LEGAL NOTICE:

This erratum to the NVM Express revision 1.2 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 NVM Express revision 1.2 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 - 2016 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 Virtual, Inc.
401 Edgewater Place, Suite 600
Wakefield, MA 01880
info@nvmexpress.org

NVM Express™ Technical Errata

Errata ID	**002**
Revision Date	**11/28/2016**
Affected Spec Ver.	**NVMe over Fabrics 1.0**
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Judy Brock	Samsung
David Black	EMC
James Smart	Broadcom
Fred Knight	NetApp
Peter Onufryk	Microsemi
Christoph Hellwig	WD

Errata Overview

The erratum includes clarifications for NVMe over Fabrics 1.0, including:

- Clarifying that message transports are allowed to use messages for data.
- Specify when Keep Alive timer is active (also applies to PCIe).
- Various editorial updates and clarifications.

Revision History

Revision Date	Change Description
7/20/2016	Initial draft
8/24/2016	Includes updates to model, InfiniBand Service ID, and more.
10/10/2016	Minor changes for HSQSIZE.
10/11/2016	Minor changes based on team discussion. Clarified that the maximum I/O Queue command capsule size is based on MDTS and IOCCSZ.
10/17/2016	Removed change for MDTS and IOCCSZ to ECN 003.
11/28/2016	Ratified

Description of Specification Changes

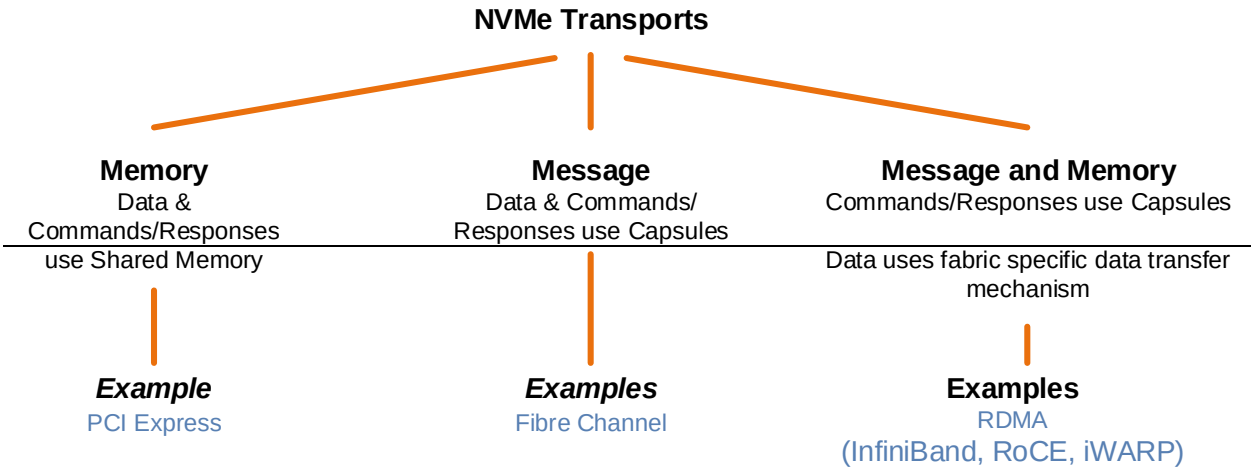
Modify a portion of Section 1.5.1 as shown below:

NVMe over Fabrics requires the underlying NVMe Transport to provide reliable NVMe command and data delivery. An NVMe Transport is an abstract protocol layer independent of any physical interconnect properties. A taxonomy of NVMe Transports along with examples is shown in Figure 1. An NVMe Transport may expose a memory model, a message model, or a combination of the two. A memory model is one in which commands, responses and data ~~is~~ are transferred between fabric nodes by performing explicit memory read and write operations while a message model is one in which only messages containing command capsules, response capsules and data are sent between fabric ~~from a source~~ node~~s to destination node~~. A message/memory model uses a combination of messages and explicit memory read and write operations to transfer command capsules, response capsules and data between fabric nodes. Data may optionally be included in command capsules and response capsules.

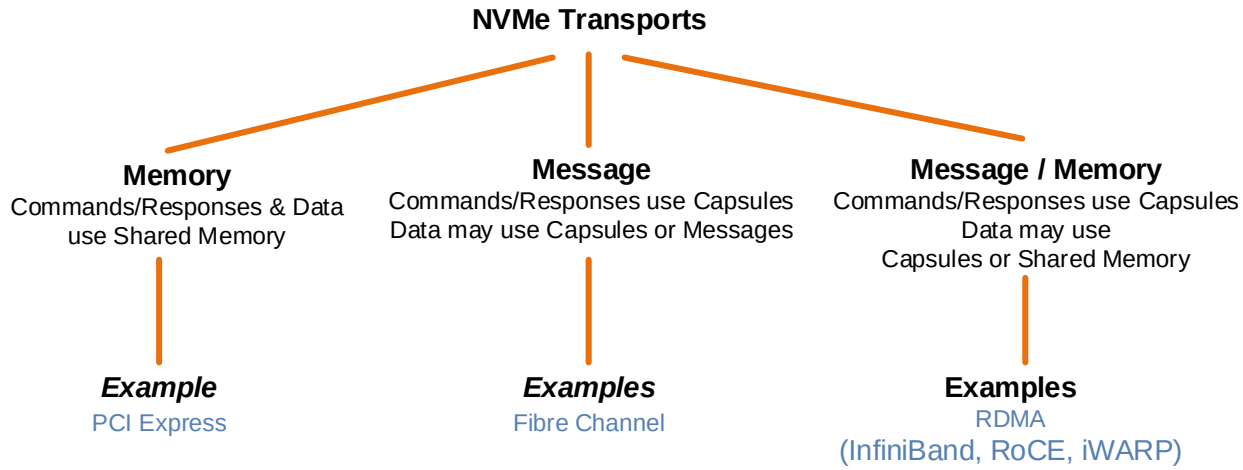
The only memory model NVMe Transport supported by NVMe is PCI Express, as defined in the NVMe Base specification. Message model and message/memory model NVMe Transports are specified in this document.

