



QEngine Technical Paper

---

## Data Driven Testing in QEngine

## Table of Contents

<u>Abstract .....</u>	<u>3</u>
<u>Introduction .....</u>	<u>4</u>
<u>PreRequisites .....</u>	<u>4</u>
<u>Preface .....</u>	<u>4</u>
<u>Data Driven Testing .....</u>	<u>4</u>
<u>Data sources in QEngine .....</u>	<u>4</u>
<u>Example for Data Driven Testing .....</u>	<u>5</u>
<u>Creating Datasource in QEngine .....</u>	<u>5</u>
<u>Creating CSV Datasource .....</u>	<u>5</u>
<u>Creating Excel Datasource .....</u>	<u>5</u>
<u>Creating Database Datasource .....</u>	<u>6</u>
<u>Creating Variable Datasource .....</u>	<u>7</u>
<u>Using Datasource inside scripts .....</u>	<u>7</u>
<u>List of JDBC Compliant Databases .....</u>	<u>8</u>
<u>Conclusion .....</u>	<u>9</u>

## Abstract:

Automation goes unfulfilled if it does not use **Data-Driven Testing** techniques. In data-driven testing, we may be required to fetch data from various data sources such as Database / CSV / XLS, to use it to input to the forms or compare against data present in the web page. The data can be fetched straight away in the cases of CSV and XLS. In case of database, we have multiple database types from which the required data needs to be fetched.

In this paper, we will see how to use data source in QEngine to create data-driven testcases.

## Introduction:

The purpose of this paper is to provide insight on using data sources in QEngine.

## Pre-Requisites:

- Working knowledge of the web application to be tested.
- Basic working knowledge of ManageEngine QEngine Web Functional Testing Tool.
- Understanding of testing concepts.

## Preface:

QEngine is a test automation tool that can be used to automate web applications by recording the user interaction with the web application and playing it back as recorded.

QEngine script is a collection of step-by-step actions carried out by the user while interacting with the web application in a script format. Upon recording the user interaction with the web application, QEngine will generate the script in a human-readable format.

QEngine uses popular scripting language “**Jython**” as the scripting language for QEngine scripts. Jython is the Java Extension of **Python** language.

## What is Data-Driven Testing ?

Data-driven testing is a methodology used in Test automation where test scripts are executed and verified based on the data values stored in one or more central data sources or databases. These data sources can range from CSV files, Excel files, JDBC databases, etc., Data-driven testing is the establishment of several interacting test scripts together with their related data results in a framework used for the methodology. In this framework, variables are used for both input values and output verification values: navigation through the program, reading of the data sources, and logging of test status and information are all coded in the test script. Thus the logic executed in the script is also dependent on the data values.

## What are the various data sources supported in QEngine ?

QEngine has support for **the** following data sources :

1. CSV files - This can be used to fetch data from the csv files. You can get all the values or a certain set of values from the CSV files.
2. Excel files – This can be used to fetch data from the Excel files. You can get all the values or a certain set of values from the Excel files.
3. Databases – This can be used to fetch data from the database using database query. QEngine supports all JDBC compliant databases. By default, QEngine supports MySQL, Oracle and MSSQL.
4. Variables – This can be used to use the variable data declared globally and certain dynamic data such as current date, current hour, etc. The variables should be declared as Environmental variables.

### Consider the following scenario to use the data source in QEngine:

In an application you need to automate in such a way that 100 unique users needs to be added to the application database. You have an application page where you will enter the user details and on submitting the data, it will store the data into the application database.

### How to automate the above scenario ?

While recording, record the script in such a way that it takes the desired input in the form and then submit the data to the application. Then, use the “Data Configuration” in QEngine to fetch 100 users data from a CSV file and input it to the form, so that it can repeatedly submit the 100 user data into the database.

### How to create Data Source in QEngine ?

To do data-driven testing in QEngine, you should create a “Data Source” in the “Data Configuration” wizard.

### Follow the below steps to create a new CSV data source :

1. Click the “Data Configuration” link present above the script editor and open the “Data Configuration” wizard.
2. In the “Data Configuration” screen, click “Add” to add a new data source.
3. Provide name for the “Data Source”, say csvdata.
4. Select the data source type as “CSV” in the data source type combo.
5. The panel below the combo will show CSV-specific input details.
6. The “CSV File Name” combo will show the list of “CSV files” already available in the suite.
7. You can Import / Create CSV files in this wizard.
8. Import the CSV file that you already have to use it inside QEngine script.
9. Select the “Fetch All Values” option to fetch all the values present in the CSV file. Alternately, you can choose “Fetch Specific Values” to fetch only the specific number of rows / columns from the CSV file.
10. After configuring the above values, click “Save” to save the configuration.

