NVMeCraft gui User Guide

NVMeCraft gui

Disclaimer of Warranties and Limitation of Liabilities

This manual and the software described in it were prepared by Lalla Soft, Inc., and are subject to change without notice. Lalla Soft, Inc. assumes no liability resulting from any inaccuracy or omissions contained herein or from the use of the information or programs contained herewith.

Lalla soft, Inc. makes no expressed or implied warranty of any kind with regard to these programs or to the supplemental documentation in this manual. In no event shall Lalla Soft, Inc. be liable for incidental or consequential damages in connection with or arising out of the furnishing, performance, or use of the program or documentation. This disclaimer includes, but is not limited to, any loss of service, loss of business, or anticipatory profits, or consequential damages resulting from the use or operation of the enclosed software.

NVMeCraft is a trademark of Lalla Soft, Inc.

Other product names are trademarks of their respective manufacturers.

Getting Started

Welcome to **NVMeCraft gui**.

We hope that using NVMeCraft gui will increase your productivity and enjoyment of NVMe device testing. It was created out of a need to navigate NVMe specification, and to help understand NVMe specification while working and planning.

Key Benefits

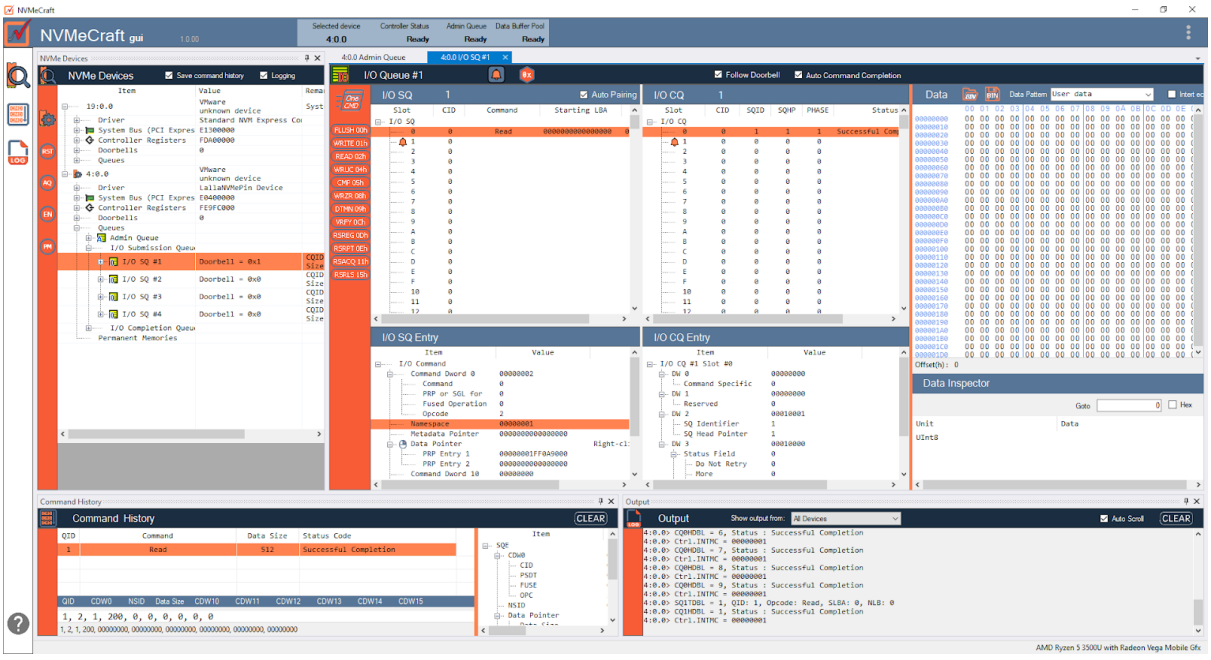
NVMeCraft gui's important benefits are:

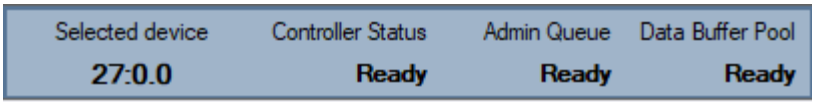
- It helps to understand the behavior of an existing NVMe windows driver.
- You can quickly navigate NVMe specification and NVMe Device behavior.
- You can find a violation of the NVMe specification.
- You can navigate and edit the admin queue, io queue and data.
- You can submit all commands including VU commands.
- It shows live windows inbox driver behavior.
- It has powerful command and data editing features.
- Dynamic information panels work together to create a productive workflow.
- It is used as part of the test set up required for NVMe compliance testing.

User Interface

This chapter describes the main NVMeCraft gui program window and general information about using different windows available in NVMeCraft gui.

In the screenshot below, the main NVMeCraft gui application window contains the NVMe device explorer at the left, an admin/io queue navigator in the middle, and log output window docked to the bottom side.

