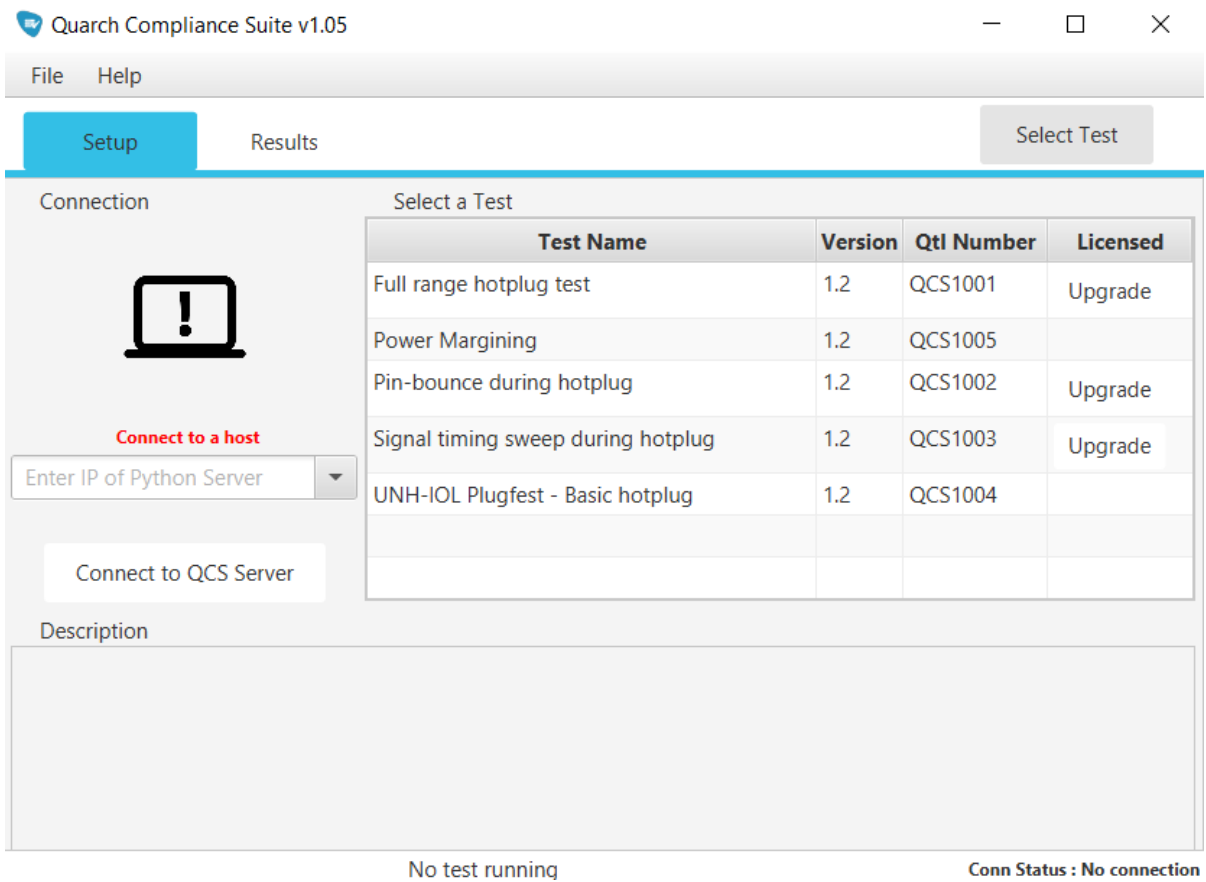


Quarch Technology Ltd

Quarch Compliance Suite

Quickstart



Quarch Compliance Suite v1.05

File Help

Setup Results Select Test

Connection

Select a Test

Test Name	Version	Qtl Number	Licensed
Full range hotplug test	1.2	QCS1001	Upgrade
Power Margining	1.2	QCS1005	
Pin-bounce during hotplug	1.2	QCS1002	Upgrade
Signal timing sweep during hotplug	1.2	QCS1003	Upgrade
UNH-IOL Plugfest - Basic hotplug	1.2	QCS1004	

Connect to a host

Enter IP of Python Server

Connect to QCS Server

Description

No test running

Conn Status : No connection

Change History

1.0	26/07/2019	Initial Release
1.2	11/12/2020	Added information on installer

QCS Version History

1.0	26/07/2019	Initial Release
1.2	30/09/2019	Introducing Power Tests
1.3	11/11/2019	General improvements
1.5	27/11/2020	Ground work for Performance Testing and general improvements

Contents

Change History.....	2
QCS Version History	3
Introduction	5
Server side.....	5
Client side	5
Requirements	6
Java	6
Python install.....	7
Required Hardware.....	8
Running QCS.....	9
Generating a Report.....	14
Example equipment configurations.....	15
Setup 1 – Hotplug Testing.....	15
Setup 2 – Power Testing.....	17
Getting help	18

Introduction

Quarch Compliance Suite is a simple solution for performing automated compliance testing on storage devices. This program consists of 2 parts; the Java client and the Python server.

