FEATURE VALIDATION

Using NVM Compare

SUMMARY

This white paper describes how to do a feature validation with SANBlaze software using NVM Compare as an example.

WRITTEN BY

Haiyan Lin

Senior Software Engineer

INTRODUCTION

When testing NVMe SSDs, some customers like to perform complex testing, tweaking multiple variables to create a variety of real-life scenarios that could potentially be replicated out in the field. For these customers, it is important for them to be able to control individual options in order to complete complex testing to validate their SSDs. Conversely, other customers want to run basic NVMe testing to ensure they comply with various specifications. For these customers, the testing process is greatly simplified using SANBlaze's pre-made scripts.

SANBlaze provides a complete system for NVMe testing, including three hardware options: The SANBlaze Express-DT (Desktop), the SANBlaze Express-RM (Rackmount), and the SANBlaze Express-RM4 (Gen4).

Figure 1: SANBlaze SBExpress-DT

Figure 2: SANBlaze SBExpress-RM / RM4

Whether you want to perform complex testing or testing that requires minimal tweaking of parameters and options, it is beneficial to learn and know what is "under the hood" when using SANBlaze. Knowing what goes on behind the scenes of simplified scripting can help you realize the potential of your testing and possibly save you many Qual hours in the process.

This white paper takes you through a SANBlaze validation process using NVM Compare as the example throughout. Understanding this process will enable you to use any SANBlaze NVMe command in the same manner - determining support in your drive for the command, running the command, checking the output, deciding if there needs to be a re-run, and viewing the output in the log files. Understanding what happens "under the hood" is useful to see the power of SANBlaze and how we can help you test your NVMe drives in the fastest and most accurate way possible.

Verify NVM Compare Feature with SANBlaze

The NVMe Compare command is one of the NVM commands. It reads the logical blocks specified by the command from the medium and compares the data read to a comparison data buffer transferred as part of the command.

If the data read from the controller and the comparison data buffer are equivalent with no miscompares, then the command completes successfully. If there is any miscompare, the command completes with an error of Compare Failure.

If metadata is provided, then a comparison is also performed for the metadata, excluding protection information. The command may specify protection information to be checked as well.

Verify NVM Compare Support

Not all SSD controllers support NVMe Compare. Users can find out through the SSD Optional NVM Command Support (ONCS) in the Identify Controller output which indicates the optional NVM commands and features supported by the controller. The bytes 521:520 are for ONCS and bit 0 of ONCS, if set to '1', this indicates the controller supports the Compare command. If cleared to '0', then the controller does not support the Compare command.

Bytes	O/M [1]	Description
521:520	M	**Optional NVM Command Support (ONCS):** This field indicates the optional NVM commands and features supported by the controller. Refer to section 6. Bits 15:8 are reserved. Bit 7 if set to '1', then the controller supports the Verify command. If cleared to '0', then the controller does not support the Verify command. Bit 6 if set to '1', then the controller supports the Timestamp feature. If cleared to '0', then the controller does not support the Timestamp feature. Refer to section 5.21.1.14. Bit 5 if set to '1', then the controller supports reservations. If cleared to '0', then the controller does not support reservations. If the controller supports reservations then the following commands associated with reservations shall be supported: Reservation Report, Reservation Register, Reservation Acquire, and Reservation Release. Refer to section 8.8 for additional requirements. Bit 4 if set to '1', then the controller supports the Save field set to a non-zero value in the Set Features command and the Select field set to a non-zero value in the Get Features command. If cleared to '0', then the controller does not support the Save field set to a non-zero value in the Set Features command and the Select field set to a non-zero value in the Get Features command. Bit 3 if set to '1', then the controller supports the Write Zeroes command. If cleared to '0', then the controller does not support the Write Zeroes command. Bit 2 if set to '1', then the controller supports the Dataset Management command. If cleared to '0', then the controller does not support the Dataset Management command. Bit 1 if set to '1', then the controller supports the Write Uncorrectable command. If cleared to '0', then the controller does not support the Write Uncorrectable command. Bit 0 if set to '1', then the controller supports the Compare command. If cleared to '0', then the controller does not support the Compare command.

