



LEGAL NOTICE:

© Copyright 2007 - 2019 NVM Express, Inc. **ALL RIGHTS RESERVED.**

This erratum to the NVMe over Fabrics revision 1.0 specification is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this erratum to the NVMe over Fabrics revision 1.0 specification subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: "© 2007 - 2019 NVM Express, Inc. **ALL RIGHTS RESERVED.**" When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

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

NVM Express™ Technical Errata

Errata ID	005
Revision Date	10/16/2019
Affected Spec Ver.	NVMe over Fabrics 1.0a
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Fred Knight	NetApp Inc.
David Black	Dell EMC
James Smart	Broadcom
Judy Brock	Samsung

Errata Overview

Add references to more easily find related text.

Correct the reference section in the base spec for the HOSTID.

Clarification that Authentication and Queue initialization error cases are queue specific.

Clarify several error codes that are returned during failure cases.

Clarify that methods for allocating specific controller to particular hosts are outside of scope.

Clarify that Connect Command HOSTID field is only used for the Admin Queue.

Clarify that the Connect Command KATO field is only used for the Admin Queue (reserved for an I/O queue).

Clarify that ANA state is not part of persistent controller state for a Dynamic controller.

Clarify shutdown process and requirements.

Clarify that transport specific methods may provide indications of discovery related changes.

Clarify SGL types used by Fabrics (in Authenticate commands, Connect command, and the Transport model description).

Revision History

Revision Date	Change Description
3/26/2019	Creation
05/03/19	Add additional changes from Judy
5/14/19	Verify text against both 1.0a and 1.0aNEXT, update overview, add Shutdown updates and other updates from Judy/David/James.
5/23/19	Complete Shutdown rewrite and review and remove QID change (move to TP).
5/30/19	Update the summary list, clarify section 4.3 that authentication is per queue pair (i.e., per connection). The KATO field changes were moved to the right place (they were incorrectly placed in the Host Identifier field section).
6/3/19	Add generic text to keep association alive for 2 minutes when controller becomes disabled (e.g. reset and shutdown).
6/4/19	Defer the 2 minute delay and handling during reset processing to the next ECN. Add SGL clarification for Authentication Receive and Authentication Send.
10/16/19	Update date for final publication
10/22/19	Ratified

Incompatible Changes

If a host specified a Controller ID other than FFFFh in a Connect command to a Dynamic subsystem the specification indicated in section 3.3 that the Invalid Field in Command status was returned, but Figure 10 indicated that Connect Invalid Parameters status was returned. The correct error status to return is Connect Invalid Parameters.

Description of Specification Changes

Modify a portion of section 1.5.2 (NVM Subsystem) as shown below:

1.5.2 NVM Subsystem

...

The NVM subsystem may support a dynamic or static controller model. In a dynamic controller model, the controller is allocated by the NVM subsystem on demand with no state (e.g., Feature settings) preserved from prior associations. In a static controller model, the host may request a particular controller based on the Controller ID where state (e.g., Feature settings) is preserved from prior associations. Refer to section 4.2.

While an association exists between a host and a controller, only that host may establish connections with I/O Queues of that controller by presenting the same Host NQN, Host Identifier, NVM Subsystem NQN and Controller ID in subsequent Connect command(s) using the same NVM subsystem port, NVMe Transport type, and NVMe Transport address.

An association exists until the controller is shutdown **as described in section 4.5**, a Controller Level Reset **occurs**, or the NVMe Transport connection is lost between the host and controller for the Admin **Queue** or any I/O Queue. There is no explicit NVMe command that breaks the NVMe Transport connection between a host

and controller. While a controller is associated ~~to~~ with a host, ~~that~~ controller is busy, and no other associations may be made ~~to~~ with that controller.

