

## QMetry Generic Plug-in (Version 3.1)

**Download:** <http://downloads.qmetry.com/QMetryLaunchers31/QMetryLaunchers.zip>

This zip contains following files:

- **QMetry Launcher library: QMetryLaunchers.jar**

This library is responsible for fetching Test Case information from QMetry Test Automation Agent and launch Generic Test Scripts. Once Test Script execution is complete, this library will upload the Test Results to QMetry.

- **Sample Test Script in Python: PythonExampleTestCase.py**

This is a simple Python script, which decides Test Case status (pass/fail) based on command line arguments passed to it

- **Configuration File: QMetryConfig.properties**

This file contains configuration related to QMetry instance. These configuration settings will be used for communicating with QMetry Server for updating the Test Results.

- **QMetry Agent Configuration Template: QMetryAgent.properties-addons**

This file contains 2 additional configuration parameters for QMetry Test Automation Agent and Generic Framework integration

### System Requirements:

- Operating System: Windows XP, Windows Vista, Windows 2003 Server, Windows 7, \*Unix
- Java Runtime: Java version 1.6 or later
- Root/administrator privilege to install QMetry Test Automation Agent as Windows Service/Unix daemon

### How to integrate QMetry Agent with Generic Framework?

1. Login to QMetry Web Application
2. In **Test Case Management**, go to **Admin->Agent** tab
3. Create an Agent by clicking the **Add Agent** button
4. Download QMetry Agent (*QMetryAgent.zip*) as instructed
5. Download QMetry Agent Configuration file (*QMetryAgent.properties*) as instructed
6. Extract QMetryAgent.zip file on your test machine. We assume that you will extract it on C:\
7. Copy *QMetryAgent.properties* (downloaded in Step #5) to C:\QMetryAgent\bin directory
8. Download [QMetry Launcher](#)
9. Extract *QMetry QMetry Launcher* zip file on your test machine, where QMetry Agent is running. We assume that you will extract it on C:\

10. Open *QMetryAgent.properties* from *C:\QMetryAgent\bin* directory and add two lines from *QMetryAgent.properties-addons* file
11. *QMetryAgent.properties-addons* file contains paths assuming that *QMetryLaunchers.zip* and *QMetryAgent.zip* are extracted on *C:\*. If you have extracted the files at some different location, then please update these paths accordingly.
12. If your Generic Test Framework has dependencies on other JAR files, then please add them under *TESTSCRIPTWRAPPER* property
13. Open *QMetryConfig.properties* and update all 4 parameters accordingly
14. Install QMetry agent by running the *InstallQMetryAgent.bat* file from *C:\QMetryAgent\bin*

**Note:** You must have administrative privileges to run this file. If you do not have administrative privileges, then run *StartQMetryAgent.bat* from *C:\QMetryAgent\bin*

If you are configuring QMetry Agent on \*unix platform, then you need to run *QmetryAgent.sh* to configure the agent as Unix Daemon. You must have root privileges to start QMetry Agent as Unix Daemon.

15. If QMetry Agent is successfully configured and started, then you will see its status as “Active” under **Admin->Agent** tab in Test Case Management in QMetry application

The steps mentioned above are for one time configuration.

## How to execute Generic Test Cases from QMetry Application?

1. Login to QMetry application
2. In **Test Case Management**, go to **TestCases** module
3. Create a Test Case and specify your Generic Test Class under the **Test Script Name** field (for example “*C:\Python25\python.exe C:\QMetry-Launchers\PythonExampleTestCase.py PASS*” – without quotes)
4. In **Test Case Management**, go to **TestSuites** module
5. Create a Test Suite and link the Test Case created above with this Test Suite
6. In **Test Case Management**, go to the **Test Suites** module and select the **Execute TestSuite** tab
7. Select a **Platform** and link it with the Test Suite
8. Click on **Assign Agent** icon under **Automation** column
9. Select an Agent on which you want to execute your Generic Test Case
10. Click the **Run Now** button to run the Generic Test Case immediately, or else, select the Date and Time to schedule the execution on future date

## Variables in Test Script Name

Additionally you can use variables generated by QMetry Generic Launcher in to your Test Script. For example, suppose you want to pass QMetry Platform Name and QMetry Test Case Id to your test script “C:\QMetry-Launchers\PythonExampleTestCase.py” as command line arguments. To do so, while specifying below as a Test Script Name in QMetry Application

```
C:\QMetry-Launchers\PythonExampleTestCase.py PASS “{QMPlatform}” {QATestCaselD}
```

While launching above Test Script, launcher will replace {QMPlatform} and {QATestCaselD} with actual values so that you can access them in your script. Please note that, if the value contains spaces (for example Platform Name may have spaces), then enclose them in double quotes, as shown in above example.

## Additional Log Files

By default, QMetry Generic Launcher attaches output of your Test Script with the Test Case execution. If you want to attach additional files, then you can do so by specifying them under QMetryConfig.properties. You can specify up to 10 additional files to attach. You can also use variables generated by Generic Launcher in your log file name. For example, your Test Script generates log file with Test Case ID in its name (like TestCase\_247.log) and you want to attach this log file with Test Case execution. To attach additional log file, please specify them under QMetryConfig.properties as below.

```
attach.logfile.1=C:/QMetryLaunchers/TestCase_{QMTestCaselD}.log
attach.logfile.2=C:/QMetryLaunchers/EnvironmentDetails.log
...
...
...
attach.logfile.10=C:/QMetryLaunchers/Errors.log
```

## Below are list of variables which can be used

- {QMProject}
- {QMRelease}
- {QMBuild}
- {QMTestSuiteId}
- {QMTestSuiteName}
- {QMTestSuiteFolder}
- {QMPlatform}
- {QMPlatformId}
- {QMTestCaselD}
- {QMTestCaseName}
- {QMTestCaseFolder}