Figure 3: Identify Controller Data Structure for ONCS

You can run the Identify Controller command with SANBlaze VLUN SW as follows:

Click the NameSpace you want to run the command on in the left menu, and select tab “Namespace 1 Generic I/O” on the right. Under “Read I/O Type [Block]” select “IdentifyController”, then click “Do Read” in the same row at the most right. Then it will output Identify Controller 4KB data in the middle as follows. All buttons and outputs are highlighted in red below.

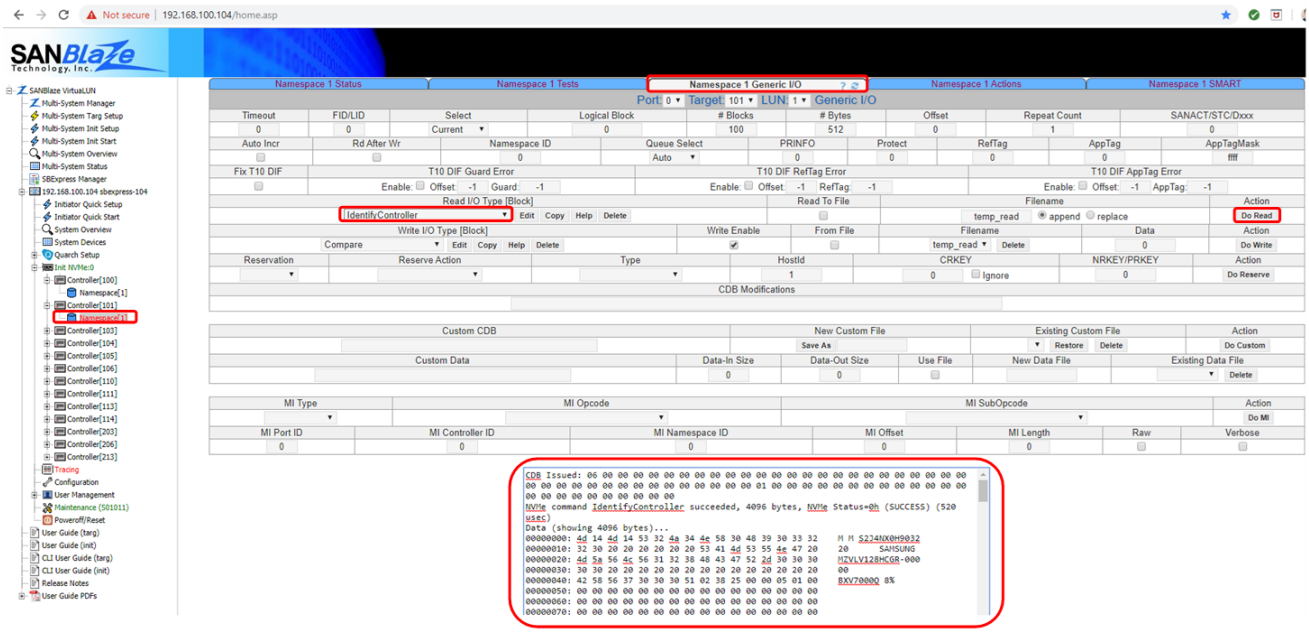

Figure 4: SANBlaze SBExpress-RM NVMe Command Identify Controller Issue

You can scroll the output window down to byte 521:520 (0x209:0x208) and you will see the ONCS = 0x001F as follows. Since bit 0 is set to 1 in ONCS, this means the NVMe controller supports the NVM Compare command.