Replace Figure 1 as shown below:

Figure 1: Taxonomy of Transports



With the update Figure 1 shown below:



Modify a portion of section 1.5.3 that follows Figure 4 as shown below:

Message and Message/Memory model NVMe Transports require all SGLs sent ~~SGL information and data~~ from the host to the controller ~~associated with a command~~ be transferred within the command capsule. Message and Message/Memory model NVMe Transports may optionally support the transfer of a portion or all ~~SGL information and~~ data within the command and response capsules.

Modify a portion of section 4.4 as shown below:

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.
    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 the number of I/O Submission Queues and I/O Completion Queues supported using the Set Features command with the Number of Queues feature identifier.
8. 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 section 4.5 as shown below:

To shutdown the controller, the host should set the Shutdown Notification (CC.SHN) field to 01b to indicate a normal shutdown operation using the Property Set command. After the host specifies a shutdown, the host may either disconnect at the NVMe Transport level or it may choose to poll CSTS.SHST to determine when the shutdown is complete (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 the shutdown.

The CC.EN field is not used to shutdown the controller (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.

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 7.1.2 as shown below:

The Keep Alive feature is defined in section 7.11 of the NVMe Base specification. The Transport binding specification indicates whether the Keep Alive feature is required.

The controller shall treat a Keep Alive Timeout in the same manner as connection loss. If the Keep Alive feature is in use and the timer expires, then the controller shall:

- stop processing commands and set the Controller Fatal Status (CSTS.CFS) bit to '1';
- terminate the NVMe Transport connection; and
- break the host to controller association.

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

Modify section 7.3.6 to add a section 7.3.6.2 that describes the RDMA transport's use of the Discovery Log Page Entry Fields as shown below:

7.3.6.2 Discovery Log Page Entry Fields

The RDMA Transport's use of the Discovery Log Page Entry fields shown in Figure 34 is dependent on the value of the RDMA_CMS field. Figure 4x shows the use of these fields. Refer to section 1.5 in the NVMe Base specification for ASCII string format requirements.

Figure 4x: RDMA Transport Discovery Log Page Entry Usage by RDMA_CMS Value

<table>
<tr><th>RDMA_CMS Value</th><th>Discovery Log Page Entry Field</th><th>Discover Log Page Entry Field Values</th></tr>
<tr><td rowspan="3">RDMA_CM</td><td>ADRFAM</td><td>IPV4, IPV6, AF_IB</td></tr>
<tr><td>TR_ADDR</td><td>The value of this field is dependent on the value of the ADRFAM Field. IPv4 and IPv6 address format is described in Figure 34. AF_IB address format is an InfiniBand™ GUID in ASCII string format.</td></tr>
<tr><td>TRSVCID</td><td>The value of this field is dependent on the value of the TR_ADDR field as shown.
<table>
<tr><th>TR_ADDR Value</th><th>TRSVCID Value (ASCII string)</th></tr>
<tr><td>IPv4 or IPv6</td><td>TCP or UDP port number: decimal number represented as an ASCII string with no leading zeroes.</td></tr>
<tr><td>AF_IB</td><td>InfiniBand™ Service ID represented as 16 case insensitive ASCII hexadecimal digits ('0'-'9', 'a'-'f', 'A-F) in groups of 4 separated by hyphens ('-'), e.g., dead-beef-0123-3210</td></tr>
</table>
</td></tr>
</table>

Modify a portion of section 7.3.4 (Capsule and Data Exchange) as shown below:

All RDMA_READ and RDMA_WRITE operations are initiated by the controller. Data transfers from a controller to a host are performed using the RDMA_WRITE operation. Data transfers from a host buffer to a controller buffer are performed using the RDMA_READ operation. Data for ~~the Fabrics, Admin, or~~ an I/O command may also be exchanged from the host to the controller using a single RDMA_SEND operation that contains the command capsule and in-capsule data within the RDMA_SEND message payload.

Modify a portion of section 7.3.2 as shown below:

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 ~~(Keyed)~~ SGL Data Block 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.

Modify a portion of Figure 43 (RDMA_CM_REQUEST Private Data Format) as shown below:

07:06	**RDMA QP Host Send Queue Size (HSQSIZE):** This field indicates the number of RDMA QP send queue entries allocated by the host's RDMA transport for capsule transmission. The value shall be set to the Submission Queue Size (SQSIZE). Refer to the SQSIZE definition in the Connect command in Figure 19. This is a 0's based value.