QIE Install Guide

Installing and Configuring
the Qvera Interface Engine

Contents

Document Confidentiality Statement

The information in this document is confidential and for the use of the Qvera Interface Engine licensed customers only. It should not be disclosed to any other person or entity. It may not be reproduced in whole, or in part, nor may any of the information contained herein be disclosed without the prior written consent of Qvera.

Any form of reproduction, dissemination, copying, disclosure, modification, distribution and or publication of this material is strictly prohibited.

Technical Support

For Technical support visit http://www.qvera.com/support
Contact Qvera Support via:
Email at support@qvera.com
Call support at (801) 335-5101 option 2

Tell us What You Think

We value your input. If you have any suggestions, comments, or questions about this document please contact us by email at support@qvera.com. Thank you.

General Contact Information

Qvera
265 N Main, Suite D140
Kaysville, UT 84037

Ph: 801.335.5101
Email: info@qvera.com

System Requirements

Hardware Requirements

Component	Small Deployment *< 5,000 message per day* *< 50 CCD/CDA's per day*	Medium Deployment *< 250K messages per day* *< 1,000 CCD/CDA's per day*	Large Deployment *< 5M message per day* *< 500K CCD/CDA's per day*	Enterprise *> 5M messages per day* *> 500K CCD/CDA's per day*
Dedicated Server	Not required	Recommended	Recommended	Required
Processors	2-4 cores*	4-6 cores	6-12 cores	12+ cores
	QIE is a multi-threaded enterprise application which can take full advantage of all available CPU cores **64-bit JVM is recommended which requires a minimum of 2 cores, but QIE will work on a single core and 32-bit JVM*			
System RAM	2 – 4 GB	4 – 8 GB	8 – 16 GB	16+ GB
QIE RAM	1 – 2 GB*	2 – 4 GB	4 – 8 GB	8+ GB
	** 32-bit JVM is limited to 1.2 GB of RAM*			
Available Hard Disk Space	20 – 50 GB	50 – 100 GB	100 – 500 GB	500+ GB
	Hard disk requirements will vary depending on message volume and retention policy, recommendation based on 14-day retention			
Virtualization	Supported	Supported	Supported	Supported
	QIE has been tested in VMware's ESX Server and Workstation environments on both Windows and Linux guest systems			

Software Requirements

Component	Small Deployment *< 5,000 message per day* *< 50 CCD/CDA's per day*	Medium Deployment *< 250K messages per day* *< 1,000 CCD/CDA's per day*	Large Deployment *< 5M message per day* *< 500K CCD/CDA's per day*	Enterprise *> 5M messages per day* *> 500K CCD/CDA's per day*
Database Managment System	MariaDB v5 – 10 MySQL v5.x MSSQL 2003 – 2019	MariaDB v5 – 10 MySQL v5.x MSSQL 2003 – 2019	MariaDB v5 – 10 MySQL v5.x MSSQL 2003 – 2019	MariaDB v5 – 10 MySQL v5.x MSSQL 2003 – 2019
	** H2 is not recommended for production environments. It should only be used in test or demo environments.* ** High Availability requires In-Memory tables which are available in MySQL, MariaDB, or MSSQL 2016+*			
Operating System *(All deployments)*	All Windows Platforms Currently Supported by Microsoft/ Ubuntu 9.04 or newer / Unix *QIE supports both 32-bit and 64-bit operating systems*			
Java *(All deployments)*	64-bit (32-bit supported)	64-bit	64-bit	64-bit
	Java Runtime Environment (JRE) version 8			
Client Browser *(All deployments)*	Google Chrome 12 or newer / Mozilla Firefox 6 or newer / Safari 6 or newer *Client browser is only used to access the QIE console login and configure the engine from any workstation.*			

QIE Express Install Instructions

If you are comfortable with the Java Runtime Environment and software installations in general, you can follow the express install steps below. Otherwise, more detailed information is contained in the remainder of this install guide for installing and configuring the QIE environment.

Install Java

Download and install the Java runtime environment (JRE) version 8. Accept all defaults during the installation. The JRE is freely available and can be downloaded from http://www.java.com.

Install QIE

Run the QIE Windows installer to install QIE to the local machine. When the QIE installer completes, click 'Start' in the QIE Service Manager dialog to launch the QIE service.

NOTE: Contact Qvera support for information on downloading the latest QIE installer.

Install a Compatible Browser

Install a compatible Browser. Refer to the Software Requirements section above for a list of compatible browsers.

Launch QIE and Activate your License

If QIE was installed on the local machine, the console can be accessed by navigating to http://localhost using a supported browser. Otherwise, if QIE was installed on a remote server, the console can be accessed by navigating to http://{hostname or ip}.

NOTE: If the Jetty port was changed (in the Java tab of the QIE Service Manager dialog) from port 80 to some other port (e.g. port 8080), the port number must be appended to the URL in order to connect to the QIE console (e.g. http://localhost:8080).

To log in, use the default username ('admin') and password ('admin'). After logging in, you will be prompted to change the default 'admin' password.

To activate a license, select '**Help -> About'** from the main menu. On the 'License Keys' tab select **'insert'**. Enter a valid QIE customer ID and password and click **'Get License'**. Click **'Save'** after it populates the license details. Click **'OK'** to exit the about dialog.

