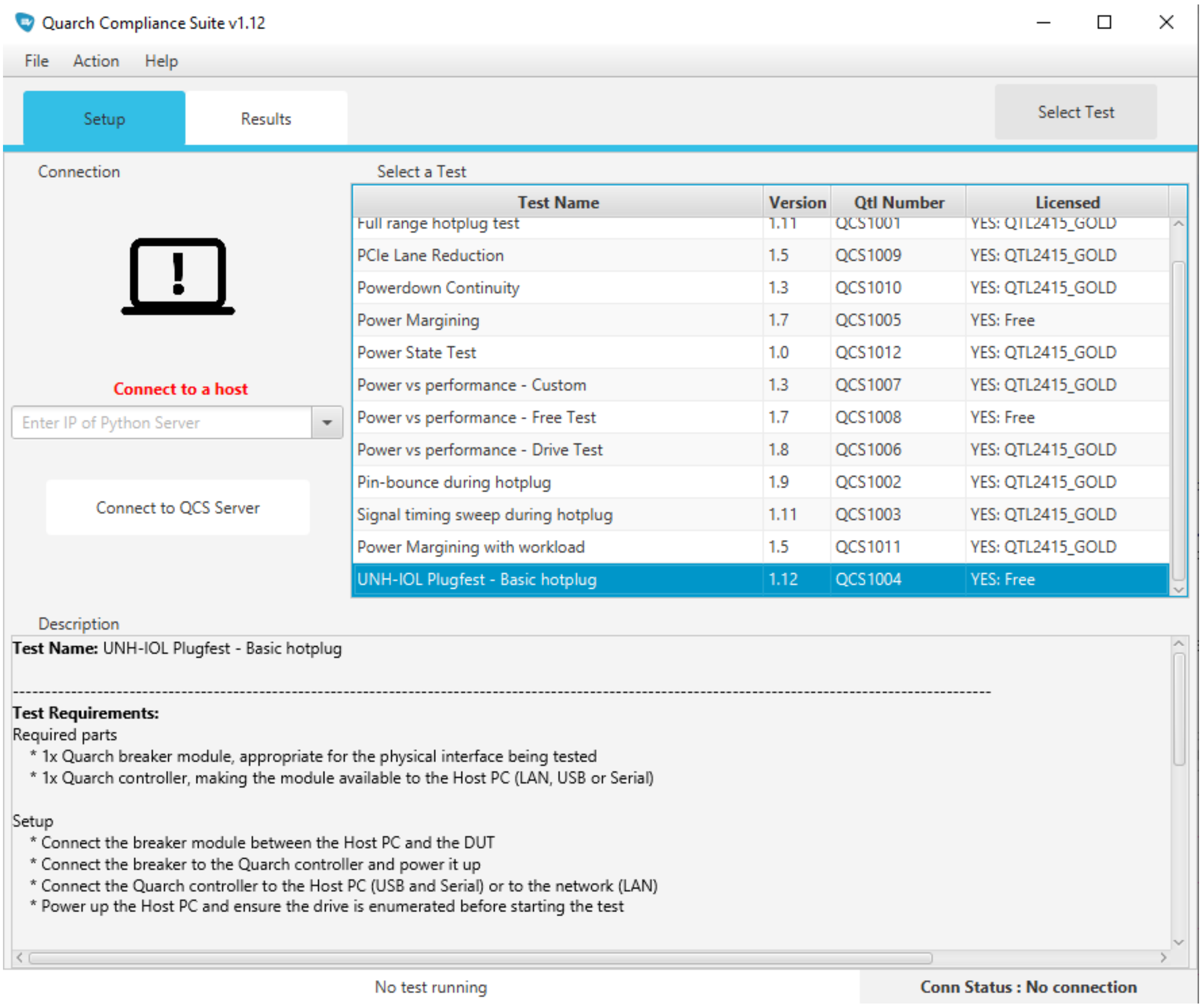
Quarch Technology Ltd

Quarch Compliance Suite

Quickstart



Test Name	Version	Qtl Number	Licensed
Full range hotplug test	1.11	QCS1001	YES: QTL2415_GOLD
PCIe Lane Reduction	1.5	QCS1009	YES: QTL2415_GOLD
Powerdown Continuity	1.3	QCS1010	YES: QTL2415_GOLD
Power Margining	1.7	QCS1005	YES: Free
Power State Test	1.0	QCS1012	YES: QTL2415_GOLD
Power vs performance - Custom	1.3	QCS1007	YES: QTL2415_GOLD
Power vs performance - Free Test	1.7	QCS1008	YES: Free
Power vs performance - Drive Test	1.8	QCS1006	YES: QTL2415_GOLD
Pin-bounce during hotplug	1.9	QCS1002	YES: QTL2415_GOLD
Signal timing sweep during hotplug	1.11	QCS1003	YES: QTL2415_GOLD
Power Margining with workload	1.5	QCS1011	YES: QTL2415_GOLD
UNH-IOL Plugfest - Basic hotplug	1.12	QCS1004	YES: Free

Test Name: UNH-IOL Plugfest - Basic hotplug

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)
* Power up the Host PC and ensure the drive is enumerated before starting the test

Change History

1.0	26/07/2019	Initial Release
1.2	11/12/2020	Added information on installer
1.3	01/09/2021	Updating for new requirements and a more detailed setup breakdown
1.4	06/04/2023	Updated for 1.13 release introducing test comparison

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
1.8	11/03/2021	Releasing numerous new tests and bug fixing.
1.9	28/07/2021	Released new PCIe lane reduction test
1.9.01	01/09/2021	Swapping QCS server to standalone python package and bug fixes
1.10	29/03/2022	Updating custom variable displays Added new Power Margining Test
1.11	09/09/2022	Updated reports Added Power state transition test
1.12	31/01/2023	Added pcie lane reduction test
1.13	04/04/2023	Added test comparison functionality (beta) Bug fixed and updated tests

Contents

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
    - Allows 1 licensed QCS client to connect (individually) to different QCS server PC's.
- 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.

Operating systems

Both QCS client and QCS server are approved for use on the following Operating systems:

- Ubuntu
- Centos
- Windows 10 (*english version*)
- Windows server 2016 (*english version*)

This includes using QCS client on one operating system and QCS server on another operating system.

Licensing

Customers can buy a QCS license in order to run the additional and extended tests.

The license is associated with the Client PC only. This means a user can run QCS server on any number of different hardware setups, but the QCS client is bound to the hardware that was used to generate a license.

Licenses are sold currently for a period of 1 year that includes full support for that 1 year time period. At the end of this year, the license expires and would require a new / renewed license from Quarch.

To read more about how to setup a license visit :

https://quarch.com/support/faqs/qcs-licensing/

To buy a license, please talk to your local reseller about buying a license from Quarch.

https://quarch.com/resellers/

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
- Quarchpy python package to be installed on Client PC.
    - Quarchpy contains a known location to launch QPS + QIS for different tests and is required for automating tests that utilise these programs.
- QuarchQCS python package to be installed on server PC. Downloaded from the public pypi repository via the command “python -m pip install quarchqcs”.
    - This package requires both the base Quarchpy package and Pandas. These are auto-installed on the download of the QuarchQcs package.
- SmartCTL (*tested up to V7.2*) required on Server PC - Download at smartmontools

Optional:

- The ‘zeroconf’ python module is required on the Server PC if you wish to use auto-discovery of QCS servers on your network.