Modify a portion of section 1.5.7 (Connection) as shown below:

1.5.7 Connection

NVMe over Fabrics uses the Connect command (refer to section 3.3) to create controller Admin or I/O Queues. The creation of an Admin Queue establishes an association between a host and the corresponding controller. NVMe over Fabrics does not support the Admin Submission Queue Base Address (ASQ), Admin Completion Queue Base Address (ACQ), and Admin Queue Attributes (AQA) properties as all information necessary to establish an Admin Queue is contained in the Connect command. NVMe over Fabrics does not support the Admin commands associated with I/O Queue creation and deletion (Create I/O Completion Queue, Create I/O Submission Queue, Delete I/O Completion Queue, Delete I/O Submission Queue) defined in the NVMe Base specification.

...

Modify a portion of section 1.5.8 (Authentication) as shown below:

1.5.8 Authentication

NVMe over Fabrics supports both fabric secure channel (that includes authentication) and NVMe in-band authentication. An NVM subsystem may require a host to use fabric secure channel, NVMe in-band authentication, or both. The Discovery Service indicates if fabric secure channel shall be used for an NVM subsystem. The Connect response indicates if NVMe in-band authentication shall be used with that controller.

A controller associated with an NVM subsystem that requires a fabric secure channel shall not accept any commands (i.e., Fabrics commands, Admin commands, or I/O commands) on an NVMe Transport until a secure channel is established. Following a Connect command, a controller that requires NVMe in-band authentication shall not accept any commands on the queue created by that Connect command other than authentication commands until NVMe in-band authentication has completed. Refer to section 6.

Modify a portion of section 2.1 (Command Capsules) as shown below:

2.1 Command Capsules

...

Figure 7: Fabrics Command Capsule – Submission Queue Entry Format

Byte	Description
00	Opcode (OPC): Set to 7Fh to indicate a Fabrics command.
01	Reserved
03:02	Command Identifier (CID): This field specifies a unique identifier for the command. The identifier shall be unique among all outstanding commands associated with a particular queue.
04	Fabrics Command Type (FCTYPE): This field specifies the Fabrics command transferred in the capsule. The Fabrics command types are defined in Figure 14. If this field is set to a reserved value, the command should be aborted with a status code of Invalid Field in Command.
39:05	Reserved

Figure 7: Fabrics Command Capsule – Submission Queue Entry Format

Byte	Description
63:40	Fabrics Command Type Specific: This field is Fabrics command type specific.

...

Modify a portion of section 2.2.1 (Status Values) as shown below:

2.2.1 Status Values

...

Figure 10: Fabrics Command Specific Status Values

Value	Description	Commands Affected
80h	Connect Incompatible Format: The NVM subsystem does not support the Connect Record Format specified by the host.	Connect
81h	Connect Controller Busy: The controller is already associated with a host. This value is also returned if there is no available controller.	Connect
82h	Connect Invalid Parameters: One or more of the parameters (e.g., Host NQN, Subsystem NQN, Host Identifier, Controller ID, Queue ID) specified are not valid.	Connect
...		
92h to BFh	Reserved	
NOTES:		
1. All commands other than Connect, Authenticate Send, and Authenticate Receive.		

...

Modify a portion of section 2.4 (Submission Queue and Completion Queue Definition) in NVMe-oF Revision 1.0a as shown below:

2.4 Submission Queue and Completion Queue Definition

...

Each Submission Queue has a Head entry pointer and a Tail entry pointer that are used to manage the queue and determine the number of **outstanding Submission Queue capsules available to the host for new submissions**. The Head and Tail entry pointers are initialized to **zero 0h** when a queue is created. All arithmetic operations and comparisons on entry pointers are performed modulo the queue size with queue wrap conditions taken into account. The host increments the Tail entry pointer when it adds a capsule to a queue. The controller increments the Head entry pointer when it removes a capsule from the queue.

...

Modify a portion of section 3 (Commands) as shown below:

3 Commands

...