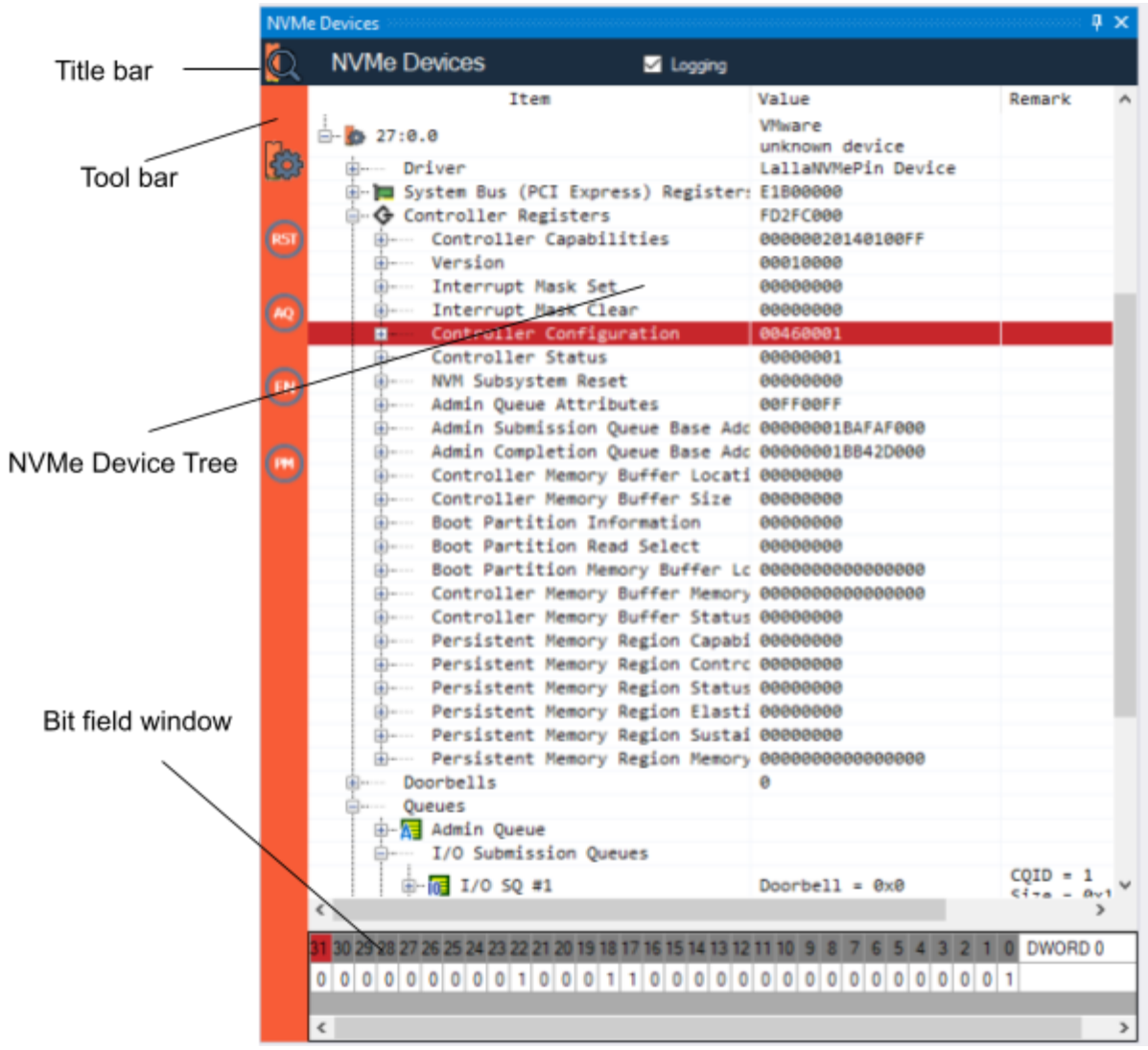




The top status bar shows the current status of NVMeCraft's resources.
[Selected device] is the current device.
[Controller Status] indicates the value of the Ctrl.CSTS.RDY bit.
[Admin Queue] indicates the status of the admin queue pair. It just checks the AQA, ACQ, ASQ register of the controller, and does not guarantee the integrity of the admin queue pair.
[Data Buffer Pool] indicates the status of internal buffer. NVMeCraft gui prepares 32 X 2M bytes memory buffer for internal data operations.

If you want to quickly check the functions, please go to the link.

NVMe Device Explorer



NVMe Device Explorer appears at the left side of NVMeCraft gui. The NVMe Device Explorer lists all the NVMe devices installed in the system to allow easy navigation within each NVMe device, and to provide a quick access of the NVMe device. It can list the NVMe device with its own driver but also Windows inbox driver. The NVMe Device Explorer displays PCI Header, Controller Registers, Doorbells, Admin/IO Queues, and persistent memory that the user grabs. The items with an icon are interactive by double click.

Title bar

The Title bar has the title and logging checkbox. The logging checkbox can allow logging the output message from the user select device. The logging file is named with the following format. [guiLogyyyymmdd.txt] And, saved in the [LogOutput] folder.

Tool bar

The Toolbar contains the basic functions to operate NVMe Device. If the NVMe Device is grabbed by the inbox driver, Toolbar is not available.

 NVMe Device Setup - Initialization to admin queue and io queue.

Reset - Reset NVMe Controller internally CC.EN to 0.

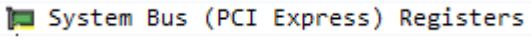
Create Admin Queue Pair - Setup the Admin Queue pair. It is activated when the NVMe Device is not ready.

Enable Controller - Enable the NVMe Controller internally CC.EN to 1.

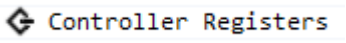
Allocate a persistent memory - Allocate a persistent memory. It can be used in the admin queue, io queue, the Controller Memory Buffer and where the user wants to use.

NVMe Device Tree

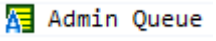
The NVMe Device Tree displays all NVMe device resources. And, If the device is grabbed by its own driver, the user can access the resources by editing value.
The node with an icon is interactive with double click.



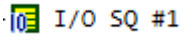
- It displays the hex data of the PCI header.



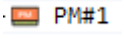
- It displays the hex data of the NVMe controller registers.



- It displays the details of admin queue



- It displays the details of the io queue.

