancel | Next ››

Click Next.

In Step 3/3, enter the following values:

Connection URL: jdbc:mysql://DB_HOST:DB_PORT/DB_NAME
Username: DB_USER
Password: DB_PASSWORD
Security Domain: leave empty

Click Done.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

Pool
Click the Edit button and enter the following values:

Min Pool Size: 2
Initial Pool Size: 2
Max Pool Size: 10
Prefil: check

Click Save.

Validation
Click the Edit button 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

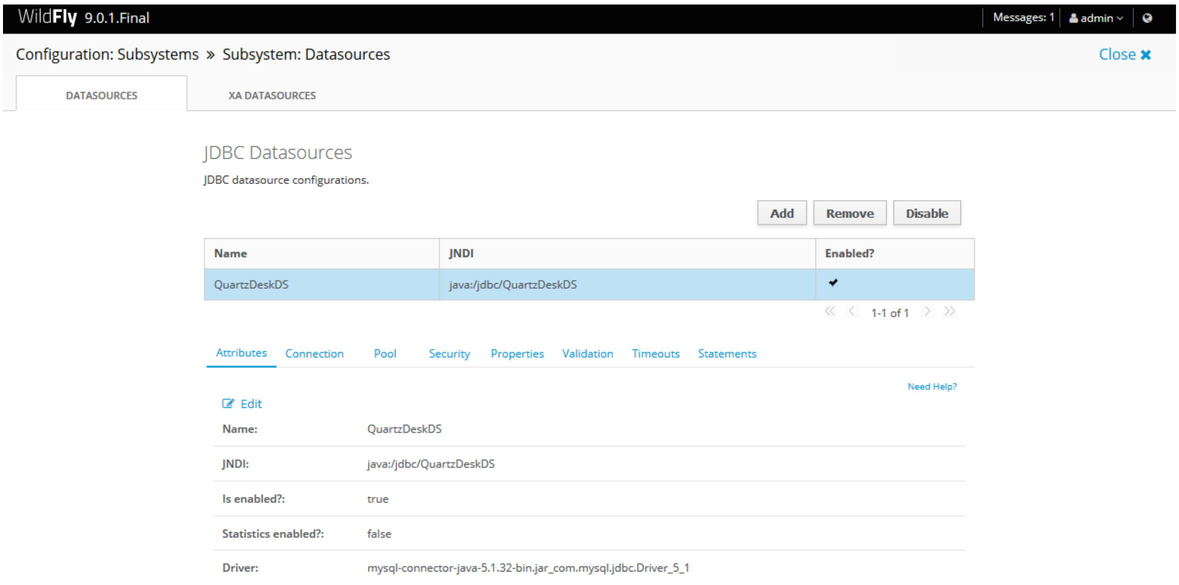
Click Save.

Statements
Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



WildFly 9.0.1.Final — Messages: 1 — admin

Configuration: Subsystems » Subsystem: Datasources — Close

DATASOURCES | XA DATASOURCES

JDBC Datasources

JDBC datasource configurations.

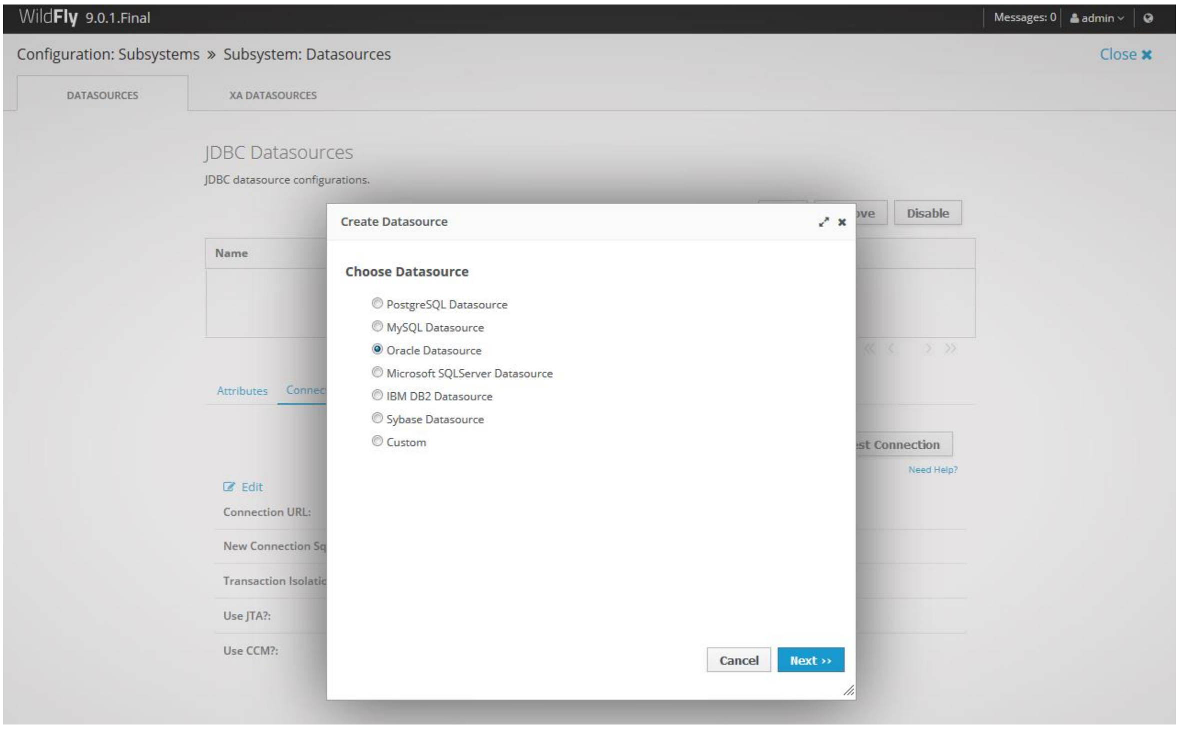
Name	JNDI	Enabled?
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

Attributes | Connection | Pool | Security | Properties | Validation | Timeouts | Statements

Name:	QuartzDeskDS
JNDI:	java:/jdbc/QuartzDeskDS
Is enabled?:	true
Statistics enabled?:	false
Driver:	mysql-connector-java-5.1.32-bin.jar_com.mysql.jdbc.Driver_5_1

4.3.5 Oracle

In the Create Datasource dialog, select the Oracle Datasource option.