### Follow the below steps to create new Excel data source:

1. Click the “Data Configuration” link present above the script editor and open the “Data Configuration” wizard.
2. In the “Data Configuration” screen, click “Add” to add a new data source.
3. Provide name for the “Data Source”, say exceldata.
4. Select the data source type as “Excel (XLS)” in the data source type combo.
5. The panel below the combo will show Excel-specific input details.

- 6.The “Excel File Name” combo will show the list of “Excel files” already available in the suite.
- 7.You can Import Excel files you already have into QEngine using this wizard.
- 8.Import the Excel file you already have to use it inside QEngine script.
- 9.Provide Sheet Name or Sheet Index, from which the required data needs to be fetched.
- 10.Select the “Fetch All Values” option to fetch all the values present in the Excel file. Alternately, you can choose “Fetch Specific Values” to fetch only the specific number of rows / columns from the Excel file.
- 11.After configuring the above values, click “Save” to save the configuration.

**Follow below steps to create database data source :**

- 1.Click the “Data Configuration” link present above the script editor and open the “Data Configuration” wizard.
- 2.In the “Data Configuration” screen, click “Add” to add a new data source.
- 3.Provide name for the “Data Source”, say dbdata.
- 4.Select the data source type as “Database” in the data source type combo.
- 5.The panel below will show the database-specific input details.
- 6.Choose the Database type from the combo box, say Oracle.
- 7.Click on the “Database Configuration” link to define various Database properties.
- 8.This will open the “Database Configuration” window.
- 9.In the database configuration window, select the database that needs to be configured, say Oracle.
- 10.The default properties of the database will be shown in the right side frame.
- 11.Check if the default properties such as “Driver Name”, “Connection URL”, “Username”, and “Password” are correct. The default properties for various databases are listed below.
- 12.Click the “Set DB Classpath” button to configure the classpath.
- 13.You can copy the appropriate jar file into the <QEngine Installation folder>/jars/ folder and then configure the classpath as ./jars/<jar file name>. Say for oracle, copy the “classes12.zip” file into <QEngine Installation folder>/jars/ and configure the classpath as ./jars/classes12.zip. Then click the “Apply” button to save the classpath.
- 14.Then configure the proper URL , Username & Password for the database.
- 15.Click on “Commit to List” to save the configuration for the database and click the “Apply” button.
- 16.This will populate the “URL” for the database in the “URL combo” in Data configuration screen.
- 17.Configure the “Query” using which the data needs to be fetched from the database.
- 18.Click the “Save” button to save the data source configurations.

## Follow the below steps to configure a Variable data source :

1. Click the "Data Configuration" link present above the script editor and open the "Data Configuration" wizard.
2. In the "Data Configuration" screen, click "Add" to add a new data source.
3. Provide name for the "Data Source", say variable\_data.
4. Select the data source type as "Variable" in the data source type combo.
5. This will list all the variables that are configured under Environmental Variables.
6. Select the appropriate variable from the "Variable List" combo and click on "Save" to save the configuration.

## How to use the data source inside QEngine scripts ?

After following the above steps and creating the data source of your required type, click the "Paste" button.

This will insert a block of code snippet into the script. The code snippet will look this,

```
initDataSet("csvdata")
result = getValuesFrom("csvdata")
#*****
#result stores the values fetched from the specified dataset.
#You can fetch values using a for loop as shown in the below sample.
#Remove the '#' in the sample, to use the code.
#for i in range(0,len(result)):
    #var1=str(result[i][0])
    #var2=str(result[i][1])
    #setText("yourname",var1,3)
    #setText("dept",var2,3)
#*****
```

In the above code snippet, the "#" symbol denotes commented lines. The values fetched from the data source will be stored in the "result" array variable. The array is a two-dimensional array. To use the data in the script, for loop should be used. Iterate each data from the result array and use it to fill the form details. In the above code snippet, the values inside the 'for' loop are fetched from a result array and stored in a local variable to use it with the setText function.

