LEGAL NOTICE:

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

NVM Express® Technical Proposal (TP)

Technical Proposal ID	4194 SGL Support Clarifications
Revision Date	2024.12.17
Builds on Specification(s)	NVM Express Base Specification 2.1
References	

Technical Proposal Author(s)

Name	Company
Judy Brock, Jiwon Chang, Mike Allison, Bill Martin	Samsung

Technical Proposal Overview

This technical proposal provides clarifications related to processing of Scatter Gather Lists.

Revision History

Revision Date	Change Description
2024.07.19	• Initial version
2024.07.31	• Updated to NVMe base spec v2.1; removed references to ECN116 and ECN120 as they are included in v2.1 • Incorporated editorial feedback from Randy

Revision Date	Change Description
2024.08.08	Addressed all feedback from 08/01 WG mtg: • Removed language associated with logical blocks, replaced with direct reference to MDTS field • Changed two occurrences of "… "data requested to be transferred" to "data to be transferred" in existing (black) text and made the same change in several places throughout the new blue text. Left one instance of existing text alone as it is functionally correct. These changes allowed text to be inclusive of commands that specify a length to be transferred that is not necessarily equal to the actual amount of data to be transferred (host may not know how much data will actually be transferred – in those cases "requested" amount represents a maximum amount to transfer rather than a fixed amount of data to be transferred (e.g., certain Get Log Page commands, Key Value Retrieve command). • Revised references to SGLErrors Figure to point to presence of status code values in that figure • Identified specific error to return if either of the two low-order bits in the Length or Address field in a SGL Data Block Descriptor are not zero when controller requires dword granularity: "SGL Data Block Granularity Invalid" status code • Moved stricken purple text to original position to make it clear where existing black text got moved to • Reverted this change: "… then the values in the Address and Length fields of all Data Block descriptors ~~shall~~ should have their two least significant bits cleared to 00b" • Minor editorial changes
2024.08.18	• Added text to address LLDTS=0 case in section 4.3.2 Scatter Gather List • Substantially revised new blue text in section 4.3.2 Scatter Gather List • removed assumption that all SGL validation is done before data transfer • reverted changes around changing "data requested to be transferred" to "data to be transferred" because, at a minimum, an SGL length should be long enough to cover the amount of data requested to be transferred – even if the actual amount of data to be transferred winds up being less than that amount. • Added new instances of "data requested to be transferred" in LLDTS field, Data SGL Length Invalid definition, and Metadata SGL Length Invalid definition.
2024.08.25	• Added text to covered LLDTS=0 case • Added clarification of what "requested data transfer length" means • Added clarifications to Data SGL Length Invalid and Metadata SGL Length Invalid status code definitions to remove misleading "may" statements • Added shall statements for the cases in the Data SGL Length Invalid and Metadata SGL Length Invalid status code definitions (SGL length too short and SGL Length too long)

Revision Date	Change Description
2024.08.27	• Clarified that the controller is not required to examine for errors the portion of the SGL, if any, that is not used during the data transfer operation – regardless of the value of the LLDTS bit. • Reverted these changes: “Added clarifications to Data SGL Length Invalid and Metadata SGL Length Invalid status code definitions to remove misleading “may” statements” and “Added shall statements for the cases in the Data SGL Length Invalid and Metadata SGL Length Invalid status code definitions for SGL length too long” • Clarified that it’s optional, when LLDTS=0, for controllers to check whether or not SGL length is equal to requested data transfer length. Controller may or may not return error if compare fails. Some controllers can’t perform this check (e.g. NVMe/FC or NVMe/TCP controllers that don’t have access to original SGL) • Replaced “amount of data to be transferred” with “requested data transfer length” globally
2024.08.28	• Replaced “shall” statement in LLDTS=0 case with “if the controller detects…” language • Miscellaneous editorial changes
2024.09.03	• Defined SGL Length in section 4.3.2 • Clarifed what is meant by an SGL Descriptor with an unsupported format in section 4.3.2
2024.09.08	• Added more clarification around when the controller is and isn’t required to validate an SGL descriptor • Clarified that SGL descriptor validation may occur out of order/data transfer may take place before SGL descriptor with error is detected • Clarified that Keyed SGL Data Block descriptors and Transport SGL Data Block descriptors are included in the calculation of SGL Length.
2024.09.23	• Changed LLDTS definition to reflect 09/12 WG decision on direction: • for LLDTS=0: host shall set SGL len = requested len and controller should not report an error if host violates the “shall” • for LLDTS=1: no error shall be returned by the controller if SGL len > requested len, host is permitted to send cmd w/SGL len > requested len • Clarified that dword alignment and granularity as indicated in the SGL Support field of the Identify Controller data structure apply to Keyed SGL Data Block descriptors and Transport SGL Data Block descriptors as well as SGL Data Block descriptors.
2024.09.27	• Integrated additional editorial feedback from Tech WG 09/26 mtg • Accepted all changes for submittal as Phase 2 exit candidate
2024.10.18	• Phase 3 draft - integrated editorial feedback from David Black
2024.10.21	• Phase 3 draft - integrated editorial feedback from Mike Allison
2024.10.24	• Accepted all changes and resolved all comments in preparation for 30 day review
2024.12.17	• Integrated