NOTE: A valid QIE customer ID and password can be obtained by contacting Qvera

NOTE: For applying multiple licenses to a single install of QIE please refer to the QIE Reference Manual under the section titled "Managing Licenses and License Groups"

Installing QIE with a Proxy Server

If QIE is being installed in an environment which requires that internet traffic be routed through a proxy server, QIE will need to be configured with the appropriate proxy host and port settings in order to gain access to external resources, including the ability to communicate with Qvera's Licensing Server. To configure the proxy settings:

1. Login to QIE and navigate to the **System Configuration** page.
2. Select the Use Proxy checkbox in the Http/Https Proxy Settings.
3. Enter the Proxy Host and Port information.
4. If you have specific hosts that should bypass the proxy, they can be entered in the Non-Proxy Host field. This is a list of patterns separated by '|'. The patterns may start or end with a '*' for wildcards. Any host matching one of these patterns will be reached through a direct connection instead of through a proxy.
5. When prompted, answer "Yes" to the QIE Service Restart Required prompt. This will restart the QIE service and enable the proxy settings.

Optimizing Performance for Large XML Messages

When doing extensive processing of XML files, you may want to use the alternate XML parsing engine to improve performance. The default DOM implementation that comes with Java is slower and uses more memory. To use the fast xml parsing engine:

1. Download the **vtd-xml-qvera-2.11.3.jar** from http://www.qvera.com/files/vtd-xml-qvera-2.11.3.jar

Note: Some browsers may warn you that jar files can be harmful and will ask you to confirm that you want to keep the jar file. Confirm to continue the download.

Note: This JAR file is freely distributed by SourceForge.net. Qvera has modified the SourceForge.net VTD JAR file to handle XPath expressions consistent with the default JRE DOM implementation. As per the terms of the GPL license under which this JAR file is distributed, the vtd-xml-qvera-2.11.3.jar source code is freely available upon request.

2. Copy the **vtd-xml-qvera-2.11.3.jar** to the ‘lib’ directory located in the QIE Home directory specified in the ***Java Options*** field of the ***Java*** tab in the QIE Service Manager.

3. Login to QIE and navigate to the **System Configuration** page. Select the **Manage External Libraries** button and make sure that the **vtd-xml-qvera-2.11.3.jar** file is checked.

4. Click **Update** to apply any changes and close the Manage External Libraries dialog.

5. When prompted, answer “Yes” to the QIE Service Restart Required prompt. This will restart the QIE service and enable the fast xml parsing engine for use.

Installing the Java Runtime Environment

QIE is a Java application which runs inside a Java Runtime Environment (JRE). For Windows based installations, the JRE is freely available and can be downloaded from www.java.com.

Linux based installations typically use the OpenJDK, although Oracle’s Java environment can be manually downloaded and installed. QIE is supported in Linux environments using both OpenJDK as well as Oracle’s Java environment.

Installing QIE on Windows

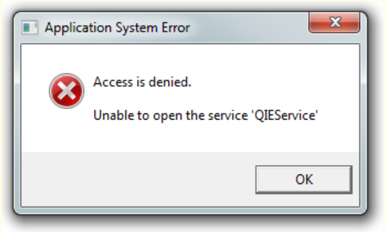
Run the QIE Windows installer to install QIE to the local machine or upgrade a current version of QIE already installed. When the QIE installer completes, the QIE Service Manager dialog will be displayed.

NOTE: Contact Qvera support for information on downloading the latest QIE installer.

Configuring the QIE Service Manager

Under Windows, QIE is configured to run as a local system service. The QIE Service Manager tool is used to configure the QIE service.

NOTE: Under Windows 7/10, If presented with an “Access is denied” dialog when launching the QIE Service Manager the shortcut will need to be configured to “Run as an administrator”. This can be done by right-clicking on the QIE Service Manager shortcut, selecting the “Compatibility” tab, and selecting the “Run this program as an administrator” option under the “Privilege Level” section.



Access is denied.

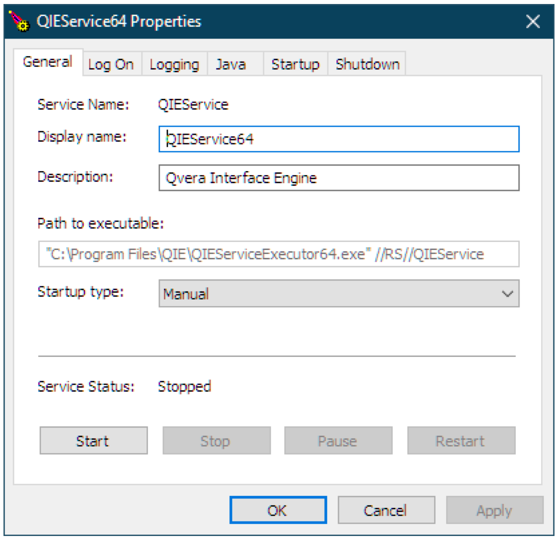
Unable to open the service 'QIEService'

General Tab

From the General tab a user can configure the following:

Display name and Description
These fields are used to configure the name and description associated with the QIE Service which are displayed in the Windows Services dialog.

Startup type
The startup type is used to configure how the QIE Service is started by the Windows operating system.



