NVM Express Technical Proposal for New Feature

Technical Proposal ID	TP8010a
Change Date	2022-05-09
Builds on Specification	**NVM Express Base Specification revision 2.0a** **NVM Express TCP Transport Specification revision 1.0a** **NVM Express Management Interface Specification 1.2b**
References	**TP8009 Automated Discovery of NVMe-oF Discovery Controllers for IP Networks** **TP8014 Discovery Controller Discovery** **TP8013 Unique Discovery Controller ID** **TP4109 Allow Host to Specify Telemetry Host-Initiated Data Areas**

Technical Proposal Author(s)

Name	Company
Babu Puttagunta, Curtis Ballard, Rupin Mohan, Matt Goepfert	HPE
Erik Smith, Claudio Desanti	Dell Technologies
Mike Allison	Samsung

This technical proposal defines new requirements and capabilities that enable discovery information (e.g., Discovery log pages) from multiple Direct Discovery controllers to be consolidated into and retrievable from a single Centralized Discovery controller. The method hosts and Direct Discovery controllers use for discovering Centralized Discovery controllers is described in TP 8009. This proposal is only applicable for IP-based fabric transports.

Revision History

Revision Date	Change Description
2020-09-28	Initial version ("skeleton" document)
2020-11-30	Push registration content added
2020-12-07	• Added existing definition for Discovery controller • Re-wording for sections 5.NEW.1, 5.NEW.1.1, and 5.NEW.1.1.1 • Added in comments from Dell

2020-12-14	- Tweaked CDC and discovery information definitions - Changed the name of the push registration command from "Discovery Information Registration" to "Discovery Information Management" - Added TAS field to Discovery Information Management command SQE so same command can be used to reg/de-reg - Removed GENCTR field from payload header - Moved TDL field up to newly vacant bytes 00 – 03 in payload header - Made EID field mandatory - Added "overwrite model" text for push registrations - Keeping basic discovery information entry for push registration (useful for DDCs) - Swapped TEL and NUMEXAT fields for logical consistency with payload header field ordering - Added new figure to show which fields in an extended discovery information entry are applicable for a given entry type (i.e., host vs. NVM subsystem)
2021-01-15	- Incorporated feedback from 1/12 FMDS discussion - Split column in Figure NEW.6 into separate host and NVM subsystem columns - Got rid of extraneous use of the word "push" in section 5.NEW.1.1.1 - Added name of fields from existing DLPE being referenced in Figure NEW.4 - Incorporated LBA Range Type style text for "overwrite model" text in 5.NEW.1.1.1 - Added distinction between I/O controller vs. discovery controller NVM subsystem discovery information entries in section 5.NEW.1.1.1 - Also added distinction between referral records vs. non-referral records and how CDCs should handle them in section 5.NEW.1.1.1
2021-01-22	- Added new field to NVMe-oF specific Identify Controller data structure (Figure 30) to indicate DDC or CDC - Made referral record content in section 5.NEW.1.1.1 open-ended (removed previous requirements) - Added dword alignment requirements for extended attributes - Added vendor specific opcode recommendation (E1h) for Discovery Information Management command - Added pull registration content: - Created sections 5.NEW.1.2 (overview), 5.NEW.1.2.1 (mDNS), 5.NEW.1.2.2 (kickstart) - Modified Figure 56 (added KDReq/KDResp definitions) - Modified Figure 61 (added KDReq/KDResp commands) - Created sections 7.4.10.NEW1 (KDReq command), 7.4.10.NEW2 (KDResp command)
2021-01-28	- Made new DCTYPE field in NVMe-oF specific Identify Controller data structure a coded value - Corrected wording in EXATLEN field in Extended Attribute figure - Cleaned up/re-organized pull registration sections so information related to the CDC performing the pull registration is contained in 5.NEW.1.2, and information related to requesting a pull registration is contained in appropriate sub-sections (mDNS or Kickstart) - Removed requirements that NVMe/TCP needs used as the transport type after Kickstart when the CDC connects to the DDC to do the pull
2021-02-08	- Defined LSP field for LID 70h (added "Link Local Entries Only" bit) and added new byte in LID 70h header to show if only link-local entries are being returned - Fixed kickstart ladder diagram - Changed KDReq PDU to account for 1…N kickstart records - Added Dell comments

2021-03-08	• Changed "link local" to "port local" for DLP • Modified text and figure in section 5.NEW.1.2.2 (Kickstart) • Added modification to section 7.4.4 related to kickstart • Added modification to Figure 73 (ICReq) for use of FLAGS field related to kickstart
2021-03-11	• Fixed kickstart discovery related content based off 3/9 FMDS feedback • Added in pull de-registration content • Added Port Local field to Discovery Information Management data portion header
2021-03-18	• "full function NVMe/TCP connection" changed to "non-kickstart discovery NVMe/TCP connection" • Fixed text in section 7.4.4 based upon 3/16 FMDS feedback • The Port Local Entries Only bit in the LSP field for LID 70h was missing an acronym, so now it's Port Local Entries Only (PLEO) • Explicitly stated for pull reg and pull de-reg that the CDC can set the PLEO bit in Get Log Page to request only port local entries be returned
2021-03-24	• Removed host-related content from PORTLCL field in DIM command data portion's header and fixed wording. Port local is only relevant for NVM subsystem discovery information entries from a DDC • Added reference to PORTLCL field in DIM command in section 5.NEW.1.1.1 • Fixed wording in PORTLCL field in DLP header to match wording changes that were made to PLEO bit in LSP field • Added CDCNQN field to KDResp • Added reference to CDCNQN field in section 5.NEW.1.2.2 • Added text to section 5.NEW.1.2 explaining how CDCNQN field is used by DDC to know CDC is performing pull registration and thus all available discovery information needs to be returned to the CDC (no filtering)

2021-04-16	• Made changes to PLEO bit in LSP field, PORTLCL field in DLP, and PORTLCL field in DIM command using Erik's proposed text
	    • Similar wording changes needed in sections 5.3, 5.NEW.1.1, and 5.NEW.1.2
	• Made small changes based off old Dell comments:
	    • Slight rewording in CDC and DDC definitions
	    • Moved "overwrite model" text for DIM command to be next to de-registration text
	    • Changed two instances of "are able to" to "may" (in section 5.NEW.1.2)
	• Changed the single instance of "Discovery Service" to "Discovery subsystem" (in section 5.NEW.1.1.1) to align with TP 8013 discussions
	• Added ENTUNRCH field to indicate if an entity unreachable or not
	• Added "immediate" de-registration capability to KDReq/KDResp

2021-05-03	• Restructured doc to be based off NVMe Base Specification 2.0 and TCP Transport Specification 1.0 • Corrected bit range for Get Log Page LSP field • Removed "immediate" de-registration capability due to security concerns • Made Commands Supported and Effects Log Page optional for Discovery controllers • ENTUNRCH field changed to ENTCONN and added as a bit to EFLAGS field added by TP 8014 • Added new field (DCOCS) to Identify Controller for Discovery controllers to indicate which optional capabilities they support (PLEO and EXTDLPE being the first two optional capabilities defined) • Added Extended Discovery Log Page Entry, which can be retrieved via LID 70h with LSP bit 8 set to '1'

2021-05-10	• Commands Supported and Effects log page made mandatory for Discovery controllers and removed note 4 from the same table (Figure 33)
	• Moved fields from DCOCS out of Identify Controller and into LID Specific Field in Supported Log Pages and shortened their descriptions
	• Made changes to LID 70h header to account for extended entries being intermixed with basic entries
	    • Added EXTEND flag
	    • Added TDLPL (only valid if EXTEND not set)
	    • Changed entry byte positioning math to account for mix of entry types
	• Added Host Discovery Log Page which DDCs can retrieve
	    • Changed Discover Restart status code to fit for Discovery Log Page and Host Discovery Log Page
	    • Added Host Discovery Log Page to I/O / Admin / Discovery controller log page support tables (M for CDCs and P for the rest)
	• Incorporated all of Claudio's zoning content
	• Added DIM command to I/O / Admin / Discovery command support tables (O for Discovery controllers and P for the rest)
	• Added Erik and Claudio as authors

2021-05-17	• Collapsed EXTEND and PORTLCL fields in DLP header into single 1 byte flags field (DLPF)
	• Removed extended attribute figures from (Subsystem) DLP and Host DLP and instead just reference existing figure NEW.5 in DIM command
	• Added Host DLP to table of LIDs in section 5.16.1
	• Moved fabric zoning commands out of section 8.NEW.2 and into section 5
	• Added Host Discovery Log Page change notification
	    • New text added to Discovery Controller requests and notifications (3.1.2.3.4.2)
	    • New text added to Async Event Informative – Notice (5.2.1)
	    • New text added to Async Event Configuration (5.27.1.8)
	• Added HDGST and DDGST to KDReq/KDResp NVMe/TCP PDUs
	• Added bit 4 to CATTR to distinguish if Discovery controller is performing the Connect
	• Added allowance for TRADDR of '0.0.0.0' in DIM payload when a host is performing the registration
	• Added ALLSUBE LSP bit to (Subsystem) DLP and ALLHOSTE LSP bit to Host DLP

2021-05-24	• Hosts can now also retrieve the Host Discovery Log Page
	• Added new zoning-related changes including:
	    • Fabric zoning definition in section 1.5.NEW4
	    • AEN content in section 8.NEW.4
	• Added intro-related text including:
	    • Discovery controller NQN discovery methodology in section 3.1.2.3
	    • Model clause intro in section 8.NEW.1

2021-05-26	• Added ALLSUBES bit to Supported Log Pages for Discovery Log Page and modified related text • Added ALLHOSTES bit to Supported Log Pages for Host Discovery Log Page and modified related text • Clarified that DDCs indicate support for Extended DLPEs via EXTDLPES bit for Discovery Log Page and that CDCs shall support Extended DLPEs • Added clarification to TRADDR field in DIM payload that only one entry can have the field set to 0.0.0.0 • Changed definition in EID field
2021-06-01	• Removed section 8.NEW.5 (DNS)
2021-06-02	• Changed text in TRADDR field in DIM payload for case with multiple entries with 0.0.0.0 to focus on controller side processing • ALLSUBE can now be requested from both CDCs and DDCs • Added text in pull registration section about how CDC uses ALLSUBE to get all entries back from DDC

2021-06-14	• Added clarity about Get Log Page LSP bit support for DDC vs. CDC
2021-06-17	• Defined PDO field in KDResp PDU
2021-06-21	• Added clarity that CDC can get AENs when DLP changes (sections 3.1.2.3.4.2 and 5.2.1)

2021-06-24	• Added "may" statement about authentication or outside configuration being required before reporting ALLSUBES = '1'
	• Added error condition text when CDC/DDC receives Get Log Page command requesting LSP functionality that is not supported by that CDC/DDC
	• Added IPv4/IPv6 address family requirement to ADRFAM field in discovery information entry being registered with DIM command
	• Added text describing usage of CNTLID field in discovery information entry being registered with DIM command
	• Revised "zeroes case" text in TRADDR field in discovery information entry being registered with DIM command
	• Changed "if this field is…" "present" to "valid" for HDGST/DDGST fields in KDReq/KDResp NVMe/TCP PDUs
	• Added Erik's EID/Entry Key related content:
	    • Entry Key definition
	    • Explanatory text in DIM command section
	    • Update task
	    • EKTYPE field in DIM command header
	• Updated O/M column and footnotes for various fields in DIM command header based upon de-registration task functionality change and update task addition

2021-07-01	• Added error conditions when 0 entries are provided in DIM command for reg/de-reg tasks and anything other than 2 entries are provided in DIM command for update tasks • Revised "zeroes case" text again in TRADDR field in DIM command to look for NULL in first byte • Included text from TP 8013 into section 3.1.2.3 to give the reader the full context about the unique Discovery subsystem NQN • Changed "Asynchronous Event Information F0h" to "Asynchronous Event Information field set to F0h" in section 3.1.2.3.4.2 • Made what can be returned in DLP/HDLP (all vs. subset) into bulleted list • Added text for DLP/HLDP about how IOS = 1 works • Clarified DLP/HDLP processing sequence by putting it into a list • Changed 00h value in DCTYPE field in Identify Controller • Renamed "Zoning Data Structure Missing" status code to "Zoning Data Structure Not Found" • Added clarity to bit 4 in CATTR field • Renamed "CDCCONN" bit in DLP entry to "NOCDCCONN" to avoid confusion • Added Capacity Exceeded as new FAILRSN for KDResp and as new FES for H2CTermReq
2021-07-12	• Added Capacity Exceeded as new command specific status code for DIM command • Propose removing new FES for H2CTermReq and replacing with new mechanism (new field in KDReq specifying how many entries will be registered so error can be caught before Get Log Page page)
2021-07-16	• Propose changing "Capacity Exceeded" KDResp FAILRSN/DIM command specific status code to "Discovery Information Database Full" • Changed "in the SQE" to "in Command Dword 10" when referring to the TAS field of the DIM command • Got rid of "shalls" in NUMENT field of DIM command for error conditions that are already covered with "shalls" in section 5.NEW.1 text • Added examples for how index offset is used for DLP/HDLP • Added clarification to NUMDIE field of KDReq that it is the maximum number of entries for this request

2021-07-19	• Propose reducing size of DCTYPE field from 2 bytes down to 1 byte
2021-07-21	• Renamed "Discovery Information Database Full" status code to "Insufficient Discovery Resources" • Included index offset 0 (i.e., header) in example for DLP/HDLP • Added rows to DLP/HDLP and DIM command data to delineate the header and entry portions • Cross-spec references now use figure/section name instead of number • Removed use of "looked up" in zoning-related text • HDLP is no longer only able to be supported by CDCs • Added clarity to NCC bit in a DLE that it only applies for CDCs • Removed "CDC or DDC" for new DLP LSP bits and their support bits and just say "Discovery controller", listing restrictions when necessary (e.g., CDC shall support EXTDLPE, PLEO is only for DDC, etc.)
2021-07-28	• Changed "has no meaning" to "should be ignored" in PLEOS bit • Corrected labels in Figure NEW.X • Incorporated various phase 3 related changes from Mike Allison
2021-07-29	• Incorporated more phase 3 related changes from Mike Allison • Turned CATTR field into bit table and named all of the bits
2021-08-04	• Removed usage of "applicable"/"not applicable" for extended discovery information entry fields in DIM command. Fields are either "used" or "ignored" (and if ignored then cleared to 0h) • Added clarity for usage of "first byte" in TRADDR field in DIM command • Added error condition when EXATLEN specified is not a multiple of 4 • Removed padding requirement from EXATVAL. Unused bytes are simply cleared to 0h • Clarified that ZDO field in FZR/FZS commands is byte offset to store transferred data (not location to transfer from) • Fixed byte range in Zone Detailed Representation figure • Added text to section 8.NEW.2 describing "implicit" registrations • Aligned with TP 4109 that moved the definition of LID Specific Fields to the definition of the log page that supports the field • Removed Model Number (from Identify Controller) example from ENAME field and replaced with generic example name ("ACME 2000") • Changed "to transfer" to "to request" in FZR command description
2021-08-05	• Moved LF bit out of FRZ SQE and into FZR CQE (CDC knows when it's the last fragment – not DDC) • Added ZGFL to FZR data header • Added 4 more reserved bytes to FZR data header for 16-byte alignment
2021-08-10	• Generalized EVER field usage (replaced prescriptive requirements with example usage) • Reworked wording in TRADDR field in DIM command for zeros case • Moved LID Specific Field support bits out of new subsections and into main DLP/HDLP sections
2021-08-23	• Renamed document for phase 3 • Converted EFLAGS field in DLPE into table of bits • Globally renumbered all figures
2021-08-24	• Made Supported Log Pages note double strikethrough in Figure 33 • Changed "will" to "shall" or "is" as appropriate globally • Changed multiplication symbol from "x" to "*" in Figure 265 and NEW.A • Added Bits column to DPTR for DIM command in Figure NEW.8 • Changed "it" to "that ZoneGroup" in Fabric Zoning section
2021-08-30	• Updated referenced TP 8009 section numbers

2021-09-01	• Removed use of "single" when referring to CDC in section 8.NEW.1 • Added "should" statement to section 8.NEW.1 about hosts/DDCs registering with each CDC • Minor sentence restructuring for PLEO/ALLSUBE LSP bit interactions in section 8.NEW.2.2 • Added "shall" statement to section 8.NEW.3.4.1 that unspecified elements for Zone member types are not enforceable • Removed "cans" and "coulds" globally • Replaced "physical port" with "fabric interface" globally in Zoning sections
2021-09-07	• Removed unmatched parenthesis in section 5.16.1.23 • Appended "authentication" text to preceding ALLHOSTES text (as done with ALLSUBES text) • Added "e" to make "NVMe Qualified Name" globally • Added comments from Dell
2021-09-08	• Removed DDC/CDC specificity globally where appropriate • Removed "for this log page" in DLP/HDLP sections • Changed "returned data" to "Generation Counter" in DLP/HDLP sections • Reworked referral entry text in DLP section • Added minor clarity when mentioning "the address" in TRADDR field in a DIE • Renamed "Port" Zone member type element to "IP Protocol Port" globally in Zoning sections • Added clarity to length fields globally in Zoning sections • Added new bits to DLPF/HDLPF fields to convey EKTYPE usage (for pull)
2021-09-10	• Added two new zone member types and renumbered section/figure numbers accordingly
2021-09-15	• Made DIM command and Fabric Zoning commands mandatory for CDCs and optional for non-CDCs • Added new command specific status code Requested Function Disabled for CDCs that haven't enabled Fabric Zoning with a pointer in section 8.NEW.3.8.1 • Moved new EKTYPE fields in DLP/HLDP from header into entry level • Added new command specific status code ZoneGroup Originator Invalid for FZS command with a pointer in section 5.NEW3 • Reworked DDC host behavior functionality text in section 8.NEW.2.1 • Reworked generation number text in section 8.NEW.3.2 to match elsewhere • Renumbered Zone member type values to move ZoneAlias to EFh • Moved field descriptions into tables globally in Zoning sections
2021-09-16	• Added Zone member types 04h, 05h, 06h, 11h, and 12h as ZoneAlias member types
2021-09-20	• Removed ZoneAlias Member Types figure and corresponding ZoneAlias Member Type definitions subsections and replaced with reference to Zone Member Types figure noting that ZoneAlias Zone Member Types are prohibited
2021-09-21	• Removed EKTYPE field from HDLP • Made NQN matching check for ZoneGroup Originator + Connect command and associated ZoneGroup Originator Invalid status code optional • Accepted all tracked changes in preparation for 30-day member review
2021-09-27	• Removed EKTYPE field from DLP as pull always overwrites • Added clarification section 8.NEW.2.2 related to additional registrations
2021-09-29	• Added vendor specific range for extended attributes
2021-10-22	• Integrated Samsung's member review comments
2021-10-27	• Made Samsung proposed editorial fixes after FMDS discussion • Added clarifying statements to GAZ and AAZ operation sections explaining how LF bit is used
2021-10-28	• Converted Fabric Zoning figures to Visio diagrams
2021-11-09	• Integrated Cisco's member review comments

2021-11-18	• Cleaned up references in change description section to only reference this TP • Changed file name to indicate integration ready
2022-01-10	• Integrated
2022-01-11	• Remove highlights to TBD values but put back in highlights to cross-references to figures and sections in new text. Updated ZMTRADD field to show bytes within the field as opposed to the entire data structure. Updated TCP Transport for showing DGST not present.
2022-01-12	• Updated highlighted text for consistency.
2022-02-09	• Changed "LID Specific Field field" to "LID Specific field". • Corrected trademarks and copyright information. • Added the Host Discovery log page support in the NVMe-MI specification.
2022-02-14	• Corrected the NVMe-MI version in title before showing document changes.
2022-02-28	• Aligned to ECN109 by changing "Log Specific field" to "Log Specific Parameter field" and "LID Specific field" to "LID Specific Parameter field".
2022-03-01	• Aligned to ECN110.
2022-03-18	• Fixed the repeated Logo
2022-05-03	• Integrated
2022-05-09	• Editorial changes implemented per Matt Goepfert

Description for Specification Changes

NVM Express Base Specification 2.0a:

- Added six new definitions: CDC, DDC, discovery information, Fabric Zoning, Entry Key, and symbolic name
- Added four new Admin commands (all of which are mandatory for CDCs and optional for other Discovery controller types): Discovery Information Management, Fabric Zoning Lookup, Fabric Zoning Send, and Fabric Zoning Receive
- Added one new log page (mandatory for CDCs and optional for other Discovery controller types): Host Discovery
- Added two new command specific status codes for the DIM command: Invalid Discovery Information and Insufficient Discovery Resources
- Added three new command specific status codes for the FZL, FZS, and FZR commands: Zoning Data Structure Locked, Zoning Data Structure Not Found, and Requested Function Disabled
- Added one new command specific status code for just the FZS command: ZoneGroup Originator Invalid
- Added new Host Discovery Log Page Change Asynchronous Event notification corresponding to the Host Discovery log page
- Defined LID Specific Parameter field for the Discovery log page and included three new bits to show support for new functionality: ALLSUBES, PLEOS, and EXTDLPES
- Defined Log Specific Parameter field for the Discovery log page and included three new bits that request the functionality corresponding with each of the three new bits included in the LID Specific Parameter field: ALLSUBE, PLEO, and EXTDLPE
- Added new NCC bit to EFLAGS field in the Discover log page to indicate whether or not the DDC describing the DLE is currently connected to the CDC
- Added new Extended Discovery Log Page Entry to the Discovery log page which includes additional information (i.e., extended attributes)

- Added new DLPF field to the Discovery log page header and included three new bits to indicate information related to new functionality: ALLSUB, PORTLCL, and EXTEND
- Added new TDLPL field to the Discovery log page header to indicate the total length of the log page
- Added new DCTYPE field to the Identify Controller data structure to indicate which type of Discovery controller the controller is
- Added new CONNENT bit to the Connect command's CATTR field to indicate what type of entity is performing the Connect command
- Added a model clause section for centralized discovery (see section 8.NEW) that defines: two different discovery information registration/de-registration models (i.e., push and pull), Fabric Zoning to control which discovery information is returned for Get Log Page, and Asynchronous Events that are generated as the result of changes to discovery information
- **New requirement / incompatible change:**
  - Commands Supported and Effects log page is now mandatory for all Discovery controllers
- References:
  - TP8010

NVM Express Management Interface Specification 1.2b:

- Added the support requirements for the Host Discovery log page.
- **New requirement / incompatible change:**
  - None
- References:
  - TP8010

NVMe TCP Transport Specification 1.0a:

- Added two new NVMe/TCP PDUs: KDReq and KDResp
- Added new KDCONN bit to the ICReq NVMe/TCP PDU's FLAGS field to request a kickstart discovery NVMe/TCP connection
- **New requirement / incompatible change:**
  - None
- References:
  - TP8010

Markup conventions

Style	Meaning
Black	Unchanged existing material
Blue	New material to be added
Red	Existing material to be deleted
Blue	New material to be renumbered as incorporated
<Green>	Note to the editor
Purple	Changes made by TP8013 and TP8014

Description of NVM Express Base Specification 2.0a Changes

1 Introduction

…

1.5 Definitions

…

…

1.5.16 Discovery controller

A controller that exposes capabilities that allow a host to retrieve a Discovery ~~L~~log ~~P~~page. A Discovery controller does not implement I/O Queues or provide access to a non-volatile memory storage medium. Refer to section 3.1.2.3.