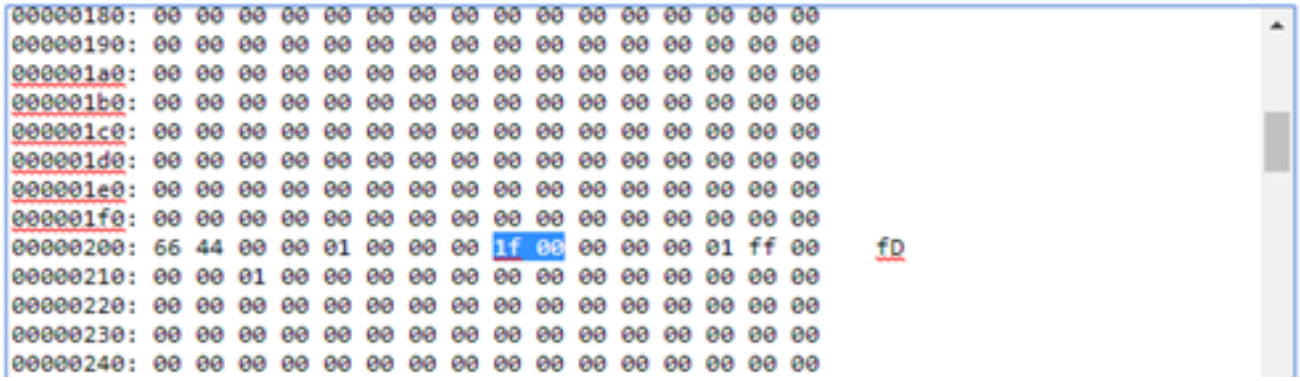

Figure 5: ONCS Value in Identify Controller Output

An upcoming release will add the test results decoding in the output window as follows:

```
VID = 0x144D
 SSVID = 0x144D
 SerialNumber = S2J4NX0H903220
ModelNumber = SAMSUNG MZVLV128HCGR-00000
FirmwareRev = BXV7000Q
RAB = 0x02
IEEE_ID_0 = 0x38
IEEE_ID_1 = 0x25
IEEE_ID_2 = 0x00
CMIC = 0x00
MDTS = 0x05
CNTLID = 0x0001
VER = 0x00000000
RTD3R = 0x00000000
RTD3E = 0x00000000
OAES = 0x00000000
CTRATT = 0x00000000
RRLS = 0x0000
CNTRLTYPE = 0x00
FGUID = 0x00000000000000000000000000000000
CRDT1 = 0x0000
CRDT2 = 0x0000
CRDT3 = 0x0000
NVMSR = 0x00
VWCI = 0x00
MEC = 0x00
OACS = 0x0007
ACL = 0x07
AERL = 0x03
FRMW = 0x06
LPA = 0x01
ELPE = 0x3F
NPSS = 0x04
AVSCC = 0x01
APSTA = 0x01
WCTEMP = 0x0000
CCTEMP = 0x0000
MFTA = 0x0000
HMPRE = 0x00000000
HMMIN = 0x00000000
TNVMCAP =
0x00000000000000000000000000000000
UNVMCAP =
0x00000000000000000000000000000000
EDSTT = 0x0000
FWUG = 0x00
KAS = 0x0000
HCTMA = 0x0000
SANICAP = 0x00000000
HMMINDS = 0x00000000
HMMAXD = 0x0000
NSETIDMAX = 0x0000
ENDGIDMAX = 0x0000
ANATT = 0x00
ANACAP = 0x00
ANAGRPMAX = 0x00000000
NANAGRPID = 0x00000000
PELS = 0x00000000
SQES = 0x66
CQES = 0x44
MAXCMD = 0x0000
NN = 0x00000001
ONCS = 0x001F
FUSES = 0x0000
FNA = 0x00
VWC = 0x01
AWUN = 0x00FF
AWUPF = 0x0000
NVSCC = 0x01
NWPC = 0x00
ACWU = 0x0000
SGLS = 0x00000000
MNAN = 0x00000000
SUBNQN =
0x0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000
IOCCSZ = 0x00000000
IORCSZ = 0x00000000
ICDOFF = 0x0000
FCATT = 0x00
MSDBD = 0x00
PSD00:
MP = 0x0258
MXPS = 0x00
NOPS = 0x00
ENLAT = 0x00000005
EXLAT = 0x00000005
RRT = 0x0
RRL = 0x0
RWT = 0x0
RWL = 0x0
IDLP = 0x0000
IPS = 0x0
ACTP = 0x0000
APW = 0x0
APS = 0x0
PSD01:
MP = 0x01A4
MXPS = 0x00
NOPS = 0x00
ENLAT = 0x0000001E
. . .
```