Java install

If you are 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.

Check that Java JRE 8 is installed on the machine you want to run the Quarch Compliance Suite client 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 on the client PC and QuarchQCS on the server PC. 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 quarchqcs
```

```
> Pip install quarchpy
```

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

```
> Pip install quarchqcs --upgrade
```

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

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

```
> python -m pip install quarchqcs –upgrade
```

```
> 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
```

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

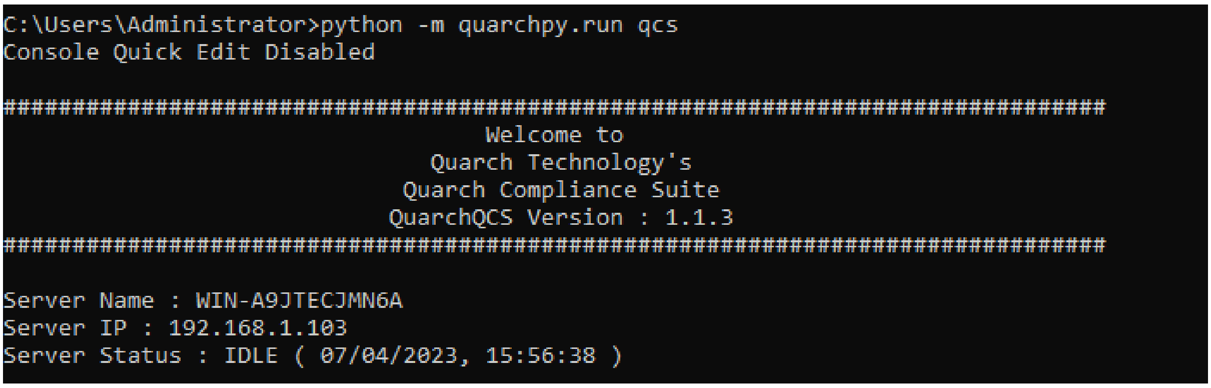
- Quarch power module: XLC, HD, or PAM
    - https://quarch.com/products/xlc-programmable-power-module/
    - https://quarch.com/products/hd-programmable-power-module/
    - https://quarch.com/products/power-analysis-module/
    - The power module must be available to the **Client** PC, either via LAN 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 QCS server on the system that the drives are connected to. Following this, launching the QCS client and lastly connecting the QCS client to the QCS server.

The QCS 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 QCS server on a Linux PC, load up a new terminal prefix the command below with “sudo”.
The command to start the qcs server is:

Python -m quarchpy.run qcs



```
C:\Users\Administrator>python -m quarchpy.run qcs
Console Quick Edit Disabled

################################################################################
                                   Welcome to
                               Quarch Technology's
                             Quarch Compliance Suite
                            QuarchQCS Version : 1.1.3
################################################################################