The client and server can run on the same PC for local testing, though more often, the server will run on a remote test PC. This provides several benefits:

- Users can run tests on remote systems
- Logs can be fully captured, even if the system under test crashes
- High load activities, such as high-resolution power capture will not add processor load to the system, under test

Server side

The Server PC is the 'system under test', which hosts the storage device you are working on.

The QCS server app (python based) must be installed and running here. The server uses very little resources so will have minimal effect on the performance of the system under test.

Client side

The Client PC is the one you use to control and monitor the tests. This can be the same as the Server PC but is generally a separate system.

The client runs the Java based monitoring and control application, allowing you to select tests and view the results.

The client also hosts the license key which is required for the more complex test suites. License keys are hardware locked, so consider which PC you wish to license when you are buying keys.

Requirements

- Java 1.8 on the PC running the QCS client
 - When using the QCS installer, a separate Java install is not required
- Python 3.x on both the Client and Server PC
- Appropriate Quarch modules, connected to the network or direct to the Server PC
- The 'zeroconf' python module is required on the Server PC if you wish to use auto-discovery of QCS servers on your network. This is an optional feature but recommended to install
- The 'pandas' python module is required on the Server PC for some test suites and is recommended to install

Quarch Compliance Suite can run on Windows and Linux systems

Java

If you NOT using the installer version of QCS (1.05 and later) then you must install an appropriate version of Java 8 on the Client PC. If you are using the installer, an appropriate version of Java is bundled and will run automatically.

Java Install

Check that Java JRE 8 is installed on the machine you want to run Quarch Compliance Suite on. To check this is installed, open up a command prompt. (*Search 'command prompt' in the start menu – Or 'terminal' for Linux users*) Once in a command prompt, enter the following command:

java -version

If the returned version does not begin with "1.8." or the command is not recognized, you will need to install Java on this machine. You can find install instructions and files here:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

If you want to use another version of Java, it must include JavaFX, see here for tips:

<https://quarch.com/support/faqs/java/>

Python install

Check that Python 3 is installed on both the Client and Server PC. To check this, open a new command prompt (or a new terminal on Linux) and enter the following command:

```
> Python -version
```

If this command is not recognized or the output does not start with “Python 3.”, you will need to install a valid Python version on this pc. Check out the following link in order to find and download Python:

<https://www.python.org/downloads/>

Once a Python 3 version is installed on the machine, you will need to download the latest Quarchpy version. This can be downloaded through Pip. To do this, open a new command prompt (or terminal for Linux users) and type the following command:

```
> Pip install quarchpy
```

If you already have quarchpy installed, you should make sure it is the latest version:

```
> Pip install quarchpy --upgrade
```

If the ‘pip’ commands fail, you can invoke them directly via Python instead:

```
> python -m pip install quarchpy --upgrade
```

On the Server PC you should ideally also install the ‘optional’ Python modules which add additional functionality and are required for some tests:

```
> Pip install zeroconf
```

```
> Pip install pandas
```

Required Hardware

As well as the client/server install, you will require appropriate Quarch modules, depending on the tests you are performing. More details will be described in the test suite documentation.

Drive hot-plug tests

- Torridon Interface kit (QTL1260) OR Torridon Controller (QTL1461, QTL1079)
 - <https://quarch.com/products/torridon-interface-kit/>
 - <https://quarch.com/products/4-port-torridon-controller/>
 - <https://quarch.com/products/28-port-torridon-controller/>
- Quarch Breaker Module (Many are supported), examples include:
 - <https://quarch.com/products/gen4-pcie-u2-module/>
 - <https://quarch.com/products/gen4-pcie-u-3-module/>
 - <https://quarch.com/products/gen4-m-2-m-key-vertical-breaker-module/>
- The storage device under test
 - This must be connected with the Quarch breaker installed between it and the Server hardware

Power analysis tests

- Quarch power module: XLC, HD, or PAM
 - <https://quarch.com/products/xlc-programmable-power-module/>
 - <https://quarch.com/products/hd-programmable-power-module/>
 - <https://quarch.com/products/power-analysis-module/>
 - The power module must be available to the Client PC, either via LAN or USB
- An appropriate power fixture, given the power module and device you are testing. These are available for all standard storage interfaces.
- The storage device under test
 - This must be connected with the Quarch breaker installed between it and the Server hardware