NVM Compare Feature Verification

In order to verify the SSD NVM Compare feature, we can complete this feature verification with the following 5 steps:

- **Step 1**: Write a range of LBAs with specified pattern.
- **Step 2**: Read the data back from the range of LBAs in step1 and save as one data pattern file.
- **Step 3**: Run NVM Compare command with the data pattern file from step2 for the range of LBAs. We expect it pass since we are using the same pattern for writing and compare.
- **Step 4**: Modify the data pattern file from Step 2 to different pattern bytes and save with the same name.
- **Step 5**: Run NVM Compare command with data pattern file from step4 for the range of LBAs. We expect it to fail since we are using different patterns for writing and compare.

Write Range of LBAs With Specified Pattern

You can write the range of LBAs with specified pattern as follows:

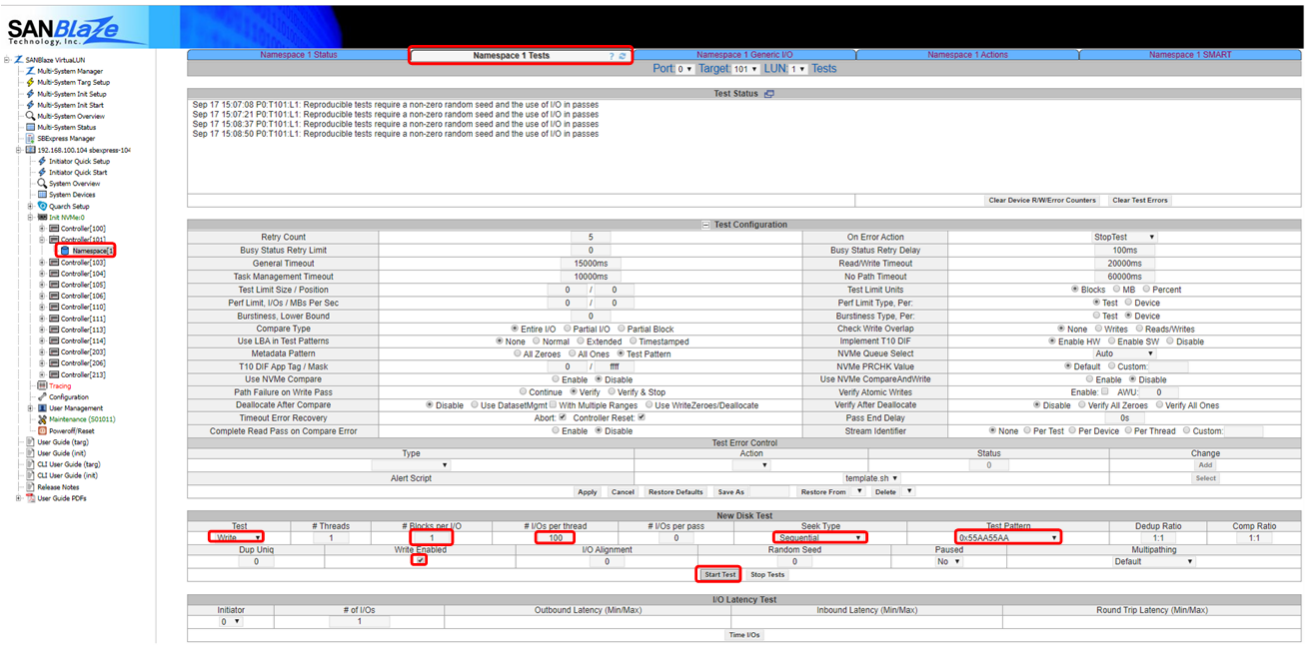

Figure 6: Write Range of LBAs with Specific Data Pattern

Click the NameSpace you want to run the command on in the left menu, then select tab “Namespace 1 Tests” on the right. In the “New Disk Test” group under “Test” select “Write”, enter “# Blocks per I/O” as 1, enter “# I/Os per thread” as 100, select “Seek Type” as “Sequential”, choose the “Test Pattern” you want (I chose 0x55AA55AA” above, check the “Write Enabled” box, then click “Start Test”. It will write LBA 0 to 99 with the specified pattern. All buttons/inputs are highlighted in red above.