Service Name: QIEService

Display name: QIEService64

Description: Qvera Interface Engine

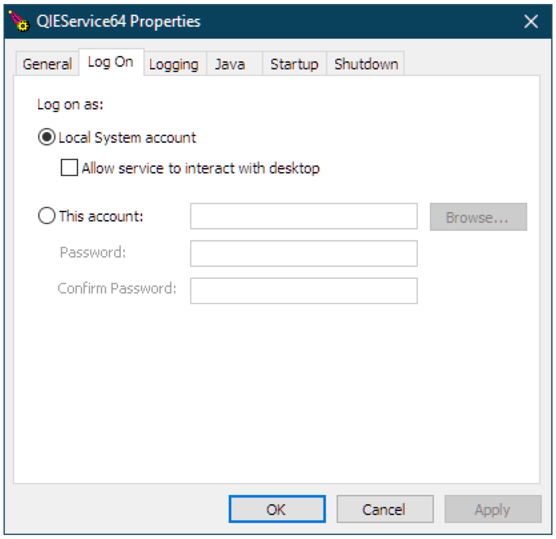
Path to executable:
"C:\Program Files\QIE\QIEServiceExecutor64.exe" //RS//QIEService

Startup type: Manual

Service Status: Stopped

Log On Tab

When a service is started under Windows, it is assigned privileges to local and remote resources based on the account selected under the "Log on as" section. If access to remote network folders is not required, then running QIE under the local system account is sufficient. Otherwise, QIE must be configured to run under a named account with appropriate privileges to the required network folders. ***This must be done using the "Windows Services" dialog and not from the QIE Service Manager.***



Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account:

Password:

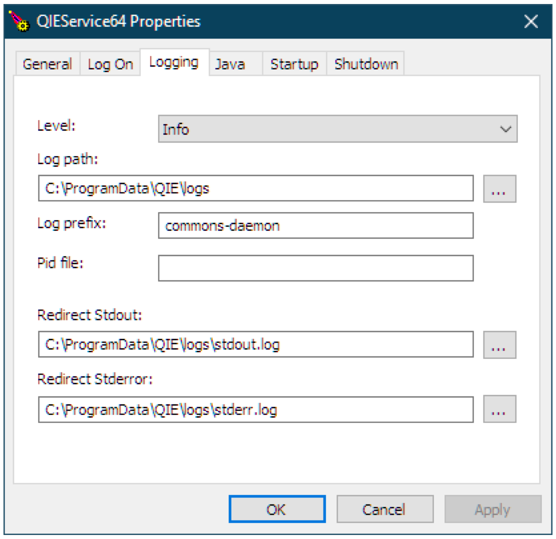
Confirm Password:

CAUTION: Never change QIE Service log on information from the QIE Service Manager dialog. *Log on configuration changes made using the QIE Service Manager dialog are not always recorded properly with the associated windows service manager. When this happens, the QIE Service does not start properly. For this reason, it is recommended that the QIE Service Log On information always be configured using the Windows Services dialog instead.*

Logging Tab

The Logging tab contains information about certain QIE Service related log files. In most cases, these setting should not need to be modified, but system administrators should take note of where the log files are located and what each log file contains.

QIE runs in a java servlet container wrapped as a service. Jetty is the Java servlet container which is used by QIE. When the QIE Service is launched, it starts the QIE Launcher. The Launcher starts the jetty instance inside the JVM. The QIE application is then loaded by that Jetty instance.



Level: Info

Log path: C:\ProgramData\QIE\logs

Log prefix: commons-daemon

Pid file:

Redirect Stdout: C:\ProgramData\QIE\logs\stdout.log

Redirect Stderror: C:\ProgramData\QIE\logs\stderr.log

commons-daemon.YYYY-MM-DD.log

The commons-daemon.YYYY-MM-DD.log records service start and stop events associated with running QIE as a service. This log file will contain information regarding any error(s) that may have prevented the service from starting properly.

stdout.log and stderr.log

These files are used by QIE Launcher and may contain errors during startup.

qieLauncher.log.x

This file is found in the log path. This is the main log file for the QIE Launcher that starts QIE application. This log file captures events and any unhandled exceptions that may occur during launching and monitoring of the QIE application.

qie.log

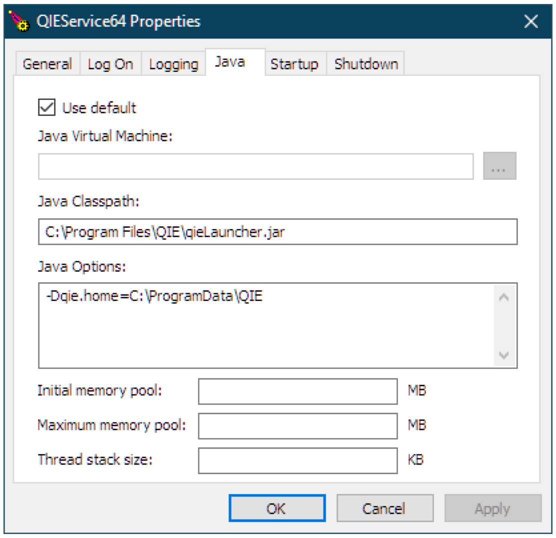
This file is found in the log path. This is the main QIE application log. This log file captures events and any unhandled exceptions that may occur in the QIE application.

Java Tab

From the Java tab a user can configure options associated with the JRE.