Description for Changes Document for the NVM Express Base Specification 2.1

New Features/Feature Enhancements/Required Changes:

- Clarified that the “requested data transfer length” is specified either in the SQE or in the elsewhere in the specification (e.g., the command definition).
- If the Length Larger than Data Transfer Support bit in the SGL Support field of Identify Controller is cleared to ‘0’, make the check for whether or not an SGL is too long optional but the recommendation is that the controller should NOT abort the command. Having this check be optional for controllers ensures that no controller is required to examine any SGL descriptor that is not used for data transfer. This provides backwards compatibility both with existing controller implementations that do perform that length check as well as with existing controller implementations that are unable to perform that length check, such as Fabrics controllers that do not have access to the original SGL (e.g., NVMe/FC and NVMe/TCP controllers).
- Clarified that the controller shall always examine for errors the portion of the SGL that is used for the data transfer operation.
- Clarified that the controller is not required to examine for errors the portion of the SGL, if any, that is not used for the data transfer operation – regardless of the value of the LLDTS bit.
- Added a figure which specifies which error status to return for various SGL validation failures.
- Called out specific error to return if either of the two low-order bits in the Length or Address field in a SGL Data Block Descriptor are not zero when controller requires dword granularity: SGL Data Block Granularity Invalid status code.
- Clarified what is meant by an SGL Descriptor with an unsupported format

Markup Conventions:

Black:	Unchanged (however, hot links are removed)
~~Red Strikethrough~~:	Deleted
Blue:	New
Purple:	Moved
Blue Highlighted:	TBD values, anchors, and links to be inserted in new text.
<Green Bracketed>:	Notes to editor or reader
Orange:	Text is pulled in from a referenced Technical Proposal or ECN

Description of Specification Changes for NVM Express Base Specification 2.1

4 Data Structures

…

4.2 Completion Queue Entry

…

4.2.3 Status Field Definition

…

4.2.3.1 Generic Command Status Definition

…

Figure 102: Status Code – Generic Command Status Values

Value	Description	I/O Command Set(s)[1]
0Fh	**Data SGL Length Invalid:** This may occur if the length of a data SGL is too short. This may occur if the length of a data SGL is too long and the controller does not support SGL ~~transfers~~ lengths longer than the ~~amount of data to be transferred~~requested data transfer length (refer to section 4.3.2) as indicated in the SGL Support field of the Identify Controller data structure.	
10h	**Metadata SGL Length Invalid:** This may occur if the length of a metadata SGL is too short. This may occur if the length of a metadata SGL is too long and the controller does not support SGL ~~transfers~~ lengths longer than the ~~amount of data to be transferred~~ requested data transfer length (refer to section 4.3.2) as indicated in the SGL Support field of the Identify Controller data structure.	

…

4.3 Data Pointer Layouts (PRPs and SGLs)

…

4.3.2 Scatter Gather List (SGL)

A Scatter Gather List (SGL) is a data structure in memory address space used to describe a data buffer. The controller indicates the SGL types that the controller supports in the Identify Controller data structure. A data buffer is either a source buffer or a destination buffer. An SGL contains one or more SGL segments.

The SGL length is t~~T~~he combined total of the values in the L~~l~~ength fields in ~~of~~ the SGL Data Block, ~~and~~ SGL Bit Bucket, Keyed SGL Data Block, and Transport SGL Data Block descriptors in an SGL. The SGL length shall be equal to or exceed the requested data transfer length ~~amount of data requested to be transferred~~.

The requested data transfer length is either defined in the specification of a command (e.g., the requested data transfer length for the Reservation Acquire command is 16 bytes); or is determined based on values in one or more command specific locations in the SQE (e.g., the requested data transfer length is specified in the NLB field for a Write command (refer to the NVM Command Set Specification) and is

specified in the NUMDL field and the NUMDU field for a Get Log Page command (refer to section 5.1.12)).
If the requested data transfer length ~~amount of data requested to be transferred~~ exceeds the SGL length ~~total length of the Data Block and Bit Bucket descriptors in an SGL,~~ then:

- data shall not be transferred to or from locations that are not described by the SGL;
- if that SGL is a data SGL, then the controller shall abort the command with a status code of Data SGL Length Invalid; and
- if that SGL is a metadata SGL, then the controller shall abort the command with a status code of Metadata SGL Length Invalid.