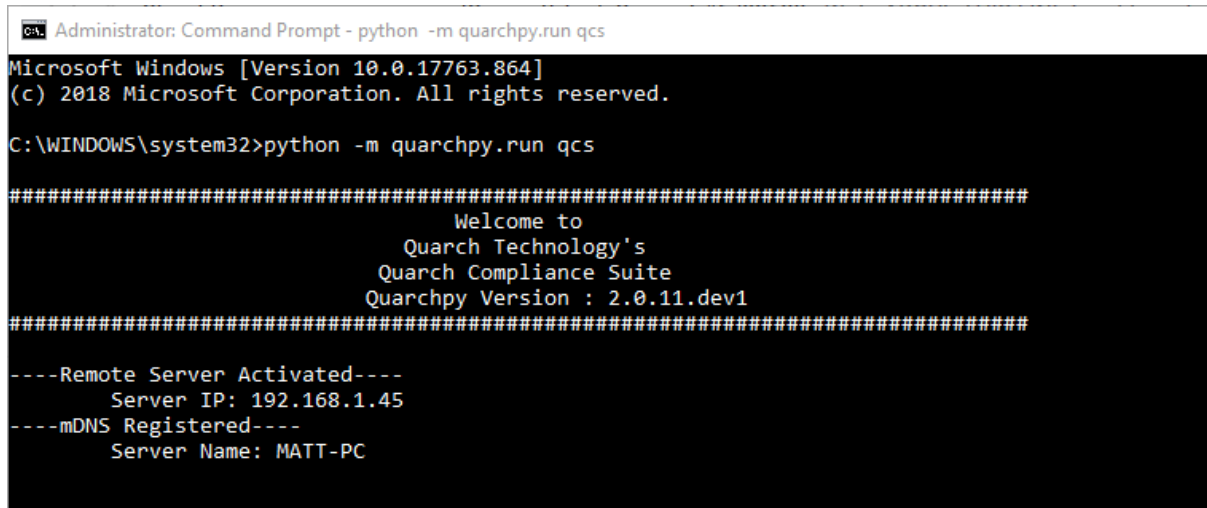
Running QCS

There are 3 steps to starting QCS. First, starting a Python server on the system that the drives are connected to. Following this, launching the Java GUI client and lastly connecting the Java client to the Python server.

The Python server needs to be launched from an **Administrator** Command Prompt. To start an Administrator command prompt in Windows, search for command prompt in the start menu, right click on the command prompt and hit run as administrator. If you are running this Python backend on a Linux PC, load up a new terminal prefix the command below with “sudo”.

The command to start the backend is:

Python -m quarchpy.run qcs

A screenshot of a Windows Administrator Command Prompt window. The title bar reads 'Administrator: Command Prompt - python -m quarchpy.run qcs'. The command prompt shows the following text:

```
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>python -m quarchpy.run qcs

#####
                        Welcome to
                        Quarch Technology's
                        Quarch Compliance Suite
                        Quarchpy Version : 2.0.11.dev1
#####

----Remote Server Activated----
      Server IP: 192.168.1.45
----mDNS Registered----
      Server Name: MATT-PC
```

If the server started correctly, the console will look something similar to the above image.

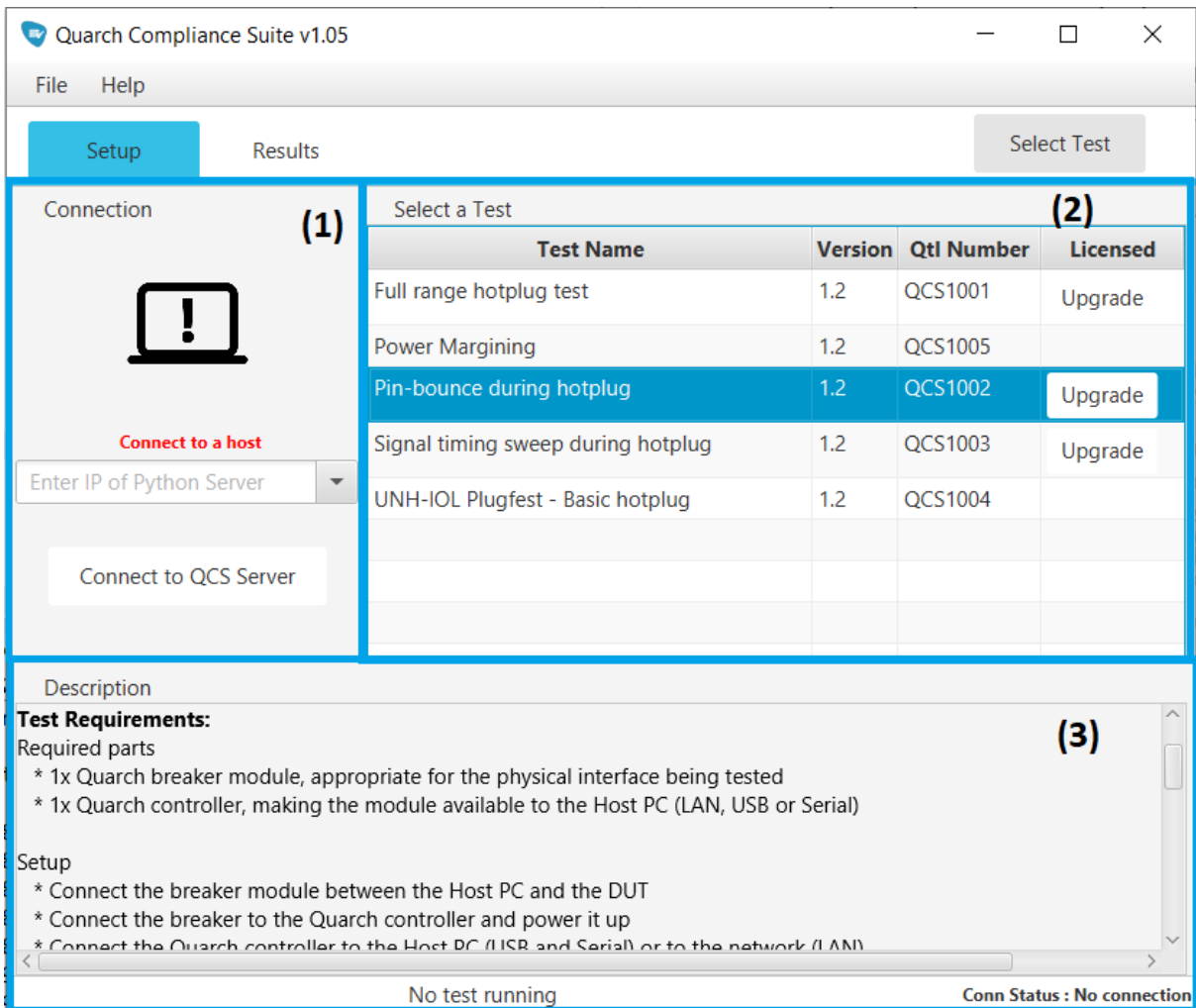
This program can utilize MDNs through the use of ZeroConf. In order for this to work correctly, the machine running this Python server must have the Python package: ZeroConf. To install this via Pip, open a new Command Prompt (or terminal for Linux users) and enter the following command:

