LEGAL NOTICE:

NVM Express Workgroup
c/o VTM, Inc.
3855 SW 153rd Drive
Beaverton, OR 97003
USA
info@nvmexpress.org

NVM Express® Technical Proposal (TP)

Technical Proposal ID	**TP4126 – NVMe-oF Boot HostNQN and HostID**
Revision Date	**2023-01-15**
Builds on Specification(s)	**NVM Express Boot Specification**
References	**TP 8012, TP 4110**

Technical Proposal Author(s)

Name	**Company**
James Smart	Broadcom
Craig Carlson	Marvell
Hannes Reinecke	SUSE

Technical Proposal Overview

This technical proposal provides a recommendation for default HostNQN and HostID values, in the absence of administratively-set values, that can be used in boot scenarios by both boot drivers and OS environments. The proposal results in a consistent behavior regardless of OS, boot device, NVMe-oF transport, and installation occurrence. The result is an easier system administration of NVMe-oF Boot Solutions

Revision History

Revision Date	**Change Description**
2021-11-03	Initial version
2022-01-11	Phase 2 draft
2022-01-27	Minor Clarifications/Whitespace from Phase 2 exit review
2022-05-27	Phase 3 changes: Modifications to merge into NVM Express Boot Specification (TP8012) Phase 3 doc.
2022-05-31	Correct review comments from review of Phase 3 document
2022-06-03	Minor revisions for review comments
2022-08-02	Migrated to TP 4.13a Template. Sync with latest 8012 section 3 text. Revisions based on 30-day Phase 3 Member Review
2023-01-15	Integrated
2023-01-25	Minor revision for preratification comments

Description for Changes Document for NVM Express Boot Specification

Markup Conventions:

Black:	Unchanged (however, hot links are removed)
~~Red Strikethrough~~:	Deleted
Blue:	New
Blue Highlighted:	TBD values, anchors, and links to be inserted in new text.
<Green Bracketed>:	Notes to editor

Description of Specification Changes for NVM Express Boot Specification

1.3 Definitions

…

1.3.x Expansion ROM Firmware

Peripheral Component Interconnect (PCI) term for firmware executed on a host processor which is used by an add-in device during the boot process. This includes Option ROM Firmware and UEFI drivers. Expansion ROM Firmware may be embedded as part of the Host Processor Boot Firmware, or may be separate (e.g., from an add-in card).

…

1.3.x Host Identifier (HostID)

The Host Identifier uniquely identifies the host associated with the controller within the NVM subsystem as defined in the NVM Express Base Specification.

1.3.x Host NVMe Qualified Name (HostNQN)

The Host NVMe Qualified Name (NQN) uniquely identifies the host as defined in the NVM Express Base Specification.

…

1.4 References

…

TCG Storage Interface Interactions Specification (SIIS). Available from https://www.trustedcomputinggroup.org.

System Management BIOS (SMBIOS) Reference Specification (DSP0134). Available from https://www.dmtf.org.

…

3 Boot Mechanisms

This section specifies different mechanisms and their requirements that may be used for booting over NVMe-oF.

3.TBD NVMe-oF Boot default values for Host NQN and HostID

During the boot process of an NVMe-oF platform, there are at least two separate NVMe host environments which must coordinate access to a NVMe-oF boot device.

A pre-OS boot environment, such as UEFI, utilizes environment features and Expansion ROM Firmware to communicate with a namespace containing the OS image. The pre-OS boot environment loads components of the OS from the namespace into platform memory and then transitions the platform to the OS environment.

The OS environment takes over ownership of platform hardware including the hardware that was used to “boot from” in the pre-OS environment. The OS environment, using its own device drivers and software, access the same namespace used by the pre-OS boot environment, but may do so with a different NVMe-oF configuration than that used by the pre-OS environment.

When an NVMe-oF connected namespace is used as a “boot” device, both the pre-OS boot environment and the OS environment create independent associations with the NVM subsystem. Key components of association creation are the HostNQN and HostID values that are passed in the Fabrics Connect command.

The HostNQN and HostID values identify the platform to the NVM subsystem and may be used by the NVM subsystem as the basis for access control lists.

Given the independence of the pre-OS boot environment from the OS environment it is very probable that both environments may use different host identification values. Additionally, in pre-OS boot environments, the associations may be created by independent Expansion ROM Firmware images, resulting in multiple and different host identifiers depending on which hardware entity is used to communicate with the NVM subsystem. OS re-installation, whether the same or different OS, may also change host identifiers for the OS environment. Each of these attributes create a situation that requires administrative intervention to configure the NVM subsystem with the various host identifiers used by the pre-OS and OS environment. In some cases, knowledge of these values may not be known prior to OS installation, resulting in difficult steps (including interrupting OS installation) to coordinate the settings in the necessary locations in order to "install and boot".