You will see the writing passed as follows in the top of the same window.

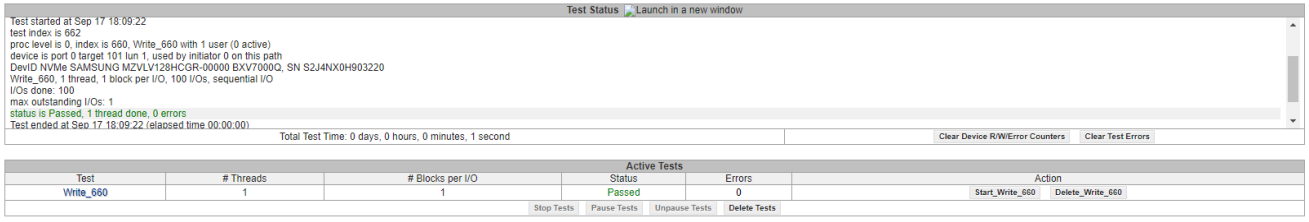

Figure 7: Command Output of Writing Range of LBAs

Read Back the Written Data Pattern

You can read back the written data pattern and save to one file as follows.

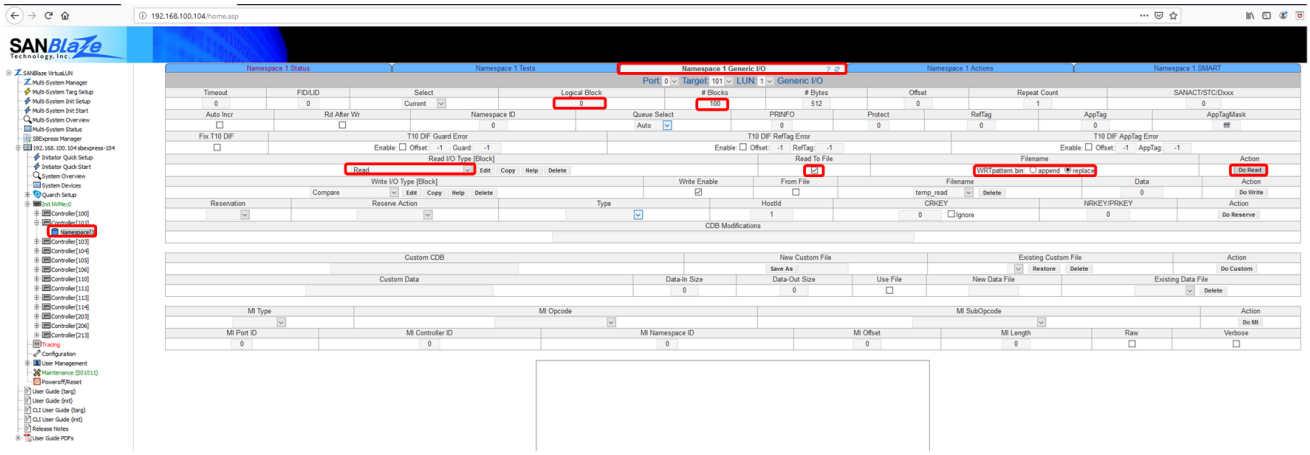

Figure 8: Read Back the Written Data Pattern and Save to File

Click the NameSpace you want to run the command on in the left menu, select tab “Namespace 1 Generic I/O” on the right. Under “Read I/O Type [Block]” select “Read”, enter “Logical Block” (start LBA) as 0, enter the “# Blocks” as 100, check the “Read To File”, Enter “Filename” as “WRTpattern.bin” and check “replace”, then click “Do Read” in the same row at the most right. Then it will output the read status and saved data pattern file location in the text box of the same window as follows. All buttons and outputs are highlighted in red above.

```
CDB Issued: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 00 00 00 00 00 00 00 00 00
00 00 00 00 ff ff
NVMe command Read succeeded, 51200 bytes, NVMe Status=0h (SUCCESS) (403 usec)
Data written to file /virtualun/lundata/initiator/WRTpattern.bin
```

Figure 9: Read Command Status and Saved Data Pattern File path

You can see the written data pattern now read back and save as file */virtualun/lundata/initiator/ WRTpattern.bin.*