…

1.5.NEW1 Centralized Discovery controller (CDC)

A Discovery controller that reports discovery information registered by Direct Discovery controllers and hosts.

1.5.NEW2 Direct Discovery controller (DDC)

A Discovery controller that is capable of registering discovery information with a Centralized Discovery controller.

1.5.NEW3 discovery information

Information about a host or NVM subsystem that is used for discovery (e.g., NVMe Transport address, NQN, etc.).

1.5.NEW4 Fabric Zoning

A technique to specify access control configurations between hosts and NVM subsystems.

1.5.NEW5 Entry Key

A set of discovery information entry fields that allow for the unique identification of each discovery information entry registered with the CDC or DDC. Refer to the Entry Key Type (EKTYPE) field.

1.5.NEW6 symbolic name

An ASCII string. Refer to section 1.4.2 for ASCII string requirements.

3 NVM Express Architecture

…

3.1.2 Controller Types

As shown in Figure 20, there are three types of controllers. An I/O controller (refer to section 3.1.2.1) is a controller that supports commands that provide access to user data stored on an NVM subsystem's non-volatile storage medium and may support commands that provide management capabilities. An Administrative controller (refer to section 3.1.2.2) is a controller that supports commands that provide management capabilities, but does not support I/O commands that access to user data stored on an NVM subsystem's non-volatile storage medium. A Discovery controller (refer to section 3.1.2.3) is a controller used in NVMe over Fabrics to provide access to a Discovery ~~L~~log ~~P~~page.

…

3.1.2.1 I/O Controller

…

3.1.2.1.1 Command Support

…

Figure 22: I/O Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Delete I/O Submission Queue	M	5.7
Create I/O Submission Queue	M	5.5
Get Log Page	M	5.16
Delete I/O Completion Queue	M	5.6
Create I/O Completion Queue	M	5.4
Identify	M	5.17
Abort	M	5.1
Set Features	M	5.27
Get Features	M	5.15
Asynchronous Event Request	M	5.2
Capacity Management	O	5.3
Namespace Management	O	5.23
Firmware Commit	O	5.12
Firmware Image Download	O	5.13
Device Self-test	O	5.9
Namespace Attachment	O	5.22
Keep Alive	NOTE 2	5.18
Directive Send	O	5.11
Directive Receive	O	5.10
Virtualization Management	O	5.28
NVMe-MI Send	O	5.21
NVMe-MI Receive	O	5.20
Doorbell Buffer Config	O	5.8
Lockdown	O	5.19
Format NVM	O	5.14
Security Send	O	5.26
Security Receive	O	5.25
Sanitize	O	5.24
Discovery Information Management	P	5.NEW1
Fabric Zoning Lookup	P	5.NEW2
Fabric Zoning Send	P	5.NEW3
Fabric Zoning Receive	P	5.NEW4

Figure 22: I/O Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Property Set	M[3]	6.6
Connect	M[3]	6.3
Property Get	M[3]	6.5
Authentication Send	O[3]	6.2
Authentication Receive	O[3]	6.1
Disconnect	O[3]	6.4
I/O Command Set specific Admin Command	Refer to the applicable I/O Command Set specification	Refer to the applicable I/O Command Set specification
Vendor Specific	O	

Notes:
1. O = Optional, M = Mandatory, P = Prohibited
2. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory.
3. For NVMe over PCIe implementations, all Fabrics commands are prohibited. For NVMe over Fabrics implementations, the commands are as noted in the table.

…

3.1.2.1.2 Log Page Support

…

Figure 24: I/O Controller – Log Page Support

Log Page Name	Log Page Support Requirements [1]
Supported Log Pages	M[3]
Error Information	M
SMART / Health Information (Controller scope)	M
SMART / Health Information (Namespace scope)	O
Firmware Slot Information	M
Changed Namespace List	O
Commands Supported and Effects	O
Device Self-test	O
Telemetry Host-Initiated	O
Telemetry Controller-Initiated	O
Endurance Group Information	O
Predictable Latency Per NVM Set	O
Predictable Latency Event Aggregate	O
Asymmetric Namespace Access	O
Persistent Event	O
Endurance Group Event Aggregate	O
Media Unit Status	O[2]
Supported Capacity Configuration List	O[2]
Feature Identifiers Supported and Effects	M[3]
NVMe-MI Commands Supported and Effects	M[3]
Command and Feature Lockdown	O
Reservation Notification	O

Figure 24: I/O Controller – Log Page Support

Log Page Name	Log Page Support Requirements [1]
Sanitize Status	O
Boot Partition	O
Rotational Media Information	O
Discovery	P
Host Discovery	P
Notes:	
1. O = Optional, M = Mandatory, P = Prohibited	
2. Mandatory for controllers that support Fixed Capacity Management (refer to section 8.3.2).	
3. Optional for NVM Express revision 1.4 and earlier.	

…

3.1.2.2 Administrative Controller

…

3.1.2.2.1 Command Support

…

**Figure 28: Administrative Controller – Admin Command
Support**

Command	Command Support Requirements [1]	Reference
Delete I/O Submission Queue	P	5.7
Create I/O Submission Queue	P	5.5
Get Log Page	M	5.16
Delete I/O Completion Queue	P	5.6
Create I/O Completion Queue	P	5.4
Identify	M	5.17
Abort	O	5.1
Set Features	O [3]	5.27
Get Features	O [3]	5.15
Asynchronous Event Request	O [4]	5.2
Capacity Management	O	5.3
Namespace Management	O	5.23
Firmware Commit	O	5.12
Firmware Image Download	O	5.13
Device Self-test	O	5.9
Namespace Attachment	O	5.22
Keep Alive	NOTE 2	5.18
Directive Send	O	5.11
Directive Receive	O	5.10
Virtualization Management	O	5.28
NVMe-MI Send	O	5.21
NVMe-MI Receive	O	5.20
Doorbell Buffer Config	O	5.8
Lockdown	O	5.19
Format NVM	O	5.14

Figure 28: Administrative Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Security Send	O	5.26
Security Receive	O	5.25
Sanitize	O	5.24
Discovery Information Management	P	5.NEW1
Fabric Zoning Lookup	P	5.NEW2
Fabric Zoning Send	P	5.NEW3
Fabric Zoning Receive	P	5.NEW4
Property Set	M[6]	6.6
Connect	M[6]	6.3
Property Get	M[6]	6.5
Authentication Send	O[6]	6.2
Authentication Receive	O[6]	6.1
Disconnect	O[6]	6.4
I/O Command Set Specific Admin Commands	P	
Vendor Specific	O	
Notes:		

Notes:
1. O = Optional, M = Mandatory, P = Prohibited
2. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory.
3. Mandatory if any of the features in Figure 30 are implemented.
4. Mandatory if Telemetry Log, Firmware Commit, or SMART/Health Critical Warnings are supported.

For NVMe over PCIe implementations, all Fabrics commands are prohibited. For NVMe over Fabrics implementations, the commands are as noted in the table.

3.1.2.2.2 Log Page Support

…

Figure 29: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements [1]
Supported Log Pages	M[4]
Error Information	M
SMART / Health Information (Controller scope)	O
SMART / Health Information (Namespace scope)	O
Firmware Slot Information	O
Changed Namespace List	O
Commands Supported and Effects	M
Device Self-test	O
Telemetry Host-Initiated	O
Endurance Group Information	O
Predictable Latency Per NVM Set	O
Predictable Latency Event Aggregate	O
Asymmetric Namespace Access	P
Persistent Event	O

Figure 29: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements [1]
Endurance Group Event Aggregate	O
Media Unit Status	P
Supported Capacity Configuration List	P
Feature Identifiers Supported and Effects	M[2,4]
NVMe-MI Commands Supported and Effects	M[3,4]
Command and Feature Lockdown	O
Reservation Notification	P
Sanitize Status	O
Rotational Media	P
Boot Partition	O
I/O Command Set Specific Log Pages	P
Discovery	P
Host Discovery	P

Notes:
1. O = Optional, M = Mandatory, P = Prohibited
2. Optional if Set Features command is not supported (refer to Figure 28).
3. Optional if NVMe-MI Send command and NVMe-MI Receive command is not supported (refer to Figure 28).
4. Optional for NVM Express revision 1.4 and earlier.

…

3.1.2.3 Discovery Controller

A Discovery controller only implements features related to Discovery ~~L~~log ~~P~~pages and does not implement I/O Queues, I/O commands, or expose namespaces. The functionality supported by the Discovery controller is defined in section 3.1.2.3.4.

If the Discovery subsystem provides a unique NQN (i.e., the NVM Subsystem NVMe Qualified Name (SUBNQN) field in that Discovery subsystem's Identify Controller data structure contains a unique NQN value), then that Discovery subsystem shall support both the unique NQN and the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery) being specified in the Connect command (refer to section 6.3) from the host.

If the Discovery subsystem does not provide a unique NQN (i.e., the SUBNQN field in that Discovery subsystem's Identify Controller data structure contains the well-known Discovery Service NQN), then that Discovery subsystem shall support the well-known Discovery Service NQN being specified in the Connect command from the host.

In the Connect command to a Discovery subsystem that provides a unique NQN, the host may use either of the following:

- the well-known Discovery Service NQN; or
- the unique NQN of that Discovery subsystem.

In the Connect command to a Discovery subsystem that does not provide a unique NQN, the host uses the well-known Discovery Service NQN.

The method that a host uses to obtain the ~~NVMe Transport~~fabric information necessary to connect to a Discovery controller using the well-known Discovery Service NQN or the unique NQN via the NVMe Transport ~~is~~may be:

a) implementation specific~~.~~;
b) fabric specific;
c) known in advance (e.g., a well-known address);

d) administratively configured; or
e) for IP-based fabrics, Automated Discovery of Discovery Controllers for IP-based Fabrics (refer to section 8.NEW.A <Note to editor: 8.NEW.A is located in TP8009>) may be used.

The Discovery ~~L~~log ~~P~~page provided by a Discovery controller contains one or more entries. Each entry specifies information necessary for the host to connect to an NVM subsystem. An entry may be associated with an NVM subsystem that exposes namespaces or a referral to another Discovery Service. There are no ordering requirements for log page entries within the Discovery ~~L~~log ~~P~~page.

<Note to editor: the purple text and "unique NQN" mentioned above comes from TP8013. It has been included here to give the reader the full context of how everything works together.>

…

A Discovery ~~L~~log ~~P~~page with multiple Discovery Log Page Entries for the same NVM subsystem indicates that there are multiple fabric paths to the NVM subsystem, and/or that multiple static controllers may share a fabric path. The host may use this information to form multiple associations to controllers within an NVM subsystem.

…

If an NVM subsystem implements the dynamic controller model, then multiple Discovery Log Page Entries (refer to Figure 264) with the Controller ID set to FFFFh may be returned for that NVM subsystem (e.g., to indicate multiple NVM subsystem ports) in the Discovery ~~L~~log ~~P~~page. If an NVM subsystem implements the static controller model, then multiple Discovery Log Page Entries that indicate different Controller ID values may be returned for that NVM subsystem in the Discovery ~~L~~log ~~P~~page. If an NVM subsystem that implements the static controller model includes any Discovery Log Page Entries that indicate a Controller ID of FFFEh, then the host should remember the Controller ID returned from the Fabrics Connect command and re-use the allocated Controller ID for future associations to that particular controller.

3.1.2.3.1 Discovery Controller Initialization

…

After initializing the Discovery controller, the host reads the Discovery ~~L~~log ~~P~~page. Refer to section 5.16.1.2~~1~~3.

3.1.2.3.2 Command Support

…

Figure 32: Discovery Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Delete I/O Submission Queue	P	5.7
Create I/O Submission Queue	P	5.5
Get Log Page	M	5.16
Delete I/O Completion Queue	P	5.6
Create I/O Completion Queue	P	5.4
Identify	M	5.17
Abort	O	5.1
Set Features	NOTE 2	5.27
Get Features	NOTE 2	5.15
Asynchronous Event Request	NOTE 2	5.2
Capacity Management	P	5.3
Namespace Management	P	5.23
Firmware Commit	P	5.12
Firmware Image Download	P	5.13

Figure 32: Discovery Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Device Self-test	P	5.9
Namespace Attachment	P	5.22
Keep Alive	NOTE 2	5.18
Directive Send	P	5.11
Directive Receive	P	5.10
Virtualization Management	P	5.28
NVMe-MI Send	P	5.21
NVMe-MI Receive	P	5.20
Doorbell Buffer Config	P	5.8
Lockdown	P	5.19
Format NVM	P	5.14
Security Send	P	5.26
Security Receive	P	5.25
Sanitize	P	5.24
Discovery Information Management	M[3]	5.NEW1
Fabric Zoning Lookup	M[3]	5.NEW2
Fabric Zoning Send	M[3]	5.NEW3
Fabric Zoning Receive	M[3]	5.NEW4
Property Set	M	6.6
Connect	M	6.3
Property Get	M	6.5
Authentication Send	O	6.2
Authentication Receive	O	6.1
Disconnect	O	6.4
I/O Command Set Specific Admin Commands	P	
Vendor Specific	O	

Notes:
1. O = Optional, M = Mandatory, P = Prohibited
2. For Discovery controllers that do not support explicit persistent connections, the command is reserved. For Discovery controllers that support explicit persistent connections, the command is mandatory.
3. Mandatory for CDCs and optional for Discovery controllers that are not a CDC.

3.1.2.3.3 Log Page Support

The Discovery controller shall support the Discovery Llog Ppage. The log pages that a Discovery controller may support are shown in Figure 33.

Figure 33: Discovery Controller – Log Page Support

Log Page Name	Command Support Requirements [1]
Supported Log Pages	M[4]
Error Information	O
SMART / Health Information (Controller scope)	P
SMART / Health Information (Namespace scope)	P
Firmware Slot Information	P

Figure 33: Discovery Controller – Log Page Support

Log Page Name	Command Support Requirements [1]
Changed Namespace List	P
Commands Supported and Effects	~~P~~M
Device Self-test	P
Telemetry Host-Initiated	P
Telemetry Controller-Initiated	P
Endurance Group Information	P
Predictable Latency Per NVM Set	P
Predictable Latency Event Aggregate	P
Asymmetric Namespace Access	P
Persistent Event	P
Endurance Group Event Aggregate	P
Media Unit Status	P
Supported Capacity Configuration List	P
Feature Identifiers Supported and Effects	M[2,~~4~~]
NVMe-MI Commands Supported and Effects	M[3,~~4~~]
Command and Feature Lockdown	P
Discovery	M
Reservation Notification	P
Sanitize Status	P
Rotational Media	P
Boot Partition	P
I/O Command Set Specific Log Pages	P
Host Discovery	M[4]

Notes
1. O = Optional, M = Mandatory, P = Prohibited
2. Optional if Set Features command is not supported (refer to Figure 32).
3. Optional if NVMe-MI Send command and NVMe-MI Receive command is not supported (refer to Figure 32).
4. ~~Optional for versions 1.1 and earlier of the NVMe over Fabrics specification~~Mandatory for CDCs and optional for Discovery controllers that are not a CDC.

…

3.1.2.3.4.2 Discovery Controller Asynchronous Event Information – Requests and Notifications

If Discovery controllers detect events about which a host has requested notification, then the Discovery controller shall send an Asynchronous Event with the:

- Asynchronous Event Type field set to Notice (i.e., 2h);
- Log Page Identifier field set to either Discovery (i.e., 70h) or Host Discovery (i.e., 71h) depending on which log page has changed; and
- Asynchronous Event Information field set as defined in Figure 146.

~~When~~As a result of a Discovery controller ~~updates~~updating Discovery ~~L~~log ~~P~~page(s), ~~the~~that Discovery controller shall send a Discovery Log Page Change Asynchronous Event notification (Asynchronous Event Information field set to F0h) to each ~~host~~entity that has requested asynchronous event notifications of this type (refer to Figure 146).

As a result of a Discovery controller updating Host Discovery log page(s), that Discovery controller shall send a Host Discovery Log Page Change Asynchronous Event notification (Asynchronous Event Information field set to F1h) to each entity that has requested asynchronous event notifications of this type (refer to Figure 146).

…

3.3.3.2.1.2 Command Specific Status Definition

…

Figure 95: Status Code – Command Specific Status Values

Value	Description	Commands Affected
00h	Completion Queue Invalid	Create I/O Submission Queue
01h	Invalid Queue Identifier	Create I/O Submission Queue, Create I/O Completion Queue, Delete I/O Completion Queue, Delete I/O Submission Queue
02h	Invalid Queue Size	Create I/O Submission Queue, Create I/O Completion Queue
03h	Abort Command Limit Exceeded	Abort
04h	Reserved	
05h	Asynchronous Event Request Limit Exceeded	Asynchronous Event Request
06h	Invalid Firmware Slot	Firmware Commit
07h	Invalid Firmware Image	Firmware Commit
08h	Invalid Interrupt Vector	Create I/O Completion Queue
09h	Invalid Log Page	Get Log Page
0Ah	Invalid Format	Format NVM, Namespace Management
0Bh	Firmware Activation Requires Conventional Reset	Firmware Commit, Sanitize
0Ch	Invalid Queue Deletion	Delete I/O Completion Queue
0Dh	Feature Identifier Not Saveable	Set Features
0Eh	Feature Not Changeable	Set Features
0Fh	Feature Not Namespace Specific	Set Features
10h	Firmware Activation Requires NVM Subsystem Reset	Firmware Commit, Sanitize
11h	Firmware Activation Requires Controller Level Reset	Firmware Commit, Sanitize
12h	Firmware Activation Requires Maximum Time Violation	Firmware Commit
13h	Firmware Activation Prohibited	Firmware Commit
14h	Overlapping Range	Firmware Commit, Firmware Image Download, Set Features
15h	Namespace Insufficient Capacity	Namespace Management
16h	Namespace Identifier Unavailable	Namespace Management
17h	Reserved	
18h	Namespace Already Attached	Namespace Attachment
19h	Namespace Is Private	Namespace Attachment
1Ah	Namespace Not Attached	Namespace Attachment
1Bh	Thin Provisioning Not Supported	Namespace Management
1Ch	Controller List Invalid	Namespace Attachment
1Dh	Device Self-test In Progress	Device Self-test
1Eh	Boot Partition Write Prohibited	Firmware Commit
1Fh	Invalid Controller Identifier	Virtualization Management
20h	Invalid Secondary Controller State	Virtualization Management
21h	Invalid Number of Controller Resources	Virtualization Management
22h	Invalid Resource Identifier	Virtualization Management
23h	Sanitize Prohibited While Persistent Memory Region is Enabled	Sanitize
24h	ANA Group Identifier Invalid	Namespace Management
25h	ANA Attach Failed	Namespace Attachment
26h	Insufficient Capacity	Capacity Management
27h	Namespace Attachment Limit Exceeded	Namespace Attachment
28h	Prohibition of Command Execution Not Supported	Lockdown

Figure 95: Status Code – Command Specific Status Values

Value	Description	Commands Affected
29h	I/O Command Set Not Supported	Namespace Attachment, Namespace Management
2Ah	I/O Command Set Not Enabled	Namespace Attachment
2Bh	I/O Command Set Combination Rejected	Set Features
2Ch	Invalid I/O Command Set	Identify, Namespace Management
2Dh	Identifier Unavailable	Capacity Management
2Fh	Invalid Discovery Information	Discovery Information Management
30h	Zoning Data Structure Locked	Fabric Zoning Lookup, Fabric Zoning Send, Fabric Zoning Receive
31h	Zoning Data Structure Not Found	Fabric Zoning Lookup, Fabric Zoning Send, Fabric Zoning Receive
32h	Insufficient Discovery Resources	Discovery Information Management
33h	Requested Function Disabled	Fabric Zoning Lookup, Fabric Zoning Send, Fabric Zoning Receive
34h	ZoneGroup Originator Invalid	Fabric Zoning Send
~~2Dh~~35h to 6Fh	Reserved	
70h to 7Fh	Directive Specific	NOTE 1
80h to BFh	I/O Command Set Specific	Refer to Figure 96
C0h to FFh	Vendor Specific	
NOTES: 1. The Directives Specific range defines Directives specific status values. Refer to section 8.7.		