Server Name : WIN-A9JTECJMN6A
Server IP : 192.168.1.103
Server Status : IDLE ( 07/04/2023, 15:56:38 )
```

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 QCS 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:

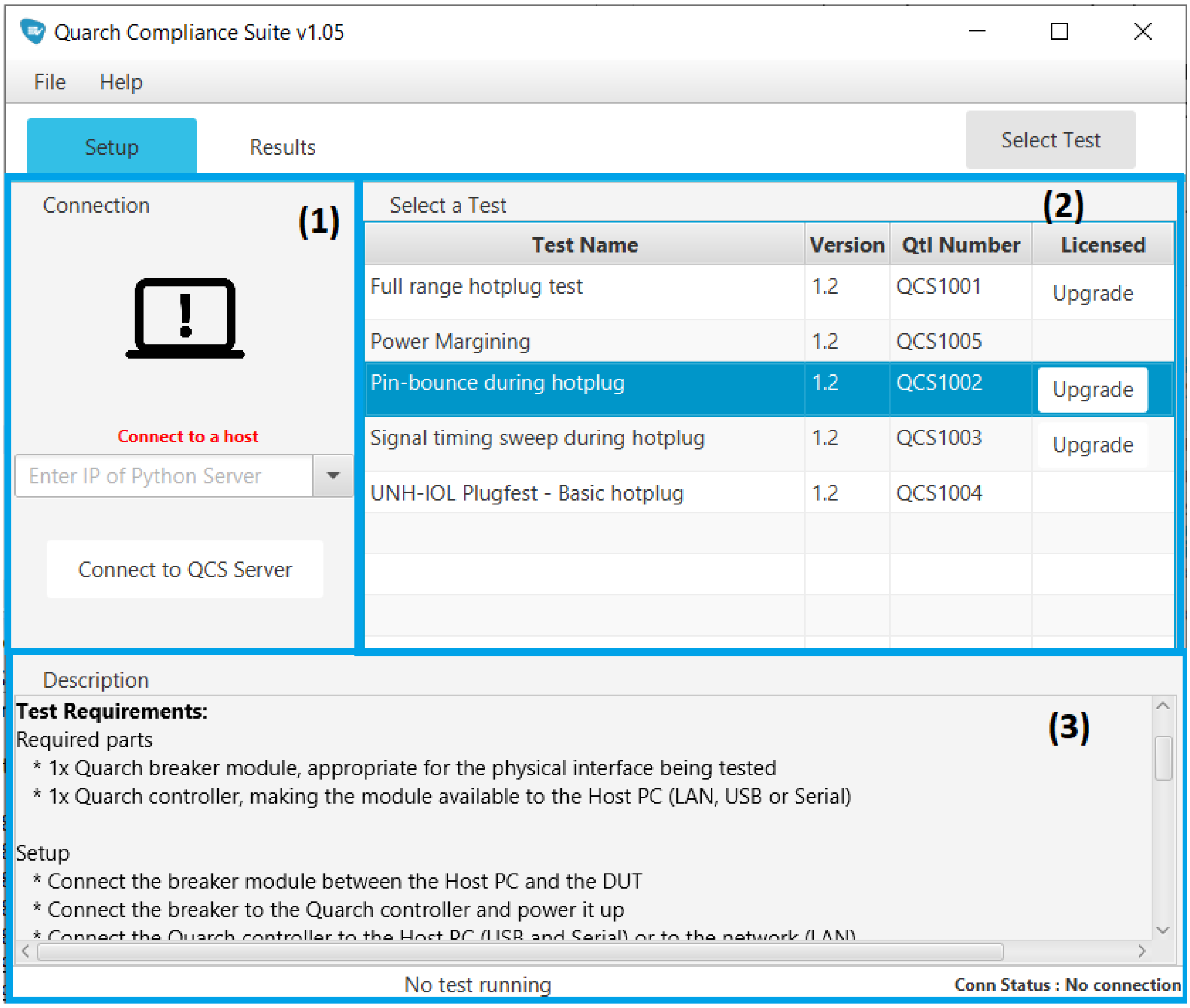
```
> Pip install zeroconf
```

If you have installed the QCS client on 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 have to open a new terminal, navigate to the directory containing QCS client and start it using the command:

java -jar QCS.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.

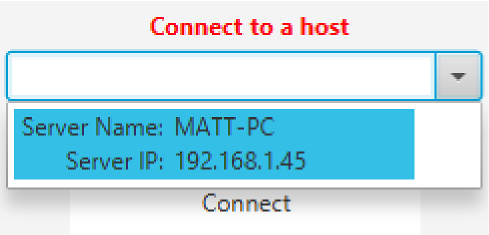


Test Name	Version	Qtl Number	Licensed
Full range hotplug test	1.2	QCS1001	Upgrade
Power Margining	1.2	QCS1005	
Pin-bounce during hotplug	1.2	QCS1002	Upgrade
Signal timing sweep during hotplug	1.2	QCS1003	Upgrade
UNH-IOL Plugfest - Basic hotplug	1.2	QCS1004	

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)

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.



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.

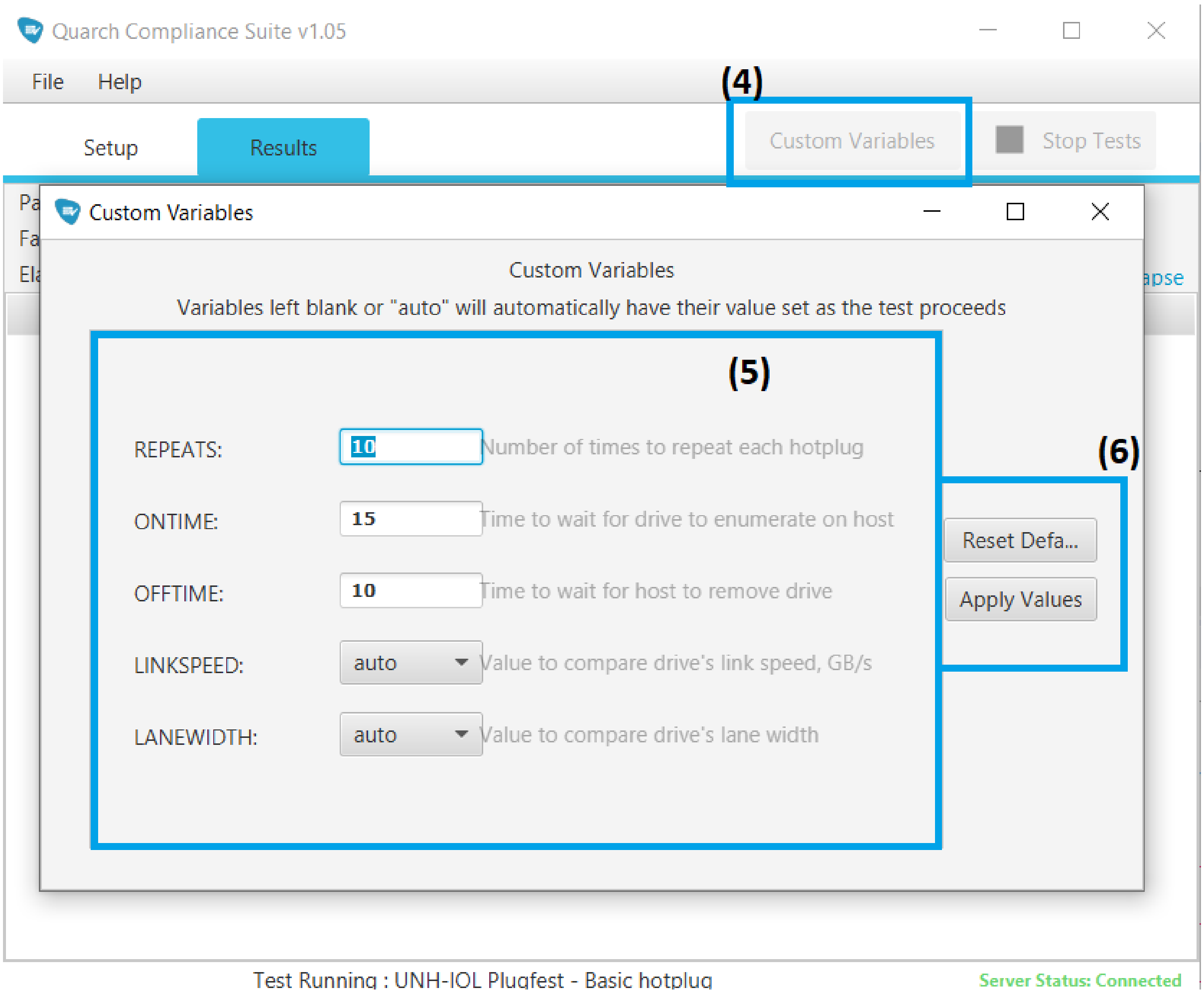
Some computers may require you to open ports 9742 and 9921 for valid communication between the Java frontend and Python backend.

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.

Once a test is selected and no errors are found, the customer can now select either to start the test or to edit the variables of the test (seen in the image below **(4)**).

Test variables are different on each test and allow different aspects of the test to be changed depending on the user's needs. For example, the amount of time to repeat the test or whether the test should stop if a failure is detected.



Custom Variables

Variables left blank or "auto" will automatically have their value set as the test proceeds

REPEATS: 10 Number of times to repeat each hotplug

ONTIME: 15 Time to wait for drive to enumerate on host

OFFTIME: 10 Time to wait for host to remove drive

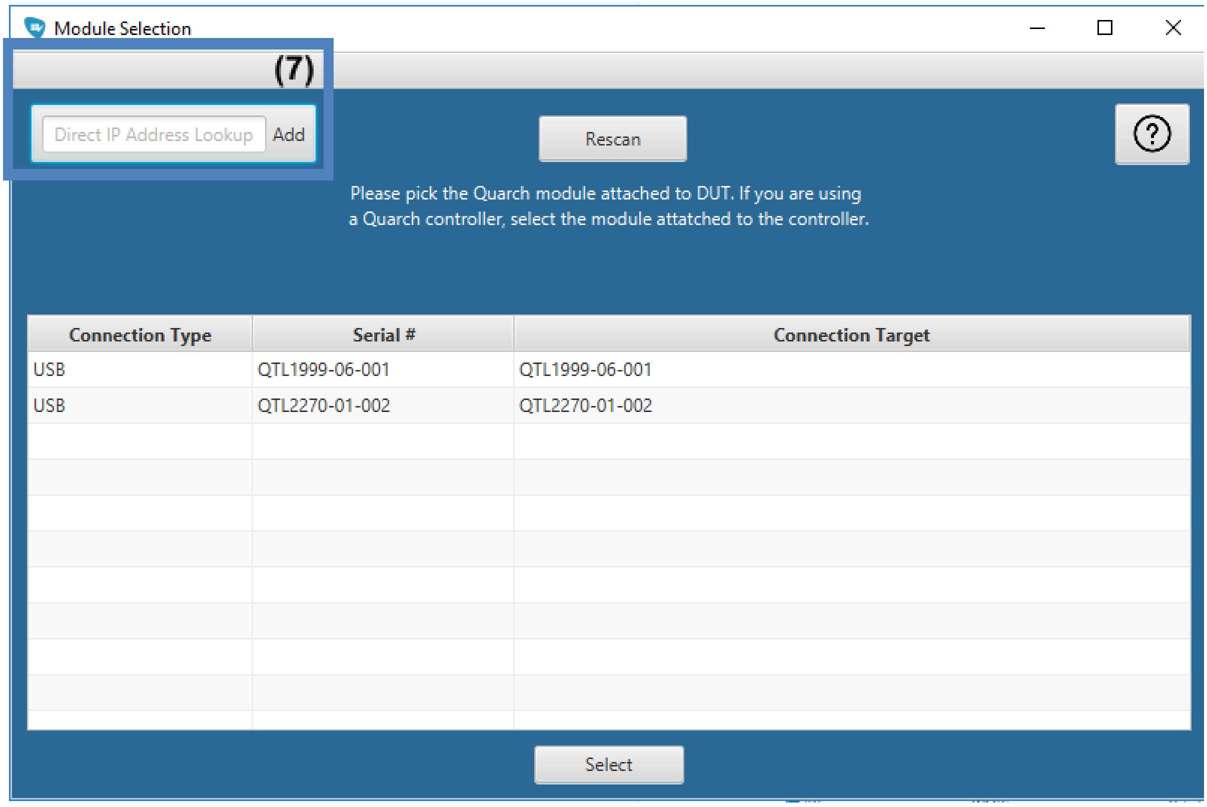
LINKSPEED: auto Value to compare drive's link speed, GB/s

LANEWIDTH: auto Value to compare drive's lane width

Test Running : UNH-IOL Plugfest - Basic hotplug

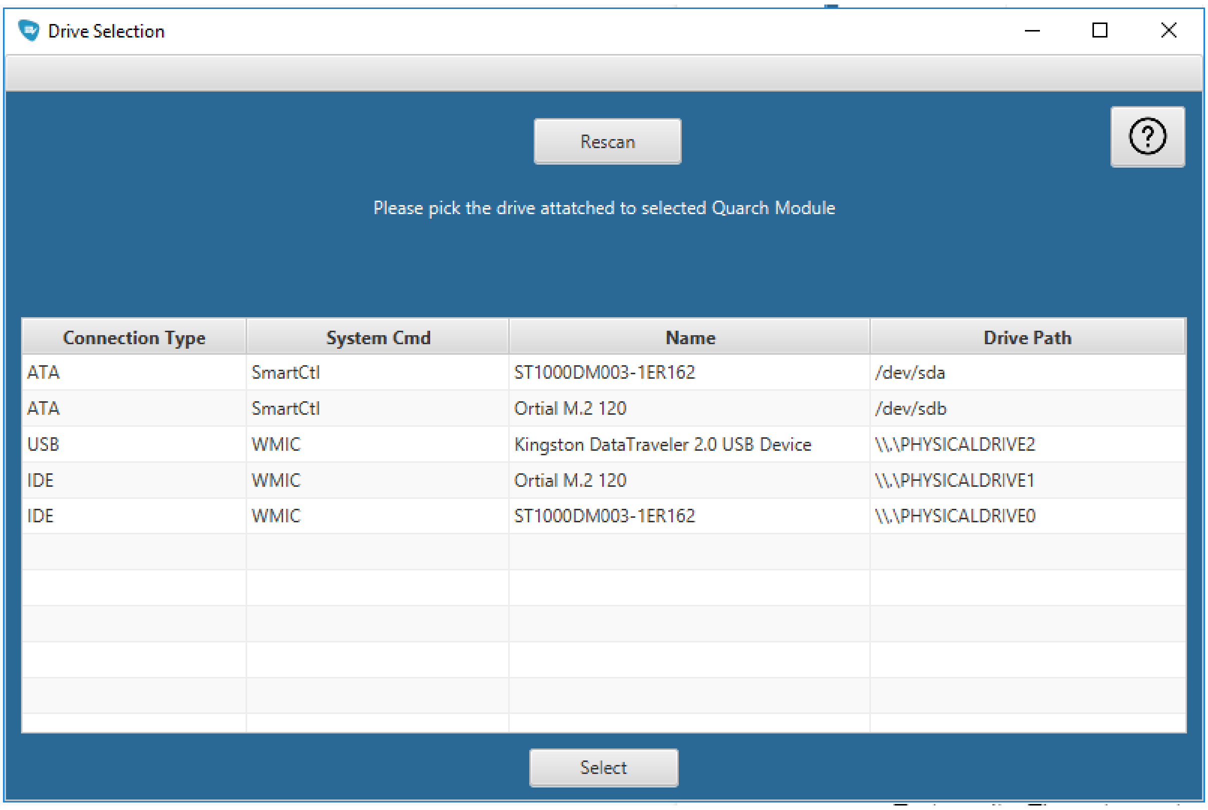
Server Status: Connected