> Pip install zeroconf

If you have installed QCS (windows) just open the application as normal. To start an executable version, just double-click on the QuarchComplianceSuite.

If you are a Linux user, you may have to open a new terminal, navigate to this directory and start the Java side using:

`java -jar QuarchComplianceSuite.jar`. It may take several seconds for QCS to launch as Java may also need to start in the background. When the program is started, the user will be presented with the setup page as seen below.



Test Name	Version	Qtl Number	Licensed
Full range hotplug test	1.2	QCS1001	Upgrade
Power Margining	1.2	QCS1005	
Pin-bounce during hotplug	1.2	QCS1002	Upgrade
Signal timing sweep during hotplug	1.2	QCS1003	Upgrade
UNH-IOL Plugfest - Basic hotplug	1.2	QCS1004	

Test Requirements:

Required parts

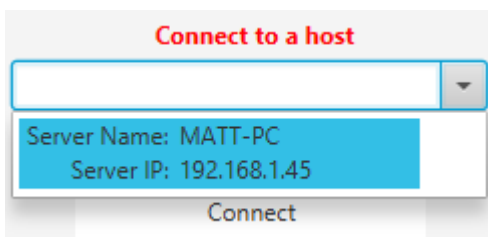
- * 1x Quarch breaker module, appropriate for the physical interface being tested
- * 1x Quarch controller, making the module available to the Host PC (LAN, USB or Serial)

Setup

- * Connect the breaker module between the Host PC and the DUT
- * Connect the breaker to the Quarch controller and power it up
- * Connect the Quarch controller to the Host PC (USB and Serial) or to the network (LAN)

No test running Conn Status : No connection

Once the Java GUI and Python Backend are both running, the last thing to do for the setup is to make a connection between them.



Server Name: MATT-PC
Server IP: 192.168.1.45

Connect

Section (1) of the above screenshot is used for connecting to the Python Backend. If you are using a remote server then enter its IP address and hit connect (1). If the remote server has ZeroConf installed, the program will attempt to utilize MDNS and if found, the remote server's