Java Virtual Machine

This option sets the path to the JRE or Java Virtual Machine. If multiple versions of the JRE have been installed, this option may need to be modified to point to a compatible JRE. Otherwise, this option should not need to be modified.



☑ Use default

Java Virtual Machine:

Java Classpath: C:\Program Files\QIE\qieLauncher.jar

Java Options: -Dqie.home=C:\ProgramData\QIE

Initial memory pool: MB

Maximum memory pool: MB

Thread stack size: KB

*Note: If you have chosen to install the **JRE version 7** in a windows environment then you will need to put the Java bin directory path into the "Path" Windows Environment System Variable. The bin directory path is usually "C:\Program Files (x86)\Java\jre7\bin". For more information on changing the "Path" system environment variable visit: http://www.java.com/en/download/help/path.xml*

*Note: If you are using **JRE version 7** and have trouble with certificates, comment out (by adding a pound symbol) the following line in the jre\lib\security\java.security file:*

#jdk.certpath.disabledAlgorithms=MD2

Java Classpath

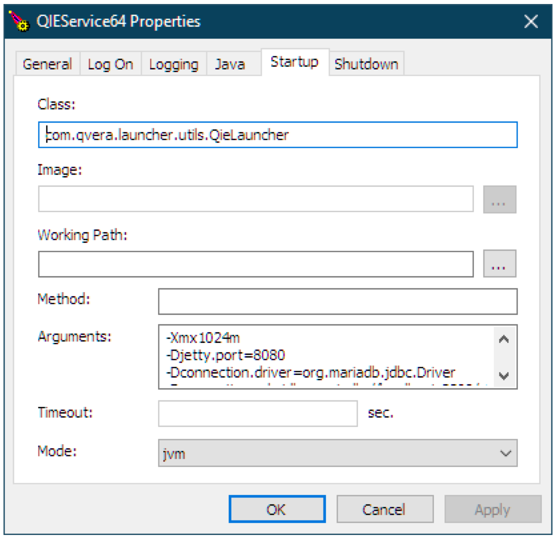
The Java classpath tells the JRE where program resources are found that should be associated with the QIE application. Normally, the only entry that needs to be included for this option is the qieLauncher.jar file.

Java Options

Options associated with the QIE application environment are configured using Java Options. The only required options on this tab are qie.home and java.security.policy. See descriptions below.

Startup Tab

From the Startup tab a user can configure Java options associated with the QIE application. These Java options are configured in the Arguments section. The rest of the settings on this tab should not change. In the event that these setting do get changed or deleted, they can be restored by re-running the QIE Installer.



Class: com.qvera.launcher.utils.QieLauncher

Arguments:

```
-Xmx1024m
-Djetty.port=8080
-Dconnection.driver=org.mariadb.jdbc.Driver
```

Mode: jvm

Arguments

This is a list of Java Options to associate with the QIE application environment.

Required Java Options

The following options are required by QIE and must be properly set in order for the QIE application to run correctly.

Option	Syntax & Default Value	Description
Java maximum memory pool	-Xmx512m	This option is used to configure the maximum amount of memory (in Megabytes) that the Jetty instance running the QIE application can use.
QIE Home	-Dqie.home=C:\ProgramData\QIE	This option is used to configure the home directory where the QIE application will store database files, log file, etc. This is the only one required to be specified on the Java Tab in Java Options. Optionally it can be specified in Arguments on the Startup tab. If configured on both tabs the values should match.
Java Security Policy	-Djava.security.policy= rmiSecurity.policy	The QIE application uses the Java RMI technology to enable the auto-update feature. In order for the RMI connection to work, this option must be present for both the QIE launcher application as well as the Jetty application.
Java Endorsed Directory	-Djava.endorsed=C:\Program Files (x86)\QIE\endorsed	**These options should not be changed**. QIE relies on some Java libraries which need to override default versions of the same libraries which are included with the JRE distribution. In order to do this, the JRE must know where to find the "override" libraries.
Jetty Port	-Djetty.port=80	The QIE console runs as a web application. The jetty port setting defines which port the QIE console will be published to. The default setting is to use port 80 which eliminates the need to include the port as part of the URL when connecting to QIE. If QIE is being installed on a system where port 80 is already in use, the jetty port should be changed to another available port (e.g. port 8080). When installing QIE on the GE Centricity DTS workstation this port number will need to be changed from 80 to another port such as port 8080.

Optional Java Options

The following options are optional and can be used if needed.

Option	Syntax & Default Value	Description
QIE Processor Thread Count	-Dqie.processorThreads=10	This option specifies the number of threads QIE should make available for processing messages. When setting this value, consideration should be given to 1) the number of physical processor cores available on the QIE server and 2) the number of channels that will be running concurrently across all QIE zones.
Proxy Host & Port	-Dhttp.proxyHost= -Dhttp.proxyPort= -Dhttp.nonProxyHosts= -Dhttps.proxyHost= -Dhttps.proxyPort= -Dhttps.nonProxyHosts=	If QIE is being installed in an environment which requires that internet traffic be routed through a proxy server, QIE will need to be configured with the appropriate proxy host and port settings in order to gain access to external resources, including the ability to communicate with Qvera's Licensing Server. If you need to communication with secure endpoints, be sure to configure the https options as well. If you have specific hosts that should bypass the proxy, they can be configured with the nonProxyHosts options. This is a list of patterns separated by '|'. The patterns may start or end with a '*' for wildcards. Any host matching one of these patterns will be reached through a direct connection instead of through a proxy.