In order to lessen this administrative overhead, the pre-OS boot environment and the OS environment should utilize the same host identifiers for the NVMe-oF associations used for the "boot" device. This may be achieved by explicit administrative actions, perhaps using vendor-specific methods such as UEFI HII pages. However, if administrative assignment does not exist or only partially exists, pre-OS boot environments and OS environments should utilize a known default behavior, based on a platform identifier, to construct a HostNQN and HostID such that both environments results in the use of the same HostNQN and HostID values.

The pre-OS boot environment and OS environment should use a fixed platform UUID to create a HostNQN and HostID. The implementation should use the System UUID found in the SMBIOS table. The System Management BIOS (SMBIOS) Reference Specification is described in DSP0134. The SMBIOS table is typically available to pre-OS firmware and Expansion ROM Firmware in the pre-OS boot environment as well as to the OS environment. The system UUID is a value that is likely to be fixed at manufacturing or virtual machine creation, perhaps displayed on system tags or via configuration menus, and visible to the system administrator ahead of any OS installation. These attributes allow the administrator to pre-generate the expected HostNQN and HostID values that will be used by the pre-OS boot environment and the OS environment. Apriori understanding of the values allow the administrator to configure the NVM subsystem in the least number of steps and before there is any attempt to boot or install the OS.

3.TBD.1 Creating HostNQN and HostID Using SMBIOS System UUID

The SMBIOS System UUID, found in the "System Information (Type 1)" structure at offset 8h, is a 16-byte value. The System UUID is composed of six fields as described in DSP0134, System Information (Type 1) System – UUID section, and illustrated in Figure TBD.

Figure TBD: System UUID Value Table

Bytes	Description	Value
3:0	Time Low	00112233h (stored little endian)
5:4	Time Mid	4455h (stored little endian)
7:6	Time Hi and versionb	6677h (stored little endian)
8	clock_seq_hi_and_rsvd	88h
9	clock_seq_l	99h
15:10	Node	Bytestream of AAh BBh CCh DDh EEh FFh

The following summary describes internal storage, interpretation, and display methodology for a System UUID value stored the table:

If the UUID is viewed as a bytestream (sequentially increasing bytes) starting at offset 8h in the System Information table, it would be seen as:

33h 22h 11h 00h 55h 44h 77h 66h 88h 99h AAh BBh CCh DDh EEh FFh

If the UUID is viewed as a string, it would be seen as:

UUID is "00112233-4455-6677-8899-aabbccddeeff"

If all 16 bytes of the System UUID are set to FFh or cleared to 0h, as specified by the System Management BIOS (SMBIOS) Reference Specification (DSP0134), the System UUID field is not present in the platform and the recommended behavior cannot be supported. The system should fall back to platform NVMe host transport device-specific methods for the pre-OS and OS environments.

If the System UUID is present, the pre-OS and OS environments are to perform the following:

Generation of the HostNQN value:

The System UUID is converted into a NQN per the NVM Express Base Specification, NVMe Qualified Names section, using the second format which creates a unique identifier without a naming authority and using a UUID.

For example, the System UUID "00112233-4455-6677-8899-aabbccddeeff" results in the NQN string:

"nqn.2014-08.org.nvmexpress:uuid:00112233-4455-6677-8899-aabbccddeeff"

Generation of the Host Identifier (HostID) value:

The pre-OS or OS environment is to choose one of the following methods for the Host Identifier:

a) HostID cleared to 0h. If this method is chosen, the Pre-OS or OS environment must be prepared for a Fabrics Connect command to fail with status code set to Connect Invalid Parameters (82h). NVM Express Base Specification revision 2.0 and earlier did not allow a HostID value of 0h and required devices to fail the Fabrics Connect command with the Connect Invalid Parameters status code. If the Fabrics Connect command fails with status code set to Connect Invalid Parameters, the pre-OS or OS environment should:
- terminate the attempt to create a host to controller association that has the HostID cleared to 0h;
- use option (b) for the HostID; and
- attempt to create a new host to controller association with the HostID set to the UUID;

or

b) HostID set to the System UUID bytes. The HostID and the System UUID are to be treated as bytestreams with each byte copied sequentially from the System UUID bytestream to the HostID bytestream.
For example, if the System UUID bytestream is seen as:
33h 22h 11h 00h 55h 44h 77h 66h 88h 99h AAh BBh CCh DDh EEh FFh
the HostID bytestream would also be seen as:
33h 22h 11h 00h 55h 44h 77h 66h 88h 99h AAh BBh CCh DDh EEh FFh