Each variable includes a description of what its purpose is within the test **(5)**. 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 to the test default by pressing the ‘Reset Defaults’ button **(6)**.



Please pick the Quarch module attached to DUT. If you are using a Quarch controller, select the module attatched to the controller.

Connection Type	Serial #	Connection Target
USB	QTL1999-06-001	QTL1999-06-001
USB	QTL2270-01-002	QTL2270-01-002



Please pick the drive attatched to selected Quarch Module

Connection Type	System Cmd	Name	Drive Path
ATA	SmartCtl	ST1000DM003-1ER162	/dev/sda
ATA	SmartCtl	Ortial M.2 120	/dev/sdb
USB	WMIC	Kingston DataTraveler 2.0 USB Device	\\.\PHYSICALDRIVE2
IDE	WMIC	Ortial M.2 120	\\.\PHYSICALDRIVE1
IDE	WMIC	ST1000DM003-1ER162	\\.\PHYSICALDRIVE0

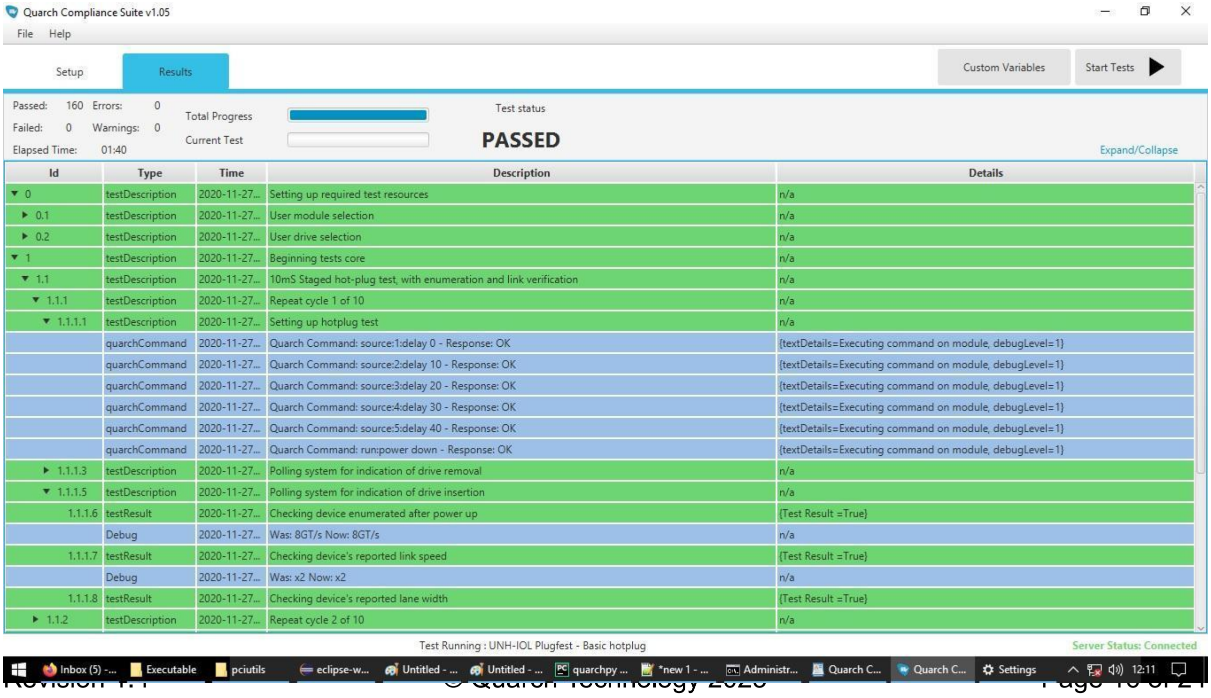
Upon test start, the user will be asked to select a module followed by a drive. These are the devices that will be used throughout the test.

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 has 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.



Passed: 160 Errors: 0
Failed: 0 Warnings: 0
Elapsed Time: 01:40
Total Progress
Current Test
Test status
PASSED

Id	Type	Time	Description	Details
▼ 0	testDescription	2020-11-27...	Setting up required test resources	n/a
▶ 0.1	testDescription	2020-11-27...	User module selection	n/a
▶ 0.2	testDescription	2020-11-27...	User drive selection	n/a
▼ 1	testDescription	2020-11-27...	Beginning tests core	n/a
▼ 1.1	testDescription	2020-11-27...	10mS Staged hot-plug test, with enumeration and link verification	n/a
▼ 1.1.1	testDescription	2020-11-27...	Repeat cycle 1 of 10	n/a
▼ 1.1.1.1	testDescription	2020-11-27...	Setting up hotplug test	n/a
	quarchCommand	2020-11-27...	Quarch Command: source:1:delay 0 - Response: OK	{textDetails=Executing command on module, debugLevel=1}
	quarchCommand	2020-11-27...	Quarch Command: source:2:delay 10 - Response: OK	{textDetails=Executing command on module, debugLevel=1}
	quarchCommand	2020-11-27...	Quarch Command: source:3:delay 20 - Response: OK	{textDetails=Executing command on module, debugLevel=1}
	quarchCommand	2020-11-27...	Quarch Command: source:4:delay 30 - Response: OK	{textDetails=Executing command on module, debugLevel=1}