…

Figure 97: Status Code – Command Specific Status Values, Fabrics Commands

Value	Description	Commands Affected
80h	**Incompatible Format:** The NVM subsystem does not support the record format specified by the host.	Connect, Disconnect
81h	**Controller Busy:** The controller is already associated with a host (Connect command). This value is also returned if there is no available controller (Connect command). The controller is not able to disconnect the I/O Queue at the current time (Disconnect command).	Connect, Disconnect
82h	**Connect Invalid Parameters:** One or more of the command parameters (e.g., Host NQN, Subsystem NQN, Host Identifier, Controller ID, Queue ID) specified are not valid.	Connect
83h	**Connect Restart Discovery:** The NVM subsystem requested is not available. The host should restart the discovery process.	Connect
84h	**Connect Invalid Host:** The host is not allowed to establish an association to any controller in the NVM subsystem or the host is not allowed to establish an association to the specified controller.	Connect
85h	**Invalid Queue Type:** The command was sent on the wrong queue type (e.g., a Disconnect command was sent on the Admin queue).	Disconnect
86h to 8Fh	Reserved	
90h	**Discover Restart:** The snapshot of the records is now invalid or out of date. If the Discovery log page was requested, then Tthe host or Discovery controller should re-read the Discovery Llog Ppage. If the Host Discovery log page was requested, then the host or Discovery controller should re-read the Host Discovery log page.	Get Log Page
91h	**Authentication Required:** NVMe in-band authentication is required and the queue has not yet been authenticated.	NOTE 1
92h to AFh	Reserved	
B0h to BFh	**Transport Specific:** The status values in this range are NVMe Transport specific. Refer to the appropriate NVMe Transport binding specification for the definition of these status values.	
NOTES: 1. All commands other than Connect, Authenticate Send, and Authenticate Receive.		

5 Admin Command Set

…

Figure 138: Opcodes for Admin Commands

Opcode by Field			Combined Opcode[1]	Namespace Identifier Used[2]	Command	Command Set Specific[8]
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[3]				
0b	000 00b	00b	00h	No	Delete I/O Submission Queue	No

Figure 138: Opcodes for Admin Commands

Opcode by Field			Combined Opcode[1]	Namespace Identifier Used[2]	Command	Command Set Specific[8]
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[3]				
0b	000 00b	01b	01h	No	Create I/O Submission Queue	No
0b	000 00b	10b	02h	Yes	Get Log Page	No
0b	000 01b	00b	04h	No	Delete I/O Completion Queue	No
0b	000 01b	01b	05h	No	Create I/O Completion Queue	No
0b	000 01b	10b	06h	NOTE 6	Identify	No
0b	000 10b	00b	08h	No	Abort	No
0b	000 10b	01b	09h	Yes	Set Features	No
0b	000 10b	10b	0Ah	Yes	Get Features	No
0b	000 11b	00b	0Ch	No	Asynchronous Event Request	No
0b	000 11b	01b	0Dh	Yes	Namespace Management	No
0b	001 00b	00b	10h	No	Firmware Commit	No
0b	001 00b	01b	11h	No	Firmware Image Download	No
0b	001 01b	00b	14h	Yes	Device Self-test	No
0b	001 01b	01b	15h	Yes [4]	Namespace Attachment	No
0b	001 10b	00b	18h	No	Keep Alive	No
0b	001 10b	01b	19h	Yes [5]	Directive Send	No
0b	001 10b	10b	1Ah	Yes [5]	Directive Receive	No
0b	001 11b	00b	1Ch	No	Virtualization Management	No
0b	001 11b	01b	1Dh	No	NVMe-MI Send	No
0b	001 11b	10b	1Eh	No	NVMe-MI Receive	No
0b	010 00b	00b	20h	No	Capacity Management	No
0b	010 00b	01b	21h	No	Discovery Information Management	No
0b	010 00b	10b	22h	No	Fabric Zoning Receive	No
0b	010 01b	00b	24h	No	Lockdown	No
0b	010 01b	01b	25h	No	Fabric Zoning Lookup	No
0b	010 10b	01b	29h	No	Fabric Zoning Send	No
0b	111 11b	00b	7Ch	No	Doorbell Buffer Config	No
0b	111 11b	11b	7Fh	NOTE 9	Fabrics Commands[9]	No

Figure 138: Opcodes for Admin Commands

Opcode by Field			Combined Opcode[1]	Namespace Identifier Used[2]	Command	Command Set Specific[8]
(07) Generic Command	(06:02) Function	(01:00) Data Transfer[3]				
1b	000 00b	00b	80h	Yes	Format NVM	No
1b	000 00b	01b	81h	NOTE 7	Security Send	No
1b	000 00b	10b	82h	NOTE 7	Security Receive	No
1b	000 01b	00b	84h	No	Sanitize	No
1b	000 01b	10b	86h	NOTE 4	Get LBA Status	NVM, ZNS
Vendor Specific						
1b	n/a	NOTE 3	C0h to FFh		Vendor specific	

NOTES:
1. Opcodes not listed are reserved.
2. A subset of commands use the Namespace Identifier (NSID) field. If the Namespace Identifier field is used, then the value FFFFFFFFh is supported in this field unless otherwise indicated in footnotes in this figure that a specific command does not support that value or supports that value only under specific conditions. When this field is not used, the field is cleared to 0h as described in Figure 87.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command does not support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.
5. Support for the Namespace Identifier field set to FFFFFFFFh depends on the Directive Operation (refer to section 8.7).
6. Use of the Namespace Identifier field depends on the CNS value in the Identify Command as described in Figure 273.
7. The use of the Namespace Identifier is Security Protocol specific.
8. No = Not I/O Command Set specific, A = All I/O Command Sets, NVM = NVM Command Set specific, ZNS = Zoned Namespace Command Set.
9. All Fabrics commands use the opcode 7Fh. Refer to section 6 for details.

…

…

5.2 Asynchronous Event Request command

…

5.2.1 Command Completion

…

Figure 146: Asynchronous Event Information – Notice

Value	Description
…	…
07h to EEh	Reserved
EFh[2]	**Zone Descriptor Changed:** I/O Command Set specific definition.
F0h	**Discovery Log Page Change:** A change has occurred to one or more of the Discovery ~~L~~log ~~P~~pages. The host or Discovery controller should submit a Get Log Page command to receive updated Discovery ~~L~~log ~~P~~pages.

Figure 146: Asynchronous Event Information – Notice

Value	Description
F1h	**Host Discovery Log Page Change:** A change has occurred to one or more of the Host Discovery log pages. The host or Discovery controller should submit a Get Log Page command to receive updated Host Discovery log pages.
~~F1h~~F2h to FFh	Reserved for future NVMe over Fabrics Asynchronous Event Notifications
NOTE: 1. Refer to the NVM Command Set specification. 2. Refer to the Zoned Namespace Command Set specification.	

…

…

5.16.1 Log Specific Information

…

Figure 202: Get Log Page – Log Page Identifiers

Log Identifier	Scope	Log Page Name	Reference Section
00h	Controller	Supported Log Pages	5.16.1.1
01h	Controller	Error Information	5.16.1.2
02h	Controller [1] Namespace [2]	SMART / Health Information	5.16.1.3
03h	Domain / NVM subsystem [6]	Firmware Slot Information	5.16.1.4
04h	Controller	Changed Namespace List	5.16.1.5
05h	Controller	Commands Supported and Effects	5.16.1.6
06h	Controller [3] Domain / NVM subsystem [4, 6]	Device Self-test [5]	5.16.1.7
07h	Vendor Specific	Telemetry Host-Initiated [5]	5.16.1.8
08h	Vendor Specific	Telemetry Controller-Initiated [5]	5.16.1.9
09h	Domain / NVM subsystem [6]	Endurance Group Information	5.16.1.10
0Ah	Domain / NVM subsystem [6]	Predictable Latency Per NVM Set	5.16.1.11
0Bh	Domain / NVM subsystem [6]	Predictable Latency Event Aggregate	5.16.1.12
0Ch	Controller	Asymmetric Namespace Access	5.16.1.13
0Dh	NVM subsystem	Persistent Event Log [5]	5.16.1.14
0Eh	I/O Command Set Specific		
0Fh	Domain / NVM subsystem [6]	Endurance Group Event Aggregate	5.16.1.15

Figure 202: Get Log Page – Log Page Identifiers

Log Identifier	Scope	Log Page Name	Reference Section
10h	Domain / NVM subsystem [5, 6]	Media Unit Status	5.16.1.16
11h	Domain / NVM subsystem [6]	Supported Capacity Configuration List	5.16.1.17
12h	Controller	Feature Identifiers Supported and Effects	5.16.1.18
13h	Controller	NVMe-MI Commands Supported and Effects	5.16.1.19
14h	NVM subsystem	Command and Feature Lockdown [5]	5.16.1.20
15h	Controller	Boot Partition	5.16.1.21
16h	Endurance Group	Rotational Media Information	5.16.1.22
17h to 6Fh	Reserved		
70h		Discovery	5.16.1.23
71h		Host Discovery	5.16.1.NEW
72h to 7Fh	Reserved		
80h	Controller	Reservation Notification	5.16.1.24
81h	NVM subsystem	Sanitize Status	5.16.1.25
82h to BFh	I/O Command Set Specific		
C0h to FFh	Vendor specific [5]		

KEY:

Namespace = The log page contains information about a specific namespace.

Endurance Group = The log page contains information about a specific Endurance Group.

Controller = The log page contains information about the controller that is processing the command.

Domain = The log page contains information about the Domain.

NVM subsystem = The log page contains information about the NVM subsystem.

Vendor Specific = The log page contains information that is vendor specific.

NOTES:
1. For namespace identifiers of 0h or FFFFFFFFh.
2. For namespace identifiers other than 0h or FFFFFFFFh.
3. Bit 0 is cleared to '0' in the DSTO field in the Identify Controller data structure (refer to Figure 275).
4. Bit 0 is set to '1' in the DSTO field in the Identify Controller data structure.
5. Selection of a UUID may be supported. Refer to section 8.25.
6. For NVM subsystems that support multiple domains (refer to the MDS bit in the Identify Controller data structure, Figure 275), Domain scope information is returned.

…

5.16.1.23 Discovery ~~Log Page~~ (Log Identifier 70h)

<Note to editor: change "Discovery Log Page" to "Discovery log page" globally.>

The Discovery ~~L~~log ~~P~~page shall only be supported by Discovery controllers. The Discovery ~~L~~log ~~P~~page shall not be supported by controllers that expose namespaces for NVMe over PCIe or NVMe over Fabrics.

The Discovery ~~L~~log ~~P~~page provides an inventory of NVM subsystems with which a host may attempt to form an association. Depending on the parameters specified in a Get Log Page command, ~~T~~the Discovery ~~L~~log ~~P~~page may contain entries for:

- all known NVM subsystems; or
- ~~be~~a subset of entries specific to the host or Discovery controller requesting the log page.

The Discovery ~~L~~log ~~P~~page is persistent across power cycles.

<Note to editor: LID Specific Parameter field support bits now live in each corresponding log page section (instead of LID 00h section) per TP4109.>

Figure NEW.1 specifies the format for the LID Specific Parameter field in the Supported Log Pages log page (refer to section 5.16.1.1) for the Discovery log page.

Figure NEW.1: Discovery Log Page – LID Specific Parameter Field

Bits	Description
15:3	Reserved
2	**All NVM Subsystem Entries Supported (ALLSUBES):** If set to '1', then the Discovery controller supports returning records for all NVM subsystem ports that are registered without filtering the list of returned records based upon the requesting HOSTNQN. If cleared to '0', then the Discovery controller does not support returning records for all NVM subsystem ports that are registered.
1	**Port Local Entries Only Supported (PLEOS):** If set to '1', then the Discovery controller supports returning only port local Discovery Log Page Entries or port local Extended Discovery Log Page Entries. If cleared to '0', then the Discovery controller does not support returning only port local Discovery Log Page Entries or port local Extended Discovery Log Page Entries. If the Discovery controller is not a DDC, then this bit should be ignored.
0	**Extended Discovery Log Page Entry Supported (EXTDLPES):** If set to '1', then the Discovery controller supports returning Extended Discovery Log Page Entries. If cleared to '0', then the Discovery controller does not support returning Extended Discovery Log Page Entries. If the Discovery controller is a CDC, then that CDC shall support returning Extended Discovery Log Page Entries.

If a Discovery controller supports returning Extended Discovery Log Page Entries (i.e., the Extended Discovery Log Page Entries Supported (EXTDLPES) bit is set to '1' in the LID Specific Parameter field as defined in Figure NEW.1) and that Discovery controller processes a Get Log Page command for this log page with the Extended Discovery Log Page Entries (EXTDLPE) bit set to '1' in the Log Specific Parameter (LSP) field (refer to Figure NEW.2), then each entry returned by that Discovery controller may be either a Discovery Log Page Entry or an Extended Discovery Log Page Entry. Extended Discovery Log Page Entries may contain zero or more extended attributes. If a Discovery controller returns records that include Extended Discovery Log Page Entries, then that Discovery controller shall set the Extended (EXTEND) bit to '1' in the Discovery Log Page Flags (DLPF) field. A Centralized Discovery controller (CDC) shall support returning Extended Discovery Log Page Entries.

If a Discovery controller supports returning only port local Discovery Log Page Entries or port local Extended Discovery Log Page Entries (i.e., the Port Local Entries Only Supported (PLEOS) bit is set to '1' in the LID Specific Parameter field as defined in Figure NEW.1) and that Discovery controller processes a Get Log Page command for this log page with the Port Local Entries Only (PLEO) bit set to '1' in the LSP field (refer to Figure NEW.2), then that Discovery controller shall return records for only NVM subsystem ports that are presented through the same NVM subsystem port that received the Get Log Page command. If a Discovery controller returns records for only NVM subsystem ports that are presented through the same NVM subsystem port that received the Get Log Page command, then that Discovery controller shall set the Port Local (PORTLCL) bit to '1' in the DLPF field. Only a Direct Discovery controller (DDC) may support returning only port local Discovery Log Page Entries or port local Extended Discovery Log Page Entries.

If a Discovery controller supports returning records for all NVM subsystem ports that are registered (i.e., the All NVM Subsystems Entries Supported (ALLSUBES) bit is set to '1' in the LID Specific Parameter field

as defined in Figure NEW.1) and that Discovery controller processes a Get Log Page command for this log page with the All NVM Subsystem Entries (ALLSUBE) bit set to '1' in the LSP field (refer to Figure NEW.2), then that Discovery controller shall return records for all NVM subsystem ports that are registered. The list of records returned by that Discovery controller shall not be filtered based upon the requesting HOSTNQN. If a Discovery controller returns records for all NVM subsystem ports that are registered, then that Discovery controller shall set the All Subsystems (ALLSUB) bit to '1' in the DLPF field. A Discovery controller may require authentication before reporting the ALLSUBES bit being set to '1' or may require configuration outside the scope of this specification.

The Log Page Offset may be used to retrieve specific records. If the Discovery controller supports an index offset for this log page (i.e., the Index Offset Supported (IOS) bit is set to '1' in the Supported Log Pages log page (refer to section 5.16.1.1) for this log page), then the index values specified in the Log Page Offset Lower (LPOL) field (refer to Figure 199) and Log Page Offset Upper (LPOU) field (refer to Figure 200) in the Get Log Page command shall be used to index into the header and list of entries within this log page (e.g., specifying an index offset of 0 returns this log page starting at the header, specifying an index offset of 1 returns this log page starting at the offset of Entry 0, etc.). The number of records is returned in the header of the log page. The format for a Discovery Log Page Entry is defined in Figure 264. The format for an Extended Discovery Log Page Entry is defined in Figure NEW.3. The format for an extended attribute is defined in Figure NEW.12. The format for the Discovery ~~L~~log ~~P~~page is defined in Figure 265.

A single Get Log Page command used to read the Discovery ~~L~~log ~~P~~page shall be atomic. If ~~the host reads~~ the Discovery ~~L~~log ~~P~~page is read using multiple Get Log Page commands, then the ~~host~~reader should validate the Generation Counter field to ensure that there has not been a change in the contents of the data. The Discovery log page contents should be processed in the following sequence:

1. ~~The host should~~ read the Discovery ~~L~~log ~~P~~page contents in order (i.e., with increasing Log Page Offset values);
2. ~~and then~~ re-read the Generation Counter field after the entire log page is transferred~~.~~; and
3. ~~I~~if the Generation Counter field does not match the original value read, then ~~the host should~~ discard the log page read, as the entries may be inconsistent, and restart the sequence~~.~~; or
4. if the Generation Counter field does match the original value read, then process the returned data.

If the log page contents change during this command sequence, then the controller may return a status code of Discover Restart.

Every record indicates via the SUBTYPE field if that record is referring to another Discovery Service or if the record indicates an NVM subsystem composed of controllers that may expose namespaces. A referral to another Discovery Service (i.e., SUBTYPE 01h) is a mechanism to find additional Discovery subsystems. An NVM subsystem entry (i.e., SUBTYPE 02h) is a mechanism to find NVM subsystems that contain controllers that may expose namespaces. ~~Referrals to another Discovery Service shall not be~~Hosts or Discovery controllers are not required to process referral entries deeper than eight levels.

<Note to editor: the struck out purple text comes from TP8014.>

If an NVM subsystem supports the dynamic controller model, then all entries for that NVM subsystem shall have the Controller ID field set to FFFFh. For a particular NVM subsystem port and NVMe Transport address in an NVM subsystem, there shall be no more than one entry with the Controller ID field set to:

- FFFFh if that NVM subsystem supports the dynamic controller model; or
- FFFEh if that NVM subsystem supports the static controller model.

Figure NEW.2: Discovery – Log Specific Parameter Field

Bits	Description
14:11	Reserved
10	**All NVM Subsystem Entries (ALLSUBE):** If set to '1', then the Discovery controller shall return records for all NVM subsystem ports that are registered without filtering the list of returned records based upon the requesting HOSTNQN. If cleared to '0', then this bit has no effect.

Figure NEW.2: Discovery – Log Specific Parameter Field

Bits	Description
09	**Port Local Entries Only (PLEO):** If set to '1', then the Discovery controller shall return records for only NVM subsystem ports that are presented through the same NVM subsystem port that received the Get Log Page command. If cleared to '0', then this bit has no effect.
08	**Extended Discovery Log Page Entries (EXTDLPE):** If set to '1', then each record returned by the Discovery controller may be either a Discovery Log Page Entry or an Extended Discovery Log Page Entry. If cleared to '0', then this bit has no effect.

Figure 264: Get Log Page – Discovery Log Page Entry

Bytes	Description
…	**…**
09:08	**Admin Max SQ Size (ASQSZ):** Specifies the maximum size of an Admin Submission Queue. This applies to all controllers in the NVM subsystem. The value shall be a minimum of 32 entries.
11:10	**Entry Flags (EFLAGS):** This field indicates additional information related to the current entry. See the following bit definitions table: <Note to editor: this field comes from TP8014. TP8010 is adding the NCC bit to this field.>
31:~~10~~12	Reserved
63:32	**Transport Service Identifier (TRSVCID):** Specifies the NVMe Transport service identifier as an ASCII string. The NVMe Transport service identifier is specified by the associated NVMe Transport binding specification.
255:64	Reserved
511:256	**NVM Subsystem Qualified Name (SUBNQN):** NVMe Qualified Name (NQN) that uniquely identifies the NVM subsystem. Refer to section 4.4. For a Discovery Service, the value returned shall be the well-known Discovery Service NQN (nqn.2014-08.org.nvmexpress.discovery).
…	**…**

Entry Flags (EFLAGS) bit definitions:

Bit	Definiton
15:~~2~~3	Reserved
2	**No CDC Connectivity (NCC):** If set to '1', then no DDC that describes this entry is currently connected to the CDC. If cleared to '0', then at least one DDC that describes this entry is currently connected to the CDC. If the Discovery controller returning this log page is not a CDC, then this bit shall be cleared to '0' and should be ignored by the host.
1	**Explicit Persistent Connection Suppport for Discovery (EPCSD):** If set to '1', then Explicit Persistent Connections are supported for the discovery controller described by this entry. If cleared to '0', then support for Explicit Persistent Connections is not reported. For entries with the Subtype field set to 2h, this bit shall be cleared to '0'.
0	**Duplicate Returned Information (DUPRETINFO):** If set to '1', then using the content of this entry to access this Discovery Service returns the same information that is returned by using the content of other entries in this log page that also have this bit set to '1'. If cleared to '0', then using the content of this entry to access this Discovery Service may or may not return different information than is returned by using the content of any other entry in this log page. For entries with the Subtype field set to a value other than 3h, this bit shall be cleared to '0'.

Figure NEW.3: Get Log Page – Extended Discovery Log Page Entry

Bytes	Description
00	**Transport Type (TRTYPE):** This field indicates the transport type as defined in the Transport Type (TRTYPE) field in Figure 264.
01	**Address Family (ADRFAM):** This field indicates the address family as defined in the Address Family (ADRFAM) field in Figure 264.
02	**Subsystem Type (SUBTYPE):** This field indicates the type of the NVM subsystem that is indicated in this entry as defined in the Subsystem Type (SUBTYPE) field in Figure 264.
03	**Transport Requirements (TREQ):** This field indicates requirements for the NVMe Transport as defined in the Transport Requirements (TREQ) field in Figure 264.
05:04	**Port ID (PORTID):** This field indicates a particular NVM subsystem port as defined in the Port ID (PORTID) field in Figure 264.
07:06	**Controller ID (CNTLID):** This field indicates the controller ID as defined in the Controller ID (CNTLID) field in Figure 264.
09:08	**Admin Max SQ Size (ASQSZ):** This field indicates the maximum size of an Admin Submission Queue as defined in the Admin Max SQ Size (ASQSZ) field in Figure 264.
11:10	**Entry Flags (EFLAGS):** This field indicates additional information related to the current entry as defined in the Entry Flags (EFLAGS) field in Figure 264.
31:12	Reserved
63:32	**Transport Service Identifier (TRSVCID):** This field indicates the NVMe Transport service identifier as an ASCII string as defined in the Transport Service Identifier (TRSVCID) field in Figure 264.
255:64	Reserved
511:256	**NVM Subsystem Qualified Name (NQN):** This field indicates the NVMe Qualified Name (NQN) that uniquely identifies the NVM subsystem as defined in the NVM Subsystem Qualified Name (SUBNQN) field in Figure 264.
767:512	**Transport Address (TRADDR):** This field indicates the address of the NVM subsystem that may be used for a Connect command as an ASCII string as defined in the Transport Address (TRADDR) field in Figure 264.
1023:768	**Transport Specific Address Subtype (TSAS):** This field indicates NVMe Transport specific information about the address as defined in the Transport Specific Address Subtype (TSAS) field in Figure 264.
1027:1024	**Total Entry Length (TEL):** This field indicates the length in bytes of the entire Extended Discovery Log Page Entry.
1029:1028	**Number of Extended Attributes (NUMEXAT):** This field indicates the number of extended attributes contained in the Extended Discovery Log Page Entry.
1031:1030	Reserved
((EXATLEN - 1) + 4) + 1032: 1032	**Extended Attribute 0 (EXAT0):** This field contains the first extended attribute as defined in Figure NEW.12 (if present), where EXATLEN is the size specified in the Extended Attribute Length (EXATLEN) field of the extended attribute.
…	…

Figure NEW.3: Get Log Page – Extended Discovery Log Page Entry

Bytes	Description
TEL - 1: TEL - (EXATLEN + 4)	**Extended Attribute N (EXATN):** This field contains the Nth extended attribute as defined in Figure NEW.12 (if present), where EXATLEN is the size specified in the Extended Attribute Length (EXATLEN) field of the extended attribute and TEL is the size specified in the Total Entry Length (TEL) field.