Create Datasource

Choose Datasource

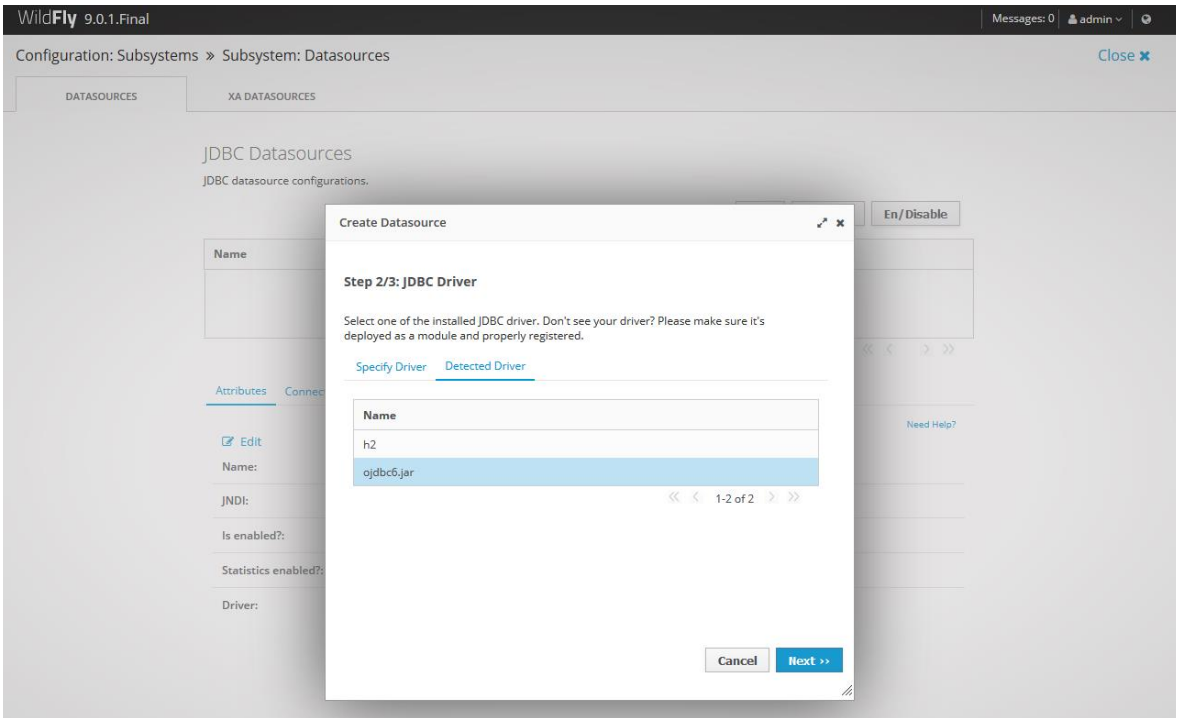
- ○ PostgreSQL Datasource
- ○ MySQL Datasource
- ● Oracle Datasource
- ○ Microsoft SQLServer Datasource
- ○ IBM DB2 Datasource
- ○ Sybase Datasource
- ○ Custom

Click Next.

In Step 1/3, enter the following datasource attributes:

Name: QuartzDeskDS
JNDI Name: java:/jdbc/QuartzDeskDS

In Step 2/3, select the Detected Driver tab and if the Oracle JDBC driver has been properly installed, it should appear in the list of detected drivers.



Step 2/3: JDBC Driver

Select one of the installed JDBC driver. Don't see your driver? Please make sure it's deployed as a module and properly registered.

Name
h2
ojdbc6.jar

Click Next.

In Step 3/3, 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

Click Done.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

Pool
Click the Edit button and enter the following values:

Min Pool Size: 2
Initial Pool Size: 2
Max Pool Size: 10
Prefil: check

Click Save.

Validation
Click the Edit button and enter the following values:

Valid Connection Checker:
org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker

Stale Connection Checker:
 org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker
Exception Sorter:
org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter

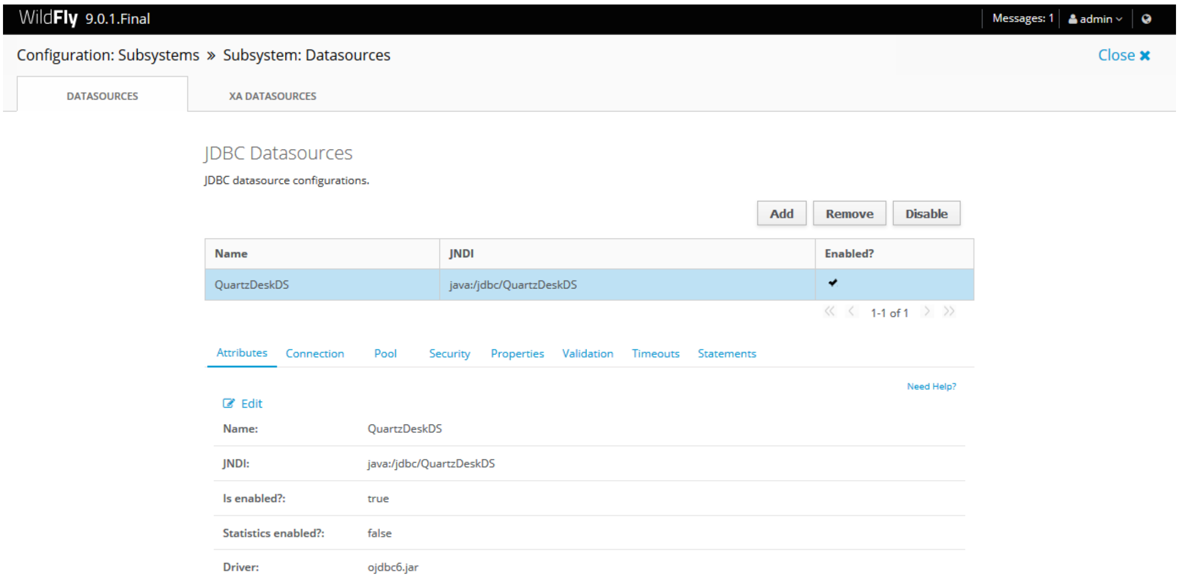
Click Save.

Statements
Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



Configuration: Subsystems » Subsystem: Datasources

JDBC Datasources

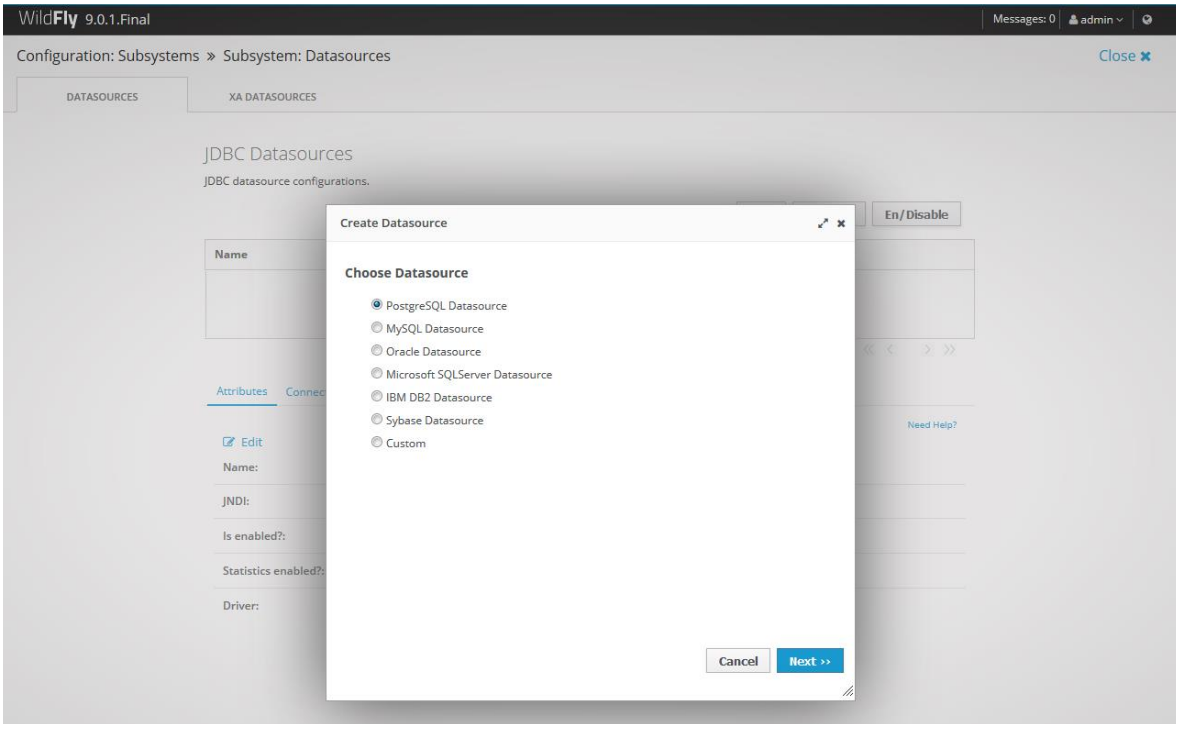
JDBC datasource configurations.