Figure 14: Fabric Command Types

Command Type by Field			Combined Command Type ²	O/M ¹	I/O Queue ³	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer ⁴				
0b	000 00b	00b	00h	M	No	Property Set
0b	000 00b	01b	01h	M	Yes	Connect ⁵
0b	000 01b	00b	04h	M	No	Property Get
0b	000 01b	01b	05h	O	Yes	Authentication Send
0b	000 01b	10b	06h	O	Yes	Authentication Receive
Vendor Specific						
1b	na	na	C0h to FFh	O		Vendor specific

NOTES:

1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. All Fabrics commands may be submitted on the Admin Queue. The I/O Queue supports Fabrics commands as specified in this column. **If a Fabrics command that is not supported on an I/O Queue is sent on an I/O Queue, that command shall be aborted with a status code of Invalid Field in Command.**
4. 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = reserved
5. The Connect command is submitted and completed on the same queue that it creates. Refer to section 1.5.7.

Modify a portion of section 3.1 (Authentication Receive Command and Response) as shown below:

3.1 Authentication Receive Command and Response

...

Figure 15: Authentication Receive Command – Submission Queue Entry

Byte	Description
...	
39:24	SGL Descriptor 1 (SGL1): This field contains a Transport SGL Data Block descriptor or Keyed SGL Data Block descriptor that describes the entire data transfer. Refer to section 4.4 of the NVMe Base specification for the definition of SGL descriptors.
...	

Modify a portion of section 3.2 (Authentication Send Command and Response) as shown below:

3.2 Authentication Send Command and Response

...

Figure 17: Authentication Send Command – Submission Queue Entry

Byte	Description
...	
39:24	SGL Descriptor 1 (SGL1): This field contains a Transport SGL Data Block descriptor or Keyed SGL Data Block descriptor that describes the entire data transfer. Refer to section 4.4 of the NVMe Base specification for the definition of SGL descriptors.
...	

Modify a portion of section 3.3 (Connect Command and Response) as shown below:

3.3 Connect Command and Response

...

If the NVM subsystem supports the dynamic controller model, then:

- the Controller ID of FFFFh shall be specified as the Controller ID in a Connect command for the Admin Queue. ~~If the controller ID is not set to FFFFh, then a status value of Invalid Field in Command~~ **Connect Invalid Parameters** is returned; ~~and~~
- the NVM subsystem shall ~~return~~ **allocate** any available controller to the host; ~~and~~
- ~~return that allocated indicating the~~ Controller ID **allocated** in the Connect response.

...

The NVM subsystem may allocate specific controllers to particular hosts. If a host requests a controller that ~~it~~ is not allocated to that host, then a status value of Connect Invalid Host is returned. ~~The mechanism for allocating specific controllers to particular hosts is outside the scope of this specification.~~

The host shall establish an association with a controller and enable the controller before establishing a connection with an I/O Queue of the controller. ~~If the host sends a Connect command specifying a Queue ID for an I/O Queue before the controller has been enabled, then a status value of Connect Invalid Parameters is returned.~~ If the host sends a Connect command specifying a Queue ID for an Admin ~~Queue~~ or I/O Queue which has already been created, then a status value of Command Sequence Error is returned.

~~If the Host Identifier, Host NQN, NVM Subsystem NQN, and Controller ID values specified for an I/O Queue are not the same as the values specified for the associated Admin Queue in which the association between the host and controller was established, then a status value of Connect Invalid Parameters is returned. If the Host NQN or NVM Subsystem NQN values do not match the values that the NVM subsystem is configured to support, then a status value of Connect Invalid Parameters is returned. If there is a syntax error in the Host NQN or NVM Subsystem NQN value (refer to section 7.9 in the NVMe Base specification), then a status value of Connect Invalid Parameters is returned. If the Host Identifier is cleared to 0h, then a status value of Connect Invalid Parameters is returned.~~

A status of Connect Invalid Parameters is returned for a Connect command if:

- the host sends a Connect command to create an I/O Queue while the controller is disabled;
- the Host Identifier, Host NQN, NVM Subsystem NQN, and the Controller ID values specified for an I/O Queue are not the same as the values specified for the associated Admin Queue in which the association between the host and controller was established;
- the Host NQN or NVM Subsystem NQN values do not match the values that the NVM subsystem is configured to support;
- there is a syntax error in the Host NQN or NVM Subsystem NQN value (refer to section 7.9 in the NVMe Base specification); or
- the Host Identifier is cleared to 0h.