	quarchCommand	2020-11-27...	Quarch Command: source:5:delay 40 - Response: OK	{textDetails=Executing command on module, debugLevel=1}
	quarchCommand	2020-11-27...	Quarch Command: run:power down - Response: OK	{textDetails=Executing command on module, debugLevel=1}
▶ 1.1.1.3	testDescription	2020-11-27...	Polling system for indication of drive removal	n/a
▼ 1.1.1.5	testDescription	2020-11-27...	Polling system for indication of drive insertion	n/a
1.1.1.6	testResult	2020-11-27...	Checking device enumerated after power up	{Test Result =True}
	Debug	2020-11-27...	Was: 8GT/s Now: 8GT/s	n/a
1.1.1.7	testResult	2020-11-27...	Checking device's reported link speed	{Test Result =True}
	Debug	2020-11-27...	Was: x2 Now: x2	n/a
1.1.1.8	testResult	2020-11-27...	Checking device's reported lane width	{Test Result =True}
▶ 1.1.2	testDescription	2020-11-27...	Repeat cycle 2 of 10	n/a

Test Running : UNH-IOL Plugfest - Basic hotplug

Server Status: Connected

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.

Additional information: Running on Linux

QCS Server

Running QCS server on a linux OS requires sudo privileges.

‘sudo python3 -m quarchpy.run qcs’

This is to ensure LSPCI can access information about the drives on the system.

QCS Client

Running QCS client on a linux OS does not require sudo privileges.
When running a Power test, QCS client will automatically launch Quarch Power Studio and Quarch Instrumental Server.

Before running a Power test, we recommend running the following quarchpy command:
“python3 -m quarchpy.run debug”

Should an error be shown stating *‘permission issues accessing USB modules’*, the user should run :

“sudo python3 -m quarchpy,run debug -fixusb”

The above command adds a new usb rule to the linux OS allowing all Quarch modules over USB to be utilised by users without requiring sudo privileges.

Additional information: Running on Windows

QCS Server

Running QCS server on a windows OS requires administrator privileges.

This is done by running the command prompt as administrator.

‘python -m quarchpy.run qcs’

This is to ensure LSPCI can access information about the drives on the system.

QCS Client

Running QCS client on a windows OS does not require administrator privileges.