Figure 265: Get Log Page – Discovery Log Page

Bytes	Description
	Header
07:00	**Generation Counter (GENCTR):** Indicates the version of the discovery information, starting at a value of 0h. For each change in the Discovery ~~L~~log ~~P~~page, this counter is incremented by one. If the value of this field is FFFFFFFF_FFFFFFFFh, then the field shall be cleared to 0h when incremented (i.e., rolls over to 0h).
15:08	**Number of Records (NUMREC):** Indicates the number of records contained in the log page.
17:16	**Record Format (RECFMT):** Specifies the format of the Discovery ~~L~~log ~~P~~page. If a new format is defined, this value is incremented by one. The format of the record specified in this definition shall be 0h.
18	**Discovery Log Page Flags (DLPF):** This field indicates additional information related to the Discovery log page. <table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:3</td><td>Reserved</td></tr><tr><td>2</td><td>**All Subsystems (ALLSUB):** If set to '1', then the records contained in the log page are for all NVM subsystem ports. If cleared to '0', then the records contained in the log page may or may not be for all NVM subsystem ports.</td></tr><tr><td>1</td><td>**Port Local (PORTLCL):** If set to '1', then the records contained in the log page are only for NVM subsystem ports that are presented through the same NVM subsystem port that is returning this log. If cleared to '0', then the records contained in the log page may be for NVM subsystem ports that are presented through any NVM subsystem port on that NVM subsystem.</td></tr><tr><td>0</td><td>**Extended (EXTEND):** If set to '1', then the records contained in the log page may include Extended Discovery Log Page Entries. If cleared to '0', then the records contained in the log page do not include Extended Discovery Log Page Entries.</td></tr></table>
19	Reserved
23:20	**Total Discovery Log Page Length (TDLPL):** This field indicates the length in bytes of the entire Discovery log page. If the value in this field is cleared to 0h, then this field is not reported.
1023:~~18~~24	Reserved

Figure 265: Get Log Page – Discovery Log Page

Bytes	Description
	Entries
~~2047~~(TEL - 1) + 1024:1024	~~**Discovery Log Page**~~ **Entry 0 (**~~DL~~**E0):** Contains either the first Discovery Log Page Entry as defined in Figure 264 or the first Extended Discovery Log Page Entry as defined in Figure NEW.3. TEL is the size indicated in the Total Entry Length (TEL) field of the Extended Discovery Log Page Entry. For Discovery Log Page Entries, TEL shall be 1024 bytes.
~~3071:2048~~	~~**Discovery Log Page Entry 1 (DLE1):** Contains the second Discovery Log Page Entry as defined in Figure 264 (if present).~~
…	…
~~(((N + 2) × 1024) - 1): ((N + 1) × 1024)~~TDLPL - 1:TDLPL - TEL	~~**Discovery Log Page**~~ **Entry N (**~~DL~~**EN):** Contains either the Nth Discovery Log Page Entry as defined in Figure 264 (if present) or the Nth Extended Discovery Log Page Entry as defined in Figure NEW.3 (if present). TEL is the size indicated in the Total Entry Length (TEL) field of the Extended Discovery Log Page Entry and TDLPL is the size indicated in the Total Discovery Log Page Length (TDLPL) field. For Discovery Log Page Entries, TEL shall be 1024 bytes. If the TDLPL field is reserved (i.e., this log page does not contain any Extended Discovery Log Page Entries), then the byte position of the Nth Discovery Log Page Entry shall be: (((N + 2) * 1024) - 1):((N + 1) * 1024).

…

5.16.1.NEW Host Discovery (Log Identifier 71h)

The Host Discovery log page shall only be supported by Discovery controllers. The Host Discovery log page shall not be supported by controllers that expose namespaces for NVMe over PCIe or NVMe over Fabrics. The Host Discovery log page provides an inventory of hosts that have registered discovery information with the Discovery controller. A host or a Discovery controller may obtain the host inventory from this log page. Depending on the parameters specified in a Get Log Page command, the Host Discovery log page may contain entries for:

- all known hosts; or
- a subset of entries specific to the host or Discovery controller requesting the log page.

Each Host Extended Discovery Log Page Entry shall contain at least one extended attribute containing a Host Identifier.

<Note to editor: LID Specific Parameter field support bits now live in each corresponding log page section (instead of LID 00h section) per TP4109.>

Figure NEW.4 specifies the format for the LID Specific Parameter field in the Supported Log Pages log page (refer to section 5.16.1.1) for the Host Discovery log page.

Figure NEW.4: Host Discovery – LID Specific Parameter Field

Bits	Description
15:1	Reserved
0	**All Host Entries Supported (ALLHOSTES):** If set to '1', then the Discovery controller supports returning records for all hosts that are registered without filtering the list of returned records based upon the requesting HOSTNQN. If cleared to '0', then the Discovery controller does not support returning records for all hosts that are registered.

If a Discovery controller supports returning records for all hosts that are registered (i.e., the All Host Entries Supported (ALLHOSTES) bit is set to '1' in the LID Specific Parameter field as defined in Figure NEW.4)

and that Discovery controller processes a Get Log Page command for this log page with the All Host Entries (ALLHOSTE) bit set to '1' in the Log Specific Parameter field (refer to Figure NEW.5), then that Discovery controller shall return records for all hosts that are registered. The list of records returned by the Discovery controller shall not be filtered based upon the requesting HOSTNQN. If a Discovery controller returns records for all hosts that are registered, then that Discovery controller shall set the All Hosts (ALLHOST) bit to '1' in the Host Discovery Log Page Flags (HDLPF) field. A Discovery controller may require authentication before reporting the ALLHOSTES bit being set to '1' or may require configuration outside the scope of this specification.

The Log Page Offset may be used to retrieve specific records. If the Discovery controller supports an index offset for this log page (i.e., the Index Offset Supported (IOS) bit is set to '1' in the Supported Log Pages log page (refer to section 5.16.1.1) for this log page), then the index values specified in the Log Page Offset Lower (LPOL) field (refer to Figure 199) and Log Page Offset Upper (LPOU) field (refer to Figure 200) in the Get Log Page command shall be used to index into the header and list of entries within this log page (e.g., specifying an index offset of 0 returns this log page starting at the header, specifying an index offset of 1 returns this log page starting at the offset of Entry 0, etc.). The number of records is returned in the header of the log page. The format for a Host Extended Discovery Log Page Entry is defined in Figure NEW.6. The format for an extended attribute is defined in Figure NEW.12. The format for the Host Discovery log page is defined in Figure NEW.7.

A single Get Log Page command used to read the Host Discovery log page shall be atomic. If the Host Discovery log page is read using multiple Get Log Page commands, then the reader should validate the Generation Counter field to ensure that there has not been a change in the contents of the data. The Host Discovery log page contents should be processed in the following sequence:

1. read the Host Discovery log page contents in order (i.e., with increasing Log Page Offset values);
2. re-read the Generation Counter field after the entire log page is transferred; and
3. if the Generation Counter field does not match the original value read, then discard the log page read, as the entries may be inconsistent, and restart the sequence; or
4. if the Generation Counter field does match the original value read, then process the returned data.

If the log page contents change during this command sequence, then the Discovery controller may return a status code of Discover Restart.

Figure NEW.5: Host Discovery – Log Specific Parameter Field

Bits	Description
14:9	Reserved
08	**All Host Entries (ALLHOSTE):** If set to '1', then the Discovery controller shall return records for all hosts that are registered without filtering the list of returned records based upon the requesting HOSTNQN. If cleared to '0', then this bit has no effect.

Figure NEW.6: Host Extended Discovery Log Page Entry

Bytes	Description
00	**Transport Type (TRTYPE):** This field indicates the transport type as defined in the Transport Type (TRTYPE) field in Figure 264.
01	**Address Family (ADRFAM):** This field indicates the address family as defined in the Address Family (ADRFAM) field in Figure 264.

Figure NEW.6: Host Extended Discovery Log Page Entry

Bytes	Description
09:02	Reserved
11:10	**Entry Flags (EFLAGS):** This field indicates additional information related to the current entry. <table><tr><td>**Bits**</td><td>**Description**</td></tr><tr><td>15:3</td><td>Reserved</td></tr><tr><td>2</td><td>**No CDC Connectivity (NCC):** If set to '1', then the host that describes this entry is not currently connected to the CDC. If cleared to '0', then the host that describes this entry is currently connected to the CDC. If the Discovery controller returning this log page is not a CDC, then this bit shall be cleared to '0' and should be ignored by the host.</td></tr><tr><td>1:0</td><td>Reserved</td></tr></table>
255:12	Reserved
511:256	**Host NVMe Qualified Name (HOSTNQN):** This field indicates the NVMe Qualified Name (NQN) that uniquely identifies the host. Refer to section 4.5 for the formatting requirments for NQNs.
767:512	**Transport Address (TRADDR):** This field indicates the address of the host that may be used for a Connect command as an ASCII string. The Address Family field describes the reference for parsing this field. Refer to section 1.4.2 for ASCII string requirements. For the definition of this field, refer to the appropriate NVMe Transport binding specification.
1023:768	**Transport Specific Address Subtype (TSAS):** This field indicates NVMe Transport specific information about the address as defined in the Transport Specific Address Subtype (TSAS) field in Figure 264.
1027:1024	**Total Entry Length (TEL):** This field indicates the length in bytes of the entire Host Extended Discovery Log Page Entry.
1029:1028	**Number of Extended Attributes (NUMEXAT):** This field indicates the number of extended attributes contained in the Host Extended Discovery Log Page Entry. This field shall be non-zero (i.e., the Host Extended Discovery Log Page Entry contains at least one extended attribute containing a Host Identifier as described in this section).
1031:1030	Reserved
((EXATLEN - 1) + 4) + 1032:1032	**Extended Attribute 0 (EXAT0):** This field contains the first extended attribute as defined in Figure NEW.12, where EXATLEN is the size specified in the Extended Attribute Length (EXATLEN) field of the extended attribute.
…	…
TEL - 1:TEL - (EXATLEN + 4)	**Extended Attribute N (EXATN):** This field contains the Nth extended attribute as defined in Figure NEW.12 (if present), where EXATLEN is the size specified in the Extended Attribute Length (EXATLEN) field of the extended attribute and TEL is the size specified in the Total Entry Length (TEL) field.

Figure NEW.7: Host Discovery Log Page

Bytes	Description
Header	
07:00	**Generation Counter (GENCTR):** This field indicates the version of the discovery information, starting at a value of 0h. For each change in the Host Discovery log page, this counter is incremented by one. If the value of this field is FFFFFFFF_FFFFFFFFh, then the field shall be cleared to 0h when incremented (i.e., rolls over to 0h).
15:08	**Number of Records (NUMREC):** Indicates the number of records contained in the log page.
17:16	**Record Format (RECFMT):** This field indicates the format of the Host Discovery log page. If a new format is defined, this value is incremented by one. The format of the record specified in this definition shall be 0h.
18	**Host Discovery Log Page Flags (HDLPF):** This field indicates additional information related to the Host Discovery log page. <table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:1</td><td>Reserved</td></tr><tr><td>0</td><td>**All Hosts (ALLHOST):** If set to '1', then the records contained in the log page are for all hosts. If cleared to '0', then the records contained in the log page may or may not be for all hosts.</td></tr></table>
19	Reserved
23:20	**Total Host Discovery Log Page Length (THDLPL):** This field indicates the length in bytes of the entire Host Discovery log page.
1023:24	Reserved
Entries	
(TEL - 1) + 1024:1024	**Host Extended Discovery Log Page Entry 0 (HEDLPE0):** This field contains the first Host Extended Discovery Log Page Entry as defined in Figure NEW.6. TEL is the size indicated in the Total Entry Length (TEL) field of the Host Extended Discovery Log Page Entry.
…	…
THDLPL - 1:THDLPL - TEL	**Host Extended Discovery Log Page Entry N (HEDLPEN):** This field contains the Nth Host Extended Discovery Log Page Entry as defined Figure NEW.6 (if present). TEL is the size indicated in the Total Entry Length (TEL) field of the Host Extended Discovery Log Page Entry and THDLPL is the size indicated in the Total Host Discovery Log Page Length (THDLPL) field.

…

5.17.2.1 Identify Controller data structure (CNS 01h)

…

Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O[1]	Admin[1]	Disc[1]	Description
…	…	…	…	**…**

Bytes	I/O[1]	Admin[1]	Disc[1]	Description
1805:1804	M[2]	M[2]	R	**Optional Fabric Commands Support (OFCS):** Indicate whether the controller supports optional fabric commands. Bits 15:1 are reserved. Bit 0 if cleared to '0' then the controller does not support the Disconnect command. Bit 0 if set to '1' then the controller supports the Disconnect command and deletion of individual I/O Queues.
1806	R	R	O	**Discovery Controller Type (DCTYPE):** This field indicates what type of Discovery controller the controller is.

Value	Definition
0h	Discovery controller type is not reported
1h	DDC
2h	CDC
3h to FFh	Reserved

|
| 2047:~~1806~~1807 | | | | Reserved |
| … | … | … | … | **…** |

NOTES:

1. O/M/R definition: O = Optional, M = Mandatory, R = Reserved.

2. Mandatory for I/O controllers using a message-based transport. Reserved for I/O controllers using a memory-based transport.

…

5.27.1.8 Asynchronous Event Configuration (Feature Identifier 0Bh)

…

Figure 326: Asynchronous Event Configuration – Command Dword 11

Bits	Description
31	**Discovery Log Page Change Notification:** This bit indicates that the Discovery controller reports Discovery Log Page Change Notifications. If set to '1', the Discovery controller shall send a notification if Discovery ~~L~~log ~~P~~page changes occur.
30	**Host Discovery Log Page Change Notification:** This bit indicates that the Discovery controller reports Host Discovery Log Page Change Notifications. If set to '1', then the Discovery controller shall send a notification if Host Discovery log page changes occur.
~~30~~29:28	Reserved
…	**…**

NOTE:
1. Refer to the NVM Command Set specification.
2. Refer to the Zoned Namespace Command Set specification.

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

5.NEW1 Discovery Information Management command

The Discovery Information Management command registers, de-registers, or updates discovery information entries.

The type of task performed by the Discovery Information Management command (i.e., registration, de-registration, or update) is determined by the value set in Task (TAS) field of Command Dword 10 (refer to Figure NEW.9).

For a register task (i.e., TAS field cleared to 0h), the Discovery Information Management command registers:

- one or more host discovery information entries; or
- one or more NVM subsystem discovery information entries.

For a de-register task (i.e., TAS field set to 1h), the Discovery Information Management command de-registers:

- one or more host discovery information entries; or
- one or more NVM subsystem discovery information entries.

For an update task (i.e., TAS field set to 2h), the Discovery Information Management command updates:

- one host discovery information entry; or
- one NVM subsystem discovery information entry.

Discovery information entries may be one of the following:

1. a basic discovery information entry (refer to Figure 264); or
2. an extended discovery information entry (refer to Figure NEW.11).

Host discovery information entries shall be extended discovery information entries.

NVM subsystem discovery information entries may be basic discovery information entries or may be extended discovery information entries. Each NVM subsystem discovery information entry may specify an NVM subsystem that exposes namespaces that hosts may access or may specify a referral to another Discovery subsystem.

The Discovery Information Management command uses the Data Pointer field as shown in Figure NEW.8 and Command Dword 10 as shown in Figure NEW.9. All other command specific fields are reserved.

The data portion of the Discovery Information Management command contains a header that identifies the entity performing the register, de-register, or update task.

If a register task is being performed and the data portion of the Discovery Information Management command does not contain one or more discovery information entries, then the controller shall abort the command with status code of Invalid Field in Command. If the number of discovery information entries contained in the data portion of the Discovery Information Management command exceeds the available capacity for new discovery information entries on the CDC or DDC, then the controller shall abort the command with a status code of Insufficient Discovery Resources. If multiple register tasks are performed by the same entity (i.e., the value set in the Entity Identifier (EID) field of the header is associated with an existing registration record contained in the CDC or DDC), and:

a) the Entry Key associated with a discovery information entry in the Discovery Information Management command matches the Entry Key associated with an existing registration record, then the existing registration record contained in the CDC or DDC shall be updated with the discovery information entry from the Discovery Information Management command; or

b) the Entry Key associated with a discovery information entry in the Discovery Information Management command does not match the Entry Key associated with an existing registration record, then the discovery information entry in the Discovery Information Management command shall be registered with the CDC or DDC.

If a de-register task is being performed and the data portion of the Discovery Information Management command does not contain one or more discovery information entries, then the controller shall abort the command with a status code of Invalid Field in Command.

If an update task is being performed and the data portion of the Discovery Information Management command does not contain two discovery information entries, then the controller shall abort the command with a status code of Invalid Field in Command. The first discovery information entry identifies the registration record contained in the CDC or DDC that is to be updated. The fields in the first discovery information entry that are not used as part of the Entry Key are ignored. The second discovery information entry replaces the existing registration record identified by the Entry Key from the first discovery information entry. The update task shall be atomic.

The format for the data portion of the Discovery Information Management command is defined in Figure NEW.10.

Based upon the value set in the Entity Type (ETYPE) field of the header, the data portion of a Discovery Information Management command shall:

- only contain host extended discovery information entries if the ETYPE field is set to 1h (i.e., a host is performing the register, de-register, or update task);
- only contain host extended discovery information entries if the ETYPE field is set to 3h (i.e., a CDC is performing the register, de-register, or update task); and
- only contain NVM subsystem basic discovery information entries or NVM subsystem extended discovery information entries if the ETYPE field is set to 2h (i.e., a DDC is performing the register, de-register, or update task). The DDC may set the Port Local (PORTLCL) field to 1h if the NVM subsystem discovery information entries being registered, de-registered, or updated are only for NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that is performing the register, de-register, or update task.

Host extended discovery information entries and NVM subsystem extended discovery information entries each contain the same set of fields, but not all of the fields are used for both extended discovery information entry types. Refer to Figure NEW.13 for the usage of the extended discovery information entry fields for each extended discovery information entry type. If the entity performing a register, de-register, or update task uses any field in an extended discovery information entry that conflicts with the value set in the ETYPE field of the Discovery Information Management command data portion's header (e.g., a host uses a field intended for only NVM subsystem extended discovery information entries), then the controller shall abort the command with a status code of Invalid Discovery Information.

Host extended discovery information entries shall contain at least one extended attribute containing a Host Identifier. NVM subsystem extended discovery information entries may contain zero or more extended attributes. The format for an extended attribute is defined in Figure NEW.12.

Figure NEW.8: Discovery Information Management – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure NEW.9: Discovery Information Management – Command Dword 10

Bits	Description
31:04	Reserved

Figure NEW.9: Discovery Information Management – Command Dword 10

Bits	Description
03:00	**Task (TAS):** This field selects the type of management task to perform. | Value	Description | |-------|-------------| | 0h | Register | | 1h | De-Register | | 2h | Update | | 3h to Fh | Reserved |

Figure NEW.9: Discovery Information Management – Command Dword 10

Bits	Description
03:00	**Task (TAS):** This field selects the type of management task to perform.

Value	Description
0h	Register
1h	De-Register
2h	Update
3h to Fh	Reserved

Figure NEW.10: Discovery Information Management – Data

Bytes	O/M [1]	Description
		Header
03:00	M	**Total Data Length (TDL):** This field specifies the length in bytes of the entire data portion of the command.
07:04		Reserved
15:08	M	**Number of Entries (NUMENT):** This field specifies the number of discovery information entries being registered, de-registered, or updated. For a register or de-register task, this field shall be non-zero. For an update task, this field shall be set to 2h.
17:16	M	**Entry Format (ENTFMT):** This field specifies the format of the discovery information entries being registered, de-registered, or updated.

Value	Definition
0h	Reserved
1h	Basic discovery information entry
2h	Extended discovery information entry
3h to FFFFh	Reserved

Bytes	O/M [1]	Description
19:18	M	**Entity Type (ETYPE):** This field specifies the type of entity performing the register, de-register, or update task.