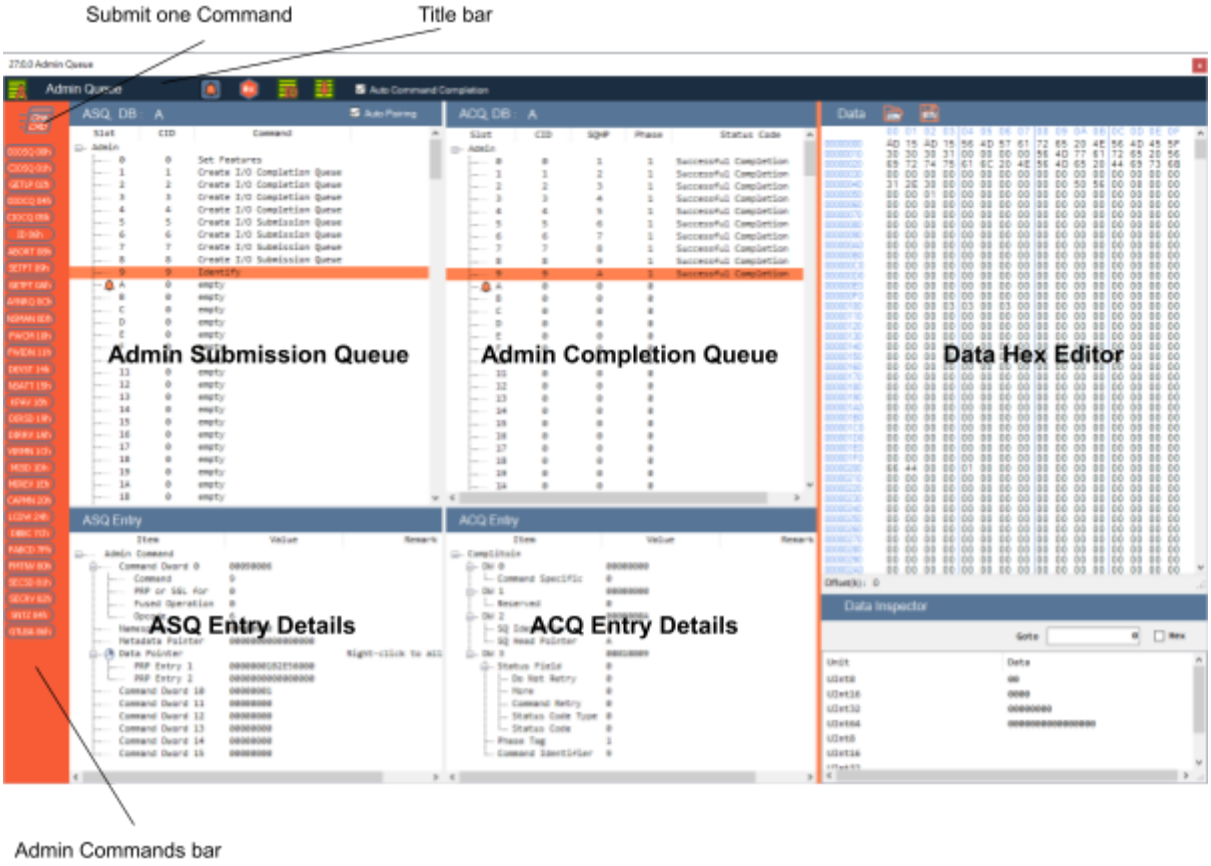


- It displays the hex data of the user allocated memory.

Bit field window

It parses the PCI Header and controller registers and displays the bit-fields.

Admin Queue Explorer



Admin Queue Explorer appears in the middle of NVMeCraft gui. It can be docked to any edge of the main application window. Admin command bar and Data Hex Editor are available with its own driver. It is activated by a double click `Admin Queue` node.

Title bar

The Title bar has the title and some functional buttons.

Goto current doorbell - Move the cursor to the current doorbell slot.

Open/Hide Data Hex Editor - Open and hide the data hex editor that can edit the data of admin commands.

Create IO Queues - It can submit the commands set to create IO Queue pairs.

Scan Admin Queue - It scans the admin submission queue to enumerate the IO Queues it has already submitted. It scans the admin queue from slot 0 to the current doorbell.

Auto Command Completion - NVMeCraft gui can atomically control the NVMe device. It is a complex job and it makes it a little easier.

Admin Command Bar

The Admin Command Bar has the Submit One Command button and several admin command configuration buttons.

Submit a command - It can submit an admin command of the current doorbell.

Several admin command configuration buttons - Users can configure the current admin command following NVMe specification.

Admin Submission Queue and Completion Queue

It displays the admin submission/completion queue in live.

Auto Pairing - According to the CID of the admin command, pairing the completion entry.

ASQ Entry Details

It displays and edits the selected admin command. Users can access all the fields of the entry.