As windows allows all USB interaction regardless of device and privileges, all Quarch modules are automatically discovered for all users.

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:

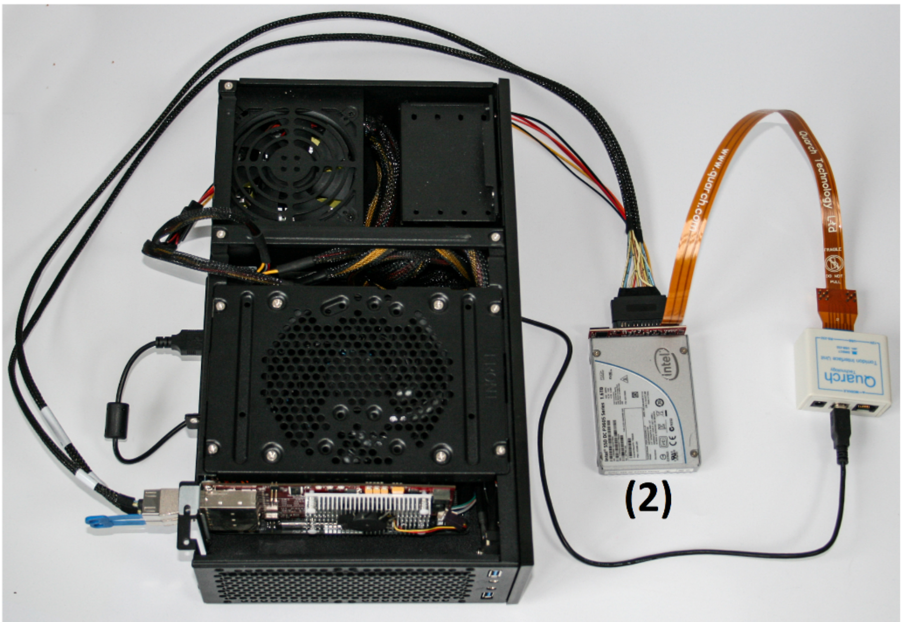
1. **Header Page**
   Includes overview of test. Details of drive under test, Quarch module in use and some information on the system the test was run on.
2. **Test Variables**
   Displays all the customizable variables used within the test alongside the description of what they are used for.
3. **Test 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.

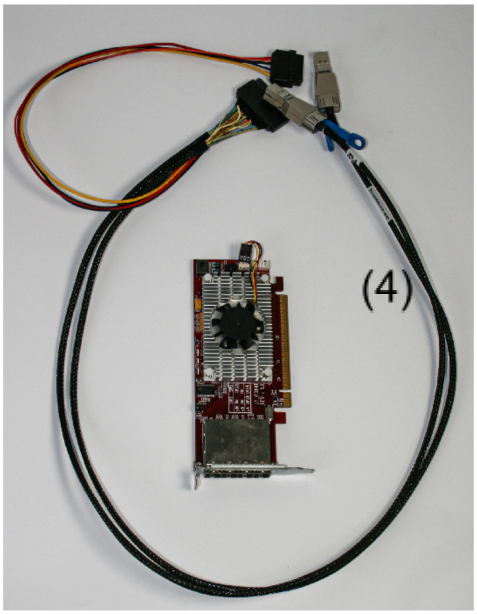
Example equipment configurations

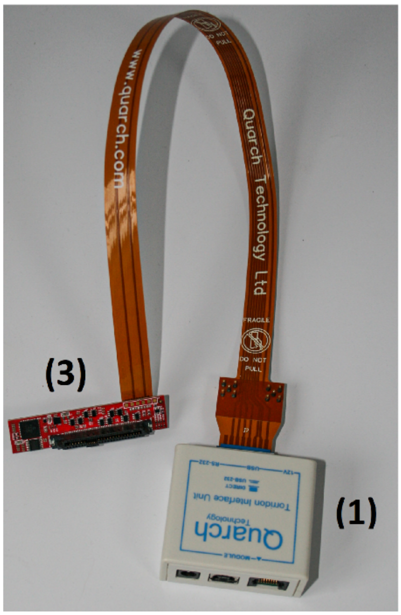
Below are the 2 main tried and tested equipment configurations tested by Quarch. These setups are for the machines running the Python backend server PC.

Setup 1 – Hotplug Testing

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





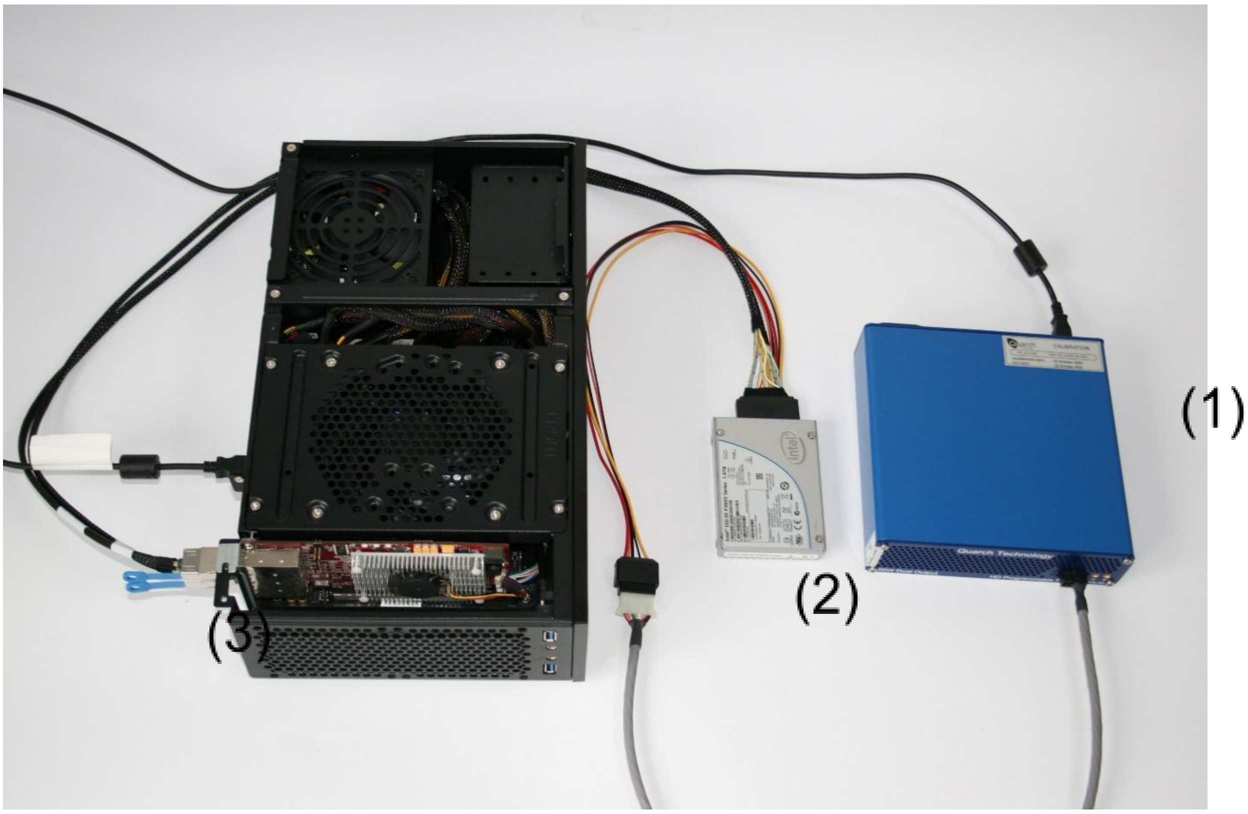


Equipment list:

- Quarch interface kit (1) – Connected to **Server** 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:

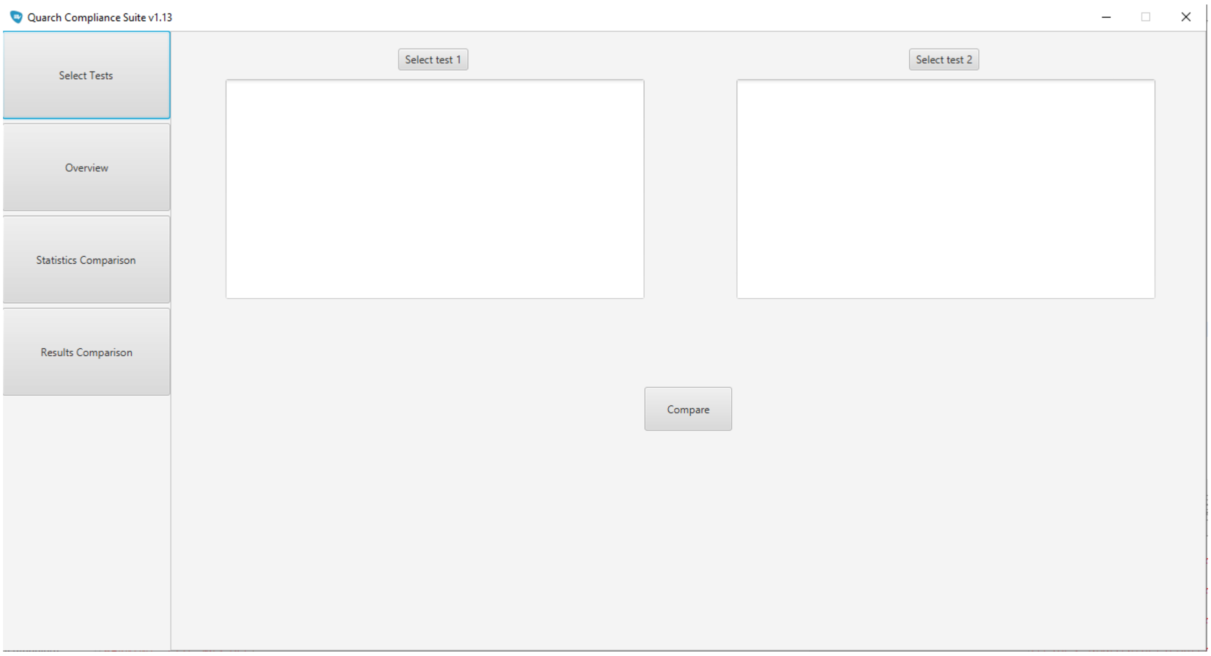
- Quarch HDPPM / Power Analysis Module (1) – Connected to **Client** PC via USB / TCP.
- Drive under test (2) – Quarch module to supply power or sit between drive and power for monitoring
- Compatible host card (3)
  (Product in image : PCIe Gen3 PLX Based Host Card)

Test comparisons

As of QCS 1.13, every test will now output a full test log to a location on the QCS client system.

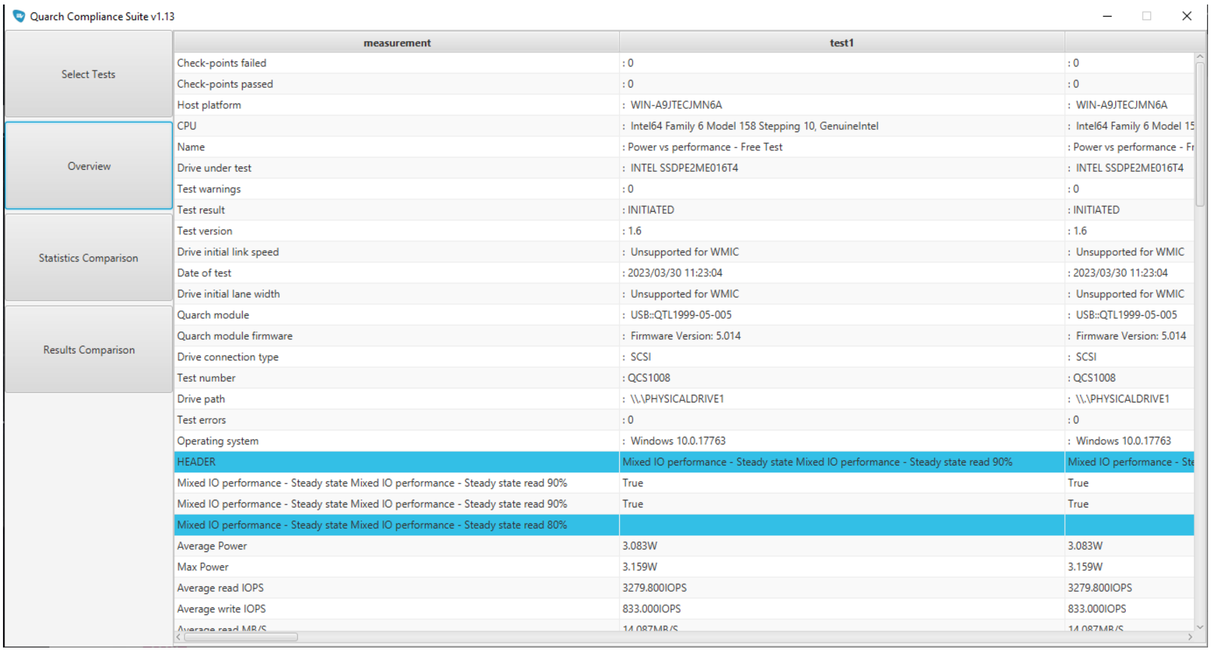
%USER_HOME%/AppData/Local/Quarch/QuarchComplianceSuite/Results/

To open the test comparison utility, go to file in the menu bar and hit ‘Test compare’ in the dropdown menu. This will bring up the following panel.



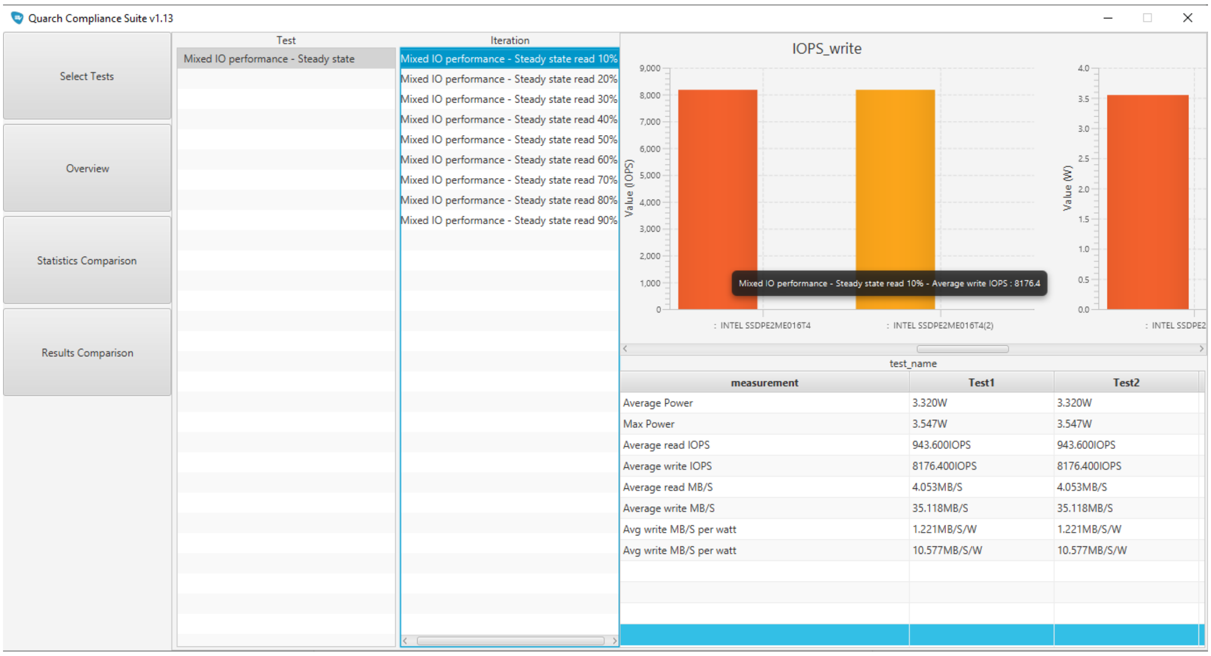
From here, the user needs to press the “select test” buttons and select the 2 tests they wish to compare.
(*Note - In the file dialog, navigate inside the qcs result directory and select the .qcs file of the relevant test)*

Once the 2 tests have been loaded in, press “compare” to compare the tests. On a successful test comparison, QCS will redirect the user to the ‘overview’ tab. This lists a side by side comparison of every data point and check-point for the user to inspect.