Value	Definition
0h	Reserved
1h	Host
2h	Direct Discovery controller
3h	Centralized Discovery controller
4h to FFFFh	Reserved

Figure NEW.10: Discovery Information Management – Data

Bytes	O/M [1]	Description
20	O[2]	**Port Local (PORTLCL):** This field specifies if the NVM subsystem discovery information entries being registered, de-registered, or updated are only for NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that is performing the register, de-register, or update task. If the Entity Type (ETYPE) field is set to any value other than 2h (i.e., an entity other than a DDC is performing the register, de-register, or update task) and this field is not cleared to 0h, then the controller shall abort the command with a status code of Invalid Field in Command. If set to 1h, then the NVM subsystem discovery information entries being registered, de-registered, or updated are only for NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that is performing the register, de-register, or update task. If cleared to 0h, then the NVM subsystem discovery information entries being registered, de-registered, or updated may be for NVM subsystem ports that are presented through any NVM subsystem port on that DDC.
21		Reserved
23:22	M	**Entry Key Type (EKTYPE):** This field specifies the fields in each discovery information entry that shall be used to determine if the entry matches an existing registration record contained in the CDC or DDC. Host discovery information entries shall use an EKTYPE of 5Fh (i.e., TRADDR Based). NVM subsystem discovery information entries may use an EKTYPE of either 3Fh (i.e., Port ID Based) or 5Fh (i.e., TRADDR Based). All other values are reserved.
279:24	M	**Entity Identifier (EID):** This field specifies an NQN that is used to uniquely identify the entity performing the registration, de-registration, or update task within the fabric in UUID-based format. Refer to section 4.5 for the formatting requirements of UUID-based format NQNs.

Bits	Definition	Port ID Based (3Fh)	TRADDR Based (5Fh)
15:7	Reserved		
6	Transport Address (TRADDR)	Not used	Used
5	Port ID (PORTID)	Used	Not used
4	Transport Type (TRTYPE)	Used	Used
3	Address Family (ADRFAM)	Used	Used
2	Transport Service Identifier (TRSVCID)	Used	Used
1	Transport Specific Address Subtype (TSAS)	Used	Used
0	NVMe Qualified Name (NQN)	Used	Used

Figure NEW.10: Discovery Information Management – Data

Bytes	O/M [1]	Description
535:280	O	**Entity Name (ENAME):** This field specifies the name associated with the entity as an ASCII string (refer to section 1.4.2 for ASCII string requirements). The value specified in this field is determined by the value set in the Entity Type (ETYPE) field. If the ETYPE field is set to 1h (i.e., a host is performing the register, de-register, or update task), then this field specifies the name associated with the host. If the ETYPE field is set to 2h (i.e., a DDC is performing the register, de-register, or update task), then this field specifies the name associated with the DDC. If the ETYPE field is set to 3h (i.e., a CDC is performing the register, de-register, or update task), then this field specifies the name associated with the CDC.
599:536	O	**Entity Version (EVER):** This field specifies the entity version as an ASCII string (refer to section 1.4.2 for ASCII string requirements). The value specified in this field is determined by the value set in the Entity Type (ETYPE) field. If the ETYPE field is set to 1h (i.e., a host is performing the register, de-register, or update task), then this field specifies the host entity version (e.g., host operating system name and version). If the ETYPE field is set to 2h (i.e., a DDC is performing the register, de-register, or update task), then this field specifies the Discovery controller entity version (e.g., currently active firmware revision of the DDC). If the ETYPE field is set to 3h (i.e., a CDC is performing the register, de-register, or update task), then this field may specify either: a) the host entity version (e.g., host operating system name and version); or b) the Discovery controller entity version (e.g., currently active firmware revision of the CDC).
1023:600		Reserved
Entries		
(TEL - 1) + 1024: 1024	M	**Discovery Information Entry 0 (DIE0):** This field contains the first discovery information entry as defined in Figure NEW.11 for extended discovery information entries and as defined in Figure 264 for basic discovery information entries, where TEL is the size specified in the Total Entry Length (TEL) field of the extended discovery information entry. For basic discovery information entries, TEL shall be 1024 bytes.
…	O	…
TDL - 1: TDL - TEL	M[3]	**Discovery Information Entry N (DIEN):** This field contains the Nth discovery information entry as defined in Figure NEW.11 for extended discovery information entries (if present) and as defined in Figure 264 for basic discovery information entries (if present), where TEL is the size specified in the Total Entry Length (TEL) field of the extended discovery information entry and TDL is the size specified in the Total Data Length (TDL) field. For basic discovery information entries, TEL shall be 1024 bytes.

NOTES:

1. O/M definition: O = Optional, M = Mandatory.

2. This field is optional for a register, de-register, or update task performed by a DDC. For a register, de-register, or update task performed by a host or CDC, this field shall be cleared to 0h.

3. This field is mandatory for an update task. For a register or de-register task, this field is optional.

Figure NEW.11: Extended Discovery Information Entry

Bytes	O/M [1]	Description
00	M	**Transport Type (TRTYPE):** This field specifies the transport type as defined in the Transport Type (TRTYPE) field in Figure 264.
01	M	**Address Family (ADRFAM):** This field specifies the address family as defined in the Address Family (ADRFAM) field in Figure 264.
02	M [2]	**Subsystem Type (SUBTYPE):** This field specifies the type of the NVM subsystem that is indicated in this entry as defined in the Subsystem Type (SUBTYPE) field in Figure 264.
03	M [2]	**Transport Requirements (TREQ):** This field specifies requirements for the NVMe Transport as defined in the Transport Requirements (TREQ) field in Figure 264.
05:04	M [2]	**Port ID (PORTID):** This field specifies a particular NVM subsystem port as defined in the Port ID (PORTID) field in Figure 264.
07:06	M [2]	**Controller ID (CNTLID):** This field specifies the controller ID as defined in the Controller ID (CNTLID) field in Figure 264. This field shall specify a controller ID that a host is able to use in a Connect command to the NVM subsystem being registered.
09:08	M [2]	**Admin Max SQ Size (ASQSZ):** This field specifies the maximum size of an Admin Submission Queue as defined in the Admin Max SQ Size (ASQSZ) field in Figure 264.
31:10		Reserved
63:32	M [2]	**Transport Service Identifier (TRSVCID):** This field specifies the NVMe Transport service identifier as an ASCII string as defined in the Transport Service Identifier (TRSVCID) field in Figure 264.
255:64		Reserved
511:256	M	**NVMe Qualified Name (NQN):** If the Entity Type (ETYPE) field in the Discovery Information Management command data portion's header is set to 2h (i.e., a DDC is performing the register, de-register, or update task), then this field specifies the NVMe Qualified Name (NQN) that uniquely identifies the NVM subsystem as defined in the NVM Subsystem Qualified Name (SUBNQN) field in Figure 264. If the ETYPE field in the Discovery Information Management command data portion's header is set to 1h or 3h (i.e., a host or CDC is performing the register, de-register, or update task), then this field specifies the NQN that uniquely identifies the host as defined in the Host NVMe Qualified Name (HOSTNQN) field in Figure NEW.6.

Figure NEW.11: Extended Discovery Information Entry

Bytes	O/M [1]	Description
767:512	M	**Transport Address (TRADDR):** If the Entity Type (ETYPE) field in the Discovery Information Management command data portion's header is set to 2h (i.e., a DDC is performing the registration, de-registration, or update task), then this field specifies the address of a fabric interface on the NVM subsystem that may be used for a Connect command as an ASCII string as defined in the Transport Address (TRADDR) field in Figure 264. If the ETYPE field in the Discovery Information Management command data portion's header is set to 1h or 3h (i.e., a host or CDC is performing the registration, de-registration, or update task), then this field specifies the address of a fabric interface on the host that may be used for a Connect command as an ASCII string as defined in the Transport Address (TRADDR) field in Figure NEW.6. If the first byte (i.e., byte 512) of this field is NULL (i.e., cleared to 00h), then the TRADDR value used by the controller shall be the remote IP address associated with the connection used to transport the Discovery Information Management command. If the host attempts to register or de-register multiple discovery information entries with the first byte of this field containing a NULL, then the controller shall abort the command with a status code of Invalid Field in Command.
1023:768	M	**Transport Specific Address Subtype (TSAS):** This field specifies NVMe Transport specific information about the address as defined in the Transport Specific Address Subtype (TSAS) field in Figure 264.
1027:1024	M	**Total Entry Length (TEL):** This field specifies the length in bytes of the entire extended discovery information entry.
1029:1028	M[3]	**Number of Extended Attributes (NUMEXAT):** This field specifies the number of extended attributes contained in the extended discovery information entry. This field shall be set to a non-zero value (i.e., the extended discovery information entry shall contain at least one extended attribute) if the Entity Type (ETYPE) field in the Discovery Information Registration command data portion's header is set to 1h or 3h (i.e., a host or CDC is performing the register, de-register, or update task). If the ETYPE field is set to 1h or 3h and this field is cleared to 0h, then the controller shall abort the command with a status code of Invalid Discovery Information.
1031:1030		Reserved
((EXATLEN - 1) + 4) + 1032: 1032	M[3]	**Extended Attribute 0 (EXAT0):** This field contains the first extended attribute as defined in Figure NEW.12 (if present), where EXATLEN is the size specified in the Extended Attribute Length (EXATLEN) field of the extended attribute.
…	O	…

Figure NEW.11: Extended Discovery Information Entry

Bytes	O/M [1]	Description
TEL - 1: TEL - (EXATLEN + 4)	O	**Extended Attribute N (EXATN):** This field contains the Nth extended attribute as defined in Figure NEW.12 (if present), where EXATLEN is the size specified in the Extended Attribute Length (EXATLEN) field of the extended attribute and TEL is the size specified in the Total Entry Length (TEL) field.

NOTES:

1. O/M definition: O = Optional, M = Mandatory.

2. This field is mandatory for NVM subsystem discovery information entries. For host discovery information entries, this field shall be cleared to 0h.

3. This field is mandatory for host discovery information entries and optional for NVM subsystem discovery information entries.

Figure NEW.12: Extended Attribute

Bytes	Description
01:00	**Extended Attribute Type (EXATTYPE):** This field specifies the type of extended attribute for the extended attribute contained in the extended discovery information entry. {table1}
03:02	**Extended Attribute Length (EXATLEN):** This field specifies the length of the Extended Attribute Value (EXATVAL) field for the extended attribute contained in the extended discovery information entry. The length specified in this field shall be a non-zero value that is a multiple of four, and is either a fixed length or within a variable range based upon the value set in the Extended Attribute Type (EXATTYPE) field. If the length specified in this field is not a multiple of four, then the controller shall abort the command with a status code of Invalid Field in Command. {table2}

table1:

Value	O/M [1]	Definition
0h		Reserved
1h	HM	Host Identifier
2h	O	Symbolic Name
3h to FEFFh		Reserved
FF00h to FFFFh	O	Vendor Specific

table2:

Extended Attribute Type	Length
Host Identifier	16 bytes
Symbolic Name	4 to 256 bytes

Figure NEW.12: Extended Attribute

Bytes	Description
(EXATLEN - 1) + 4:04	**Extended Attribute Value (EXATVAL):** This field specifies the value for the extended attribute contained in the extended discovery information entry. The value specified in this field is based upon the value set in the Extended Attribute Type (EXATTYPE) field. Unused bytes, if any, shall be cleared to 0h. <table><tr><th>Extended Attribute Type</th><th>Value</th></tr><tr><td>Host Identifier</td><td>This value specifies the Host Identifier of the host, as defined in section 5.27.1.25.</td></tr><tr><td>Symbolic Name</td><td>This value specifies the symbolic name of the host or NVM subsystem. The symbolic name is used to identify a host or an NVM subsystem.</td></tr></table>
NOTES: 1. O/M definition: O = Optional, M = Mandatory, HM = Mandatory for hosts and prohibited for NVM subsystems.	

Figure NEW.13: Extended Discovery Information Entry – Applicable Fields

Field Name	Extended Discovery Information Entry Type:	
	Host	**NVM Subsystem**
Transport Type (TRTYPE)	Used	Used
Address Family (ADRFAM)	Used	Used
Subsystem Type (SUBTYPE)	Ignored[1]	Used
Transport Requirements (TREQ)	Ignored[1]	Used
Port ID (PORTID)	Ignored[1]	Used
Controller ID (CNTLID)	Ignored[1]	Used
Admin Max SQ Size (ASQSZ)	Ignored[1]	Used
Transport Service Identifier (TRSVCID)	Ignored[1]	Used
NVMe Qualified Name (NQN)	Used	Used
Transport Address (TRADDR)	Used	Used
Transport Specific Address Subtype (TSAS)	Used	Used
Total Entry Length (TEL)	Used	Used
Number of Extended Attributes (NUMEXAT)	Used	Used
Extended Attribute 0 (EXAT0)…Extended Attribute N (EXATN)	Used	Used

Figure NEW.13: Extended Discovery Information Entry – Applicable Fields

Field Name	Extended Discovery Information Entry Type:	
	Host	**NVM Subsystem**
NOTES:		
1. This field shall be cleared to 0h.		

5.NEW1.1 Command Completion

Upon completion of the Discovery Information Management command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status of the command. Discovery Information Management command specific status values are defined in Figure NEW.14.

Figure NEW.14: Discovery Information Management – Command Specific Status Values

Value	Description
2Fh	**Invalid Discovery Information:** The discovery information provided in one or more extended discovery information entries is not applicable for the type of entity selected in the Entity Type (ETYPE) field of the Discovery Information Management command data portion's header. Refer to Figure NEW.13 for restrictions, if any, that apply to each field in the extended discovery information entry based upon the type of extended discovery information entry being registered, de-registered, or updated.
32h	**Insufficient Discovery Resources:** The number of discovery information entries provided in the data portion of the Discovery Information Management command for a registration task (i.e., TAS field cleared to 0h) exceeds the available capacity for new discovery information entries on the CDC or DDC. This may be a transient condition.

5.NEW2 Fabric Zoning Lookup command

The Fabric Zoning Lookup (FZL) command is used to lookup a key associated with a Zoning data structure. The FZL command uses the Data Pointer field, as shown in Figure NEW.15.

Figure NEW.15: Fabric Zoning Lookup (FZL) – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

5.NEW2.1 Command Completion

Upon completion of the Fabric Zoning Lookup (FZL) command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status of the command. Command specific status values for the FZL command are defined in Figure NEW.16.

Figure NEW.16: Fabric Zoning Lookup (FZL) – Command Specific Status Values

Value	Description
30h	**Zoning Data Structure Locked:** The requested Zoning data structure is locked on the CDC.

Value	Description
31h	**Zoning Data Structure Not Found:** The requested Zoning data structure does not exist on the CDC.
33h	**Requested Function Disabled:** Fabric Zoning is not enabled on the CDC.

The key associated with the Zoning data structure that matches the specified FZL data (refer to section 8.NEW.3.8) is returned in the Completion Queue Entry Dword 0, as shown in Figure NEW.17.

Figure NEW.17: Fabric Zoning Lookup (FZL) – Completion Queue Entry Dword 0

Bits	Description
31:00	**Zoning Data Key (ZDK):** The key associated with the Zoning data structure that matches the specified FZL data.

5.NEW3 Fabric Zoning Send command

The Fabric Zoning Send (FZS) command is used to transfer a Zoning data structure to the CDC. The FZS command uses the Data Pointer, Command Dword 10, Command Dword 11, and Command Dword 12 fields, as shown in Figure NEW.18, Figure NEW.19, Figure NEW.20, and Figure NEW.21 respectively.

The CDC may check if the NQN contained in the ZoneGroup Originator field of the transferred ZoneGroup data structure matches the NQN contained in the HOSTNQN field of the Connect command sent from the DDC to that CDC. If the CDC performs this check and finds that the NQN contained in the ZoneGroup Originator field of the transferred ZoneGroup data structure does not match the NQN contained in the HOSTNQN field of the Connect command sent from the DDC to that CDC, then that CDC may abort the FZS command with a status code of ZoneGroup Originator Invalid.

Figure NEW.18: Fabric Zoning Send (FZS) – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure NEW.19: Fabric Zoning Send (FZS) – Command Dword 10

Bits	Description
31:00	**Zoning Data Key (ZDK):** This field specifies the key identifying a Zoning data structure in the Zoning database of the CDC.

Figure NEW.20: Fabric Zoning Send (FZS) – Command Dword 11

Bits	Description
31:00	**Zoning Data Offset (ZDO):** This field specifies the byte offset within a Zoning data structure to store the transferred data. The offset shall be dword aligned, indicated by bits 1:0 being cleared to 00b. The CDC is not required to check that bits 1:0 are cleared to 00b. The CDC may return a status code of Invalid Field in Command if bits 1:0 are not cleared to 00b. If the CDC does not return a status code of Invalid Field in Command, then the CDC shall operate as if bits 1:0 are cleared to 00b.

Figure NEW.21: Fabric Zoning Send (FZS) – Command Dword 12

Bits	Description
31	**Last Fragment (LF):** This bit specifies if the transferred data buffer contains the last fragment of this Zoning data structure. If set to '1', then the transferred data buffer contains the last fragment; if cleared to '0', then the transferred data buffer does not contain the last fragment.
30:28	Reserved
27:00	**Number of Dwords (NUMD):** This field specifies the number of dwords to transfer. If this field is cleared to 0h, then the controller shall abort the command with a status code of Invalid Field in Command.

5.NEW3.1 Command Completion

Upon completion of the Fabric Zoning Send (FZS) command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status of the command. Command specific status values for the FZS command are defined in Figure NEW.22.

Figure NEW.22: Fabric Zoning Send (FZS) – Command Specific Status Values

Value	Description
30h	**Zoning Data Structure Locked:** The requested Zoning data structure is locked on the CDC.
31h	**Zoning Data Structure Not Found:** The requested Zoning data structure does not exist on the CDC.
33h	**Requested Function Disabled:** Fabric Zoning is not enabled on the CDC.
34h	**ZoneGroup Originator Invalid:** The NQN contained in the ZoneGroup Originator field does not match the Host NQN used by the DDC to connect to the CDC.

5.NEW4 Fabric Zoning Receive command

The Fabric Zoning Receive (FZR) command is used to request a Zoning data structure from the CDC. The FZR command uses the Data Pointer, Command Dword 10, Command Dword 11, and Command Dword 12 fields, as shown in Figure NEW.23, Figure NEW.24, Figure NEW.25, and Figure NEW.26 respectively.

Figure NEW.23: Fabric Zoning Receive (FZR) – Data Pointer

Bits	Description
127:0	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure NEW.24: Fabric Zoning Receive (FZR) – Command Dword 10

Bits	Description
31:00	**Zoning Data Key (ZDK):** This field specifies the key identifying a Zoning data structure in the Zoning database of the CDC.

Figure NEW.25: Fabric Zoning Receive (FZR) – Command Dword 11

Bits	Description
31:00	**Zoning Data Offset (ZDO):** This field specifies the byte offset within a Zoning data structure to store the transferred data. The offset shall be dword aligned, indicated by bits 1:0 being cleared to 00b. The CDC is not required to check that bits 1:0 are cleared to 00b. The CDC may return a status code of Invalid Field in Command if bits 1:0 are not cleared to 00b. If the CDC does not return a status code of Invalid Field in Command, then the controller shall operate as if bits 1:0 are cleared to 00b. If an offset greater than the size of the requested Zoning data structure is specified, then the CDC shall abort the command with a status code of Invalid Field in Command.

Figure NEW.26: Fabric Zoning Receive (FZR) – Command Dword 12

Bits	Description
31:28	Reserved
27:00	**Number of Dwords (NUMD):** This field specifies the number of dwords to transfer. If this field is cleared to 0h, then the controller shall abort the command with a status code of Invalid Field in Command.

5.NEW4.1 Command Completion

Upon completion of the Fabric Zoning Receive (FZR) command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status of the command. Command specific status values for the FZR command are defined in Figure NEW.27.

Figure NEW.27: Fabric Zoning Receive (FZR) – Command Specific Status Values

Value	Description
30h	**Zoning Data Structure Locked:** The requested Zoning data structure is locked on the CDC.
31h	**Zoning Data Structure Not Found:** The requested Zoning data structure does not exist on the CDC.
33h	**Requested Function Disabled:** Fabric Zoning is not enabled on the CDC.

The last fragment indication is returned in Dword 0 of the completion queue entry, as defined in Figure NEW.28.

Figure NEW.28: Fabric Zoning Receive (FZR) – Completion Queue Entry Dword 0

Bits	Description
31	**Last Fragment (LF):** This bit specifies if the transferred data buffer contains the last fragment of this Zoning data structure. If set to '1', then the transferred data buffer contains the last fragment; if cleared to '0', then the transferred data buffer does not contain the last fragment.
30:00	Reserved

6 Fabrics Command Set

…

6.3 Connect Command and Response

…

Figure 380: Connect Command – Submission Queue Entry

Bytes	Description
…	…
45:44	**Submission Queue Size (SQSIZE):** This field indicates the size of the Submission Queue to be created. If the size is 0h or larger than the controller supports, then a status code of Connect Invalid Parameters shall be returned. The maximum size of the Admin Submission Queue is specified in the Discovery Log Page Entry for the NVM subsystem. Refer to Figure 264. This is a 0's based value.

Figure 380: Connect Command – Submission Queue Entry

Bytes	Description
	Connect Attributes (CATTR): This field indicates attributes for the connection. <table><tr><th>Bits</th><th>Description</th></tr><tr><td>~~Bits~~ 7:~~4~~5</td><td>~~are r~~Reserved~~.~~</td></tr><tr><td>4</td><td>**Connecting Entity (CONNENT):** Indicates the type of entity performing the Connect command. If this bit is set to '1', then the entity performing the Connect command is a Discovery controller. If this bit is cleared to '0', then the entity performing the Connect command is a host.</td></tr><tr><td>~~Bit~~ 3</td><td>**Individual I/O Queue Deletion Support (INDIVIOQDELS):** ~~i~~Indicates support for deleting individual I/O Queues. If this bit is set to '1', then the host supports the deletion of individual I/O Queues. If this bit is cleared to '0', then the host does not support the deletion of individual I/O Queues.</td></tr><tr><td>~~Bit~~ 2</td><td>**Disable SQ Flow Control (DISSQFC):** ~~i~~If set to '1', then the host is requesting that SQ flow control be disabled. If cleared to '0', then SQ flow control shall not be disabled.</td></tr><tr><td>~~Bits~~ 1:0</td><td>**Priority Class (PRIOCLASS):** ~~i~~Indicate~~s~~ the priority class to use for commands within this Submission Queue. This field is only used when the weighted round robin with urgent priority class is the arbitration mechanism selected, the field is ignored if weighted round robin with urgent priority class is not used. Refer to section 3.4.4. This field is only valid for I/O Queues. ~~It~~This field shall be cleared to 00b for Admin Queue connections. <table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Urgent</td></tr><tr><td>01b</td><td>High</td></tr><tr><td>10b</td><td>Medium</td></tr><tr><td>11b</td><td>Low</td></tr></table></td></tr></table>
47	Reserved
…	…

…

8 Extended Capabilities

…

8.NEW Centralized Discovery for IP-based Fabrics

8.NEW.1 Overview

In configurations that consist of multiple NVM subsystems, the burden on administrators is reduced by enabling hosts to automatically retrieve the list of NVM subsystem ports the host has been allowed to

access from a centralized location. This centralized location is referred to as a Centralized Discovery controller (CDC).