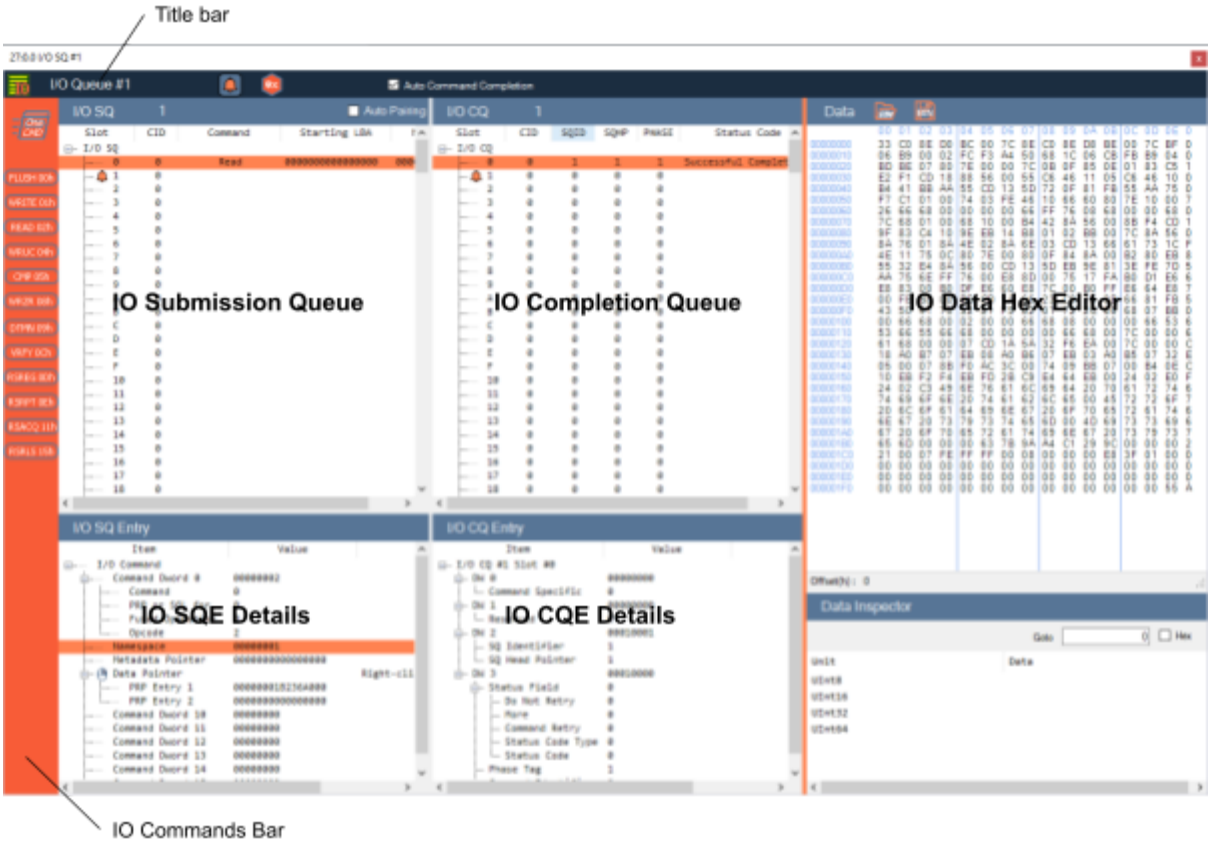
Data Hex Editor

It displays and edits the data of the admin submission entry. It supports only prp data.

Open binary data - Open a binary data and update the prp data of ASQE.

Save binary data - Save the binary data of ASQE to a binary data file.

IO Queue Explorer



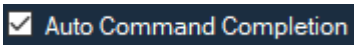
IO Queue Explorer appears in the middle of NVMeCraft gui. It can be docked to any edge of the main application window. The IO command bar and Data Hex Editor are available with its own driver. It is activated by a double click `I/O SQ #1` node.

Title bar

The Title bar has the title and some functional buttons.


Goto current doorbell - Move the cursor to the current doorbell slot.


Open/Hide Data Hex Editor - Open and hide the data hex editor that can edit the data of admin commands



Auto Command Completion - NVMeCraft gui can atomically control the NVMe device. It is a complex job and it makes it a little easier.

IO Command Bar

The IO Command Bar has the Submit One Command button and several IO command configuration buttons.

Submit a command - It can submit an IO command of the current doorbell.

Several IO command configuration buttons - Users can configure the current IO command following NVMe specification.

IO Submission Queue and Completion Queue

It displays the IO submission/completion queue in live.

Auto Pairing - According to the CID of the IO command, pairing the completion entry.

IO Submission Queue Entry Details

It displays and edits the selected IO command. Users can access all the fields of the entry.

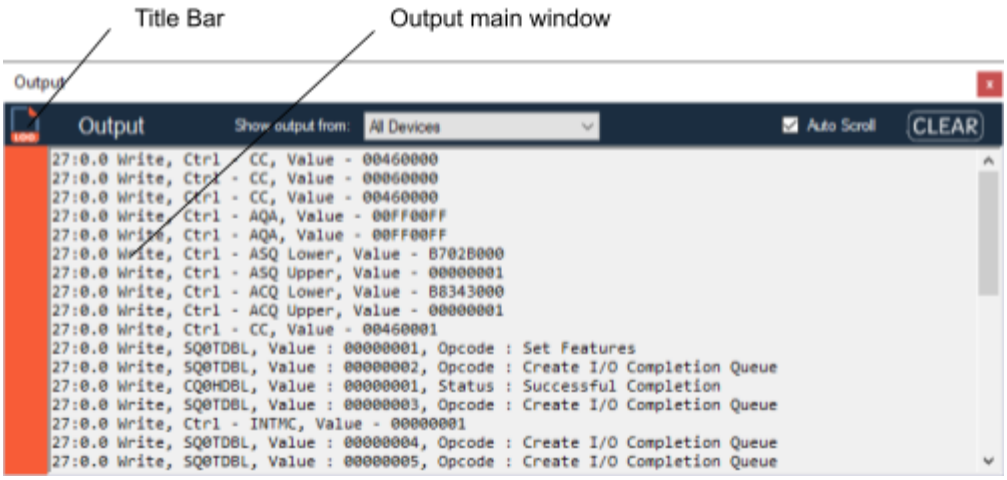
Data Hex Editor

It displays and edits the data of the IO submission entry. It supports only prp data.

Open binary data - Open a binary data and update the prp data of IO SQE.

Save binary data - Save the binary data of IO SQE to a binary data file.

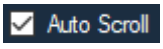
Log Output



Title bar

The Title bar has the title and some functional buttons.

Show output from - User can select the NVMe device which outputs the message.



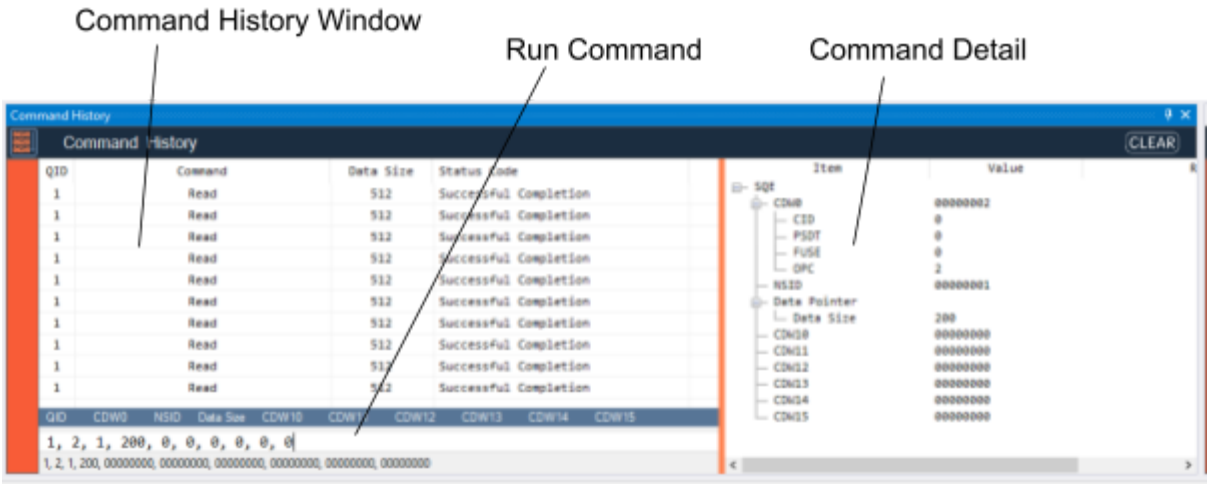
Auto Scroll - Follow the tail message.