TLS Security Related Java Options

When configuring QIE to use TLS security for communication with external systems, the following Java options may be required.

Option	Syntax	Description
Key Store	-Djavax.net.ssl.keyStore=???	The path and filename of the key store to be used by QIE for TLS communication
Key Store Password	-Djavax.net.ssl.keyStorePassword=???	The password associated with the key store file specified above
Key Store Type	-Djavax.net.ssl.keyStoreType=???	The type or format of the key store file being used (e.g. JKS or PKCS12)
Trust Store	-Djavax.net.ssl.trustStore=???	The path and filename of the trust store to be used by QIE for TLS communication
Trust Store Password	-Djavax.net.ssl.trustStorePassword=??	The password associated with the trust store file specified above
HTTPS Protocols	-Dhttps.protocols=TLSv1	The HTTPS protocols to be used by QIE for TLS communication
HTTPS Cipher Suites	-Dhttps.cipherSuites= TLS_RSA_WITH_AES_128_CBC_SHA	The cipher suites to be used by QIE for TLS communication

Database-Related Java Options

When configuring QIE to run against an alternate database management system the following Java options may be required.

Option	Syntax	Description
Connection Username	-Dconnection.username=???	The username to be used by QIE when connecting to the alternate DBMS system
Connection Password	-Dconnection.password=???	The password to be used by QIE when connecting to the alternate DBMS system
JDBC Driver	-Dconnection.driver=???	The JDBC driver class name associated with the selected DBMS system
Connection URL	-Dconnection.url=???	The JDBC connection URL associated with the selected DBMS system
Hibernate Dialect	-Dhibernate.dialect=???	The class name associated with the Hibernate SQL dialect to be used with the selected DBMS system
Validation Query *(optional)*	-Dconnection.validationQuery=???	The validation query is used to "validate" a connection before it is used to perform queries or updates against the database. The default validation query of "SELECT 1" works for every DMBS.

Shutdown Tab

The Shutdown tab of the QIE Service Manager dialog contains QIE specific configuration settings which should not be changed or deleted. In the event that these setting do get changed or deleted, they can be restored by re-running the QIE Installer.

Installing QIE on Linux/Mac

The QIE war zip file contains everything required to install QIE on a Linux or Mac OS server.

NOTE: Contact Qvera support for information on downloading the latest QIE war zip.

Linux Installation (Using systemd on Ubuntu 16.04 or later)

1. Create the required directory:

   ```
   sudo mkdir /java/qie
   ```

2. Extract the zip file to the new directory(/java/qie).

3. Copy '/java/qie/linux_install/qie.systemd.service.dist' to '/etc/systemd/system/qie.service'.

   ```
   sudo cp java/qie/linux_install/qie.systemd.service.dist
   /etc/systemd/system/qie.service
   ```

4. Grant permissions(rwxrwxrw):

   ```
   sudo chmod 777 /etc/systemd/system/qie.service
   ```

5. Create a symbolic link. This makes it easier to find the service definition file if you need to configure QIE startup options.

   ```
   sudo ln -s /etc/systemd/system/qie.service /java/qie/qie.service
   ```

6. Register the QIE service:

   ```
   sudo systemctl daemon-reload
   ```

7. Enable the QIE service to autostart on boot:

   ```
   sudo systemctl enable qie
   ```

   OPTIONAL Use this command to disable autostart:

   ```
   sudo systemctl disable qie
   ```

8. Create the log folder:

   ```
   sudo mkdir /java/qie/logs/
   ```

Starting and Stopping QIE service from a Linux Command Prompt

1. Start Command:

   ```
   sudo systemctl start qie
   ```

2. Stop Command:

   ```
   sudo systemctl stop qie
   ```

Linux Installation (Using upstart on Ubuntu 14.04 or earlier)

1. Create the required directory:

   ```
   sudo mkdir /java/qie
   ```

2. Extract the zip file to the new directory (/java/qie).

3. Copy '/java/qie/linux_install/start.upstart.sh.dist' to '/java/qie/start.sh':

   ```
   sudo cp /java/qie/linux_i8nstall/start.upstart.sh.dist /java/qie/start.sh
   ```

4. Grant permissions (rwxrwxrw):

   ```
   sudo chmod 755 /java/qie/start.sh
   ```

5. Copy '/java/qie/linux_install/qie.upstart.conf' to '/etc/init/qie.conf':

```
sudo cp /java/qie/linux_install/qie.upstart.conf /etc/init/qie.conf
```

6. Ubuntu 8.04 ONLY. This version of Ubuntu uses an older version of Upstart. It requires the file to be in the etc/event.d directory:

```
sudo mv /etc/init/qie.conf /etc/event.d/qie
```

Starting and Stopping QIE service from a Linux command prompt:

1. Start Command

```
sudo start qie
```

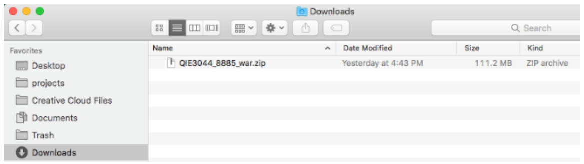
2. Stop Command

```
sudo stop qie
```

Mac OS Installation

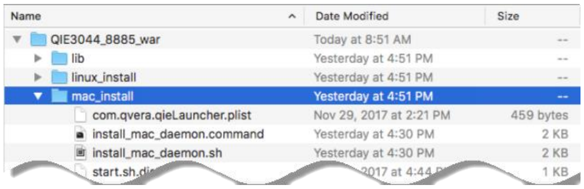
The name of the zip file contains the QIE version and build number (QIEVVVV_BBBB_war.zip). For example, if the zip file is QIE3044_8884_war.zip, the version is 3044 and the build number is 8884.