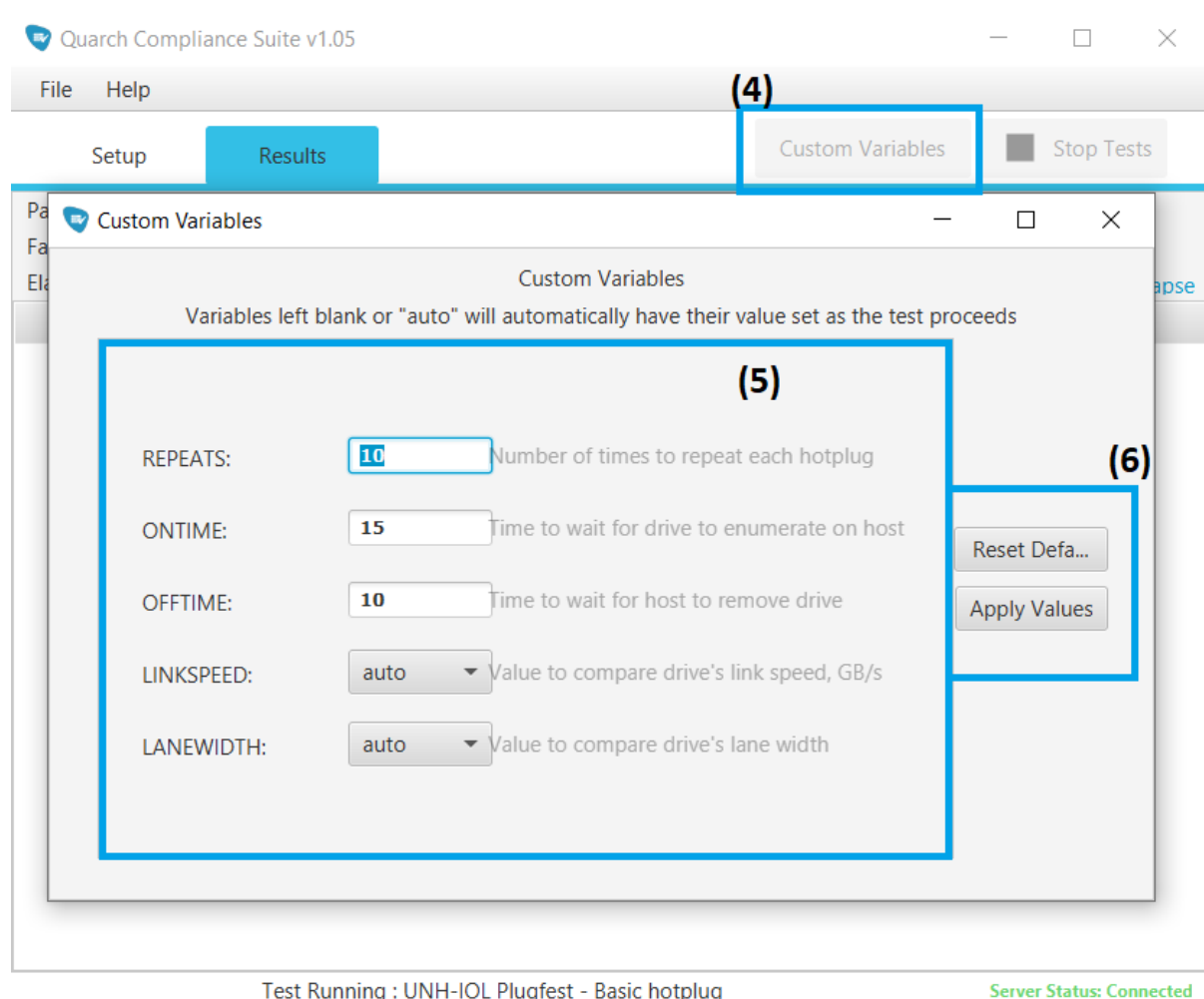
connection details will appear within the drop-down list surrounding the text field used to enter an IP Address.

By default, if no IP address is entered into the text field, the program will attempt to find a server being run on the local machine and connect, instead, to this. Upon a successful connection, the icon will change to the green connection sign showing a valid connection has been made.

Some computers may require you to open ports 9742 and 9921 for valid communication between the Java frontend and Python backend.

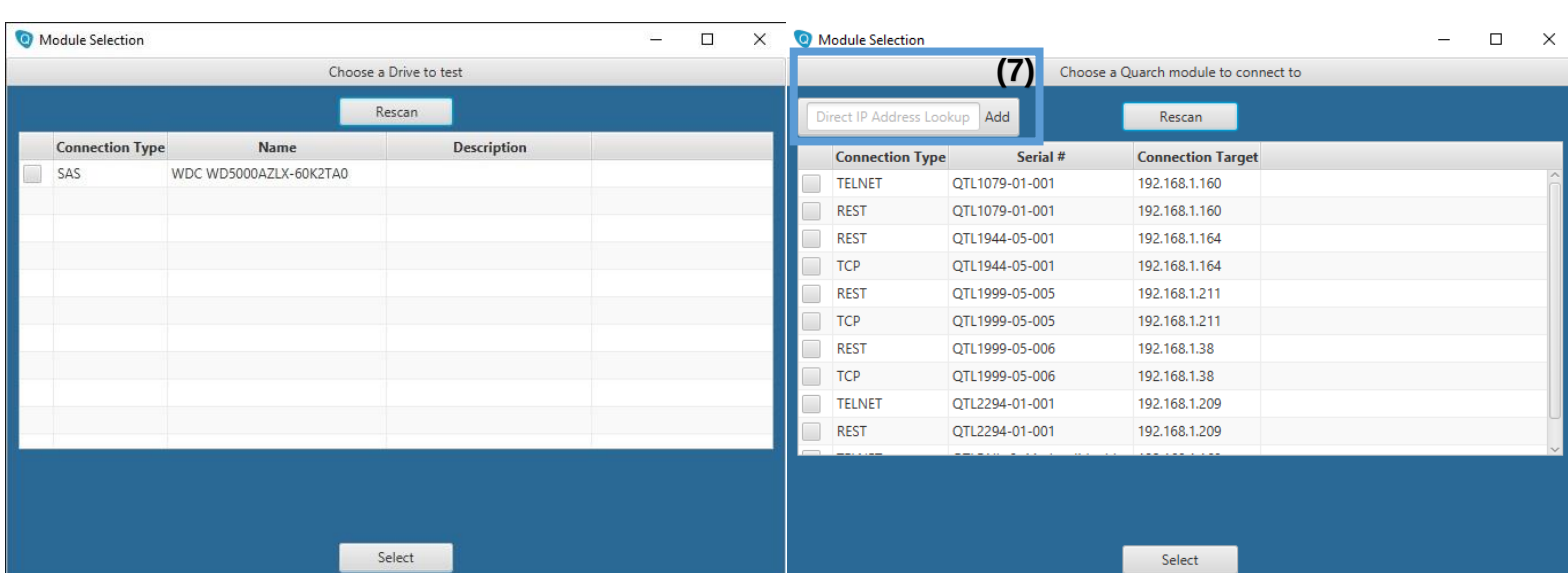
You can browse through the tests **(2)** looking at the description of each in the description window **(3)**. Tests that are unable to be selected require an upgraded license to allow usage. To unlock the specific test, use the ‘Upgrade’ button and follow the instructions.