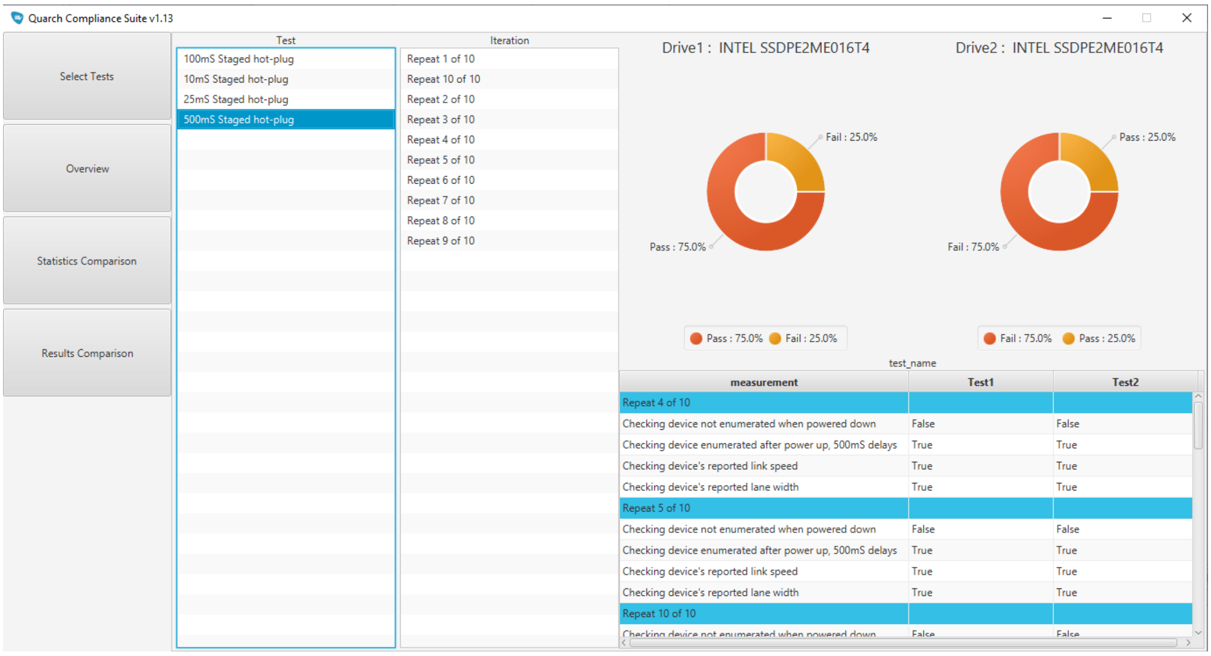
measurement	test1	
Check-points failed	: 0	: 0
Check-points passed	: 0	: 0
Host platform	: WIN-A9JTECJMN6A	: WIN-A9JTECJMN6A
CPU	: Intel64 Family 6 Model 158 Stepping 10, GenuineIntel	: Intel64 Family 6 Model 15
Name	: Power vs performance - Free Test	: Power vs performance - Fr
Drive under test	: INTEL SSDPE2ME016T4	: INTEL SSDPE2ME016T4
Test warnings	: 0	: 0
Test result	: INITIATED	: INITIATED
Test version	: 1.6	: 1.6
Drive initial link speed	: Unsupported for WMIC	: Unsupported for WMIC
Date of test	: 2023/03/30 11:23:04	: 2023/03/30 11:23:04
Drive initial lane width	: Unsupported for WMIC	: Unsupported for WMIC
Quarch module	: USB::QTL1999-05-005	: USB::QTL1999-05-005
Quarch module firmware	: Firmware Version: 5.014	: Firmware Version: 5.014
Drive connection type	: SCSI	: SCSI
Test number	: QCS1008	: QCS1008
Drive path	: \\.\PHYSICALDRIVE1	: \\.\PHYSICALDRIVE1
Test errors	: 0	: 0
Operating system	: Windows 10.0.17763	: Windows 10.0.17763
HEADER	Mixed IO performance - Steady state Mixed IO performance - Steady state read 90%	Mixed IO performance - Ste
Mixed IO performance - Steady state Mixed IO performance - Steady state read 90%	True	True
Mixed IO performance - Steady state Mixed IO performance - Steady state read 90%	True	True
Mixed IO performance - Steady state Mixed IO performance - Steady state read 80%		
Average Power	3.083W	3.083W
Max Power	3.159W	3.159W
Average read IOPS	3279.800IOPS	3279.800IOPS
Average write IOPS	833.000IOPS	833.000IOPS
Average read MB/S	14.087MB/S	14.087MB/S

Navigating to the ‘Statistics overview’ will display only the test's statistical values. These are also visually represented using bar charts, which can be further broken down into individual test iterations as seen below.


Quarch Compliance Suite v1.13
Select Tests
Overview
Statistics Comparison
Results Comparison
Test
Mixed IO performance - Steady state
Iteration
Mixed IO performance - Steady state read 10%
Mixed IO performance - Steady state read 20%
Mixed IO performance - Steady state read 30%
Mixed IO performance - Steady state read 40%
Mixed IO performance - Steady state read 50%
Mixed IO performance - Steady state read 60%