Hosts and NVM subsystems may become known to the CDC by explicitly registering their discovery information as the result of a push registration (refer to section 8.NEW.2.1). If a host or Direct Discovery controller (DDC) detects the presence of multiple CDCs, then that host or DDC should register their discovery information with each CDC. Alternatively, the CDC may implicitly register discovery information as a result of processing a Connect command from a host, or as a result of receiving a pull registration request from a DDC (refer to section 8.NEW.2.2).

The CDC may filter the list of NVM subsystem ports returned in response to a Get Log Page command from the host that requests the Discovery log page to include only the NVM subsystem ports that provide access to namespaces allocated to that host. The process used to configure this filtering function is known as Fabric Zoning (refer to section 8.NEW.3).

An overview of the registration and discovery process is described in section 8.NEW.1.1.

8.NEW.1.1 Registration and Discovery Example

The following example assumes the CDC's effective Fabric Zoning configuration allows Host A to access both NVM subsystem A and NVM subsystem B.

Figure NEW.29: Registration and Discovery Example
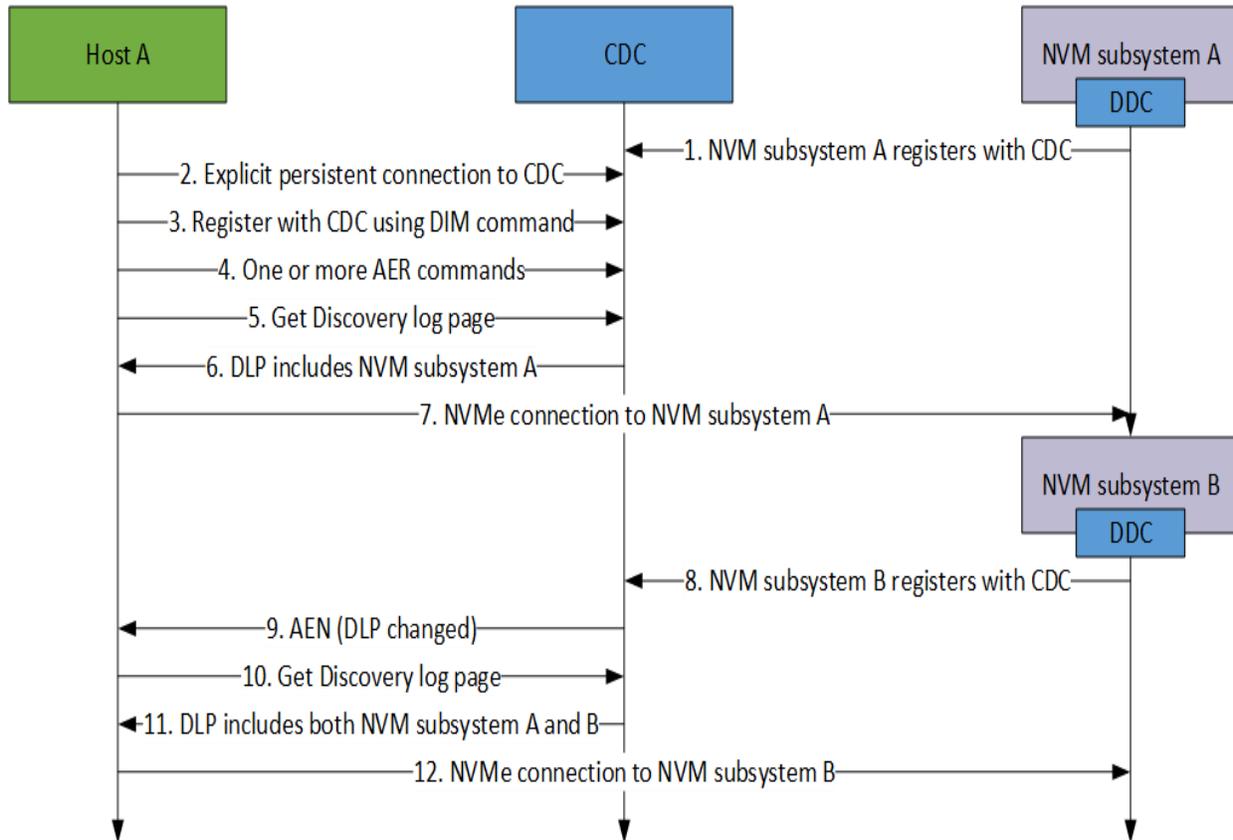
Each of the numbered steps in Figure NEW.29 are described below:

1. A fabric interface on NVM subsystem A is registered with the CDC. This is accomplished by either:
   a. using the Discovery Information Management command (refer to section 5.NEW1) to perform a push registration (refer to section 8.NEW.2.1);
   b. notifying the CDC that a pull registration (refer to section 8.NEW.2.2) is required and the NVM subsystem port must be implicitly registered by the CDC; or
   c. administrative configuration (e.g., the NVM subsystem port was manually configured on the CDC).
2. Host A should establish an explicit persistent connection with the CDC. The method that a host uses to obtain the information necessary to connect to the CDC via the NVMe Transport may be:
   a. implementation specific;
   b. fabric specific;
   c. known in advance (e.g., a well-known address);
   d. administratively configured; or
   e. for IP based fabrics, Automated Discovery of Discovery Controllers for IP Based Fabrics (refer to section 8.NEW.A <Note to editor: 8.NEW.A is located in TP8009>) may be used.
3. Host A uses the Discovery Information Management command to perform a push registration and explicitly register its discovery information with the CDC.

4. Host A should send one or more Asynchronous Event Request commands to the CDC in order to be notified about any changes that occur to the Discovery log page.
5. Host A uses the Get Log Page command to retrieve the Discovery log page from the CDC.
6. The CDC responds to the Get Log Page command with a Discovery log page containing one Discovery Log Page Entry for the interface on NVM subsystem A.
7. If the SUBTYPE field of the Discovery Log Page Entry for NVM subsystem A returned from the CDC is set to 02h (i.e., NVM subsystem), then Host A should connect to the I/O controller on NVM subsystem A.
8. A fabric interface on NVM subsystem B is registered with the CDC. This is accomplished by either:
    a. using the Discovery Information Management command to perform a push registration;
    b. notifying the CDC that a pull registration is required and the NVM subsystem port must be implicitly registered by the CDC; or
    c. administratively configured (e.g., the NVM subsystem port was manually configured on the CDC).
9. The CDC sends a Discovery Log Page Change Asynchronous Event notification (Asynchronous Event Information F0h) to notify Host A that the Discovery log page has changed.
10. Host A uses the Get Log Page command to retrieve the Discovery log page from the CDC.
11. The CDC responds to the Get Log Page command with a Discovery log page containing two Discovery Log Page Entries: one for the interface on NVM subsystem A and another for the interface on NVM subsystem B.
12. If the SUBTYPE field of the Discovery log page for NVM subsystem B returned from the CDC is set to 02h (i.e., NVM subsystem), then Host A should connect to the I/O controller on NVM subsystem B. The connection between Host A and NVM subsystem A remains unchanged.

8.NEW.2 Discovery Information Registration and Discovery Information De-Registration

Discovery information registration is the process of registering host or NVM subsystem discovery information with a Centralized Discovery controller (CDC) or registering host discovery information with a Direct Discovery controller (DDC).

Discovery information registration may be performed:

- using a push registration (refer to section 8.NEW.2.1); or
- using a pull registration (refer to section 8.NEW.2.2).

Information from a Connect command (e.g., Host NQN) may be retained by a Discovery controller to provide a limited set of discovery information. If the Discovery controller determines that the host that submitted the Connect command is the same as a host for which the Discovery controller has retained host discovery information from a previous push registration, then the Discovery controller should continue to retain that host discovery information.

A host:

- should use push registrations to register host discovery information with a CDC or DDC; and
- is not able to use pull registrations to register discovery information.

A CDC:

- may use push registrations to register host discovery information with a DDC; and
- shall not use pull registrations to register discovery information.

A DDC:

- should use push registrations when registering NVM subsystem discovery information with a CDC; or
- may use pull registrations when registering NVM subsystem discovery information with a CDC.

Discovery information de-registration is the process of de-registering host or NVM subsystem discovery information with a CDC or de-registering host discovery information with a DDC.

Discovery information de-registration may be performed using one of the following methods:

- a push de-registration (refer to section 8.NEW.2.1); or
- a pull de-registration (refer to section 8.NEW.2.2).

A host:

- may use push de-registrations to de-register host discovery information with a CDC or DDC; and
- is not able to use pull de-registrations to de-register discovery information.

A CDC:

- may use push de-registrations to de-register host discovery information with a DDC; and
- shall not use pull de-registrations to de-register discovery information.

A DDC:

- should use push de-registrations to de-register NVM subsystem discovery information with a CDC; or
- may use pull de-registrations to de-register NVM subsystem discovery information with a CDC.

8.NEW.2.1 Push Registrations and Push De-Registrations

A push registration is performed using the Discovery Information Management command (refer to section 5.NEW1) with the Task (TAS) field (refer to Figure NEW.9) cleared to 0h to explicitly register discovery information for a host or NVM subsystem with a Centralized Discovery controller (CDC) or explicitly register discovery information for a host with a Direct Discovery controller (DDC).

A push de-registration is performed using the Discovery Information Management command with the TAS field set to 1h to explicitly de-register discovery information for a host or NVM subsystem with a CDC or explicitly de-register discovery information for a host with a DDC.

A DDC that performs push registrations or push de-registrations implements host functionality (e.g., submitting commands, processing command completions, providing a Host NQN, etc.).

8.NEW.2.2 Pull Registrations and Pull De-Registrations

A pull registration may be requested using either:

- an mDNS request and response sequence (refer to section 8.NEW.2.2.1); or
- a kickstart discovery request and response sequence (refer to section 8.NEW.2.2.2).

If the Centralized Discovery controller (CDC) and the Direct Discovery controller (DDC) are both contained in the same broadcast domain, then either an mDNS request and response sequence or a kickstart discovery request and response sequence may be used to request a pull registration.

If the CDC and the DDC are contained in separate broadcast domains, then a kickstart discovery request and response sequence may be used to request a pull registration.

Upon completion of acknowledging a pull registration request (e.g., after a KDResp NVMe/TCP PDU (refer to the Kickstart Discovery Response PDU section in the NVMe TCP Transport Specification) has been sent by the CDC), the CDC performs the pull registration by:

1. sending a Connect command to the DDC to establish an NVMe connection with that DDC; and
2. sending a Get Log Page command to the DDC with the Log Page Identifier (LID) field set to 70h to retrieve NVM subsystem discovery information contained in the associated Discovery log page from that DDC. The CDC may:
    a) set the Port Local Entries Only (PLEO) bit to '1' in the Log Specific Parameter (LSP) field in that Get Log Page command to request that NVM subsystem discovery information entries for only NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that received the Get Log Page command be returned; and
    b) set the All NVM Subsystem Entries (ALLSUBE) bit to '1' in the LSP field in that Get Log Page command to request that records for all NVM subsystem ports be returned.

If kickstart discovery was used to request the pull registration, then the DDC should match the NQN contained in the CDC NVMe Qualified Name (CDCNQN) field of the KDResp NVMe/TCP PDU with the NQN contained in the Host NVMe Qualified Name (HOSTNQN) field in the data portion of the Connect command to know that a CDC is performing a pull registration.

If the PLEO bit is cleared to '0' and the ALLSUBE bit is set '1' in the LSP field of the Get Log Page command, then the DDC shall return all NVM subsystem discovery information entries in the resulting Discovery log page sent to the CDC (i.e., the DDC shall not filter out any NVM subsystem discovery information entries).

If the PLEO bit is set to '1' and the ALLSUBE bit is set '1' in the LSP field of the Get Log Page command, then the DDC shall return all NVM subsystem discovery information entries for NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that received the Get Log Page command in the resulting Discovery log page sent to the CDC (i.e., the DDC shall not filter out any NVM subsystem discovery information entries for NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that received the Get Log Page command).

An additional pull registration or a pull de-registration may be requested by either:

- sending a Discovery Log Page Change Asynchronous Event notification (Asynchronous Event Information F0h) (refer to section 3.1.2.3.4.2); or
- requesting another pull registration.

If the CDC established an explicit persistent connection with the DDC when performing the previous pull registration, then the DDC should request the additional pull registration or the pull de-registration by sending a Discovery Log Page Change Asynchronous Event notification.

If the CDC did not establish an explicit persistent connection with the DDC when performing the previous pull registration, then the DDC requests the additional pull registration or the pull de-registration by requesting another pull registration.

Upon receiving a Discovery Log Page Change Asynchronous Event notification, the CDC performs the additional pull registration or the pull de-registration by using the following sequence:

1. sending a Get Log Page command to the DDC with the LID field set to 70h to retrieve NVM subsystem discovery information contained in the associated Discovery log page from that DDC. The CDC may:
    a. set the Port Local Entries Only (PLEO) bit to '1' in the Log Specific Parameter (LSP) field in that Get Log Page command to request that NVM subsystem discovery information entries for only NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that received the Get Log Page command be returned; and
    b. set the All NVM Subsystem Entries (ALLSUBE) bit to '1' in the LSP field in that Get Log Page command to request that records for all NVM subsystem ports be returned;

    and

2. replacing all existing NVM subsystem discovery information from the previous pull registration with the newly retrieved NVM subsystem discovery information (i.e., only the most recent NVM subsystem discovery information from that DDC is retained by the CDC).

Upon receiving another pull registration request, the CDC performs the additional pull registration or the pull de-registration by using the following sequence:

1. acknowledging that pull registration request (refer to section 8.NEW.2.2.1 for mDNS and section 8.NEW.2.2.2 for kickstart discovery); and
2. sending a Connect command to the DDC to establish an NVMe connection with that DDC ; and
3. sending a Get Log Page command to the DDC with the LID field set to 70h to retrieve NVM subsystem discovery information contained in the associated Discovery log page from that DDC. The CDC may:
    a. set the Port Local Entries Only (PLEO) bit to '1' in the Log Specific Parameter (LSP) field of the Get Log Page command to request that NVM subsystem discovery information entries for only NVM subsystem ports that are presented through the same NVM subsystem port on the DDC that received the Get Log Page command be returned; and
    b. set the All NVM Subsystem Entries (ALLSUBE) bit to '1' in the LSP field of the Get Log Page command to request that records for all NVM subsystem ports be returned;

    and

4. replacing all existing NVM subsystem discovery information from the previous pull registration with the newly retrieved NVM subsystem discovery information (i.e., only the most recent NVM subsystem discovery information from that DDC is retained by the CDC).

8.NEW.2.2.1 mDNS Pull Registration Requests

A Centralized Discovery controller (CDC) queries the local broadcast domain for any Direct Discovery controller (DDC) that requires a pull registration by sending an mDNS query for the service name of "_ddcpull._sub._nvme-disc._<protocol>.local" as described in section 8.NEW.A.4.3 <Note to editor: 8.NEW.A.4.3 is located in TP8009>. If the mDNS query from the CDC is completed by an mDNS response from a DDC containing a DNS SRV record with the service name set to "_ddcpull._sub._nvme-disc._<protocol>.local" (refer to section 8.NEW.A.4.4 <Note to editor: 8.NEW.A.4.4 is located in TP8009>), then the CDC performs a pull registration with the responding DDC as described in section 8.NEW.2.2.

A DDC may request a pull registration during initialization by sending an mDNS announcement containing a DNS SRV record with the service name set to "_ddcpull._sub._nvme-disc._<protocol>.local" as described in section 8.NEW.A.3.1 <Note to editor: 8.NEW.A.3.1 is located in TP8009>. If a CDC receives an mDNS announcement from a DDC containing a DNS SRV record with the service name set to "_ddcpull._sub._nvme-disc._<protocol>.local" (refer to section 8.NEW.A.4.4 <Note to editor: 8.NEW.A.4.4

is located in TP8009>), then the CDC performs a pull registration with the requesting DDC as described in section 8.NEW.2.2.

### 8.NEW.2.2.2 Kickstart Discovery Pull Registration Requests

Figure NEW.30 illustrates the process used during a kickstart discovery request and response sequence. The first step is to establish a TCP connection between a Direct Discovery controller (DDC) and a Centralized Discovery controller (CDC). For kickstart discovery, the CDC acts as the passive side of the TCP connection and is set to "listen" for DDC-initiated TCP connection establishment requests. The IP address used by the DDC to establish the TCP connection with the CDC may be obtained from the A or AAAA record provided in an mDNS response from the CDC, as described in section 8.NEW.A.3.4 <Note to editor: 8.NEW.A.3.4 is located in TP8009>.

Once a TCP connection has been established, if an ICReq NVMe/TCP PDU (refer to the Initialize Connection Request PDU section in the NVMe TCP Transport Specification) from a DDC with Bit 7 set to '1' in the FLAGS field is received by a CDC, then the CDC shall respond with an ICResp NVMe/TCP PDU (refer to the Initialize Connection Response PDU section in the NVMe TCP Transport Specification) to establish a kickstart discovery NVMe/TCP connection.

Once a kickstart discovery NVMe/TCP connection has been established, if a KDReq NVMe/TCP PDU (refer to the Kickstart Discovery Request PDU section in the NVMe TCP Transport Specification) from a DDC is received by a CDC, then the CDC shall respond with a KDResp NVMe/TCP PDU (refer to the Kickstart Discovery Response PDU section in the NVMe TCP Transport Specification) to accept the pull registration request and connection parameters. The KDResp NVMe/TCP PDU sent by the CDC shall include the NQN of that CDC in the CDC NVMe Qualified Name (CDCNQN) field. After sending a KDResp NVMe/TCP PDU to the DDC where the Kickstart Status (KSSTAT) field is set to SUCCESS, the kickstart discovery NVMe/TCP connection should be torn down, and the CDC performs a pull registration with the DDC as described in section 8.NEW.2.2 using the connection parameters obtained from the KDReq NVMe/TCP PDU.

If the total number of discovery information entries being registered by the pull registration (i.e., as specified by the Number of Discovery Information Entries (NUMDIE) field in the KDReq NVMe/TCP PDU) exceeds the available capacity for new discovery information entries on the CDC, then the CDC shall send a KDResp NVMe/TCP PDU to the DDC where the KSSTAT field is set to FAILURE and the Failure Reason (FAILRSN) field is set to Insufficient Discovery Resources.

**Figure NEW.30: Kickstart Discovery Request and Response Sequence**

DDC   CDC

TCP SYN →

TCP SYN-ACK ←

NVMe/TCP ICReq PDU →

NVMe/TCP ICResp PDU ←

NVMe/TCP KDReq PDU →

NVMe/TCP KDResp PDU ←

- - - NVMe/TCP connection tear down - - -
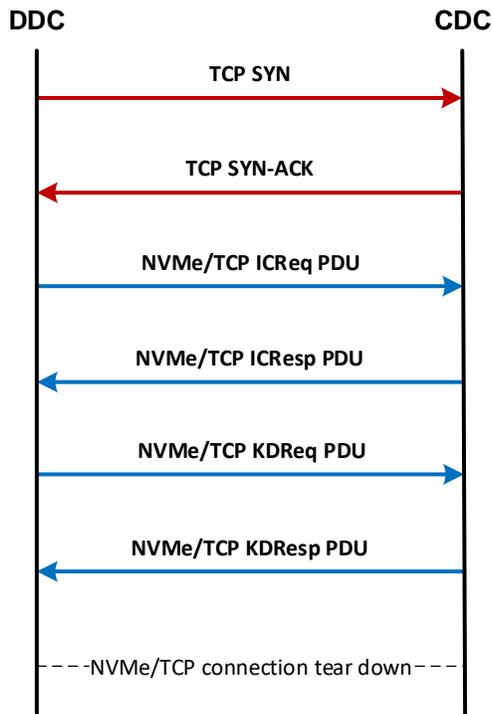
8.NEW.3 Fabric Zoning

8.NEW.3.1 Model

A Centralized Discovery controller (CDC) may provide centralized access control services (i.e., Fabric Zoning) for an NVMe-oF IP-based fabric. CDC-based Fabric Zoning provides a way to control the Discovery log pages provided in response to a Get Log Page command issued to the CDC. Fabric Zoning should be based on a Zoning database maintained by the CDC and containing two fundamental data structures: the ZoneDBConfig and the ZoneDBActive, as shown in Figure NEW.31.

Figure NEW.31: Zoning DB Abstract Representation

Zoning DB

ZoneDBConfig   ZoneDBActive
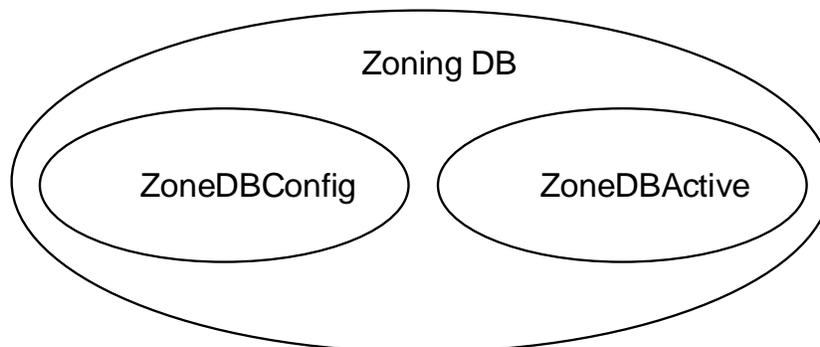
The ZoneDBActive is a list of enforced ZoneGroups. The enforcement actions include:

- Discovery log page filtering;
- Discovery Log Page Change Asynchronous Event notifications (refer to section 8.NEW.4); and
- optionally, network level restrictions (refer to section 8.NEW.3.5.2).