Tests are selected in 2 phases. The first stage is to select the test you wish to run. This can be done either via a double click on the test, or a single click and pressing the “select test” button. This initial phase sets up the test, it checks for any requirements missing on the system and will display this if any are not found.



Upon selecting the test, the layout will change to allow customization of the test variables (4).

This tab allows the user to change specific variables associated with the selected test (5). Each variable includes a description of what its purpose is within the test. Once a desired value has been entered, hit the ‘Apply Values’ button in order to save the value for the test (6). The values are only saved for the current instance of QuarchComplianceSuite and upon restart of the program will be returned to default test specifications. The values can also be reset by pressing the ‘Reset Defaults’ button (6).



Upon test start, the user will be asked to select a module followed by a drive. These are the devices that will be used throughout the test.

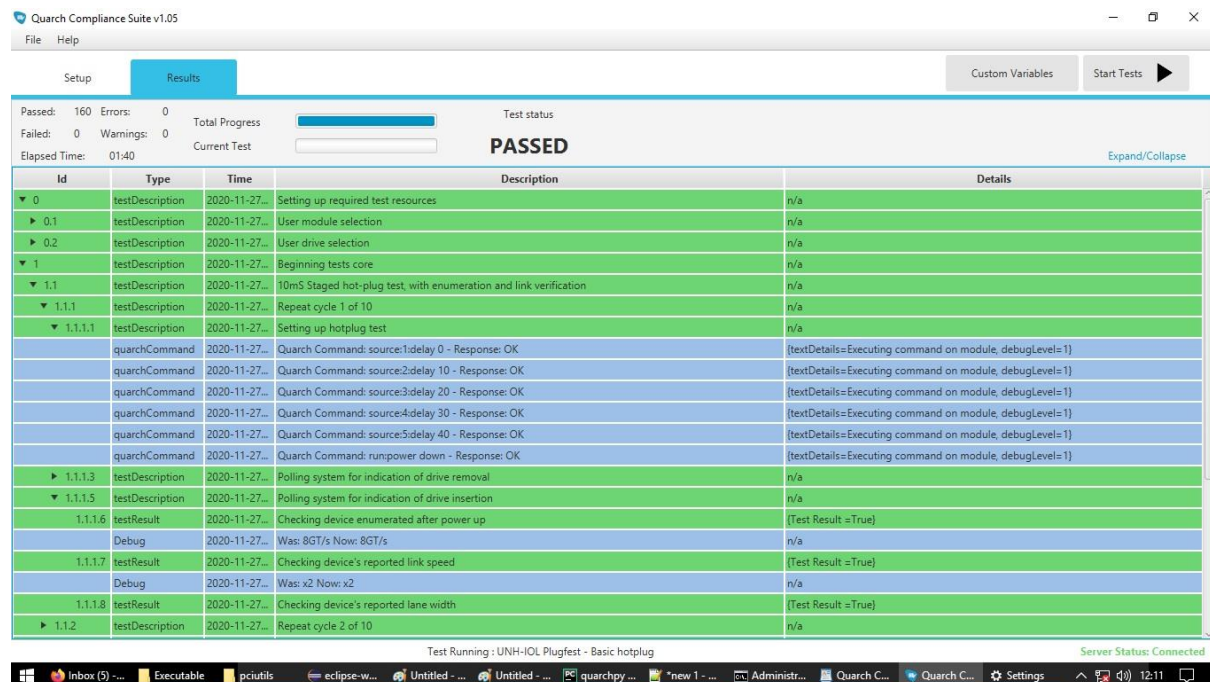
If you are connecting over TCP and your device is not automatically detected, you can attempt to locate the device directly through the IP Address lookup (7). This will send a discovery message directly to the IP address specified upon clicking ‘Add’. If a Quarch Module is detected on this IP, it will refresh the list with all found devices and the newly located device.

Next the user will be redirected to the Results tab. A test’s output will be displayed in a tree encapsulating several items of data about the ongoing test. This contains:

- Debug information: Some additional information on metrics of test
- Test descriptions: A brief explanation of the test being run.

- Test results: The outcome to a test point (e.g. has the drive been re-enumerated on the system?)
- Test Statistics: Data on a specific measurement (e.g. Max power during test)
- Quarch Commands: Commands sent to Quarch module and their result.
- Test Errors / Warnings: Anything that may be a cause for concern with test (e.g. Quarch Command not executing as expected)

If a test displays in green, it has passed a test. If a test is displayed in blue, it shows the test is currently ongoing. If a test displays in red, a test point within it failed. A yellow test point means an error has occurred. If you are unsure of what this error could mean for your test results, contact us with a copy of the test report or a screen capture and we will attempt to provide a solution / explanation.