Run NVM Compare with Specified Data Pattern

Now you can run the NVM Compare command with the data pattern read back in the section above. Click the NameSpace you want to run on the left menu, select tab “Namespace 1 Generic I/O” on the right. Enter “Logical Block” (start LBA) as 0, enter the “# Blocks” as 100. Under “Write I/O Type [Block]” select “Compare”, check the “Write Enable” and “From File”, select “Filename” as WRTpattern.bin, then click “Do Write” the same row at the most right. Then it will run the NVM Compare command for LBA 0 to 99 against the specified data pattern WRTpattern.bin. All buttons are highlighted in red below.

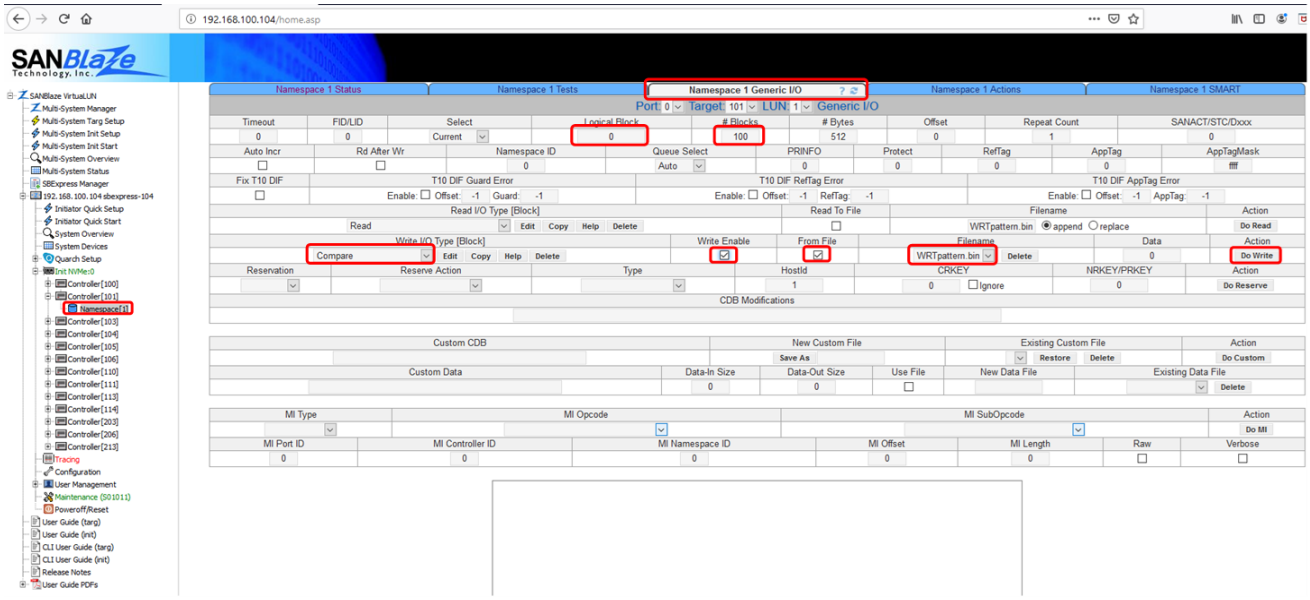

Figure 10: Run NVM Compare with Specified Data Pattern

```
CDB Issued: 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 00 00 00 00 00 00 00 00 00
00 00 ff ff 00 00
NVMe command Compare succeeded, 51200 bytes, NVMe Status=0h (SUCCESS) (3815 usec)
```