Name	JNDI	Enabled?
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

Name: QuartzDeskDS
JNDI: java:/jdbc/QuartzDeskDS
Is enabled?: true
Statistics enabled?: false
Driver: ojdbc6.jar

4.3.6 **PostgreSQL**

In the Create Datasource dialog, select the PostgreSQL Datasource option.



Create Datasource

Choose Datasource

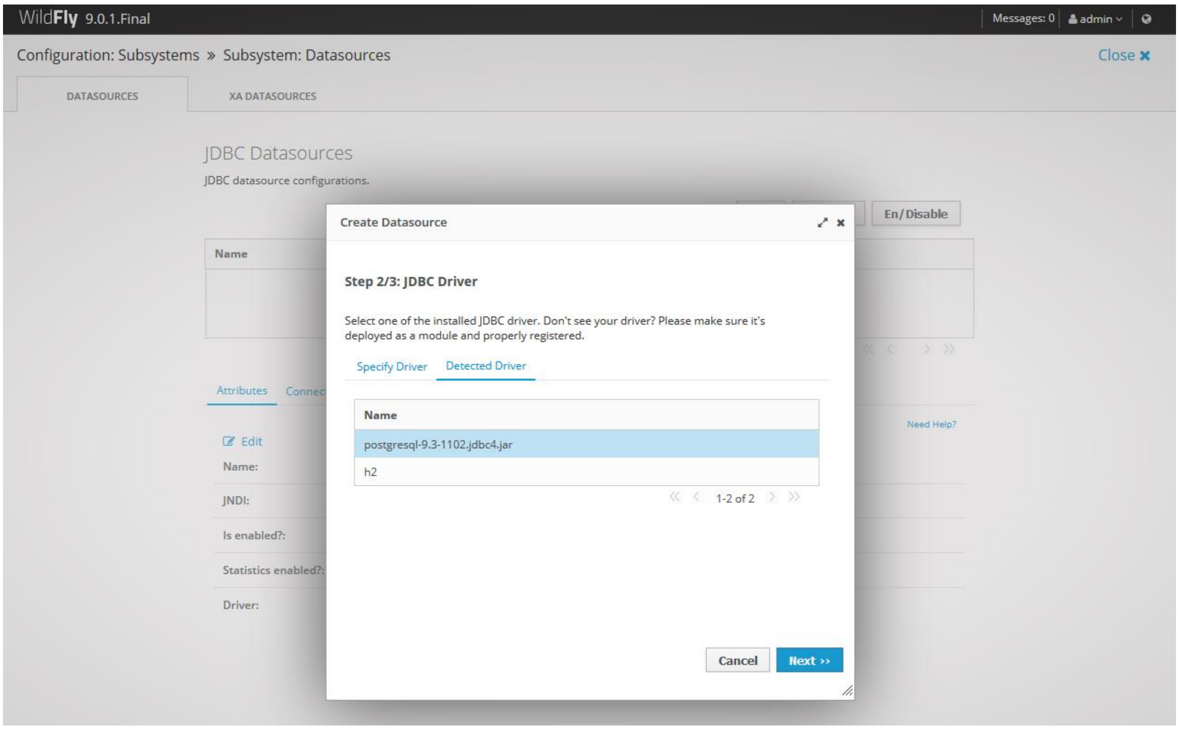
- ● PostgreSQL Datasource
- ○ MySQL Datasource
- ○ Oracle Datasource
- ○ Microsoft SQLServer Datasource
- ○ IBM DB2 Datasource
- ○ Sybase Datasource
- ○ Custom

Click Next.

In Step 1/3, enter the following datasource attributes:

Name: QuartzDeskDS
JNDI Name: java:/jdbc/QuartzDeskDS

In Step 2/3, select the Detected Driver tab and if the PostgreSQL JDBC driver has been properly installed, it should appear in the list of detected drivers.



Create Datasource

Step 2/3: JDBC Driver

Select one of the installed JDBC driver. Don't see your driver? Please make sure it's deployed as a module and properly registered.

Specify Driver | Detected Driver

Name
postgresql-9.3-1102.jdbc4.jar
h2

1-2 of 2

Click Next.

In Step 3/3, enter the following values:

Connection URL: jdbc:postgresql://DB_HOST:DB_PORT/DB_NAME
Username: DB_USER
Password: DB_PASSWORD
Security Domain: leave empty

Click Done.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

Pool
Click the Edit button and enter the following values:

Min Pool Size: 2
Initial Pool Size: 2
Max Pool Size: 10
Prefil: check

Click Save.

Properties
Add the following properties:

Key: applicationName
Value: QuartzDesk

Validation
Click the Edit button 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