Mixed IO performance - Steady state read 70%
Mixed IO performance - Steady state read 80%
Mixed IO performance - Steady state read 90%
IOPS_write
Value (IOPS)
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
Mixed IO performance - Steady state read 10% - Average write IOPS : 8176.4
: INTEL SSDPE2ME016T4
: INTEL SSDPE2ME016T4(2)
Value (W)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
: INTEL SSDPE2
test_name

measurement	Test1	Test2
Average Power	3.320W	3.320W
Max Power	3.547W	3.547W
Average read IOPS	943.600IOPS	943.600IOPS
Average write IOPS	8176.400IOPS	8176.400IOPS
Average read MB/S	4.053MB/S	4.053MB/S
Average write MB/S	35.118MB/S	35.118MB/S
Avg write MB/S per watt	1.221MB/S/W	1.221MB/S/W
Avg write MB/S per watt	10.577MB/S/W	10.577MB/S/W

Lastly, clicking the 'result's overview' will navigate to another panel displaying only the pass/fail results from both tests side by side.
These are also visually represented using doughnut charts for each test to give the user a simple view of which tests passed and failed.


Quarch Compliance Suite v1.13
Select Tests
Overview
Statistics Comparison
Results Comparison
Drive1 : INTEL SSDPE2ME016T4
Drive2 : INTEL SSDPE2ME016T4
Fail : 25.0%
Pass : 75.0%
Pass : 75.0% Fail : 25.0%
Pass : 25.0%
Fail : 75.0%
Fail : 75.0% Pass : 25.0%

Test
100mS Staged hot-plug
10mS Staged hot-plug
25mS Staged hot-plug
500mS Staged hot-plug

Iteration
Repeat 1 of 10
Repeat 10 of 10
Repeat 2 of 10
Repeat 3 of 10
Repeat 4 of 10
Repeat 5 of 10
Repeat 6 of 10
Repeat 7 of 10
Repeat 8 of 10
Repeat 9 of 10

test_name

measurement	Test1	Test2
Repeat 4 of 10		
Checking device not enumerated when powered down	False	False
Checking device enumerated after power up, 500mS delays	True	True
Checking device's reported link speed	True	True
Checking device's reported lane width	True	True
Repeat 5 of 10		
Checking device not enumerated when powered down	False	False
Checking device enumerated after power up, 500mS delays	True	True
Checking device's reported link speed	True	True
Checking device's reported lane width	True	True
Repeat 10 of 10		
Checking device not enumerated when powered down	False	False

Getting help

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

We will be happy to support you