Clear - Clear all messages from the window. But, The messages are saved in a log file.

Output main windows

It is limited to 100 lines. You can confirm the full output message with the log file in LogOutput directory.

Command History



Command History Windows

Shows the most recently executed READ/WRITE IO commands.(The record depth is 10.)
The upper item was most recently run. You can select an item to execute a command.

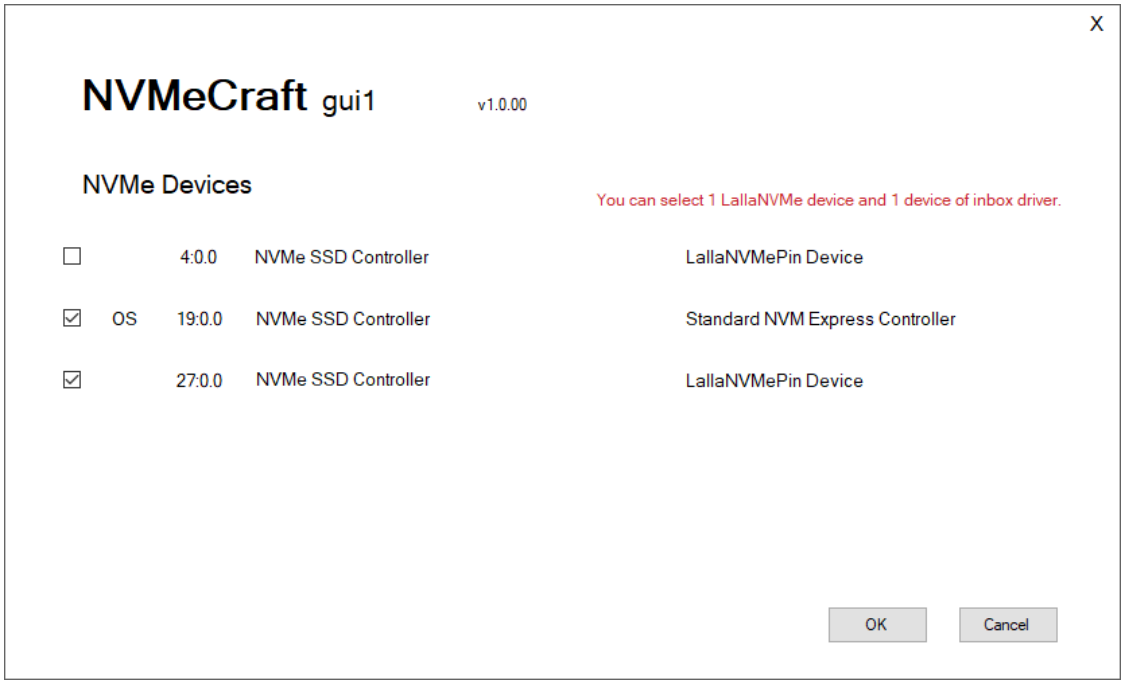
Run Command

Type [ENTER] key in this edit box to submit a command.

Usage Example

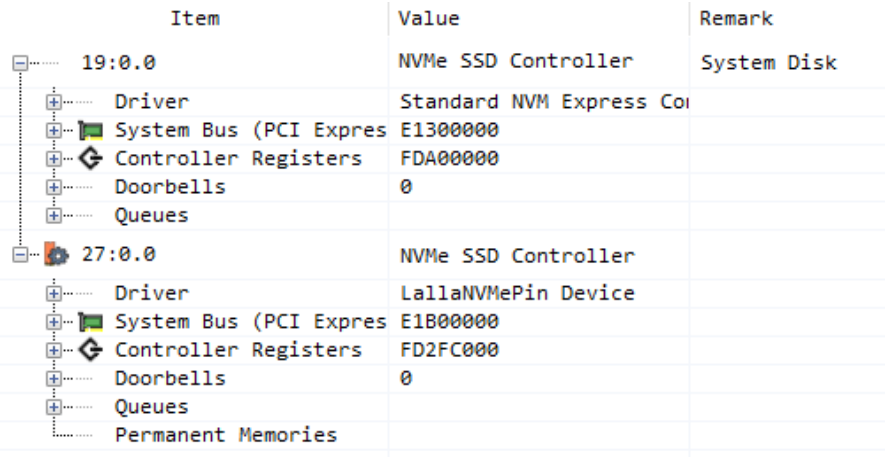
NVMe Device Configuration

There are three NVMe devices in the PC.

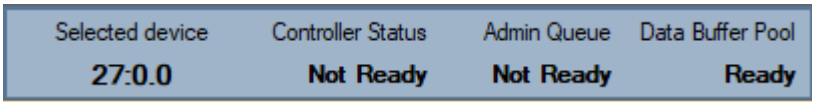


2 LallaNVMePin device - 4:0.0, 27:0.0
1 Inbox driver device including OS - 19:0.0
*** You can get the LallaNVMePin device by updating the driver to LallaNVMePin driver.
Refer to the "Update driver to LallaNVMePin".

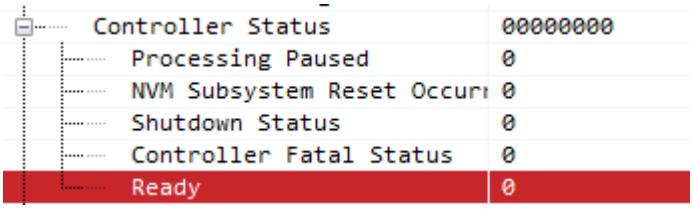
Select the target device you want to test.
I selected two devices, one is an inbox driver device(19:0.0) and the other is LallaNVMe driver device(27:0.0).



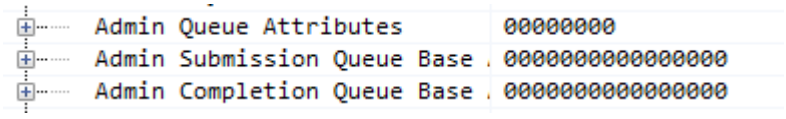
Select the LallaNVMePin device(27:0.0). and check the status of the selected NVMe device.