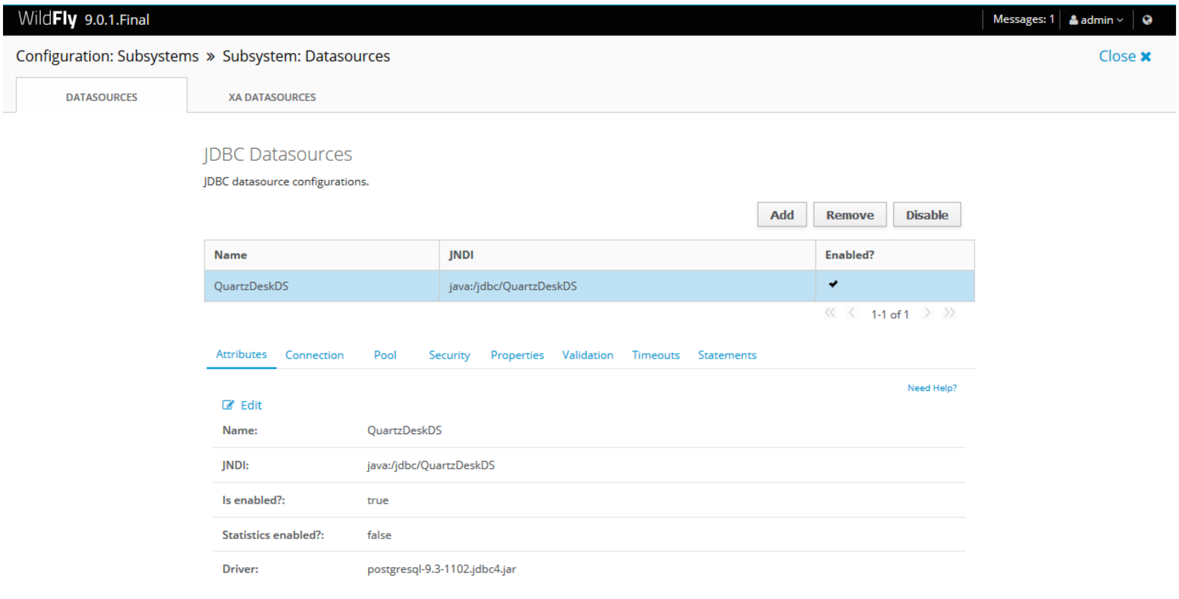
Click Save.

Statements
Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



JDBC Datasources

JDBC datasource configurations.

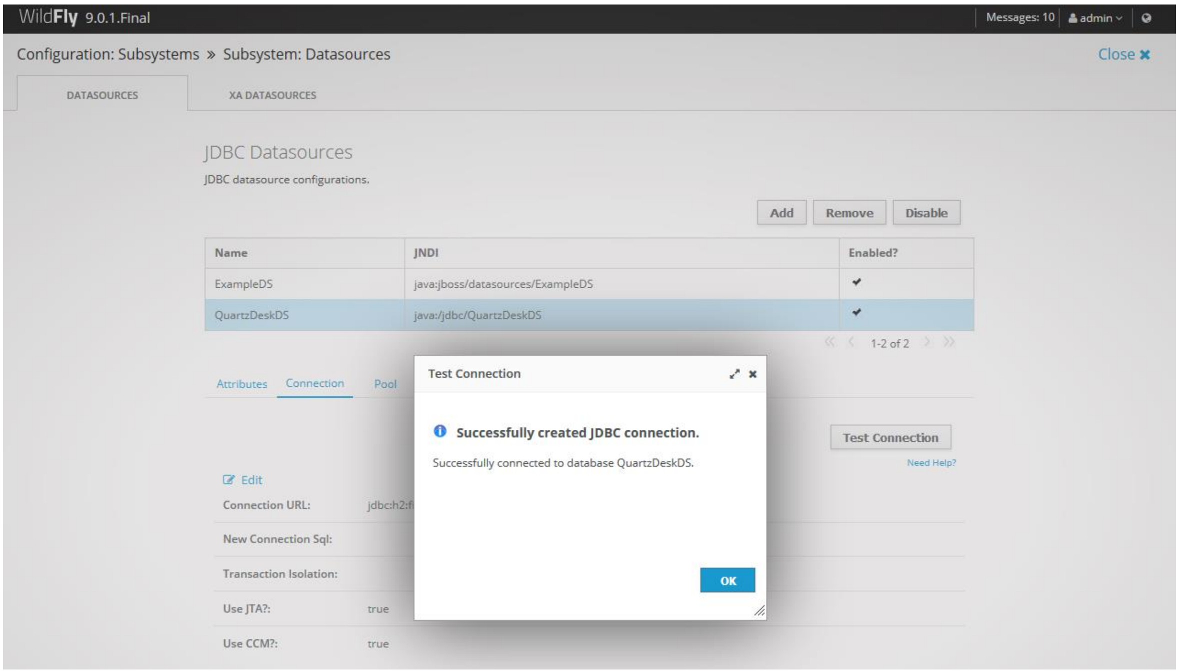
Name	JNDI	Enabled?
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

Name:	QuartzDeskDS
JNDI:	java:/jdbc/QuartzDeskDS
Is enabled?:	true
Statistics enabled?:	false
Driver:	postgresql-9.3-1102.jdbc4.jar

4.4 Test JDBC Datasource

Select the registered QuartzDeskDS datasource and open the Connection tab and click the Test Connection button.

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



JDBC Datasources

JDBC datasource configurations.

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓
QuartzDeskDS	java:/jdbc/QuartzDeskDS	✓

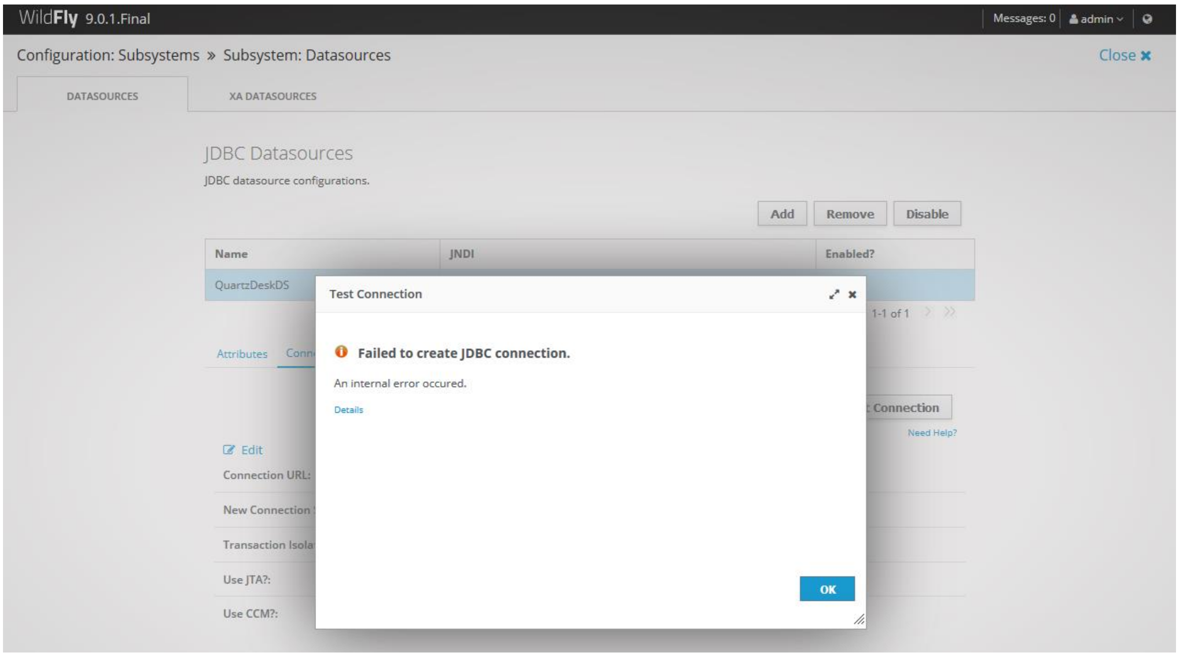
Test Connection

Successfully created JDBC connection.