The abstract representation of the ZoneDBActive is shown in Figure NEW.32.

Figure NEW.32: ZoneDBActive Abstract Representation
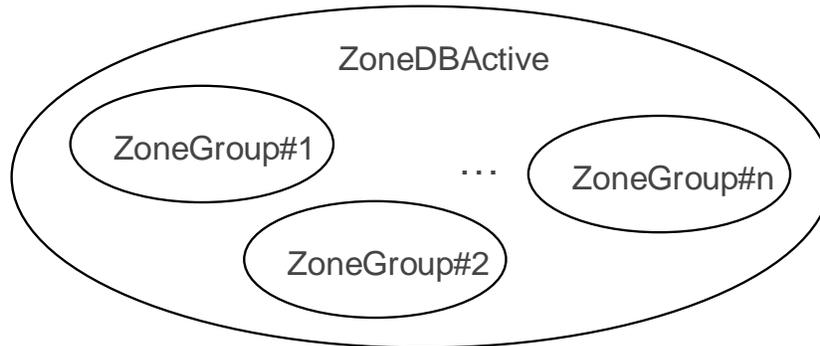
The ZoneDBConfig is a list of configured ZoneGroups and ZoneAliases, as shown in Figure NEW.33.

Figure NEW.33: ZoneDBConfig Abstract Representation
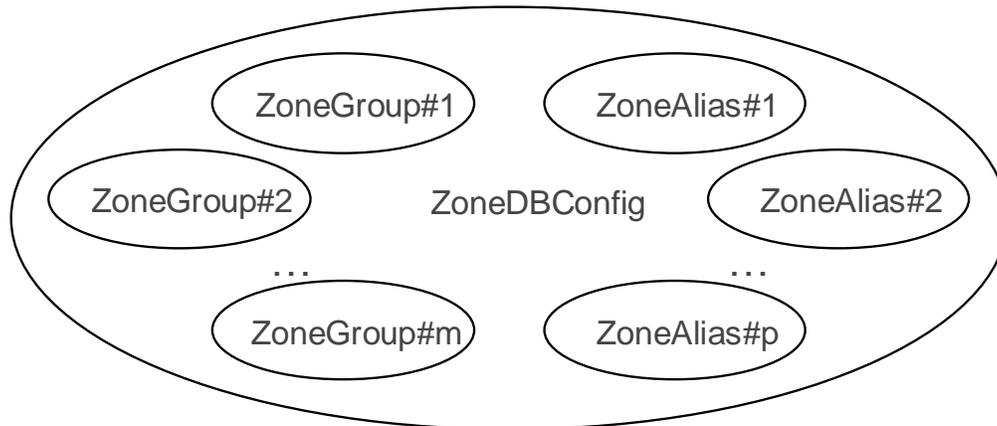
8.NEW.3.2 ZoneGroup

A ZoneGroup is the unit of activation (i.e., a set of access control rules enforceable by the CDC). The detailed representation of a ZoneGroup is shown in Figure NEW.34.

Figure NEW.34: ZoneGroup Detailed Representation

Bytes	Description
223:00	**ZoneGroup Originator (ZGORIG):** This field contains the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of either: • the CDC, if this ZoneGroup was created on the CDC; or • the DDC that created this ZoneGroup, if this ZoneGroup was created by a DDC.
253:224	**ZoneGroup Name (ZGNAME):** This field contains an ASCII encoded null-terminated string, NULL padded as necessary to the 30-byte maximum length.
255:254	**Number of Zones (n) (NUMZ):** This field specifies the number of Zones contained in this ZoneGroup.
$255+LenZ_1:256$	**Zone 1 (Length LenZ$_1$) (Z1):** This field contains the first Zone contained in this ZoneGroup as defined in Figure NEW.35.
$255+LenZ_1+LenZ_2:256+LenZ_1$	**Zone 2 (Length LenZ$_2$) (Z2):** This field contains the second Zone contained in this ZoneGroup as defined in Figure NEW.35 (if present).
	…
$255+LenZ_1+…+LenZ_n:256+LenZ_1+…+LenZ_{n-1}$	**Zone N (Length LenZ$_n$) (ZN):** This field contains the Nth Zone contained in this ZoneGroup as defined in Figure NEW.35 (if present).

A ZoneGroup is uniquely identified by the pair {ZoneGroup Name, ZoneGroup Originator}. For each ZoneGroup, the CDC shall maintain:

• a unique ZoneGroup key, used as a compact ZoneGroup identifier in the FZS and FZR commands (refer to sections 5.NEW3 and 5.NEW4 respectively). ZoneGroup keys should not be reused; and
• a generation number, incremented each time a ZoneGroup is updated and used by the GAZ operation (refer to section 8.NEW.3.8.2). If the value of the generation number is FFFFFFFFh, then the generation number shall be set to 1h when incremented (i.e., rolls over to 1h).

8.NEW.3.3 Zone

A Zone is the unit of access control. Zone members belonging to the same Zone are allowed to communicate between each other, according to the rules specified in section 8.NEW.3.4.1. The detailed representation of a Zone is shown in Figure NEW.35.

Figure NEW.35: Zone Detailed Representation

Bytes	Description
123:00	**Zone Name (ZNAME):** This field contains an ASCII encoded null-terminated string, NULL padded as necessary to the 124-byte maximum length.
125:124	**Number of Zone Members (n) (NUMZM):** This field specifies the number of Zone members contained in this Zone.
127:126	**Number of Zone Properties (k) (NUMZP):** This field specifies the number of Zone properties contained in this Zone.

Bytes	Description
383:128	**Zone Member 1 (ZM1):** This field contains the first Zone member contained in this Zone.
639:384	**Zone Member 2 (ZM2):** This field contains the second Zone member contained in this Zone.
	…
$256*n+127:256*n-128$	**Zone Member N (ZMN):** This field contains the Nth Zone member contained in this Zone (if present).
$256*n+127+LenP_1:256*n+128$	**Zone Property 1 (Length LenP$_1$) (ZP1):** This field contains the first Zone property contained in this Zone (if present).
$256*n+127+LenP_1+LenP_2:256*n+128+LenP_1$	**Zone Property 2 (Length LenP$_2$) (ZP2):** This field contains the second Zone property contained in this Zone (if present).
	…
$256*n+127+LenP_1+LenP_2+…+LenP_k:$ $256*n+128+LenP_1+LenP_2+…+LenP_{k-1}$	**Zone Property K (Length LenP$_k$) (ZPK):** This field contains the Kth Zone property contained in this Zone (if present).

8.NEW.3.4 Zone Members

8.NEW.3.4.1 Overview

Zone members are represented as type-length-value (TLV) data structures. Figure NEW.36 shows the defined Zone member types. If a Zone member type does not include a particular element in the pairing for that Zone member type, then the element values of that type shall not be used for enforcement of Zoning restrictions for that Zone (e.g., Zone member type 1h does not include IP addresses in the enforcement of that Zone member type).

Figure NEW.36: Zone Member Types

Type	Value	Reference
1h	The pair {NQN, Role}	8.NEW.3.4.2
2h	The pair {(NQN, IP, Protocol), Role}	8.NEW.3.4.3
3h	The pair {(NQN, IP, Protocol, IP Protocol Port), Role}	8.NEW.3.4.4
4h	The pair {(NQN, IP, Protocol, IP Protocol Port, PortID), Role}	8.NEW.3.4.5
5h	The pair {(NQN, PortID), Role}	8.NEW.3.4.6
6h	The pair {(NQN, Protocol, PortID, ADRFAM), Role}	8.NEW.3.4.7
11h	The pair {(IP, Protocol), Role}	8.NEW.3.4.8
12h	The pair {(IP, Protocol, IP Protocol Port), Role}	8.NEW.3.4.9
EFh	A ZoneAlias name	8.NEW.3.4.10
F0h to FEh	Vendor specific	
All others	Reserved	

The members of a Zone may communicate with each other using the following rules:

- hosts may communicate with NVM subsystems;
- NVM subsystems may communicate with hosts;
- hosts shall not communicate with hosts; and
- NVM subsystems shall not communicate with NVM subsystems.

A Zone of a ZoneGroup belonging to the ZoneDBConfig may use all defined Zone member types. When a ZoneGroup belonging to the ZoneDBConfig is activated and becomes part of the ZoneDBActive, all ZoneAlias name Zone members are resolved in the group of NVMe entities referenced by the ZoneAlias name.

8.NEW.3.4.2 {NQN, Role} Zone Member Type (Type 1h)

This Zone member type identifies all fabric interfaces, all IP protocols (e.g., TCP or UDP), and all IP protocol ports (e.g., TCP port 4420) that may be used by the NVMe-oF entity identified by the Zone member's NQN. The format of this Zone member type is shown in Figure NEW.37.

Figure NEW.37: {NQN, Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 1h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.
31:04	Reserved
255:32	**Zone Member NQN (ZMNQN):** This field specifies the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the referenced NVMe-oF entity.

8.NEW.3.4.3 {(NQN, IP, Protocol), Role} Zone Member Type (Type 2h)

This Zone member type identifies a specific fabric interface (i.e., through the IP address) and the specific IP protocol (e.g., TCP) used by the NVMe-oF entity identified by the Zone member's NQN over that fabric interface. The format of this Zone member type is shown in Figure NEW.38.

Figure NEW.38: {(NQN+IP+Protocol), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 2h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.

Bytes	Description
04	**Zone Member Address Family (ZMADRFAM):** This field specifies the IP address family. This field shall be set to one of the following values: • 1h: IPv4 address family; or • 2h: IPv6 address family.
05	**Zone Member IP Protocol (ZMIPPROTO):** This field specifies the IP protocol. This field shall be set to one of the following values: • 6h: TCP; or • 11h: UDP.
07:06	Reserved
23:08	**Zone Member Transport Address (ZMTRADDR):** This field specifies the IP address. An IPv6 address is encoded in binary as follows:<table><tr><td>**Bytes**</td><td>**Definition**</td></tr><tr><td>15:00</td><td>Used</td></tr></table>An IPv4 address is encoded in binary as follows:<table><tr><td>**Bytes**</td><td>**Definition**</td></tr><tr><td>03:00</td><td>Used</td></tr><tr><td>15:04</td><td>Reserved</td></tr></table>
31:24	Reserved
255:32	**Zone Member NQN (ZMNQN):** This field specifies the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the referenced NVMe-oF entity.

8.NEW.3.4.4 {(NQN, IP, Protocol, IP Protocol Port), Role} Zone Member Type (Type 3h)

This Zone member type identifies a specific fabric interface (i.e., through the IP address), the specific IP protocol (e.g., TCP), and IP protocol port (e.g., TCP port 4420) used by the NVMe-oF entity identified by the Zone member's NQN over that fabric interface. The format of this Zone member type is shown in Figure NEW.39.

Figure NEW.39: {(NQN+IP+Protocol+IP Protocol Port), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 3h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.
04	**Zone Member Address Family (ZMADRFAM):** This field specifies the IP address family. This field shall be set to one of the following values: • 1h: IPv4 address family; or • 2h: IPv6 address family.
05	**Zone Member IP Protocol (ZMIPPROTO):** This field specifies the IP protocol. This field shall be set to one of the following values: • 6h: TCP; or • 11h: UDP.
07:06	**Zone Member Transport Service Identifier (ZMTRSVCID):** This field specifies the TCP or UDP port.

Bytes	Description
23:08	**Zone Member Transport Address (ZMTRADDR):** This field specifies the IP address. An IPv6 address is encoded in binary as follows: | Bytes	Definition
15:00	Used | An IPv4 address is encoded in binary as follows: | Bytes	Definition
03:00	Used
15:04	Reserved	
31:24	Reserved
255:32	**Zone Member NQN (ZMNQN):** This field specifies the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the referenced NVMe-oF entity.

8.NEW.3.4.5 {(NQN, IP, Protocol, IP Protocol Port, PortID), Role} Zone Member Type (Type 4h)

This Zone member type identifies a specific fabric interface (i.e., through the IP address), the specific IP protocol (e.g., TCP), IP protocol port (e.g., TCP port 4420), and PortID used by the NVMe-oF entity identified by the Zone member's NQN over that fabric interface. The format of this Zone member type is shown in Figure NEW.40.

Figure NEW.40: {(NQN+IP+Protocol+IP Protocol Port+PortID), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 4h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.
04	**Zone Member Address Family (ZMADRFAM):** This field specifies the IP address family. This field shall be set to one of the following values: • 1h: IPv4 address family; or • 2h: IPv6 address family.
05	**Zone Member IP Protocol (ZMIPPROTO):** This field specifies the IP protocol. This field shall be set to one of the following values: • 6h: TCP; or • 11h: UDP.
07:06	**Zone Member Transport Service Identifier (ZMTRSVCID):** This field specifies the TCP or UDP port.
23:08	**Zone Member Transport Address (ZMTRADDR):** This field specifies the IP address. An IPv6 address is encoded in binary as follows: | Bytes	Definition
15:00	Used | An IPv4 address is encoded in binary as follows: | Bytes	Definition
03:00	Used
15:04	Reserved	
25:24	**Zone Member PortID (ZMPORTID):** This field specifies the NVM subsystem PortID.
31:26	Reserved

Bytes	Description
255:32	**Zone Member NQN (ZMNQN):** This field specifies the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the referenced NVMe-oF entity.

8.NEW.3.4.6 {(NQN, PortID), Role} Zone Member Type (Type 5h)

This Zone member type identifies a specific PortID used by the NVMe-oF entity identified by the Zone member's NQN. The format of this Zone member type is shown in Figure NEW.41.

Figure NEW.41: {(NQN+PortID), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 5h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.
05:04	**Zone Member PortID (ZMPORTID):** This field specifies the NVM subsystem PortID.
31:06	Reserved
255:32	**Zone Member NQN (ZMNQN):** This field specifies the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the referenced NVMe-oF entity.

8.NEW.3.4.7 {(NQN, Protocol, PortID, ADRFAM), Role} Zone Member Type (Type 6h)

This Zone member type identifies the specific IP protocol (e.g., TCP), PortID, and Address Family used by the NVMe-oF entity identified by the Zone member's NQN over that fabric interface. The format of this Zone member type is shown in Figure NEW.42.

Figure NEW.42: {(NQN+Protocol+PortID+ADRFAM), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 6h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.
04	**Zone Member Address Family (ZMADRFAM):** This field specifies the IP address family. This field shall be set to one of the following values: • 1h: IPv4 address family; or • 2h: IPv6 address family.

Bytes	Description

05	**Zone Member IP Protocol (ZMIPPROTO):** This field specifies the IP protocol. This field shall be set to one of the following values:
	
	    • 6h: TCP; or
	    • 11h: UDP.

07:06	**Zone Member PortID (ZMPORTID):** This field specifies the NVM subsystem PortID.
31:08	Reserved
255:32	**Zone Member NQN (ZMNQN):** This field specifies the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the referenced NVMe-oF entity.

8.NEW.3.4.8 {(IP, Protocol), Role} Zone Member Type (Type 11h)

This Zone member type identifies the fabric interface (i.e., through the IP address) of an NVMe-oF entity and the specific IP protocol (e.g., TCP) used by the NVMe-oF entity over that fabric interface. The format of this Zone member type is shown in Figure NEW.43.

Figure NEW.43: {(IP+Protocol), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 11h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.

03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values:
	
	    • 1h: host; or
	    • 2h: NVM subsystem.

04	**Zone Member Address Family (ZMADRFAM):** This field specifies the IP address family. This field shall be set to one of the following values:
	
	    • 1h: IPv4 address family; or
	    • 2h: IPv6 address family.

05	**Zone Member IP Protocol (ZMIPPROTO):** This field specifies the IP protocol. This field shall be set to one of the following values:
	
	    • 6h: TCP; or
	    • 11h: UDP.

07:06	Reserved
23:08	**Zone Member Transport Address (ZMTRADDR):** This field specifies the IP address. An IPv6 address is encoded in binary as follows: <table><tr><th>Bytes</th><th>Definition</th></tr><tr><td>15:00</td><td>Used</td></tr></table> An IPv4 address is encoded in binary as follows: <table><tr><th>Bytes</th><th>Definition</th></tr><tr><td>03:00</td><td>Used</td></tr><tr><td>15:04</td><td>Reserved</td></tr></table>
255:24	Reserved

8.NEW.3.4.9 {(IP, Protocol, IP Protocol Port), Role} Zone Member Type (Type 12h)

This Zone member type identifies the fabric interface (i.e., through the IP address) of an NVMe-oF entity, the specific IP protocol (e.g., TCP), and IP protocol port (e.g., TCP port 4420) used by the NVMe-oF entity over that fabric interface. The format of this Zone member type is shown in Figure NEW.44.

Figure NEW.44: {(IP+Protocol+IP Protocol Port), Role} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to 12h.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
03	**Zone Member Role (ZMROLE):** This field specifies what type of entity the Zone member is. This field shall be set to one of the following values: • 1h: host; or • 2h: NVM subsystem.
04	**Zone Member Address Family (ZMADRFAM):** This field specifies the IP address family. This field shall be set to one of the following values: • 1h: IPv4 address family; or • 2h: IPv6 address family.
05	**Zone Member IP Protocol (ZMIPPROTO):** This field specifies the IP protocol. This field shall be set to one of the following values: • 6h: TCP; or • 11h: UDP.
07:06	**Zone Member Transport Service Identifier (ZMTRSVCID):** This field specifies the TCP or UDP port.
23:08	**Zone Member Transport Address (ZMTRADDR):** This field specifies the IP address. An IPv6 address is encoded in binary as follows:<table><tr><th>Bytes</th><th>Definition</th></tr><tr><td>15:00</td><td>Used</td></tr></table>An IPv4 address is encoded in binary as follows:<table><tr><th>Bytes</th><th>Definition</th></tr><tr><td>03:00</td><td>Used</td></tr><tr><td>15:04</td><td>Reserved</td></tr></table>
255:24	Reserved

8.NEW.3.4.10 {ZoneAlias} Zone Member Type

This Zone member type is used to identify a ZoneAlias in a Zone definition. The format of this Zone member type is shown in Figure NEW.45.

Figure NEW.45: {ZoneAlias} Zone Member Format

Bytes	Description
00	**Zone Member Type (ZMTYPE):** This field specifies the Zone member type and shall be set to EFh.
02:01	**Zone Member Length (ZMLEN):** This field specifies the length in bytes of this Zone member type and shall be set to 100h.
31:03	Reserved
155:32	**Zone Member ZoneAlias Name (ZMZANAME):** This field specifies an ASCII encoded null-terminated string, NULL padded as necessary to the 124-byte maximum length.

Bytes	Description
255:156	Reserved

8.NEW.3.5 Zone Properties

8.NEW.3.5.1 Overview

Zone properties are represented as type-length-value (TLV) data structures. Unrecognized Zone properties may be ignored. Figure NEW.46 shows the defined Zone property types.

Figure NEW.46: Zone Property Types

Type	Value
1h	Fabric Enforced Zone
FEh to F0h	Vendor specific
All others	Reserved

8.NEW.3.5.2 Fabric Enforced Zone Property

This Zone property specifies if a Zone is intended to be enforced through packet-by-packet network level restrictions. The format of this Zone property is shown in Figure NEW.47.

Figure NEW.47: Fabric Enforced Zone Property Format

Bytes	Description
00	**Zone Property Type (ZPTYPE):** This field specifies the Zone property type and shall be set to 1h.
02:01	**Zone Property Length (ZPLEN):** This field specifies the length in bytes of this Zone property type and shall be set to 4h.
03	**Zone Property Value (ZPVAL):** This field specifies the value of this Zone property. This field shall be set to the following value: • 1h: Network level enforcement requested. All other values are ignored

8.NEW.3.6 ZoneAlias

A ZoneAlias is a convenient grouping of NVMe entities identified and is referenceable by a ZoneAlias name. The detailed representation of a ZoneAlias is shown in Figure NEW.48.

Figure NEW.48: ZoneAlias Detailed Representation

Bytes	Description
123:00	**ZoneAlias Name (ZANAME):** This field contains an ASCII encoded null-terminated string, NULL padded as necessary to the 124-byte maximum length.
125:124	**Number of ZoneAlias Members (n) (NUMZAM):** This field specifies the number of ZoneAlias members contained in this ZoneAlias.
127:126	Reserved

Bytes	Description
383:128	**ZoneAlias Member 1 (ZAM1):** This field contains the first ZoneAlias member contained in this ZoneAlias.
639:384	**ZoneAlias Member 2 (ZAM2):** This field contains the second ZoneAlias member contained in this ZoneAlias (if present).
	…
256*n+127:256*n-128	**ZoneAlias Member N (ZAMN):** This field contains the Nth ZoneAlias member contained in this ZoneAlias (if present).

8.NEW.3.7 ZoneAlias Members

8.NEW.3.7.1 Overview

ZoneAlias members are represented as type-length-value (TLV) data structures. ZoneAlias members may be any of the Zone member types specified in Figure NEW.36 except the ZoneAlias Zone member type (i.e., ZMTYPE field set to EFh).

8.NEW.3.8 Zoning Operations

8.NEW.3.8.1 Overview

ZoneGroups are data structures maintained and managed (i.e., created, read, modified, or deleted) in the Zoning database of the CDC. During a 'write' access on a ZoneGroup (e.g., the ZoneGroup is created or modified) through an administrative interface outside the scope of this specification, the ZoneGroup becomes locked to any access from any other administrative interface outside the scope of this specification.

Management of ZoneGroups generally happens through an administrative interface outside the scope of this specification on the CDC. Direct Discovery controllers (DDCs) may be able to manage their own ZoneGroups (i.e., ZoneGroups having the DDC NQN as ZoneGroup Originator) through the Get Active ZoneGroup (GAZ) operation (refer to section 8.NEW.3.8.2), the Add/Replace Active ZoneGroup (AAZ) operation (refer to section 8.NEW.3.8.3), and the Remove Active ZoneGroup (RAZ) operation (refer to section 8.NEW.3.8.4).