If [Data Buffer Pool] is not ready, you will need to reboot your PC.



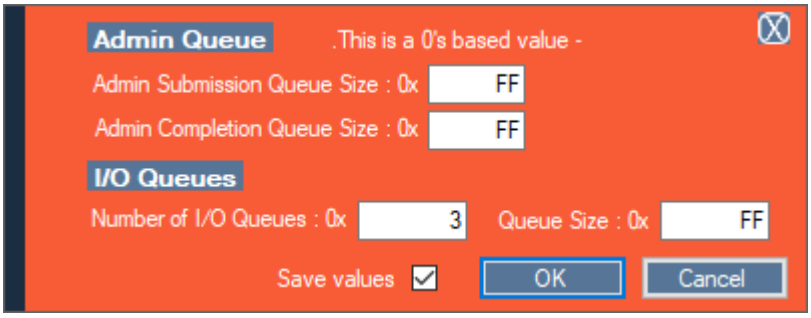
[Controller Status] is not ready because Ctrl.CSTS.RDY is zero.



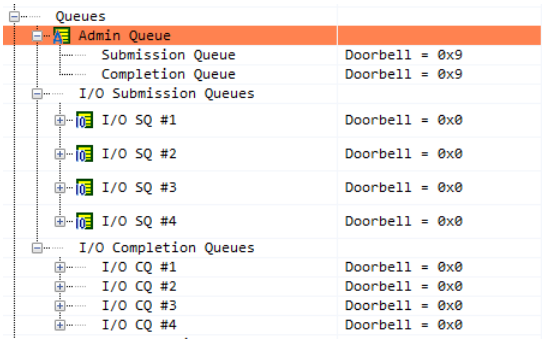
The registers related to the admin queue are zero. And you can see the status of [Admin Queue].

Setup the admin and io Queues

Click the [Setup NVMe Device] button or double click the root node of NVMe device tree to initialize the NVMe device.

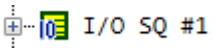


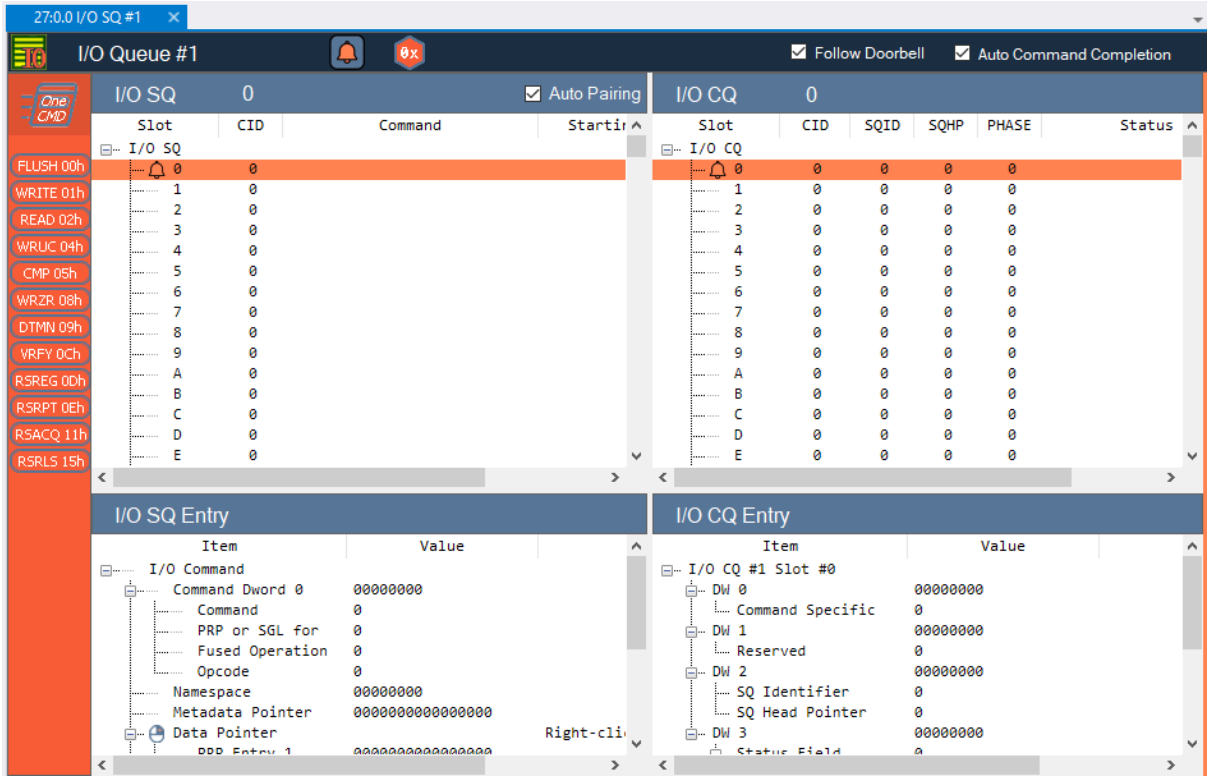
Fill the proper values of the dialog box. refer to the NVMe specification sheet.
And, click [OK] button.



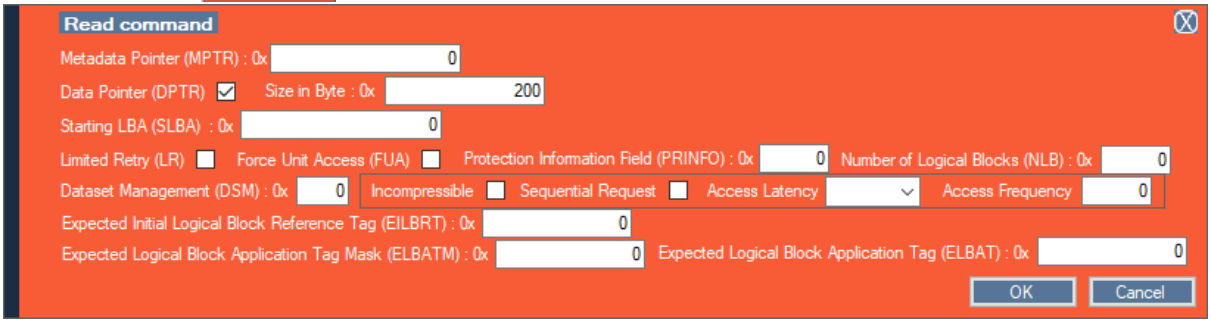
You can confirm the IO Queues you created.