Successfully connected to database QuartzDeskDS.

Connection URL:	jdbc:h2:f
New Connection Sql:	
Transaction Isolation:	
Use JTA?:	true
Use CCM?:	true

If the connection test fails, an error dialog appears and the error is logged in the WFAS log (`WFAS_INSTALL_ROOT/WFAS_CONFIG/log/server.log`).



Test Connection

Failed to create JDBC connection.

An internal error occured.

Details

4.5 Application Work Directory

Create a QuartzDesk web application work directory (WORK_DIR) anywhere on the local file system. The directory must be readable and writeable by the user the WFAS process is running under.

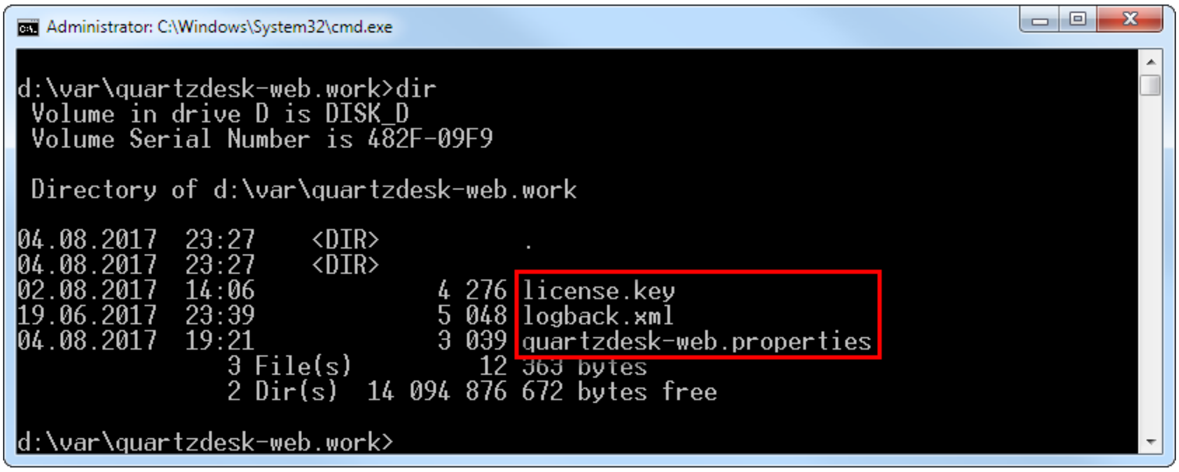
Copy your QuartzDesk license key file (`license.key`) to WORK_DIR.

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

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

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

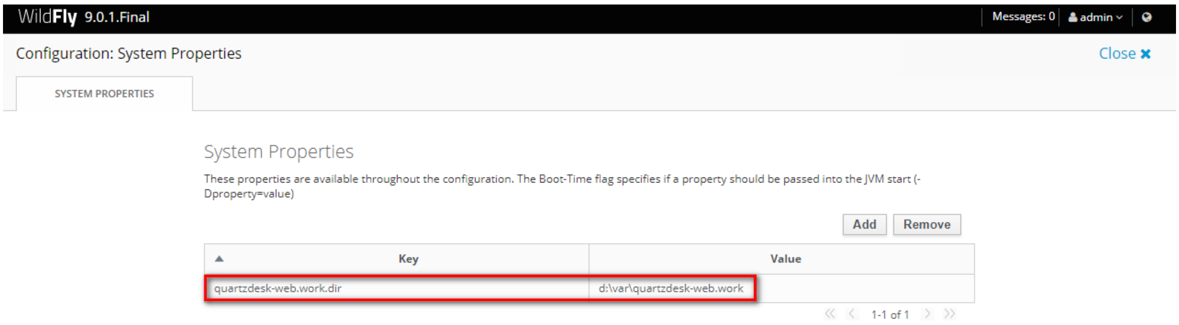
In the following figure you can see an example of a QuartzDesk web application work directory correctly set up on a Microsoft Windows machine.



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

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

04.08.2017  23:27    <DIR>          .
04.08.2017  23:27    <DIR>          ..
02.08.2017  14:06             4 276 license.key
19.06.2017  23:39             5 048 logback.xml
04.08.2017  19:21             3 039 quartzdesk-web.properties
               3 File(s)         12 363 bytes
               2 Dir(s)  14 094 876 672 bytes free

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

In WFAC go to Configuration → System Properties → View. Click the Add button to add a new boot-time system property:

Name: quartzdesk-web.work.dir
Value: WORK_DIR



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)

Key	Value
quartzdesk-web.work.dir	d:\var\quartzdesk-web.work

4.6 Application Configuration

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

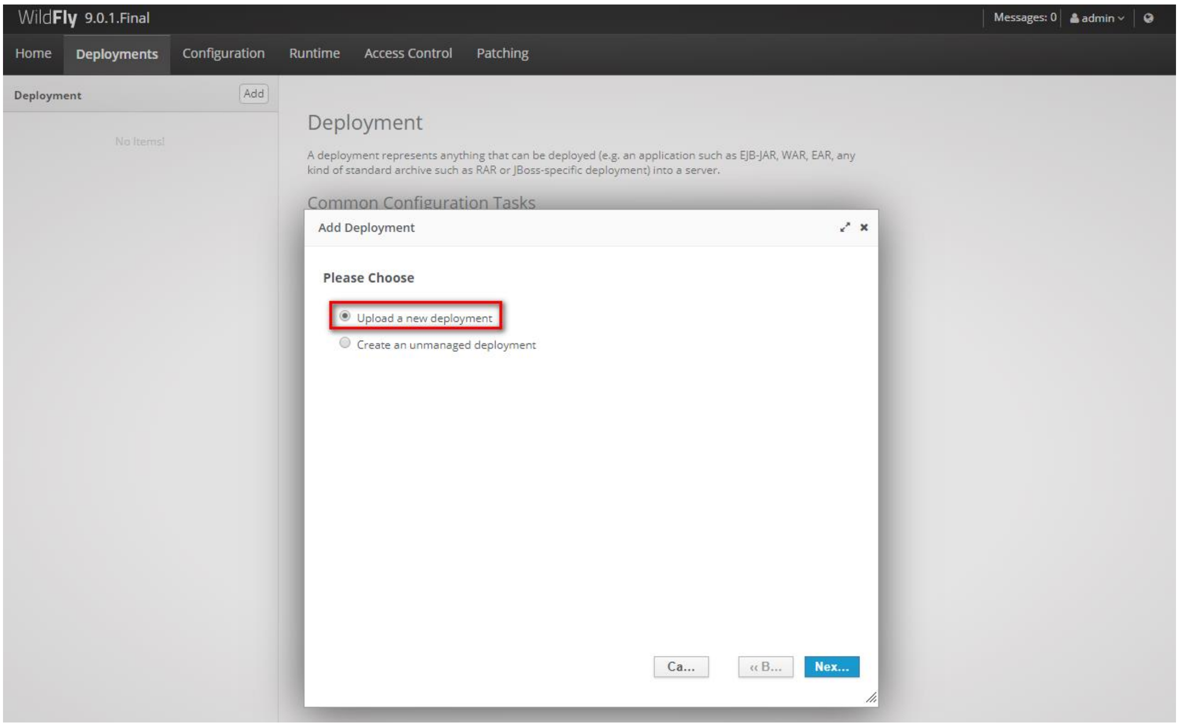
Based on the type and version of the database created in step 4.1, change the value of the `db.profile` configuration property according to the following table.