Figure 11: NVM Compare Command Status

Modify NVM Compare Data Pattern to Verify Failure Case

You can modify the data pattern used for NVM Compare to verify the failure case of this command. In section "Read Back the Written Data Pattern" you can see the data pattern is saved in file /virtualun/lundata/initiator /WRTpattern.bin. You can modify it on the tester with vim or other editors. Please note this data pattern file is a binary file, and you may need to use ":%!xxd" for vim binary view and modification. For example, I modified the first byte from 0x55aa to 0x66aa in this data pattern file below and saved it:

```
[root@sbexpress-104 ~]# cd /virtualun/lundata/initiator/
[root@sbexpress-104 initiator]# ls -lrt
total 280
lrwxrwxrwx 1 root root      9 Mar 16  2015 zero -> /dev/zero
lrwxrwxrwx 1 root root     12 Mar 16  2015 random -> /dev/urandom
-rw-rw-r-- 1 root root  51200 Sep 17 23:23 WRTpattern.bin
[root@sbexpress-104 initiator]# vim WRTpattern.bin
```

```
root@sbexpress-104:/virtualun/lundata/initiator
0000000: 66aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000010: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000020: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000030: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000040: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000050: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000060: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
0000070: 55aa 55aa 55aa 55aa 55aa 55aa 55aa 55aa  U.U.U.U.U.U.U.
```

Figure 12: Data Pattern Modification

Rerun NVM Compare with Modified Data Pattern to Verify Failure Case

Now you can run the NVM Compare command with the modified data pattern in section "Modify NVM Compare Data Pattern to Verify Failure Case" above. Click the NameSpace you want to run the command on in the left menu, select tab “Namespace 1 Generic I/O” on the right. Enter “Logical Block” (start LBA) as 0, enter the “# Blocks” as 100. Under “Write I/O Type [Block]” select “Compare”, check the “Write Enable” and “From File”, select “Filename” as WRTpattern.bin, then click “Do Write” the same row at the most right. Then it will run the NVM Compare command for LBA 0 to 99 against the modified data pattern “WRTpattern.bin”. All buttons are highlighted in red below.

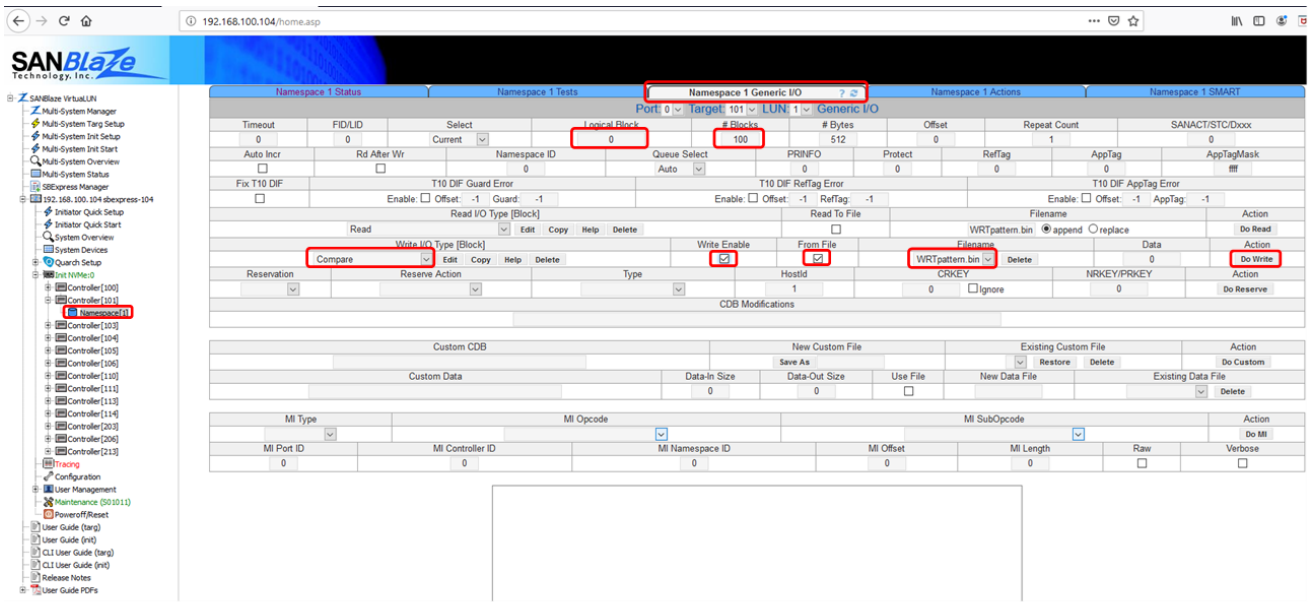

Figure 13: Rerun NVM Compare with Modified Data Pattern

After clicking the “Do Write” button it will output the NVM Compare command status in the text box of the same window as follows. You can see it failed the NVM Compare command as expected because the data pattern has been changed.

```
CDB Issued: 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 00 00 00 00 00 00 00 00 00
00 00 ff ff 00 00
NVMe command Compare failed, NVMe Status=2285h (COMPARE_FAILED, M=1, SCT=2h, SC=85h) (706
usec)

Compare failed on device P0:T101:L1
```