Read data LBA 0, Sector count 1.

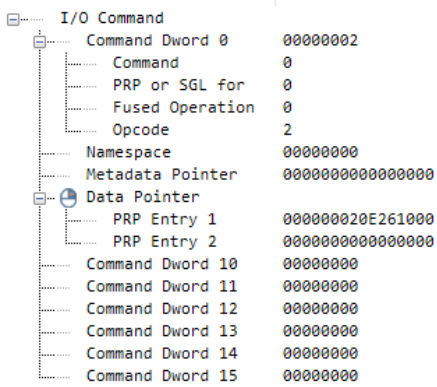
Double click an IO Node to open the IO Queue explorer. 



If you Click the  button, you can see the [Read command] dialog as below.



Fill the proper values, refer to the NVMe specification.
In this case, [Starting LBA] is 0, and the [NLB] is 0.
NLB = 0 means sector count is 1, data byte size is 512(0x200).
Click the [OK] button.

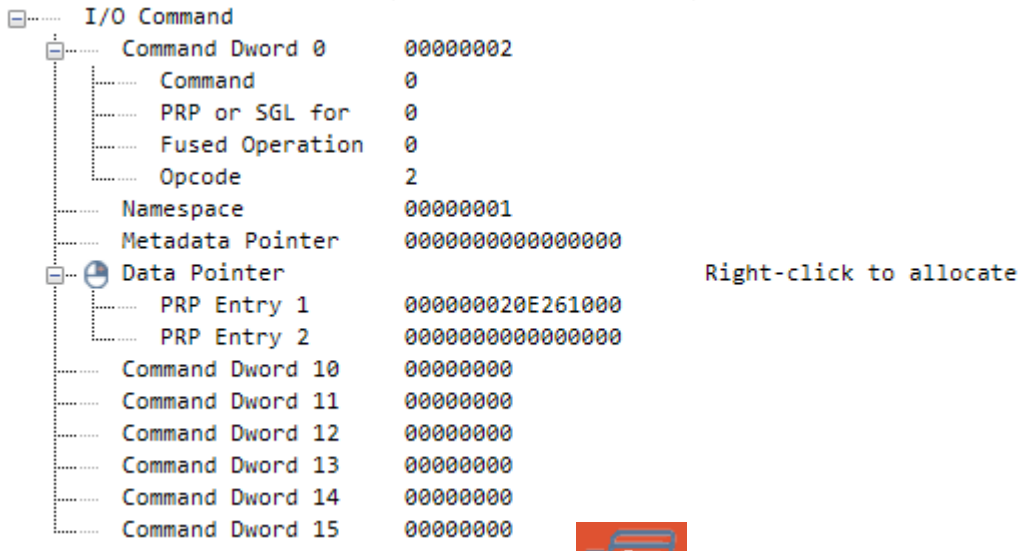


You can confirm the values are filled.

Typically, NVMe SSD has only one namespace.
You need to indicate proper namespace you will use.

On the value field of [Namespace] node 

If you select the value field of [Namespace] and click again, you can update the value.
Type in “1” to use namespace 1.



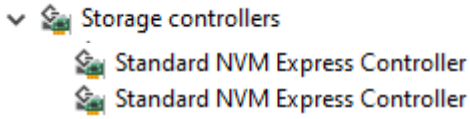
Finally, click the [Submit a command] button. 

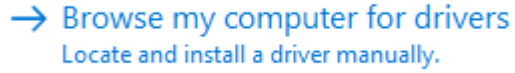
Confirm the complete queue. and the data from the target SSD.

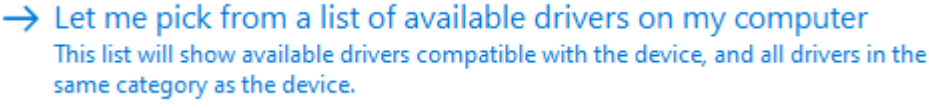
Click  button to check the data.

Update driver to LallaNVMePin

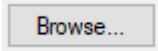
You need to update the NVMe device driver to LallaNVMePin on Windows Device Manager.
You have to make sure the target NVMe device is not a primary disk, it includes an OS.

1. Select target NVMe device.
   [Device Manager]
   

2. Right click and click [Update driver].
3. Click 

   , on popup dialog.
4. Click 

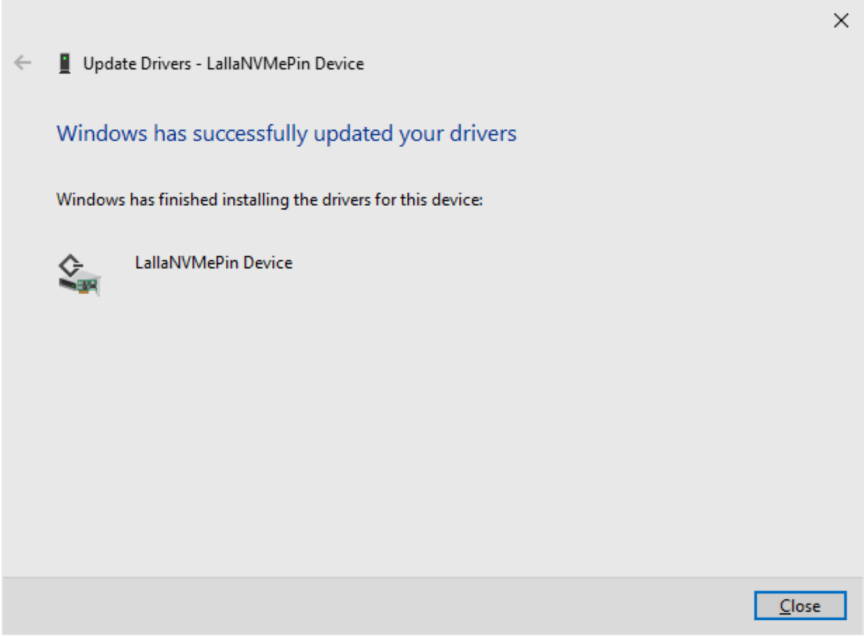
5. Click 

   button, and 

   button.