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 web application logging parameters by editing the `WORK_DIR/logback.xml` configuration file. The default sample `logback.xml` configuration file makes QuartzDesk web application log under the `WORK_DIR/logs` directory that is automatically created when the web application starts. Please refer to the Logback Manual for Logback configuration details.

4.7 Deploy Application

In WFAC go to Deployments and click the Add button. In the Add Deployment dialog select the “Upload a new deployment” option.



Deployment

A deployment represents anything that can be deployed (e.g. an application such as EJB-JAR, WAR, EAR, any kind of standard archive such as RAR or JBoss-specific deployment) into a server.

Add Deployment

Please Choose

● Upload a new deployment

○ Create an unmanaged deployment

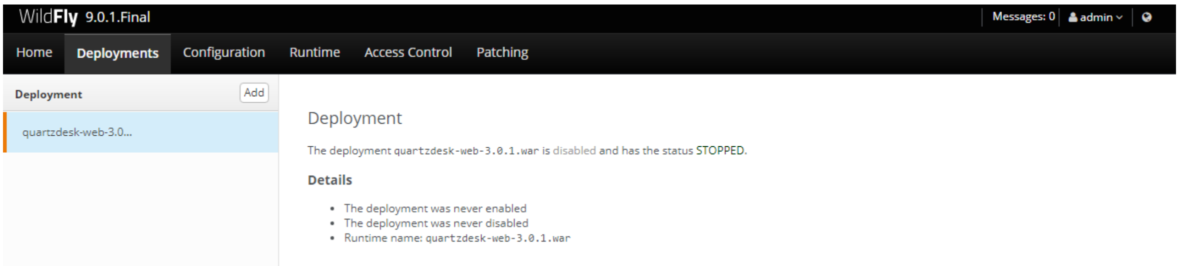
Click Next.

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

In the next step, make no changes and click Finish.

The WildFly deployment descriptor packaged in the application sets the QuartzDesk web application context root to /quartzdesk. Therefore, there is no need to modify deployment names.

Deployed quartzdesk-web-x.y.z.war file should appear in the list of deployments.



Deployment

The deployment `quartzdesk-web-3.0.1.war` is disabled and has the status STOPPED.

Details

- The deployment was never enabled
- The deployment was never disabled
- Runtime name: `quartzdesk-web-3.0.1.war`

4.8 Start Application

In WFAC go to Deployments, select the QuartzDesk web application and click on the (En/Dis)able button to enable the application. Confirm this action in a dialog window that opens and wait for the action to complete.

The application should be starting now.

Monitor the WFAS `server.log` log file under `WFAS_INSTALL_ROOT/WFAS_CONFIG/logs` for errors and wait for the application start operation to complete.