Figure 14: Verify Failure Case - Outputn

The SCT=2h means Status Code Type = 2 and it is the Media and Data Integrity Error category. Any media specific errors that occur in the NVM or data integrity type errors shall be of this type. SC=85h means Status Code = 0x85 and it is Compare Failure. The command failed due to a miscompare during a Compare command as below:

Value	Description
84h	**End-to-end Reference Tag Check Error:** The command was aborted due to an end-to-end reference tag check failure.
85h	**Compare Failure:** The command failed due to a miscompare during a Compare command.
86h	**Access Denied:** Access to the namespace and/or LBA range is denied due to lack of access rights. Refer to the appropriate security specification (e.g., TCG Storage Interface Interactions specification).
87h	**Deallocated or Unwritten Logical Block:** The command failed due to an attempt to read from or verify an LBA range containing a deallocated or unwritten logical block.
88h to BFh	Reserved

Figure 15: Status Code – Media and Data Integrity Error Values, NVM Command Set

Run NVM Compare on SSD Not Supporting This Feature

If you tried to run the NVM Compare command on SSDs that don't support this feature, it will fail with "Invalid Opcode". For example, I have one drive not supporting NVM Compare and its ONCS = 0x4 as follows from Identify Controller output:

```
00000180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200: 66 44 00 00 01 00 00 00 04 00 00 00 06 00 ff ff    fD
00000210: 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
```

. . .

MAXCMD =0x0000
NN = 0x00000001
ONCS = 0x0004
FUSES = 0x0000
FNA = 0x06

. . .

Figure 16: Identify Controller Output from SSD Not Supporting NVM Compare

When you run the NVM Compare command it will display “INVALID_OPCODE” as below.

```
CDB Issued: 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 00 00 00 00 00 00 00 00 00
00 00 ff ff 00 00
NVMe command Compare failed, NVMe Status=1h (INVALID_OPCODE, SCT=0h, SC=1h) (109 usec)

Compare failed on device P0:T100:L1
```

Figure 17: Command Status When Run NVM Compare on SSD Not Supporting This Feature

Conclusion

Now that you have seen how to use the NVM Compare command manually, you can always use this same process using any NVM command that SANBlaze supports.

Alternatively, you can use the SANBlaze Certified Automated Testing Scripts (or "canned scripts") provided within the software to eliminate the above steps and to run your tests quickly. Either way, you will have everything you need in order to validate and test your NVM SSDs quickly and accurately so that you can get them into your customers' hands without delay.

SANBlaze

NVMe over Everything™

Over 450 Tests!!

Let us certify your NVMe drives *before* they go to market so you can rest easy that your customers are taken care of.

What Do We Test?

NVMe Generic I/O Commands
NVMe I/O Tests
NVMe Resets (all supported reset methods)
NVMe Namespace Management
NVMe Basic Management Commands
NVMe-MI Full Command Set
NVMe Dual Port Drive Tests
NVMe SBExpress Hotplug and Link Testing
NVMe Quarch Testing Pull/Plug Glitch
NVMe Miscellaneous Commands (e.g. SR-IOV)
IOL NVMe Certification
IOL NVMe-MI Certification
SSD Endurance JEDEC Spec. (long runtime)

Certified by SANBlaze™

Tel.: (978) 679-1400
sales@sanblaze.com
support@sanblaze.com

Get Started Today!

For more information please contact:

SANBlaze, Inc.
One Monarch Drive, Suite 204
Littleton, MA 01460
www.sanblaze.com
sales@sanblaze.com | support@sanblaze.com