### Various JDBC-Compliant Database details:

In QEngine, you can fetch data from any JDBC-compliant database. The details of various JDBC-compliant databases can be found in the below table:

Database Type	Driver Name	JDBC URL	Jar file to use
Oracle Thin	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<HOST>:<PORT>:<SID>	Classes12.zip
Oracle OCI 8i	oracle.jdbc.driver.OracleDriver	jdbc:oracle:oci8:@<SID>	Classes12.zip
Oracle OCI 9i	oracle.jdbc.driver.OracleDriver	jdbc:oracle:oci:@<SID>	Classes12.zip
Microsoft SQL Server 2000 (Microsoft Driver)	com.microsoft.jdbc.sqlserver.SQLServerDriver	jdbc:microsoft:sqlserver://<HOST>:<PORT>[;DatabaseName=<DB>]	Msbases.jar,msutil.jar,mssqlserver.jar
Microsoft SQL Server 2005 (Microsoft Driver)	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<HOST>:<PORT>[;DatabaseName=<DB>]	Sqjjdbc.jar
MySQL (MM.MySQL Driver)	org.gjt.mm.mysql.Driver	jdbc:mysql://<HOST>:<PORT>/<DB>	mysql_connector.jar
Microsoft SQL Server (Sprinta Driver)	com.inet.tds.TdsDriver	jdbc:inetdae:<HOST>:<PORT>?database=<DB>	
Microsoft SQL Server (JTurbo Driver)	com.ashna.jturbo.driver.Driver	jdbc:JTurbo://<HOST>:<PORT>/<DB>	
Microsoft SQL Server	weblogic.jdbc.mssqlserver4.Driver	jdbc:weblogic:mssqlserver4:<DB>@<HOST>:<PORT>	
JDBC-ODBC Bridge	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:<DB>	
IBM DB2	COM.ibm.db2.jdbc.app.DB2Driver	jdbc:db2://<HOST>:<PORT>/<DB>	
PointBase Embedded Server	com.pointbase.jdbc.jdbcUniversalDriver	jdbc:pointbase://embedded[:<PORT>]/<DB>	
Cloudscape	COM.cloudscape.core.JDBCDriver	jdbc:cloudscape:<DB>	
Cloudscape RMI	RmiJdbc.RJDriver	jdbc:rmi://<HOST>:<PORT>/jdbc:cloudscape:<DB>	
Firebird (JCA/JDBC Driver)	org.firebirdsql.jdbc.FBDriver	jdbc:firebirdsql://<HOST>[:<PORT>]/<DB>	
IDS Server	ids.sql.IDSDriver	jdbc:ids://<HOST>:<PORT>/conn?dsn='<ODBC_DSN_NAME>'	
Informix Dynamic Server	com.informix.jdbc.IfxDriver	jdbc:informix-sqli://<HOST>:<PORT>/<DB>:INFORMIXSERVER=<SERVER_NAME>	
InstantDB (v3.13 and earlier)	jdbc.idbDriver	jdbc:idb:<DB>	

Database Type	Driver Name	JDBC URL	Jar file to use
InstantDB (v3.14 and later)	org.enhydra.instantdb.jdbc.idbDriver	jdbc:ldb:<DB>	
Interbase (InterClient Driver)	interbase.interclient.Driver	jdbc:interbase://<HOST>/<DB>	
Hypersonic SQL (v1.2 and earlier)	hsqldb.HDriver	jdbc:HypersonicSQL:<DB>	
Hypersonic SQL (v1.3 and later)	org.hsqldb.jdbcDriver	jdbc:HypersonicSQL:<DB>	
PostgreSQL (v6.5 and earlier)	postgresql.Driver	jdbc:postgresql://<HOST>:<PORT>/<DB>	
PostgreSQL (v7.0 and later)	org.postgresql.Driver	jdbc:postgresql://<HOST>:<PORT>/<DB>	
Sybase (jConnect 4.2 and earlier)	com.sybase.jdbc.SybDriver	jdbc:sybase:Tds:<HOST>:<PORT>	
Sybase (jConnect 5.2)	com.sybase.jdbc2.jdbc.SybDriver	jdbc:sybase:Tds:<HOST>:<PORT>	

### Conclusion:

This concludes the paper on creating data-driven test scripts with QEngine. We hope that you now have a better understanding of the features and capabilities of ManageEngine QEngine.

Visit our website [www.qengine.com](http://www.qengine.com) for more information.

For any clarification on this article, e-mail to [srinivasar@zohocorp.com](mailto:srinivasar@zohocorp.com)

ManageEngine Applications Manager ([www.appmanager.com](http://www.appmanager.com)) provides out of the box performance monitoring for J2EE Application Servers, Databases and Servers.

ManageEngine is the leader in low-cost enterprise IT management software. The ManageEngine suite offers enterprise IT management solutions including Network Management, HelpDesk & ITIL, Bandwidth Monitoring, Application Management, Desktop Management, Security Management, Password Management, Active Directory reporting, and a Managed Services platform. ManageEngine products are easy to install, setup and use and offer extensive support, consultation, and training. More than 30,000 organizations from different verticals, industries, and sizes use ManageEngine to take care of their IT management needs cost effectively. ManageEngine is a division of ZOHOO Corporation. For more information, please visit [www.manageengine.com](http://www.manageengine.com).