Quarch Technology Ltd

AN-016

Application Note

QPS automation with Iometer

For use with:

**XLC Power Modules  
HD Power Modules**

**Quarch Power Studio (QPS)**



# Change History

|  |  |  |
| --- | --- | --- |
| 1.0 |  | Initial Release |
| 1.1 |  | Significant improvements and new control options |
| 1.2 | Feb 2019 | Moving to Python 3.x |
|  |  |  |
|  |  |  |

Contents

[Change History 2](#_Toc521058869)

[Introduction 4](#_Toc521058870)

[Modules Supported 5](#_Toc521058871)

[System Supported 5](#_Toc521058872)

[Application Note Example Files 5](#_Toc521058873)

[Installation and setup 6](#_Toc521058874)

[Python install 6](#_Toc521058875)

[QuarchPy library install 6](#_Toc521058876)

[Java install 8](#_Toc521058877)

[QPS install 8](#_Toc521058878)

[Additional library install 8](#_Toc521058879)

[Power module setup 9](#_Toc521058880)

[Running the example 10](#_Toc521058881)

# Introduction

Quarch Power Studio (QPS) is a unique system for recording and analyzing power consumption of storage devices.

This application note demonstrates the ability to automate QPS and bring in user data from another source

In this example, we will use Iometer to drive traffic to a device and plot both power and performance as part of a fully automated test.

This example was demonstrated at FMS2018

# Modules Supported

XLC power modules

* QTL1824 (QTL1824-02A modules do NOT support streaming, but can be upgraded by Quarch)
* QTL1847

HD power modules

* QTL1995
* QTL1999

# System Supported

This example is written for windows, though could be converted to Linux if the drive identifying functions were re-written.

It currently requires Python 2.x

## Application Note Example Files

The **AN-016.zip** should be extracted to your preferred location.

|  |  |
| --- | --- |
| IometerExample.py | Main python file to execute |
| IOmeter.exe | IOmeter application |
| Dynamo.exe | Part of IOmeter |
| Iometer License.txt | License for IOmeter |
| /conf | Folder to place .conf and .icf files for execution |
| /conf\_examples | Provided examples of .icf and .conf files |
| Csv\_example.csv | Example csv file with workloads for use by the example script |

# Installation and setup

## Python install

If you do not already have Python installed, download and install it from:

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

Under Windows it is helpful to make sure the Python installation directory and PythonXX\Scripts are included in the PATH environment variable. See:

<https://docs.python.org/2/using/windows.html#excursus-setting-environment-variables>

## QuarchPy library install

The Quarch Python package can be installed from the Python web repository (assuming you have internet access) or via the download from our website.

Quarchpy will also install a version of Quarch Power Studio

### Web Install

From the command line:

**>pip install quarchpy**

If this fails, your path to “pip” may not be set, you can instead run:

**>python –m pip install quarchpy**

### Local Install

If you want to install from a downloaded folder, ensure the folder is unzipped to a local disk, navigate to the folder containing the setup.py file and run (noting the ‘.’ on the end):

**>pip install quarchpy .**

If this fails, your path to ‘pip’ may not be set, you can instead run:

**>python –m pip install quarchpy .**

### Upgrade

If you already have QuarchPy installed, you will get a failure message. If you want to upgrade to a new version, you need to add the ‘--upgrade’ command:

**>pip install --upgrade quarchpy**

The --upgrade command can similarly be used in any of the other examples, to load from a local install folder.

## Java install

Check that the Java JRE is installed

You can find install instructions and files here:  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## QPS install

Current versions of QPS are provided as a portable (non-install) .jar file

The latest version can be downloaded from here:  
<https://quarch.com/products/quarch-power-studio>

## Additional library install

The example script makes use of win32 calls in order to identify the Iometer targets

You will need to install wmi and pywin32

**>pip install wmi**

**>pip install pywin32**

## Power module setup

Connect the power module to the test PC and the output of the power module to the drive under test.

Customer Drive

USB/LAN Connection

Power Module

Customer Test PC

# Running the example

You must run the script with **administrator** privileges to work

* When run, you will be prompted to select a target to run on, ensure that you select the correct drive, or the Iometer test may erase important data!

Select the drive that the power module is attached to.

* You will be prompted to select the Quarch power module you want to connect to.
* The script will now launch Power Studio, this may take a few seconds
* The script will iterate through every .conf file in the root /conf folder of the application note folder.

Each .conf file is a renamed Iometer .icf file, with no target information

The script will take each .conf file, add the required target information for each worker and then execute it with Iometer. This means you can create any number of your own .conf files and drop them in the folder to be executed.

* As each Iometer run is executed, the script will:
  + Display the real time power consumption
  + Annotate the start and end of each test
  + Parse the iometer ‘instresults’ file and add the second-by-second performance data onto the chart



Note that by default, the script will process every .conf file in the /conf folder into .icf files and then run every .icf file in order. If you have .conf AND .icf files in the folder, all of them will be run. If you have any corrupt or incorrectly formatted files in the folder, the test may not run correctly.

See the following sections on the correct format for files.

# Creating Iometer Workloads

This example code can take instructions in several forms

* Placing .icf files in the /Conf folder and the script will iterate through them (using the target set in the file)
* Placing .conf files in the /Conf folder and the script will iterate through them, converting them to .icf automatically, filling in the host and target information so they will work in a cross-platform and cross-device way
* Creating a .csv file describing the test parameters to use and calling it from the example code. This will auto-generate a set of iometer workload tests in a simple way, but not all Iometer options are available.

## Generating .icf files

The application note code can run any .icf file BUT there are some critical options in the file that must be set in order for it to process correctly.

.icf files contain target information for both the host name and the device targes, so are not easily portable between tests.

* 1. Open Iometer and create the workload setup you require, selecting the workers, targets and access specifications you want.
  2. On the ‘Results Display’ tab:
     1. Check the box: ‘**Record last update to file’**
     2. Ensure the ‘**Update Frequency**’ is set to 1
  3. On the ‘Test Results’ tab:
     1. Enter a short descriptive **test name** for the test, such as “64k 100% Read”. This will be displayed on the chart and statistics later.
     2. Set a limited run time for the workload in the ‘**Run Time**’ area
  4. Save the file, placing the ones you want to run into the /Conf folder

## Using .conf files

Several example .conf files are supplied in the /conf\_examples folder. Any .conf file placed in the /conf folder will be converted to a .icf file at run time. This involves replacing the access specifications, host and target information. Using .conf files is more portable thatn fixed .icf files, as you can move between host systems or target devices.

A .conf file is a simple IOmeter file, with placeholders for the sections that will be changed. You can edit the existing .conf files, or create new ones:

* 1. Generate a valid .icf file for IOmeter, following the previous instructions
  2. Locate the ‘Manager List’ section here:

**'Manager ID, manager name**

**1,MY-PC-NAME**

**'Manager network address**

Change it to:

**'Manager ID, manager name**

**1,[\*MANAGER\*]**

**'Manager network address**

* 1. Next locate each of the ‘Worker’ sections (normally there are 8).

For each worker, edit:

**'Target assignments**

**'End target assignments**

Change it to:

**'Target assignments**

**'Target**

**[\*TARGET\*]**

**'Target type**

**DISK**

**'End target**

**'End target assignments**

When the script runs, the manager and target information will be set automatically.