~~If the NVMe Subsystem Port, NVMe Transport Type or NVMe Transport Address used by the NVMe Transport (refer to section 1.5.1) are not the same as the values used for the associated Admin Queue in which the association between the host and controller was established, then it is possible that the Connect command is not received by an NVM subsystem. If the Connect command is received by an NVM subsystem, then:~~

- ~~the NVM subsystem that receives the command may not be the same NVM subsystem to which the association between the host and controller was established (i.e., the NVMe Transport Type and NVMe Transport Address are unique to an NVM Subsystem Port); and~~
- ~~the values of the NVM Subsystem NQN or Controller ID may not be valid at that NVM Subsystem Port (e.g., the NVM Subsystem NQN may specify a different NVM subsystem than the one that received~~

that Connect command, or the Controller ID may specify a controller that is already bound to a different NVM Subsystem Port).

If this situation occurs and a status value is returned for the Connect command, then that status value is Connect Invalid Parameters. There is no requirement that such a Connect command be received by an NVM subsystem (e.g., if the NVMe Transport Address is not a valid transport address, or is the address of a fabric endpoint that does not support NVMe over Fabrics, then the resulting error, if any, is specific to the fabric).

...

Figure 19: Connect Command – Submission Queue Entry

Byte	Description
...	
39:24	SGL Descriptor 1 (SGL1): This field contains a Transport SGL Data Block descriptor or Keyed SGL Data Block descriptor that describes the entire data transfer. Refer to section 4.4 of the NVMe Base specification for the definition of SGL descriptors.
41:40	Record Format (RECFMT): Specifies the format of the Connect command capsule. If a new Connect command format is defined, this value is incremented by one. The format of the record specified in this definition shall be 0h. If the NVM subsystem does not support the value specified, then a status value of Connect Incompatible Format shall be returned.
43:42	Queue ID (QID): Specifies the Queue Identifier for the Admin Queue or I/O Queue to be created . The identifier is used for both the Submission and Completion Queue. The identifier for the Admin Submission Queue and the Completion Queue identifier is 0h. The identifier for an I/O Submission and Completion Queue identifiers are in the range 1 to 65534.
...	
51:48	Keep Alive Timeout (KATO): In the Connect command for the Admin Queue, this This field has the same definition as the Keep Alive Timeout defined in section 5.15.1.14 (Keep Alive Timer) of the the NVMe Base specification. The controller shall set the Keep Alive Timeout Feature to this value. Upon successful completion of the Connect command the controller shall enable and activate the Keep Alive timer as described in section 7.11 (Keep Alive) of the NVMe Base specification. In the Connect command for an I/O Queue, this field is reserved.
...	

Figure 20: Connect Command – Data

Byte	Description
15:00	Host Identifier (HOSTID): This field has the same definition as the Host Identifier defined in section 5.14.1.16 5.15.1.16 (Host Identifier) of the the NVMe Base specification. The controller shall set the Host Identifier Feature to this value.
17:16	Controller ID (CNTLID): Specifies the controller ID requested. This field corresponds to the Controller ID (CNTLID) value returned in the Identify Controller data structure for a particular controller. If the NVM subsystem uses the dynamic controller model, then the value shall be FFFFh for the Admin Queue and any available controller may be returned. If the NVM subsystem uses the static controller model and the value is FFFEh for the Admin Queue, then any available controller may be returned.
255:18	Reserved
511:256	NVM Subsystem NVMe Qualified Name (SUBNQN): NVMe Qualified Name (NQN) that uniquely identifies the NVM subsystem. Refer to section 7.9 (NVMe Qualified Names) of the NVMe Base specification.
767:512	Host NVMe Qualified Name (HOSTNQN): NVMe Qualified Name (NQN) that uniquely identifies the host. Refer to section 7.9 (NVMe Qualified Names) of the NVMe Base specification.
1023:768	Reserved

...

Modify a portion of section 4.2 (Controller Model) as shown below:

4.2 Controller Model

The NVM subsystem may support a dynamic or static controller model. All controllers in the NVM subsystem shall follow the same controller model. A Discovery Controller shall support the dynamic controller model.