1. Use Finder to navigate to the directory where the zip file was downloaded and double-click the zip file to extract.



Name	Date Modified	Size	Kind
QIE3044_8885_war.zip	Yesterday at 4:43 PM	111.2 MB	ZIP archive

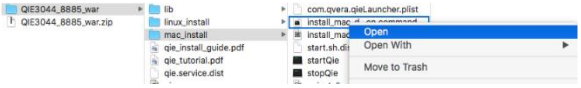
2. Navigate to the mac_install sub-directory. Hold down the <Ctrl> key and click on 'install_mac_daemon.command'.



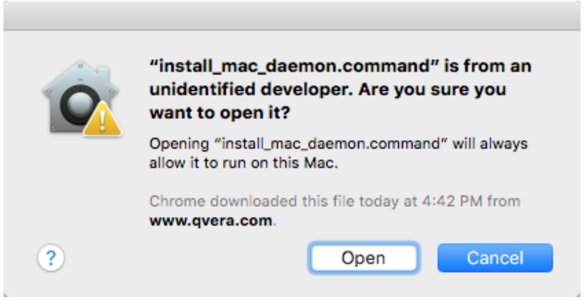
Name	Date Modified	Size
QIE3044_8885_war	Today at 8:51 AM	--
lib	Yesterday at 4:51 PM	--
linux_install	Yesterday at 4:51 PM	--
mac_install	Yesterday at 4:51 PM	--
com.qvera.qieLauncher.plist	Nov 29, 2017 at 2:21 PM	459 bytes
install_mac_daemon.command	Yesterday at 4:30 PM	2 KB
install_mac_daemon.sh	Yesterday at 4:30 PM	2 KB
start.sh.di	2017 at 4:44	1 KB

NOTE: If you Do Not hold down the <CTRL> key while clicking on the file it won't allow you to launch it.

3. This will display a list of options, select Open.



4. Click Open again in the dialog box.

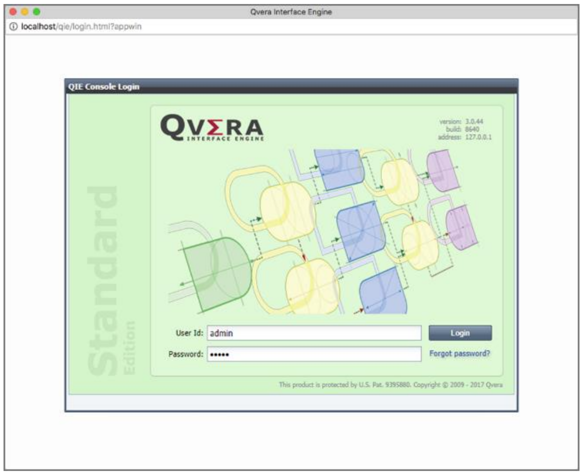


"install_mac_daemon.command" is from an unidentified developer. Are you sure you want to open it?

Opening "install_mac_daemon.command" will always allow it to run on this Mac.

Chrome downloaded this file today at 4:42 PM from **www.qvera.com**.

5. This will open the Terminal app and install QIE. It will request your Mac user password to continue. When the install is complete, the browser will open and populate the user name and password with the default username/password of admin/admin. Click login or press <Enter> to log into QIE.



Qvera Interface Engine

localhost/qie/login.html?appwin

QIE Console Login

version: 3.0.44
build: 8640
address: 127.0.0.1

User Id: admin

Password: •••••

This product is protected by U.S. Pat. 9395880. Copyright © 2009 - 2017 Qvera

6. To access QIE after the installation, open the browser and enter localhost in the URL.

Starting and Stopping the QIE Service from the Mac Terminal Application

The QIE service must be started for access to the application. The installation process automatically starts the service. Use the commands below to manually start or stop the QIE service.

1. Start Command

```
startQie
```

2. Stop Command

```
stopQie
```

Configuring QIE with an Alternate Database Management System

QIE uses a standard SQL hierarchal database to store system configuration, channel configuration, message data and audit logging information. By default, QIE uses an in-memory database called H2. H2 databases are very fast and efficient at processing information as they run in memory with the native application. Because H2 databases are persisted to disk using a single file for the entire database, it is recommended that large and enterprise QIE deployment environments be configured to run against a MariaDB, MySQL, or MSSQL database management system.

Configuring QIE with Microsoft SQL Server

Create a QIE Database Schema

Using Microsoft SQL Server Management Studio or a similar tool, connect to the MSSQL database server where the QIE database will reside and create a database schema called 'qie' or whichever name you wish to use for the QIE related database.