By default, a ZoneGroup should be accessible only to an administrative interface outside the scope of this specification on the CDC and to the ZoneGroup originator DDC.

If Fabric Zoning is not enabled on the CDC, then that CDC shall abort all Fabric Zoning commands (i.e., any Fabric Zoning command that is issued as part of a Zoning operation) with a status code of Requested Function Disabled.

8.NEW.3.8.2 Get Active ZoneGroup (GAZ)

The Get Active ZoneGroup (GAZ) operation allows a DDC to retrieve from the CDC an active ZoneGroup associated with the DDC issuing the GAZ request. The GAZ operation is mapped to an FZL command to lookup the key of the ZoneGroup to retrieve in the Zoning database of the CDC, followed by one or more FZR commands to retrieve that ZoneGroup from the CDC.

The identifier of the ZoneGroup to get is provided in the FZL buffer, as shown in Figure NEW.49.

Figure NEW.49: FZL Data for GAZ

Bytes	Description
00	**Operation type:** 1h (i.e., Lookup for Get Active ZoneGroup)
03:01	Reserved
227:04	ZoneGroup Originator
257:228	ZoneGroup Name

The FZL command returns the key of the ZoneGroup to retrieve as a Zoning Data key value in the Completion Queue. For the FZL command of a GAZ operation:

- if the requested ZoneGroup does not exist on the CDC, then the CDC shall abort the command with a status code of Zoning Data Structure Not Found; or
- if the requested ZoneGroup is locked on the CDC (i.e., another administrative interface outside the scope of this specification is modifying that ZoneGroup), then the CDC shall abort the command with a status code of Zoning Data Structure Locked.

The Zoning Data key value returned by the FZL command is used in the Command Dword 10 of the FZR commands (refer to Figure NEW.24) to retrieve that ZoneGroup or a fragment of that ZoneGroup. The ZoneGroup definition is retrieved through one or more subsequent FZR commands and is returned in the FZR buffer, as shown in Figure NEW.50. The FZR completion queue entry sending the buffer containing the last fragment shall have the Last Fragment (LF) bit set to '1' in Completion Queue Entry Dword 0 (refer to Figure NEW.28).

Figure NEW.50: FZR Data for GAZ

Bytes	Description
00	**Operation type:** 2h (i.e., Get Active ZoneGroup)
03:01	Reserved
07:04	ZoneGroup generation number
11:08	ZoneGroup fragment length (ZGFL)
15:12	Reserved
ZGFL+15:16	ZoneGroup fragment

For the FZR command of a GAZ response operation:

- if the ZoneGroup identified by the specified ZoneGroup key does not exist on the CDC, then the CDC shall abort the command with a status code of Zoning Data Structure Not Found ; or
- if the ZoneGroup identified by the specified ZoneGroup key is locked on the CDC (i.e., another administrative interface outside the scope of this specification is modifying that ZoneGroup), then the CDC shall abort the command with a status code of Zoning Data Structure Locked.

When a ZoneGroup is transferred in multiple fragments, the receiver shall verify that the ZoneGroup generation number stays constant across all FZR commands. If the ZoneGroup generation number changes, then the GAZ operation shall be aborted. The DDC shall not process received ZoneGroup information until the full ZoneGroup (i.e., all of the fragments of the ZoneGroup) is received.

8.NEW.3.8.3 Add/Replace Active ZoneGroup (AAZ)

The Add/Replace Active ZoneGroup (AAZ) operation allows a DDC to add or replace in the CDC an active ZoneGroup associated with the DDC issuing the GAZ request. The AAZ operation is mapped to an FZL

command to lookup the key of the ZoneGroup to add or replace in the Zoning database of the CDC, followed by one or more FZS commands to provide the CDC with the ZoneGroup to add or replace.

The identifier of the ZoneGroup to add or replace is provided in the FZL buffer, as shown in Figure NEW.51.

Figure NEW.51: FZL Data for AAZ

Bytes	Description
00	**Operation type:** 3h (i.e., Lookup for Add/Replace Active ZoneGroup)
03:01	Reserved
227:04	ZoneGroup Originator
257:228	ZoneGroup Name

The FZL command returns the key of the ZoneGroup to add or replace as a Zoning Data key value in the Completion Queue. For the FZL command of an AAZ operation:

- if the ZoneGroup that matches the specified FZL data exists on the Zoning database of the CDC and is locked by another administrative interface outside the scope of this specification, then the CDC shall abort the command with a status code of Zoning Data Structure Locked;
- if the ZoneGroup that matches the specified FZL data exists on the Zoning database of the CDC and is not locked by another administrative interface outside the scope of this specification, then that ZoneGroup should be locked by the submitting DDC and its key shall be returned as the Zoning Data key in the Completion Queue; or
- if the ZoneGroup that matches the specified FZL data does not exist on the Zoning database of the CDC, then an empty ZoneGroup having the provided identifier shall be created on the Zoning database of the CDC, that ZoneGroup should be locked by the submitting DDC, and its key shall be returned as the Zoning Data key in the Completion Queue.

Successful completion of the FZL command for the AAZ operation results in the identified ZoneGroup on the CDC being locked by the DDC performing the operation.

The ZoneGroup to add or replace is provided in the FZS buffer of subsequent FZS commands with the appropriate Zoning Data key in Command Dword 10 (refer to Figure NEW.19) and is transferred in one or more fragments, as needed, as shown in Figure NEW.52. The FZS command sending the buffer containing the last fragment shall have the Last Fragment (LF) bit set to '1' in Command Dword 12 (refer to Figure NEW.21).

Figure NEW.52: FZS Data for AAZ

Bytes	Description
00	**Operation type:** 4h (i.e., Add/Replace Active ZoneGroup)
03:01	Reserved
NUMD*4+03:04	ZoneGroup fragment

For the FZS command of an AAZ request operation:

- if the ZoneGroup identified by the specified ZoneGroup key does not exist on the CDC, then the CDC shall abort the command with a status code of Zoning Data Structure Not Found; or
- if the ZoneGroup identified by the specified ZoneGroup key is locked by another entity on the CDC (i.e., another administrative interface outside the scope of this specification is modifying that ZoneGroup), then the CDC shall abort the command with a status code of Zoning Data Structure Locked.

The CDC shall update a ZoneGroup in its Zoning database and increment its generation number only upon successful reception of the full ZoneGroup (i.e., all of the fragments of the ZoneGroup). Successful receipt of the full ZoneGroup for the AAZ operation shall unlock the identified ZoneGroup on the CDC. If the full ZoneGroup is not received within 30 seconds from the establishment of the lock (during processing of the related FZL command), then all the received data shall be discarded and the lock shall be released (i.e., the Zoning database is not changed).

8.NEW.3.8.4 Remove Active ZoneGroup (RAZ)

The Remove Active ZoneGroup (RAZ) operation allows a DDC to remove from the CDC an active ZoneGroup associated with the DDC issuing the RAZ request. The RAZ operation is mapped to an FZL command to provide to the CDC with the identifier of the ZoneGroup to remove.

The identifier of the ZoneGroup to remove is provided in the FZL buffer, as shown in Figure NEW.53.

Figure NEW.53: FZL Data for RAZ

Bytes	Description
00	**Operation type:** 5h (i.e., Lookup for Remove Active ZoneGroup)
03:01	Reserved
227:04	ZoneGroup Originator
257:228	ZoneGroup Name

For the FZS command of an RAZ request operation:

- if the requested ZoneGroup does not exist on the CDC, then the CDC shall abort the command with a status code of Zoning Data Structure Not Found; or
- if the requested ZoneGroup is locked on the CDC (i.e., another administrative interface outside the scope of this specification is modifying that ZoneGroup), then the CDC shall abort the command with a status code of Zoning Data Structure Locked.

8.NEW.4 Asynchronous Events

A Centralized Discovery controller (CDC) reports a Discovery Log Page Change Asynchronous Event notification (Asynchronous Event Information field set to F0h) to each host that has requested asynchronous event notifications of this type (refer to Figure 146) as specified in section 3.1.2.3.4.2 when a Fabric Zoning configuration changes. In particular:

- if a Zone member with the Role set to 2h (i.e., an NVM subsystem) is added or removed from a Zone, then the CDC shall report an AEN to all Zone members of that Zone having the Role set to 1h (i.e., all hosts in that Zone);
- if a Zone member with the Role set to 1h (i.e., a host) is added or removed from a Zone, then the CDC shall report an AEN to all Zone members of that Zone having the Role set to 2h (i.e., all NVM subsystems in that Zone); and
- If a Zone is added or removed, then the CDC shall report an AEN to all Zone members of the added or removed Zone.

Description of NVM Express Management Interface Specification 1.2b Changes

6 NVM Express Admin Command Set

…

6.3   Get Log Page

…

Figure 122: Management Endpoint - Log Page Support

Log Page Name[3]	Log Identifier	Requirements[1]	
		NVMe Storage Device	**NVMe Enclosure**
…			
Discovery	70h	O	O
Host Discovery	71h	O	O
…			

Description of NVM Express TCP Transport Specification 1.0a Changes

1 Introduction

…

1.6 Acronyms

Figure 2: NVMe/TCP Acronym Descriptions

Term	Definition
CH	PDU Common Header
CPDA	Controller PDU Data Alignment
C2H	Controller to Host Direction
C2HTermReq	Controller to Host Terminate Connection Request
DATA	PDU Data
DATAL	H2CData and C2HData PDU Data Length
DATAO	H2CData and C2HData PDU Data Offset
DDGST	PDU Data Digest
HDGST	PDU Header Digest
HDR	PDU Header
HPDA	Host PDU Data Alignment
H2C	Host to Controller Direction
H2CTermReq	Host to Controller Terminate Connection Request
ICReq	Initialize Connection Request
ICResp	Initialize Connection Response
KDReq	Kickstart Discovery Request
KDResp	Kickstart Discovery Response
PAD	PDU padding bytes (before DATA starts)
PDU	Protocol Data Unit
PFV	Protocol Format Version
PSH	PDU Specific Header
R2T	Ready to Transfer (PDU)
R2TO	Ready to Transfer PDU Data Offset
R2TL	Ready to Transfer PDU Data Length
TTAG	Transfer Tag

3 Transport Binding

3.1 Setup & Initialization

…

Once a TCP connection has been established, the host sends an Initialize Connection Request (ICReq) PDU to the controller. When a controller receives an ICReq PDU, that controller responds with an Initialize Connection Response (ICResp) PDU. The exchange is used to both establish a connection and exchange connection configuration parameters. When a connection is established, the host and controller are ready to exchange capsules and command data. If the Kickstart Discovery Connection (KDCONN) bit is cleared to '0' in the ICReq PDU, then a non-kickstart discovery NVMe/TCP connection is established, and Tthe first capsule exchange is the NVMe-oF Connect request/response sequence. The NVMe/TCP PDUs listed in Figure 10 except for KDReq (refer to section 3.6.2.NEW1) and KDResp (refer to section 3.6.2.NEW2) may be exchanged over non-kickstart discovery NVMe/TCP connections. If the KDCONN bit is set to '1' in the ICReq PDU, then a kickstart discovery (refer to the Kickstart Discovery Pull Registration Requests section in the NVMe Base Specification) NVMe/TCP connection is established, and only KDReq NVMe/TCP PDUs and KDResp NVMe/TCP PDUs shall be exchanged over that kickstart discovery NVMe/TCP connection.

…

…

3.3 Data Transfer Model

…

Figure 10: NVMe/TCP PDU Types

| PDU Name | Opcode by field | | Combined Opcode [2] | Section | PDU Description |
	Function (07:01)	PDU Direction [1] (00)			
ICReq	0000000b	0b	00h	3.6.2.2	**Initialize Connection Request:** A PDU sent from a host to a controller to communicate NVMe/TCP connection parameters and establish an NVMe/TCP connection
ICResp	0000000b	1b	01h	3.6.2.3	**Initialize Connection Response:** A PDU sent from a controller to a host to accept a connection request and communicate NVMe/TCP connection parameters
H2CTermReq	0000001b	0b	02h	3.6.2.4	**Host to Controller Terminate Connection Request:** A PDU sent from a host to a controller in response to a fatal transport error
C2HTermReq	0000001b	1b	03h	3.6.2.5	**Controller to Host Terminate Connection Request:** A PDU sent from a controller to a host in response to a fatal transport error
CapsuleCmd	0000010b	0b	04h	3.6.2.6	**Command Capsule:** A PDU sent from a host to a controller to transfer an NVMe over Fabrics Command Capsule
CapsuleResp	0000010b	1b	05h	3.6.2.7	**Response Capsule:** A PDU sent from a controller to a host to transfer an NVMe over Fabrics Response Capsule
H2CData	0000011b	0b	06h	3.6.2.8	**Host to Controller Data:** A PDU sent from a host to a controller to transfer data to the controller
C2HData	0000011b	1b	07h	3.6.2.9	**Controller to Host Data:** A PDU sent from a controller to a host to transfer data to the host

Figure 10: NVMe/TCP PDU Types

PDU Name	Opcode by field		Combined Opcode [2]	Section	PDU Description
	Function (07:01)	PDU Direction [1] (00)			
R2T	0000100b	1b	09h	3.6.2.10	**Ready to Transfer:** A PDU sent from a controller to a host to indicate that the controller is ready to accept data
KDReq	0000101b	0b	0Ah	3.6.2.NEW1	**Kickstart Discovery Request:** A PDU sent from a DDC to a CDC to request a pull registration and communicate connection parameters to be used during the subsequent pull registration (refer to the Kickstart Discovery Pull Registration Requests section in the NVMe Express Base Specification).
KDResp	0000101b	1b	0Bh	3.6.2.NEW2	**Kickstart Discovery Response:** A PDU sent from a CDC to a DDC to accept a pull registration request and connection parameters (refer to the Kickstart Discovery Pull Registration Requests section in the NVMe Express Base Specification).

NOTES:
1. Indicates the opcode encoded direction of the PDU. All PDUs shall follow this convention:
    a. 0b = Host to Controller (H2C); and
    b. 1b = Controller to Host (C2H).
2. Opcodes not listed are reserved.

…

3.6.2.2 Initialize Connection Request PDU (ICReq)

Figure 20: Initialize Connection Request PDU (ICReq)

Bytes	PDU Section	Description
00	CH	**PDU-Type:** 00h
01	CH	**FLAGS:** ~~Reserved~~ <table><tr><th>Bits</th><th>Description</th></tr><tr><td>7</td><td>**Kickstart Discovery Connection (KDCONN):** If set to '1', then a kickstart discovery (refer to the Kickstart Discovery Pull Registration Requests section in the NVMe Base Specification) NVMe/TCP connection is established, as described in section 3.1. If cleared to '0', then a non-kickstart discovery NVMe/TCP connection is established, as described in section 3.1.</td></tr><tr><td>6:0</td><td>Reserved</td></tr></table>
02	CH	**HLEN:** Fixed length of 128 bytes (80h).
03	CH	**PDO:** Reserved
07:04	CH	**PLEN:** Fixed length of 128 bytes (80h).
09:08	PSH	**PDU Format Version (PFV):** Specifies the format version of NVMe/TCP PDUs. The format of the record specified in this definition shall be cleared to 0h.
10	PSH	**Host PDU Data Alignment (HPDA):** Specifies the data alignment for all PDUs transferred from the controller to the host that contain data. This value is 0's based value in units of dwords in the range 0 to 31 (e.g., values 0, 1, and 2 correspond to 4 byte, 8 byte, and 12 byte alignment).
11	PSH	**DGST:** Host PDU heasdder and data digest enable options. <table><tr><th>Bits</th><th>Definition</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1</td><td>**DDGST_ENABLE:** If set to '1', the use of data digest is requested by the host for the connection. If cleared to '0', data digest shall not be used for the connection.</td></tr><tr><td>0</td><td>**HDGST_ENABLE:** If set to '1', the use of header digest is requested by the host for the connection. If cleared to '0', header digest shall not be used for the connection.</td></tr></table>
15:12	PSH	**Maximum Number of Outstanding R2T (MAXR2T):** Specifies the maximum number of outstanding R2T PDUs for a command at any point in time on the connection. This is a 0's based value.
127:16	PSH	Reserved

…

3.6.2.NEW1 Kickstart Discovery Request PDU (KDReq)

Figure NEW.A: Kickstart Discovery Request PDU (KDReq)

Bytes	PDU Section	Description
00	CH	**PDU-Type:** 0Ah
01	CH	**FLAGS:** <table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1</td><td>**DDGSTF:** If set to '1', then the DDGST field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present.</td></tr><tr><td>0</td><td>**HDGSTF:** If set to '1', then the HDGST field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.</td></tr></table>
02	CH	**HLEN:** Fixed length of 12 bytes (i.e., Ch).
03	CH	**PDO:** Data offset within PDU. This value shall be a multiple of the data alignment specified by the CPDA field in the ICResp PDU (refer to section 3.6.2.3) that was previously sent by the CDC on this TCP connection.

Figure NEW.A: Kickstart Discovery Request PDU (KDReq)

Bytes		PDU Section	Description
07:04			**PLEN:** Variable length in bytes. If HDGST and DDGST are both not present, then the length will be (NUMKR * 290 bytes) + 12 bytes. If only HDGST or only DDGST is present, then the length will be (NUMKR * 290 bytes) + 16 bytes. If HDGST and DDGST are both present, then the length will be (NUMKR * 290 bytes) + 20 bytes.
09:08		PSH	**Number of Kickstart Records (NUMKR):** This field specifies the number kickstart records included in the PDU DATA field.
11:10			**Number of Discovery Information Entries (NUMDIE):** This field specifies the maximum number of discovery information entires that the DDC is expected to return if a pull registration is requested. This field shall be cleared to 0h if a pull de-registration is being requested. Refer to the Pull Registrations and Pull De-Registrations section in the NVMe Base Specification.
HDGSTF=1	HDGSTF=0		
15:12	Not present	HDGST	**HDGST:** If HDGSTF is set to '1' in the FLAGS field, this field is valid and contains the header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.
305:16	301:12	DATA	**Kickstart Record 0 (KR0):** This field specifies the first kickstart record as defined in Figure NEW.B.
…	…		…
(((N + 1) * 290) – 1) + 16:(N * 290) + 16	(((N + 1) * 290) – 1) + 12:(N * 290) + 12		**Kickstart Record N (KRN):** This field specifies the Nth kickstart record as defined in Figure NEW.B (if present).
DDGSTF=1	DDGSTF=0		
M + 3:M	Not Present	DDGST	**DDGST:** If DDGSTF is set to '1' in the FLAGS field, this field is valid and contains the data digest (refer to section 3.3.1.1). If the DDGSTF bit is cleared to '0', then this field is not present.

Figure NEW.B: Kickstart Record

Bytes	Description
00	**Transport Type (TRTYPE):** This field specifies the transport type as defined in the Transport Type (TRTYPE) field in the Discovery Log Page Entry data structure in the NVMe Base Specification.
01	**Address Family (ADRFAM):** This field specifies the address family as defined in the Address Family (ADRFAM) field in the Discovery Log Page Entry data structure in the NVMe Base Specification.
33:02	**Transport Service Identifier (TRSVCID):** This field specifies the NVMe Transport service identifier as an ASCII string as defined in the Transport Service Identifier (TRSVCID) field of the Discovery Log Page Entry data structure in the NVMe Base Specification.
289:34	**Transport Address (TRADDR):** This field specifies the address of the DDC that may be used for a Connect command as an ASCII string as defined in the Transport Address (TRADDR) field of the Discovery Log Page Entry data structure in the NVMe Base Specification.

3.6.2.NEW2 Kickstart Discovery Response PDU (KDResp)

Figure NEW.C: Kickstart Discovery Response PDU (KDResp)

Bytes		PDU Section	Description		
00		CH	**PDU-Type:** 0Bh		
01			**FLAGS:**		
			Bits	**Description**	
			7:2	Reserved	
			1	**DDGSTF:** If set to '1', then the DDGST field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present	
			0	**HDGSTF:** If set to '1', then the HDGST field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.	
02			**HLEN:** Fixed length of 10 bytes (i.e., Ah).		
03			**PDO:** Data offset within PDU. This value shall be a multiple of the data alignment specified by the HPDA field in the ICReq PDU (refer to section 3.6.2.2) that was previously sent by the DDC on this TCP connection.		
07:04			**PLEN:** Fixed length of 274 bytes (i.e., 112h).		
08		PSH	**Kickstart Status (KSSTAT):**		
			Bits	**Description**	
			7:2	Reserved	
			1	**FAILURE:** If set to '1', then the CDC shall not perform a pull registration due to the reason indicated in the Failure Reason (FAILRSN) field.	
			0	**SUCCESS:** If set to '1', then the CDC shall perform a pull registration.	
09			**Failure Reason (FAILRSN)**		
			Bits	**Description**	
			7	Reserved	
			6	Insufficient Discovery Resources	
			5	TRSVCID does not match TRTYPE	
			4	TRADDR does not match ADRFAM	
			3	Invalid ADRFAM	
			2	Invalid TRTYPE	
			1	No additional information	
			0	Reserved	
HDGSTF=1	HDGSTF=0				
13:10	Not Present	HDGST	**HDGST:** If HDGSTF is set to '1' in the FLAGS field, this field is valid and contains the header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.		
269:14	265:10	DATA	**CDC NVMe Qualified Name (CDCNQN):** This field indicates the NVMe Qualified Name (NQN) that uniquely identifies the CDC. Refer to the NVMe Qualified Names section in the NVMe Base Specification for the formatting requirements of NQNs.		
DDGSTF=1	DDGSTF=0				
273:270	Not Present	DDGST	**DDGST:** If DDGSTF is set to '1' in the FLAGS field, this field is valid and contains the data digest (refer to section 3.3.1.1). If the DDGSTF bit is cleared to '0', then this field is not present.		