The screenshot shows the Quarch Compliance Suite v1.05 interface. The top bar includes 'File' and 'Help' menus, and buttons for 'Custom Variables' and 'Start Tests'. The main area displays test statistics: Passed: 160, Errors: 0, Failed: 0, Warnings: 0, Total Progress: 100%, and Elapsed Time: 01:40. The test status is 'PASSED'. Below this is a table with columns: Id, Type, Time, Description, and Details. The table lists various test points, including 'Setting up required test resources', 'User module selection', 'User drive selection', 'Beginning tests core', '10mS Staged hot-plug test, with enumeration and link verification', 'Repeat cycle 1 of 10', 'Setting up hotplug test', 'Quarch Command: source:1:delay 0 - Response: OK', 'Quarch Command: source:2:delay 10 - Response: OK', 'Quarch Command: source:3:delay 20 - Response: OK', 'Quarch Command: source:4:delay 30 - Response: OK', 'Quarch Command: source:5:delay 40 - Response: OK', 'Quarch Command: run:power down - Response: OK', 'Polling system for indication of drive removal', 'Polling system for indication of drive insertion', 'Checking device enumerated after power up', 'Was: 8GT/s Now: 8GT/s', 'Checking device's reported link speed', 'Was: x2 Now: x2', 'Checking device's reported lane width', and 'Repeat cycle 2 of 10'. The table is expandable, as indicated by the 'Expand/Collapse' button in the top right corner of the table area.

The tree is an expandable view. If the user requires more information on what happened within the test, they can click the arrow on the far left that is adjacent to the item they wish to see more details on.

Generating a Report

Upon a test's completion you may wish to generate a report. This will allow you to preserve the current test's results alongside any statistics or errors that may have occurred during this. To generate a report, go to the 'Help' and click 'Generate Report'. By default this is saved to your user home directory. However this location can be changed within the setting menu.

A report consists of the following:

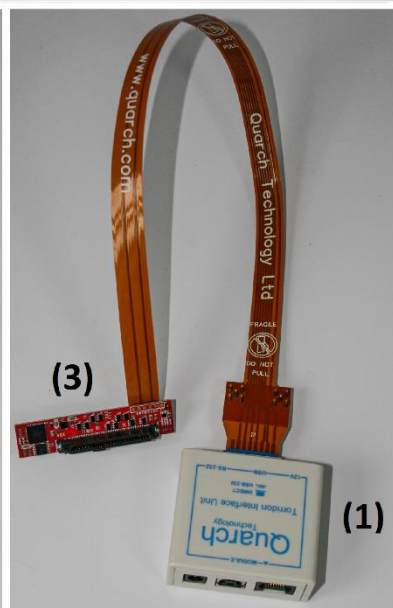
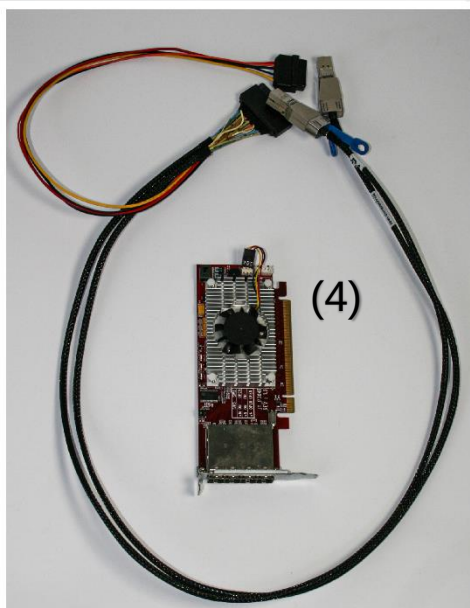
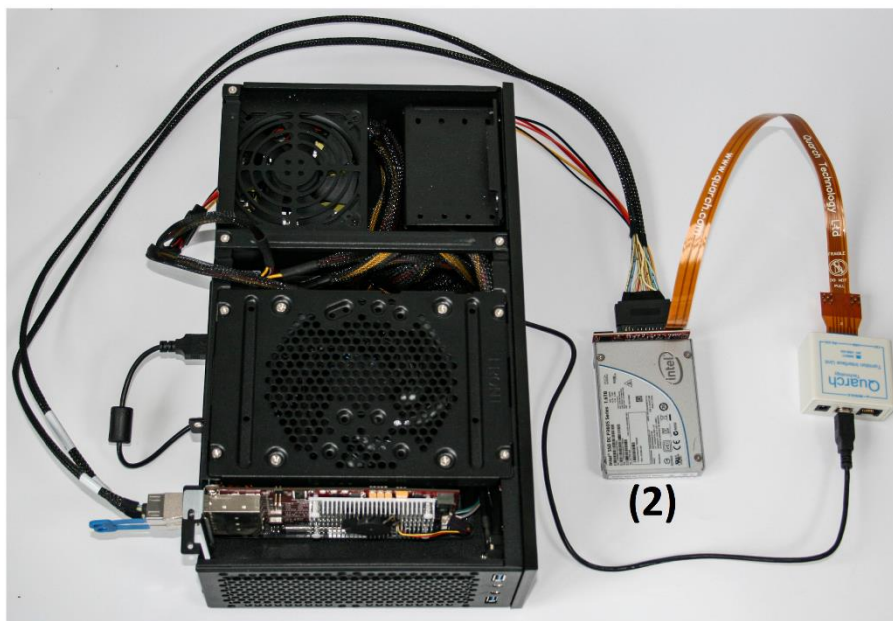
1. **Header Page**
Includes overview of test. Details of drive under test, Quarch module in use and some information on the system the test was run on.
2. **Test Variables**
Displays all the customizable variables used within the test alongside the description of what they are used for.
3. **Test Results**
A table view consisting of the test point ID, a description of what it was testing and the result of this test.
4. **Statistics (specific tests)**
A table view consisting of the tests run and the ID of this, followed by some rows containing the statistic measured and its value (to 3dp)
5. **Failed Quarch commands**
If any Quarch commands were not executed correctly, these will be displayed on the bottom of the report.