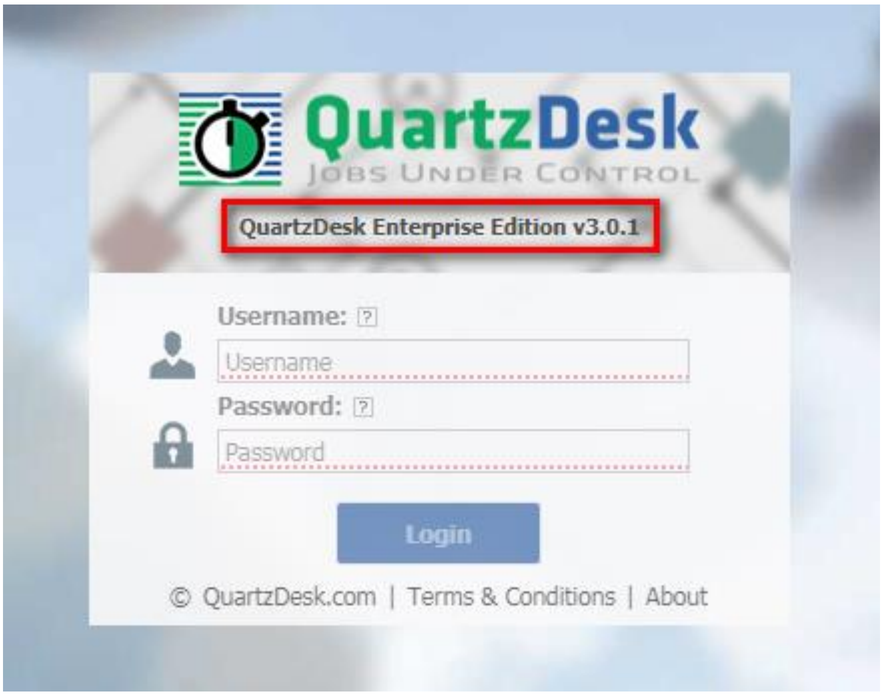
You can safely ignore the following warning messages:

```
2017-08-04 14:41:38,997 WARN  [org.jboss.as.server.deployment]
(MSC service thread 1-2) WFLYSRV0059: Class Path entry
lib/<library>.jar in /d:/Java/wildfly-
9.0.1.Final/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 located in the `WORK_DIR/logs` directory) for errors.

If there are no errors, point your browser to `http://WFAS_HTTP_HOST:WFAS_HTTP_PORT/quartzdesk/` and verify that the QuartzDesk web application GUI is accessible.

Check the version number of the deployed QuartzDesk web application.



QuartzDesk Enterprise Edition v3.0.1

To log in, use the default administrator login credentials:

Username: admin

Password: admin123

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

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

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

5. Upgrading

5.1 Stop Existing Application

In WFAC go to Deployments, select the QuartzDesk web application and click on the (En/Dis)able button to disable the application. Confirm this action in a dialog window that opens and wait for the action to complete.

The application should be stopping now.

Monitor the WFAS `server.log` log file under `WFAS_INSTALL_ROOT/WFAS_CONFIG/logs` for errors and wait for the application stop operation to complete.

5.2 Backup

Backup the QuartzDesk web application database. We recommend performing a **full database backup**.

Backup the contents of the QuartzDesk web application 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 existing QuartzDesk web application version if the need arises.

5.3 Remove Existing Application

In WFAC go to Deployments, select the deployed QuartzDesk web application. Click on the down arrow in the (En/Dis)able button and select the Remove menu 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 Deployment list.

5.4 Deploy New Application

Deploy the new version of the QuartzDesk web application by following the deployment steps outlined in 4.7.

⚠ Some WFAS 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 `WFAS_INSTALL_ROOT/WFAS_CONFIG/tmp/quartzdesk-web-x.y.z` Once the cache has been purged, start the WFAS instance.

5.5 Start New Application

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

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

To upgrade QuartzDesk web application 2.x to 3.x, follow the upgrade steps outlined in 5.

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

6.1 Minimum Required Java Version

QuartzDesk web application 3.x requires Java 7 or higher. Java 6 is no longer supported.

Make sure WFAS is configured to use Java 7 or higher.

6.2 Rename Configuration File

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

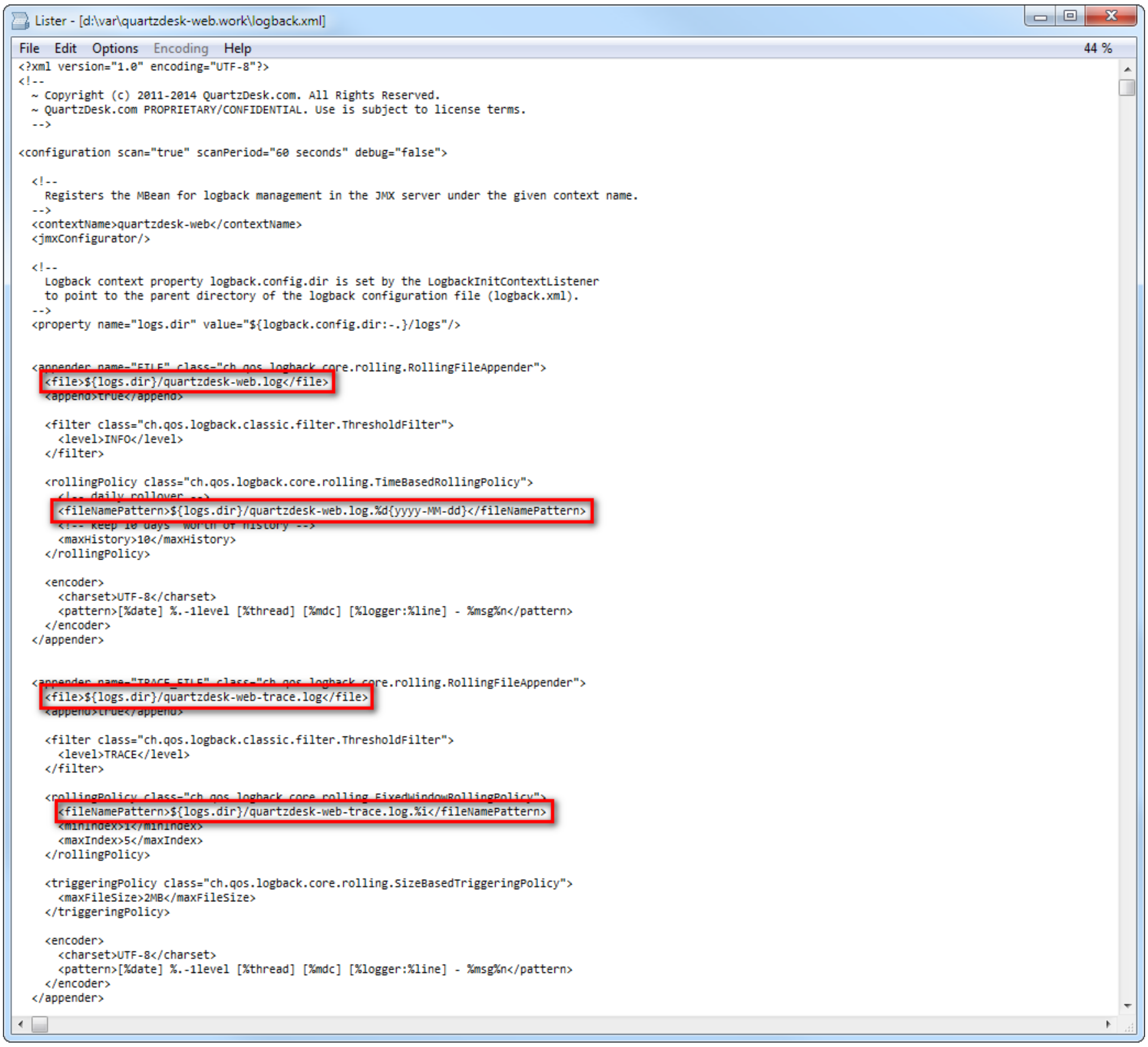
Rename the existing configuration file `quartzdesk.properties` located in the QuartzDesk web application work directory.

6.3 Rename Log Files

The names of QuartzDesk web application 3.x log files have changed.

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

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



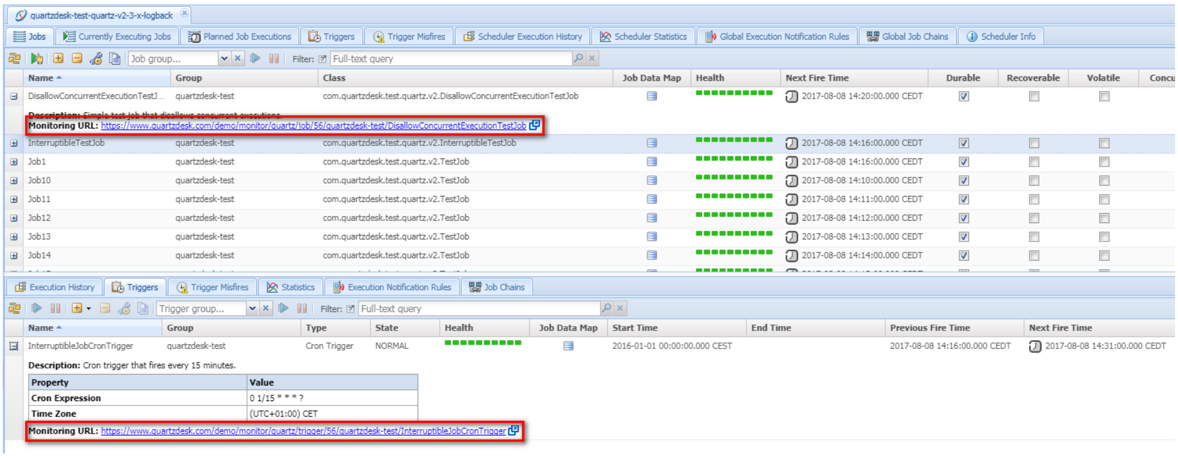
Lister - [d:\var\quartzdesk-web.work\logback.xml]
File Edit Options Encoding Help
44 %

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

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

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

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


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

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

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

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


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

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

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

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

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

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

6.4 Access to Monitoring URLs (REST API)

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



	Name	Group	Class	Job Data Map	Health	Next Fire Time	Durable	Recoverable	Volatile	Concu
⊟	DisallowConcurrentExecutionTestJ...	quartzdesk-test	com.quartzdesk.test.quartz.v2.DisallowConcurrentExecutionTestJob	▤	🟩	2017-08-08 14:20:00.000 CEDT	☑	□	□	
⊞	InterruptibleTestJob	quartzdesk-test	com.quartzdesk.test.quartz.v2.InterruptibleTestJob	▤	🟩	2017-08-08 14:16:00.000 CEDT	☑	□	□	
⊞	Job1	quartzdesk-test	com.quartzdesk.test.quartz.v2.TestJob	▤	🟩	2017-08-08 14:16:00.000 CEDT	☑	□	□	
⊞	Job10	quartzdesk-test	com.quartzdesk.test.quartz.v2.TestJob	▤	🟩	2017-08-08 14:10:00.000 CEDT	☑	□	□	