Create a Database User with DB Owner Privileges to the QIE Database Schema

Create a database user to be used by QIE for connecting to the QIE database schema created above. This user must be configured with full privileges to the QIE database schema as the QIE application handles the creation and management of all database tables, indexes, etc. Ensure that the default database associated with the user is set to the QIE database schema create above. When you create the new user, uncheck the "User must change password at next login". If this was checked, login to SQL Server with the new user and change the password before using it with QIE.

Note: The database user used by QIE must be an actual SQL Server user (not an active directory user) as QIE cannot be configured to connect to SQL Server using integrated authentication.

Setup In-Memory Tables (High Availability Only)

In-Memory tables are required for High Availability in QIE. For Microsoft SQL Server, In-Memory tables were introduced in 2016. Versions of MSSQL prior to 2016 are not supported with QIE High Availability. Here is a script to enable In-Memory table MSSQL:

```
-- High Availability support
ALTER DATABASE CURRENT SET AUTO_CLOSE OFF
GO

-- 1. validate that In-Memory OLTP is supported
IF SERVERPROPERTY(N'IsXTPSupported') = 0
BEGIN
PRINT N'Error: In-Memory OLTP is not supported for this server
edition or database pricing tier.';
END

IF DB_ID() < 5
BEGIN
PRINT N'Error: In-Memory OLTP is not supported in system databases.
Connect to a user database.';
END
ELSE
BEGIN
BEGIN TRY;

-- 2. add MEMORY_OPTIMIZED_DATA filegroup when not on Azure SQL DB
IF SERVERPROPERTY('EngineEdition') != 5
```

```
BEGIN
DECLARE @SQLDataFolder nvarchar(max) =
cast(SERVERPROPERTY('InstanceDefaultDataPath') as nvarchar(max))
DECLARE @MODName nvarchar(max) = DB_NAME() + N'_mod';
DECLARE @MemoryOptimizedFilegroupFolder nvarchar(max) =
@SQLDataFolder + @MODName;

DECLARE @SQL nvarchar(max) = N'';

-- Add a MEMORY_OPTIMIZED_DATA filegroup.
IF NOT EXISTS (SELECT 1 FROM sys.filegroups WHERE type = N'FX')
BEGIN
SET @SQL = N'
ALTER DATABASE CURRENT
ADD FILEGROUP ' + QUOTENAME(@MODName) + N' CONTAINS
MEMORY_OPTIMIZED_DATA;';
EXECUTE (@SQL);
END;

-- Add container in the filegroup in the default data folder.
IF NOT EXISTS (SELECT * FROM sys.database_files WHERE data_space_id
IN (SELECT data_space_id FROM sys.filegroups WHERE type = N'FX'))
BEGIN
SET @SQL = N'
ALTER DATABASE CURRENT
ADD FILE (name = N''' + @MODName + ''', filename = '''
+ @MemoryOptimizedFilegroupFolder + N''')
TO FILEGROUP ' + QUOTENAME(@MODName);
EXECUTE (@SQL);
END
END

-- 3. Set compat level to 130 if it is lower.
IF (SELECT compatibility_level FROM sys.databases WHERE
database_id=DB_ID()) < 130
ALTER DATABASE CURRENT SET COMPATIBILITY_LEVEL = 130

-- 4. Enable MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT for the database.
ALTER DATABASE CURRENT SET MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT =
ON;
END TRY
BEGIN CATCH
PRINT N'Error enabling In-Memory OLTP';
IF XACT_STATE() != 0
ROLLBACK;
THROW;
END CATCH;
END;
GO
```

For More details about In-Memory Tables for MSSQL, please see the following: https://www.qvera.com/kb/index.php/2265

Configure the QIE Service Manager to connect to Microsoft SQL Server

Add the following ***Arguments*** on the ***Startup*** tab of the QIE Service Manager:

```
-Dconnection.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
-Dconnection.url=jdbc:sqlserver://{server}:{port, default=1433};databaseName={qie}
-Dhibernate.dialect=com.qvera.qie.persistence.SQLServer2014UnicodeDialect
-Dconnection.username=???
-Dconnection.password=???
```

Note: For a Linux or MacOS install the ***Arguments*** *are located in the start.sh script located in the home directory. The above* ***Arguments*** *must be added to the script after*
–Djetty.port=80
The Connection URL must be modified to escape the ';' as it is a control character. For example:

```
-Dconnection.url=jdbc:sqlserver://{server}:{port, default=1433}\;databaseName={qie}
```

Note: Use the following list to select your correct SQL Server dialect-

Microsoft SQL Server 2019, use SQLServer2019UnicodeDialect
Microsoft SQL Server 2017, use SQLServer2017UnicodeDialect
Microsoft SQL Server 2016, use SQLServer2016UnicodeDialect
Microsoft SQL Server 2014, use SQLServer2014UnicodeDialect
Microsoft SQL Server 2012, use SQLServer2012UnicodeDialect
Microsoft SQL Server 2008, use SQLServer2008UnicodeDialect
Microsoft SQL Server 2005, use SQLServer2005UnicodeDialect

For SQL Server versions prior to 2005, use SQLServerDialectEx.

Note: Refer to the ***Database Related Java Options*** *section of this guide for a complete description of each of the above options.*

Configuring QIE with MySQL

Create a QIE Database Schema

Using the MySQL Workbench or a similar tool, connect to the MySQL database server where the QIE database will reside and create a database schema called 'qie' or whichever name you wish to use for the QIE related database.