6. Select [LallaNVMePin.inf] in LallaNVMePin folder, and click [OK].
7. Click Next button.

8. Wait for a second.
9. You can see the shown below dialog, and clic [Close] button.



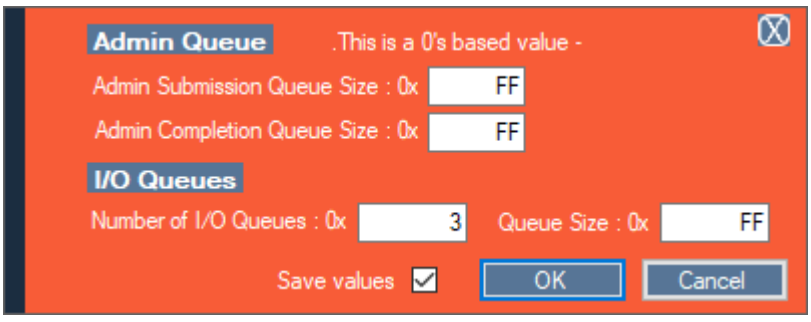
10. Reboot.

Examples of Operations

Setup the admin and io Queues

Select the NVMe device which is grabbed by the Lallasoft own driver

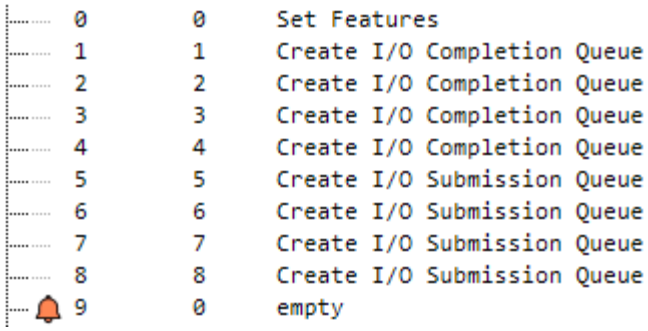
1. Click the  button on the NVMe Device explorer.
2. Fill the values below the dialog. And click [OK].



If the [Save values] is checked, the values are permanently saved as system parameters.

3. Users can confirm the created admin/io queue in the NVMe device tree and the log output.
4. Log output

```
27:0.0> SQ0TDBL = 1, Opcode: Set_Features
27:0.0> CQ0HDBL = 1, Status : Successful Completion
27:0.0> SQ0TDBL = 2, Opcode: Create_IO_Completion_Queue
27:0.0> SQ0TDBL = 3, Opcode: Create_IO_Completion_Queue
27:0.0> SQ0TDBL = 4, Opcode: Create_IO_Completion_Queue
27:0.0> SQ0TDBL = 5, Opcode: Create_IO_Completion_Queue
27:0.0> SQ0TDBL = 6, Opcode: Create_IO_Submission_Queue
27:0.0> SQ0TDBL = 7, Opcode: Create_IO_Submission_Queue
27:0.0> SQ0TDBL = 8, Opcode: Create_IO_Submission_Queue
27:0.0> SQ0TDBL = 9, Opcode: Create_IO_Submission_Queue
27:0.0> CQ0HDBL = 2, Status : Successful Completion
27:0.0> CQ0HDBL = 3, Status : Successful Completion
27:0.0> CQ0HDBL = 4, Status : Successful Completion
27:0.0> CQ0HDBL = 5, Status : Successful Completion
27:0.0> CQ0HDBL = 6, Status : Successful Completion
27:0.0> CQ0HDBL = 7, Status : Successful Completion
27:0.0> CQ0HDBL = 8, Status : Successful Completion
27:0.0> CQ0HDBL = 9, Status : Successful Completion
```

5. 

Manually setup the admin queue

1. Select the NVMe device which is grabbed by the Lallasoft own driver.
2. Reset the NVMe device with writing 0 to Ctrl.CC.EN. And wait until Ctrl.CSTA.RDY to 0.
3. Allocate a persistent memory for ASQ. The size is 0x4000 = 0x100(ASQ size) X 0x40(ASQE size)

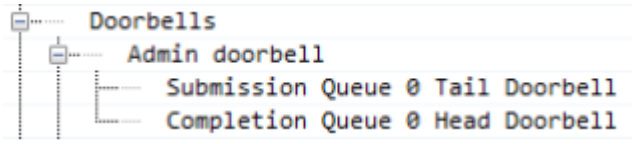
4. Allocate a persistent memory for ACQ. The size of 0x1000 = 0x100(ACQ size) X 0x10(ACQE size)

5. Copy the addresses of ASQ and ACQ to Ctrl.ASQ and Ctrl.ACQ registers.
6. Write the Ctrl.CC.IOCQES to 4. And the Ctrl.CC.IOSQES to 6.
7. Write the Ctrl.AQA.ACQS to 0xFF. And the Ctrl.AQA.ASQS to 0xFF.
8. Finally, Set the Ctrl.CC.EN to 1. And Wait until Ctrl.CSTS.RDY to 1.

⛔ Please pay the careful to allocate the memories, It would be the cause of system crash.

Manually submit an admin command (Identify 06h)

1. Uncheck the [Auto Command Completion] on the title bar of Admin Queue Window.
2. Select the admin submission queue entry of the current doorbell.
3. Write the opcode and CID with correct value in the ASQE panel. (OPC - 0x06)
4. Allocate the prp data for the command with right click on the PRP node of ASQE panel..

5. Ring ASQ doorbell with writing the proper value.
6. Check the completion queue entry from the device. And, write ACQ doorbell with the correct value.
7. 

8. Write Ctrl.IVMC to 1 to clear the interrupt.(right click on the IVMC node, and click [Write])