…
The SGL Identifier Descriptor Sub Type field may indicate additional information about a descriptor. As an example, the Sub Type may indicate that the Address field is an offset rather than an absolute address. The Sub Type may also indicate NVMe Transport specific information.
~~The controller shall abort a command if:~~

- ~~an SGL segment contains an SGL Segment descriptor or an SGL Last Segment descriptor in other than the last descriptor in the segment;~~
- ~~a last SGL segment contains an SGL Segment descriptor, or an SGL Last Segment descriptor; or~~
- ~~an SGL descriptor has an unsupported format.~~

~~The controller should abort a command if:~~
~~an SGL Data Block descriptor contains Address or Length fields with either of the two least significant bits set to 1b and the controller supports only dword alignment and granularity as indicated in the SGL Support field of the Identify Controller data structure. Refer to Figure 312.~~

The controller shall examine an SGL descriptor for errors before using that SGL descriptor for data transfer. SGL descriptor error checking may occur in any order and data may be transferred using a valid SGL descriptor before an SGL descriptor error is detected in another SGL descriptor. The controller is not required to examine for errors any SGL descriptor that is not used for data transfer.

If the LLDTS bit is cleared to ‘0’ and the length of an SGL is greater than the requested data transfer length, then:

- the controller should not abort the command for this reason; and
- if the controller does abort the command for this reason, then the controller should abort the command with a status code of:
  - Data SGL Length Invalid if that SGL is a data SGL; and
  - Metadata SGL Length Invalid if that SGL is a metadata SGL.

For all SGL descriptors examined for errors by the controller:

- If the controller detects an error condition listed in Figure SGLErrors, then the controller shall abort the command with the corresponding status code listed in that figure; and
- If:
  - an SGL Data Block descriptor, Keyed SGL Data Block descriptor, or a Transport SGL Data Block descriptor contains Address or Length fields with either of the two least-significant bits set to 1b; and
  - the controller supports only dword alignment and granularity as indicated in the SGL Support field of the Identify Controller data structure~~. R~~(refer to Figure 312),

  then the controller should abort the command with a status code of SGL Data Block Granularity Invalid.

Figure SGLErrors: SGL Validation Error Conditions

SGL Validation Error Condition	Status Code
An SGL segment contains: • an SGL Segment descriptor; or • an SGL Last Segment descriptor, in other than the last descriptor in the segment.	Invalid Number of SGL Descriptors

SGL Validation Error Condition	Status Code
A last SGL segment contains an SGL Segment descriptor or an SGL Last Segment descriptor.	Invalid SGL Segment Descriptor
An SGL descriptor has: • an unsupported ~~format~~ type; or • a combination of type and subtype.	SGL Descriptor Type Invalid

5 Identify Data Structures

…

5.1 Common Admin Commands

…

5.1.13 Identify command

…

5.1.13.2 Identify Data Structures

…

5.1.13.2.1 Identify Controller Data Structure (CNS 01h)

…

Figure 276: Identify – Identify Controller Data Structure, I/O Command Set Independent

<table>
<tr><th>Bytes</th><th>I/O[1]</th><th>Admin[1]</th><th>Disc[1]</th><th>Description</th></tr>
<tr><td colspan="5">Controller Capabilities and Features</td></tr>
<tr><td>539:536</td><td>O</td><td>O</td><td>M</td><td>SGL Support (SGLS): This field indicates if SGLs are supported and the particular SGL types supported. Refer to section 4.1.2.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>18</td><td>Length Larger than Data Transfer Support (LLDTS): This bit indicates the SGL length supported by the controller. If this bit is set to ‘1’, then ~~the controller supports commands that contain a data or metadata SGL~~ the SGL length in a command is permitted to be ~~of a length~~ larger than the ~~amount of data to be transferred~~ requested data transfer length (refer to section 4.3.2).
If this bit is cleared to ‘0’, then the SGL length shall be equal to the ~~amount of data to be transferred~~ requested data transfer length. If the controller detects that the SGL length is larger than the requested data transfer length, then:
• the controller should not abort the command for this reason; and
• if the controller does abort the command for this reason, the controller should abort the command with the status specified in section 4.3.2.</td></tr>
<tr><td>…</td><td>…</td></tr>
</table></td></tr>
</table>