In a dynamic controller model, the controller is allocated by the NVM subsystem on demand. In this model, all controllers allocated to a specific host have the same state at the time the association is established, including attached namespaces and Feature settings. Changes to a controller (e.g., attached namespaces, feature settings) after the association is established do not impact other dynamic controllers. **ANA state (refer to TP4004) describes a relationship between a controller and a namespace and is not persistent controller state.** The host ~~shall specify~~ **specifies** a Controller ID of FFFFh when using the Fabrics Connect command (refer to section 3.3) to establish an association with an NVM subsystem using the dynamic controller model.

In a static controller model, controllers that may be allocated to a particular host may have different state at the time the association is established. The controllers within an NVM subsystem are distinguished by their Controller ID. The state that persists across associations is any state that persists across a Controller Level Reset. In a static controller model, different controllers may present different Feature settings or namespace attachments to the same host. The NVM subsystem may allocate particular controllers to specific hosts.

...

Modify a portion of section 4.3 (Queue Initialization and Queue State) as shown below:

4.3 Queue Initialization and Queue State

...

If a controller requires or is undergoing NVMe in-band authentication **for a queue pair, then** a controller shall abort all commands **received on that queue** other than authentication commands with a status of Authentication Required. After the NVMe in-band authentication has been performed successfully **on a queue, then** a controller shall abort all authentication commands **on that queue** with a status of Command Sequence Error.

When an Admin Queue is first created, the associated controller is disabled (i.e., CC.EN is initialized to '0'). A disabled controller shall abort all commands other than Fabrics commands on the Admin Queue with a status of Command Sequence Error. After the controller is enabled, it shall accept all supported Admin commands in addition to Fabrics commands.

Modify a portion of section 4.4 (Initialization) as shown below:

4.4 Initialization

...

The controller initialization steps after an association is established are described below. For determining capabilities or configuring properties, the host uses the Property Get and Property Set commands, respectively.

1. NVMe in-band authentication is performed if required (refer to section 6.2);
2. The host determines the controller capabilities;
3. The host configures controller settings. Specific settings include:
 - a. The arbitration mechanism should be selected in CC.AMS;

- b. The memory page size should be initialized in CC.MPS; and
 - c. The I/O Command Set that is to be used should be selected in CC.CSS;
- 4. The controller should be enabled by setting CC.EN to '1';
- 5. The host should wait for the controller to indicate it is ready to process commands. The controller is ready to process commands when CSTS.RDY is set to '1';
- 6. The host should determine the configuration of the controller by issuing the Identify command, specifying the Controller data structure. The host should then determine the configuration of each namespace by issuing the Identify command for each namespace, specifying the Namespace data structure;
- 7. The host should determine:
 - a. the maximum I/O Queue size using CAP.MQES; and
 - b. the number of I/O Submission Queues and I/O Completion Queues supported using the response from the Set Features command with the Number of Queues feature identifier;
- 8. The host should use the Connect command (refer to section 3.3) to create I/O Submission and Completion Queue pairs; and
- 9. If the host desires asynchronous notification of optional events, the host should issue a Set Features command specifying the events to enable. If the host desires asynchronous notification of events, the host should submit an appropriate number of Asynchronous Event Request commands. This step may be done at any point after the controller signals it is ready (i.e., CSTS.RDY is set to '1').

The association may be removed if step 4 (set CC.EN to '1') is not completed within 2 minutes after establishing the association.

Modify a portion of section 4.5 (Shutdown) as shown below:

4.5 Shutdown

To initiate a shutdown ~~the~~ of a controller, the host should use the Property Set command (refer to section 3.5) to set the Shutdown Notification (CC.SHN) field to:

- 01b to ~~indicate~~ initiate a normal shutdown operation; or
- 10b to initiate an abrupt shutdown ~~using the Property Set command.~~

After the host ~~specifies~~ initiates a shutdown, the host may either disconnect at the NVMe Transport level or ~~it~~ the host may choose to poll CSTS.SHST to determine when the shutdown is complete (i.e., the controller should not initiate a disconnect at the NVMe Transport level). It is an implementation choice whether the host aborts all outstanding commands prior to ~~initiating~~ the shutdown.