Create a Database User with DB Owner Privileges to the QIE Database Schema

Create a database user to be used by QIE for connecting to the QIE database schema created above. This user must be configured with full privileges to the QIE database schema as the QIE application handles the creation and management of all database tables, indexes, etc. Ensure that the default database associated with the user is set to the QIE database schema created above.

Setup In-Memory Tables (High Availability Only)

In-Memory tables are required for High Availability in QIE. MySQL allows for In-Memory tables by default.

Download and Install MySQL JDBC Driver

Due to licensing restrictions in MySQL, QIE is not able to ship with the JDBC driver. For this reason, you will need to download and install the JDBC driver before using QIE with the MySQL database.

To download the latest version, browse to https://dev.mysql.com/downloads/connector/j/ and select the '*Platform Independent*' option from the operating system download. You can then select the '*Download*' button for the '*Zip Archive*'.

> *Note: You will receive a prompt to "Login" or "Sign Up", below this box you can just select "No thanks, just start my download".*

Once the download is complete, open the zip file and browse find the '*mysql-connector-java-{version}.jar*' file inside the first folder. Copy this file to the '*jdbcDriver*' directory found in your QIE home directory. The default location will be '*C:\ProgramData\QIE\jdbcDriver*'.

Finally, extract the .jar file (as if it was a zip file) to the same folder. Once the extraction is done, you should see two different folders in the '*jdbcDriver*' folder. '*com*' and '*META-INF*'.

> *Note: Windows Users, if you don't have a zip program installed that can extract the .jar file, just rename the .jar file to have an extension of .zip and then use windows explorer to unzip the file.*

Configure the QIE Service Manager to connect to MySQL

Add the following *Java Options* in the ***Arguments*** on the ***Startup*** tab of the QIE Service Manager:

```
-Dconnection.driver=com.mysql.cj.jdbc.Driver
-Dconnection.url=jdbc:mysql://{server}:{port, default=3306}/{database, default=qie}
-Dhibernate.dialect=com.qvera.qie.persistence.MySQL55UnicodeDialect
-Dconnection.username=???
-Dconnection.password=???
```

Note: Use the following list to select your correct MySQL Server dialect-

MySQL 5.5, use MySQL55UnicodeDialect
MySQL 5.6, use MySQL55UnicodeDialect
MySQL 5.7, use MySQL57UnicodeDialect
MySQL 8, use MySQL8UnicodeDialect

For MySQL versions prior to 5.5, use MySQLUnicodeDialect.

*Note: Refer to the **Database Related Java Options** section of this guide for a complete description of each of the above options.*

Configuring QIE with MariaDB

Create a QIE Database Schema

Using the MySQL Workbench or a similar tool, connect to the MariaDB database server where the QIE database will reside and create a database schema called 'qie' or whichever name you wish to use for the QIE related database.

Create a Database User with DB Owner Privileges to the QIE Database Schema

Create a database user to be used by QIE for connecting to the QIE database schema created above. This user must be configured with full privileges to the QIE database schema as the QIE application handles the creation and management of all database tables, indexes, etc. Ensure that the default database associated with the user is set to the QIE database schema created above.

Setup In-Memory Tables (High Availability Only)

In-Memory tables are required for High Availability in QIE. MariaDB allows for In-Memory tables by default.

Configure the QIE Service Manager to connect to MariaDB

Add the following *Java Options* in the ***Arguments*** on the ***Startup*** tab of the QIE Service Manager:

-Dconnection.driver=org.mariadb.jdbc.Driver
-Dconnection.url=jdbc:mariadb://{server}:{port, default=3306}/{database, default=qie}
-Dhibernate.dialect=com.qvera.qie.persistence.MySQLUnicodeDialect
-Dconnection.username=???
-Dconnection.password=???

Note: Use the following list to select your correct MariaDB Server dialect-

MariaDB 5.3, use MariaDB53UnicodeDialect
MariaDB 10, use MariaDB10UnicodeDialect
MariaDB 10.2, use MariaDB102UnicodeDialect
MariaDB 10.3, use MariaDB103UnicodeDialect
MariaDB 10.4, use MariaDB103UnicodeDialect
MariaDB 10.5, use MariaDB103UnicodeDialect

For MariaDB versions prior to 5.3, use MariaDB53UnicodeDialect.

Note: Refer to the ***Database Related Java Options*** *section of this guide for a complete description of each of the above options*

High Availability Java Options

Once In-Memory tables are setup, add the following *Java Option* in the ***Arguments*** on the ***Startup*** tab of the QIE Service Manager:

-Dqie.haEngine=EnterpriseHAServiceImpl

This Java option allows multiple QIE instances to run as a cluster. For additional High Availability configurations, see the QIE Disaster Recovery Guide.

If running QIE in an HA cluster, the environment can be configured with a primary/failover configuration by setting the '*qie.instancePriority*' java option on the primary node. Setting the priority to 1 on the primary node will cause all the other nodes to defer the work to the primary node. If the primary node goes down, then the other nodes will start working. Once the primary node comes back, it will automatically take over the processing of messages.

-Dqie.instancePriority=1

The instance priority on all node's defaults to 99, and they are all equal if not defined by a java option. A QIE cluster can be configured with multiple nodes each with a different priority, giving the cluster a primary/secondary/ect. hierarchy.