**What is QuartzDesk?**

In one sentence, QuartzDesk is a highly flexible enterprise-class management and monitoring platform for Quartz schedulers embedded in Java applications.

**TYPICAL USERS**

**DEVELOPERS**

As a developer you can use QuartzDesk to ad-hoc execute individual jobs, off their regular schedule. Therefore you can debug, test and tune your Quartz jobs any time you want. Prior to executing a job, you can modify job's start-up parameters (job data map properties in Quartz scheduler speak) and thus simulate job execution in various contexts.

Last but not least, you can use QuartzDesk to view and analyze log messages produced by executed jobs without the need to download and filter lengthy application log files cluttered with messages produced by parallelly running JVM threads.

**TESTERS**

As a tester you can use QuartzDesk to inspect execution history of individual Quartz jobs and see when these jobs were executed and with what results and confront it with the application specification and use-cases.

Using the execution history you can easily identify all job execution failures and view the relevant job execution logs that can be passed back to developers for further analysis and fixing.

**OPERATIONS**

As an operations team member you can use QuartzDesk to closely monitor Quartz schedulers, their jobs and triggers in all your Quartz scheduler enabled applications. You can either plugin your existing network/system monitoring software (Nagios, Cacti,...) to QuartzDesk monitoring REST API, or you can create sophisticated notification rules in QuartzDesk and be notified by email, instant message, or web-service invocation whenever a defined job execution condition occurs.

By inspecting QuartzDesk job execution statistics you can easily **identify problematic jobs.**

QuartzDesk has been designed with the following goals in mind:

**EASY INSTALLATION**

QuartzDesk is a Java-based application distributed as a single standard web application WAR (Web Application Archive) file and two optional JAR (Java Archive) files. QuartzDesk can be easily deployed to any light-weight Java servlet container such **Apache Tomcat, Jetty, Resin, etc. as well as to any heavy weight J2EE application server such as IBM WebSphere, RedHat JBoss, Oracle WebLogic, Oracle GlassFish,** etc. QuartzDesk can run on any operating system that provides the Java runtime environment.

QuartzDesk supports all major database engines such as **DB2, H2, MySQL, MS SQL Server, Oracle and PostgreSQL.**

**ADVANCED FEATURES**

QuartzDesk provides some unique and powerful features that are not available in any of the competing products which focus primarily on the basic Quartz scheduler management functionality (start/stop/pause scheduler, listing of jobs/triggers, etc.). For a detailed overview of QuartzDesk features, please refer to our Features page.

**NON-INTRUSIVE INSTALLATION**

One of the most important aspect that we considered during the QuartzDesk design-phase was the impact on managed Quartz-enabled applications. QuartzDesk requires only a minimum set of changes to JVM and application configuration. Most importantly, users do not have to modify their application code in order to use their application with QuartzDesk.

**MODERN USER INTERFACE**

QuartzDesk features an attractive RIA (Rich Internet Application) user interface that requires no browser plugin. QuartzDesk runs in all modern browsers with enabled JavaScript support.

**FLEXIBLE LICENSING MODEL**

QuartzDesk is available in three editions **- Lite (free), Standard and Enterprise.**

Lite edition is a free-of-charge edition that provides just the basic scheduler, job and trigger management features and it imposes limits on the maximum number of managed jobs and triggers. This edition comes with no technical support.

Standard and Enterprise editions are unlimited editions that differ only in the number of available features. Both editions automatically come with 12 months of Software Maintenance that includes access to our technical support and free software upgrades.

For a detailed comparison of features available in individual QuartzDesk editions and pricing options, please refer to our Pricing & Licensing Terms page.

**HOW IT WORKS**

The QuartzDesk platform consists of the following three downloadable components:

* QuartzDesk Web Application (quartzdesk-web-x.y.z.war)
* QuartzDesk JVM Agent (quartzdesk-agent-x.y.z.jar)
* QuartzDesk Public API Library (quartzdesk-api-x.y.z.jar)

**QUARTZDESK WEB APPLICATION**

This is the main component whose installation is required for all QuartzDesk editions. The QuartzDesk Web Application is a standard Java web-application distributed as a WAR file that can be deployed to any light-weight Java servlet container (Tomcat, Jetty, etc.) or any full-fledged Java application server (IBM WebSphere, RedHat JBoss, etc.). For the complete list of all currently supported servlet containers and application servers, please refer to our Supported Platforms list. The QuartzDesk Web Application provides the Graphical User Interface (GUI) for the QuartzDesk platform. Using this GUI, users can connect via one of the supported JMX protocols to remote Quartz scheduler instances and perform all management and monitoring operations.

The QuartzDesk Web Application exposes multiple JAX-WS SOAP endpoints of web-services through which you can access most of the functionality that is available in the GUI.

The installation of the QuartzDesk Web Application component is described in the QuartzDesk Web-Application Installation and Upgrade Guide.

**QUARTZDESK JVM AGENT**

This component is required for the QuartzDesk Standard and Enterprise editions only. The QuartzDesk JVM Agent component works as a specialized JVM plugin that provides the following functionality:

* Discovery of all Quartz scheduler instances running on the JVM.
* Automatic registration of job listeners for all discovered Quartz scheduler instances.
* On-the-fly instrumentation of the Quartz scheduler API's Java bytecode in order to add new methods to the Quartz scheduler MBean API. These injected methods are invoked by the QuartzDesk Web Application and they expose functionality that is not available in the standard Quartz scheduler MBean API. For example, the agent injects methods to access extended, trigger-type specific attributes and methods to create and update triggers.
* Registration of agent-specific MBeans to access job and trigger execution history, statistics, execution notification rules, job chains etc.
* Execution notification rules framework for a real-time job and trigger monitoring.
* Job chaining engine for job orchestration purposes.
* Accessing messages enqueued by execution notification rules.
* Recording the job & trigger execution history including the interception of log messages that are produced by executing jobs.
* Providing real-time access to the intercepted log messages produced by currently executing jobs. These messages can be viewed in the Currently Executing Jobs tab in the QuartzDesk GUI for all currently executing jobs.
* Full-text indices and search features for jobs, triggers, execution history records, notification rules, job chains etc.
* Optional JMXMP and JMX/RMI connectors for the JVM's platform MBean server.

The installation of QuartzDesk JVM Agent is described in the QuartzDesk JVM Agent Installation and Upgrade Guide.

**QUARTZDESK PUBLIC API LIBRARY**

This component is required for the QuartzDesk Standard and Enterprise editions only. The QuartzDesk Public API Library works primarily as an interface between managed Quartz scheduler instances and the QuartzDesk JVM Agent. The library contains:

* Quartz job listeners for all supported Quartz scheduler versions.
* Specialized logging Appenders and Handlers for all popular Java logging frameworks (log4j, log4j2, logback, jul). These Appenders and Handlers can be configured to intercept log messages produced by executed jobs. These messages are persisted in the execution history and can be viewed in the QuartzDesk GUI. They can be parsed and analyzed by custom (JavaScript-coded) execution notification rules.
* SOAP client APIs for all JAX-WS web-services exposed by the QuartzDesk web application.

The QuartzDesk Public API Library is distributed as a single JAR file and Quartz scheduler enabled applications only need to include this library on their runtime classpath. For web applications this typically involves copying the library to the web application's WEB-INF\lib folder. There are no other installation steps required to use this library, just drop it on the classpath and that is it.

Please note that the QuartzDesk Public API Library is available in the Maven Central repository so that developers can easily add it as a runtime dependency in their Maven-based projects.

**Work Flow Diagram**