The CC.EN field is not used to shutdown the controller (i.e., it is used for Controller Reset). ~~As part of a shutdown, the CC.EN field is cleared to '0'. After a shutdown has been initiated and while:~~

- ~~• the CC.EN field is cleared to '0' after a shutdown; or~~
- ~~• the CSTS.RDY field is cleared to '0' after a shutdown;~~

~~only Fabrics commands are processed by the controller and the Keep Alive timer (if supported) is disabled.~~

From the time a shutdown is initiated until:

- a Controller Level Reset occurs; or
- the controller, if dynamic, is removed from the NVM subsystem,

the controller shall:

- process only Fabric commands (refer to Figure 14); and
- disable the Keep Alive timer, if supported.

After CC.EN transitions to '0' (due to shutdown or reset), the association between the host and controller shall be preserved for at least 2 minutes. After this time, the association may be removed if the controller has not been re-enabled.

Modify a portion of section 5 (Discovery Service) as shown below:

5 Discovery Service

...

The host uses the well-known Discovery Service NQN (nqn.2014-08.org.nvmexpress.discovery) in the Connect command (refer to section 3.3) to a Discovery Service. The method that a host uses to obtain the NVMe Transport information necessary to connect to the well-known Discovery Service is implementation specific.

...

Multiple entries for the same NVM subsystem with different Port ID values indicates that the resulting NVMe Transport connections are independent with respect to NVM subsystem port hardware failures. A host that uses a single association should pick the first / best record to attach to an NVM subsystem. A host that uses multiple associations should choose different ports. In NVMe over Fabrics revision 1.0, there is no indication of preferred ports or active/passive port state.

A transport specific method may exist to indicate changes to a Discovery controller.

...

Modify a portion of section 5.4.1 (Asynchronous Event Configuration) as shown below:

5.4.1 Asynchronous Event Configuration (Feature Identifier 0Bh), (Optional)

Discovery controllers that support Asynchronous Event Notifications shall implement the Get Features and Set Features commands. A Discovery controller shall enable Asynchronous Discovery Log Event Notifications, if a non-zero KATO value is received in the Connect command (refer to section 3.3) sent to that controller.

...

Modify a portion of section 6 (Authentication) as shown below:

6 Authentication

NVMe over Fabrics supports both fabric secure channel (that includes authentication) and NVMe in-band authentication. Fabric authentication is part of establishing a fabric secure channel via an NVMe Transport specific protocol that provides authentication, encryption, and integrity checking (e.g., IPsec; see RFC 4301). NVMe in-band authentication is performed immediately after a Connect command (refer to section 3.3) succeeds using the Authentication Send and Authentication Receive commands (refer to section 3) to tunnel authentication protocol commands between the host and the controller.

...

Modify a portion of section 7.1.2 (Keep Alive) as shown below:

7.1.2 Keep Alive

...

After completing these steps, a controller may accept a Connect command (refer to section 3.3) for the Admin Queue from the same or another host in order to form a new association.

...

Modify a portion of section 7.3.2 (Capsules and SGLs) as shown below:

7.3.2 Capsules and SGLs

...

Admin command data is transferred using host-resident data buffers specified in Keyed SGL Data Block descriptor entries. I/O command data is transferred using host-resident data buffers specified in Keyed SGL Data Block descriptor entries or within the capsule. The RDMA Transport supports the SGL Data Block with Sub Type Offset, SGL Last Segment with Sub Type Offset, and Keyed SGL Data Block descriptors only. The RDMA Transport does not support SGLs in host memory; all SGLs shall be contained in the command capsule. Fabrics and Admin commands have one **Transport** SGL Data Block **descriptor** or Keyed SGL Data Block descriptor (i.e., there are no SGL descriptors following the Submission Queue Entry). I/O commands may have more than one SGL descriptor.