⊞	Job11	quartzdesk-test	com.quartzdesk.test.quartz.v2.TestJob	▤	🟩	2017-08-08 14:11:00.000 CEDT	☑	□	□	
⊞	Job12	quartzdesk-test	com.quartzdesk.test.quartz.v2.TestJob	▤	🟩	2017-08-08 14:12:00.000 CEDT	☑	□	□	
⊞	Job13	quartzdesk-test	com.quartzdesk.test.quartz.v2.TestJob	▤	🟩	2017-08-08 14:13:00.000 CEDT	☑	□	□	
⊞	Job14	quartzdesk-test	com.quartzdesk.test.quartz.v2.TestJob	▤	🟩	2017-08-08 14:14:00.000 CEDT	☑	□	□	

Description: Simple test job that disallows concurrent executions.
Monitoring URL: https://www.quartzdesk.com/demo/monitor/quartz/job/56/quartzdesk-test/DisallowConcurrentExecutionTestJob

	Name	Group	Type	State	Health	Job Data Map	Start Time	End Time	Previous Fire Time	Next Fire Time
⊟	InterruptibleJobCronTrigger	quartzdesk-test	Cron Trigger	NORMAL	🟩	▤	2016-01-01 00:00:00.000 CEST		2017-08-08 14:16:00.000 CEDT	2017-08-08 14:31:00.000 CEDT

Description: Cron trigger that fires every 15 minutes.

Property	Value
Cron Expression	0 1/15 * * * ?
Time Zone	(UTC+01:00) CET

Monitoring URL: https://www.quartzdesk.com/demo/monitor/quartz/trigger/56/quartzdesk-test/InterruptibleJobCronTrigger

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

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

6.5 Access to JAX-WS Endpoints

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

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

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

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

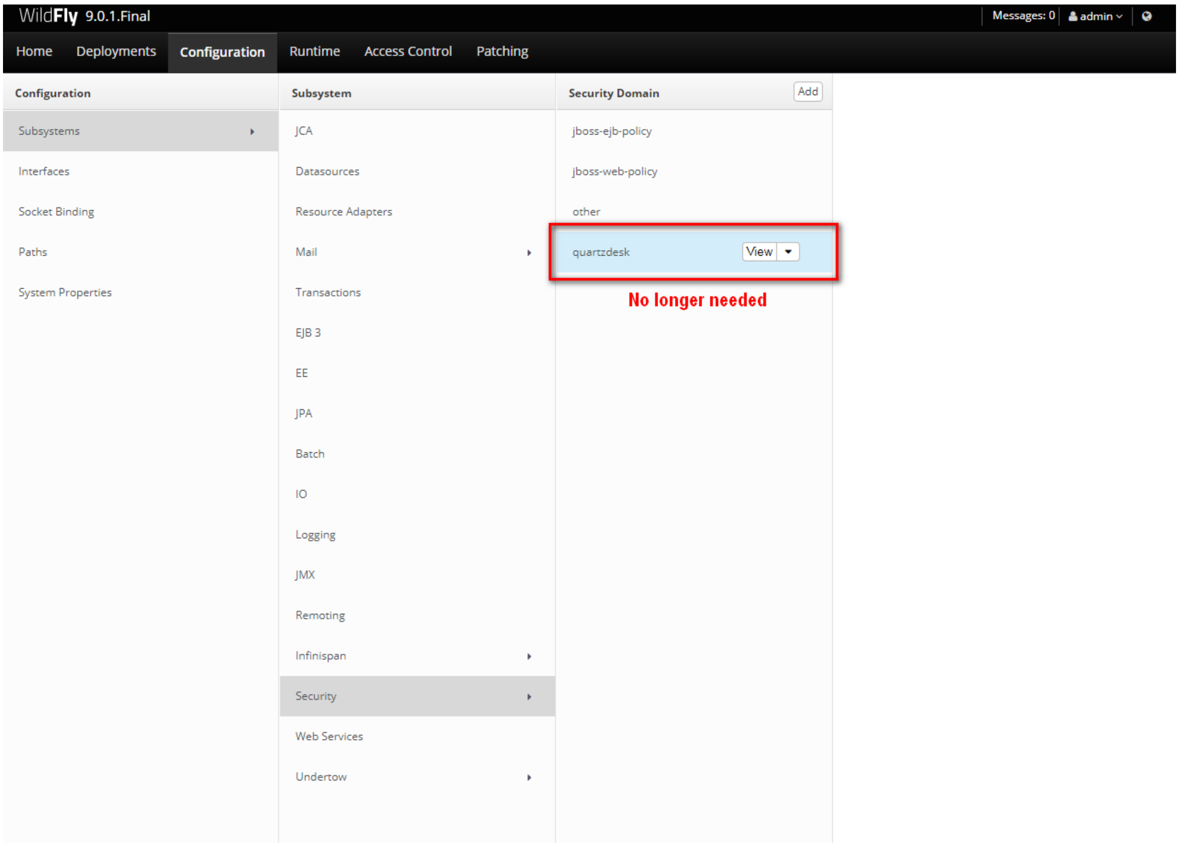
All JAX-WS web service endpoints in QuartzDesk 3.x support the HTTP Basic authentication scheme where the user's authentication credentials are passed in the `Authorization` HTTP header. Please note that the same authentication scheme was used by JAX-WS endpoints in QuartzDesk 2.x.

6.6 Remove Unused Security Domain

WFAS 8.x:
In WFAC go to Configuration → Security → Security Domains

WFAS 9.x, 10.x:
In WFAC go to Configuration → Subsystems → Security → Security Domain

Select **quartzdesk** security domain and remove it by clicking on the Remove menu option in the menu next to the security domain name.



Remove the users and roles properties files that were used by the removed security domain:

```
WFAS_INSTALL_ROOT/WFAS_CONFIG/quartzdesk-users.properties
```

```
WFAS_INSTALL_ROOT/WFAS_CONFIG/quartzdesk-roles.properties
```

7. Cluster Deployment Notes

When deploying the QuartzDesk web application to a WildFly 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.

7.1 HTTP Session Replication and Affinity

The 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 the 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 WildFly instance that handled the first user request that established the session.

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

7.2 Shared Work Directory

We recommend that you put the QuartzDesk web application 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 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.

7.3 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.

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

7.3.1 Using Shared Log Files

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

```
...

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


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

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

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

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

...
```

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

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

7.3.2 Using Separate Log Files

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

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

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

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

```
...

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

  ...

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

  ...

</appender>


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

  ...

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

  ...

</appender>

...
```

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

```
...

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

...
```

7.4 Internal Quartz Scheduler

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

Implementation	Description
`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. This is the default implementation used by the QuartzDesk web application. No 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. This implementation extracts the Quartz scheduler instance ID from the `org.quartz.scheduler.instanceId` JVM system property that must be explicitly set. Please refer to the WildFly 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.