Example equipment configurations

Below are the 2 main tried and tested equipment configurations tested by Quarch. These setups are for the machines running the Python backend server PC.

Setup 1 – Hotplug Testing

- UNH-IOL Plugfest – Basic hotplug
- Full range hotplug test
- Pin-Bounce during hotplug
- Signal timing sweep during hotplug

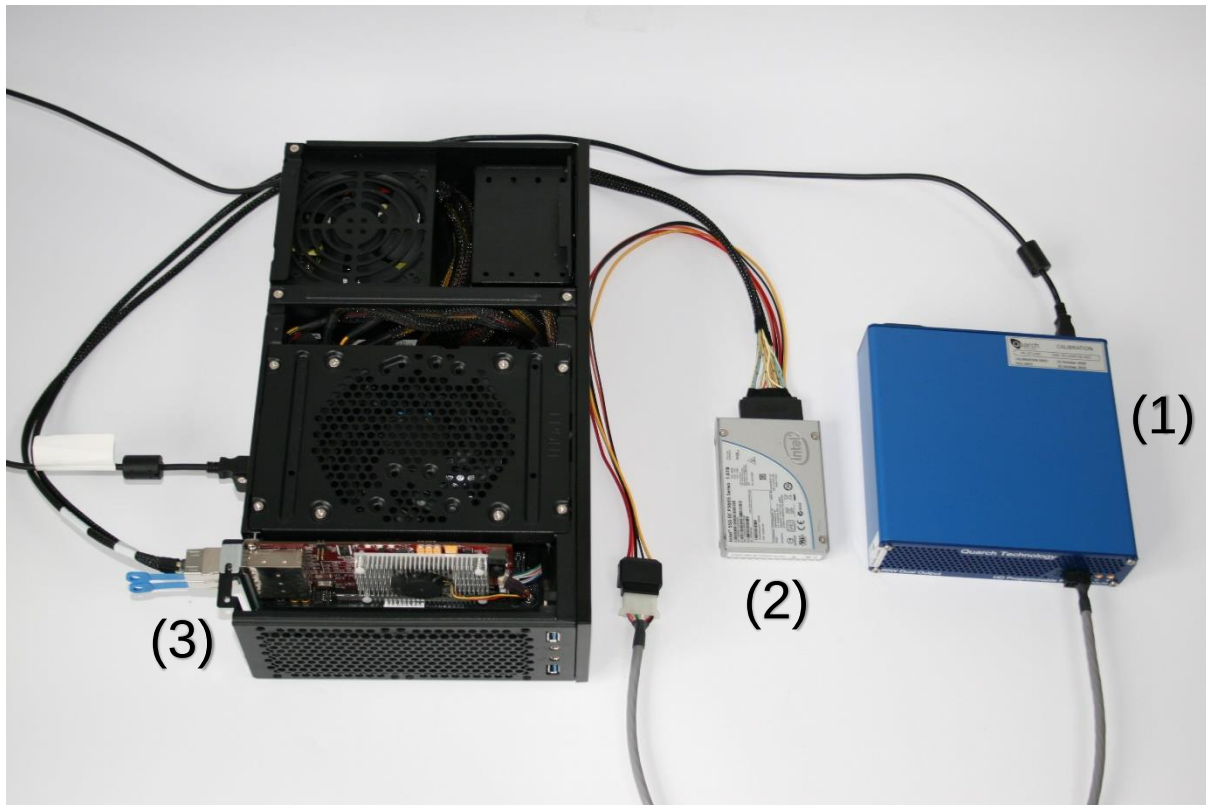


Equipment list:

- Quarch interface kit (1) – Connected to PC via USB / Serial cable
- Drive under test (2)
- Quarch Breaker Module (3) – Sits between PC and Drive under test. Connected to the Quarch interface kit, or any Quarch array module.
- Compatible host card (4)
(*Product in image : PCIe Gen3 PLX Based Host Card*)

Setup 2 – Power Testing

- Power Margining
- Drive power vs performance



Equipment list:

- Quarch HDPPM / Power Analysis Module (1) – Connected to PC via USB / TCP.
- Drive under test (2) – Quarch module to supply power or sit between drive and power for monitoring
- Compatible host card (3)
(*Product in image : PCIe Gen3 PLX Based Host Card*)

Getting help

If you have any questions, please email support@quarch.com

We will be happy to support you