



LEGAL NOTICE:

© **Copyright 2008 to 2022 NVM Express®, Inc. ALL RIGHTS RESERVED.**

This Technical Proposal 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 Technical Proposal 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: “© 2008 to 2022 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.

The NVM Express® design mark is a registered trademark of NVM Express, Inc.

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

NVM Express® Technical Proposal (TP)

Technical Proposal ID	TP8020 – UTF-8 Admin Labels
Revision Date	2022.12.06
Builds on Specification(s)	NVM Express Base Specification 2.0b, TP8010a NVM Express Management Interface Specification 1.2b
References	RFC 3629

Technical Proposal Author(s)

Name	Company
Erik Smith	Dell
David Black	Dell
Curtis Ballard	HPE

Technical Proposal Overview

This TP defines the concept of a UTF-8 string and proposes to replace the existing term “symbolic name” with “Admin Label” and defines how Hosts and NVM subsystems may register and retrieve an Admin Label in UTF-8 format in addition to ASCII.

Registration will be accomplished by defining a new extended attribute in the Discovery Information Management (DIM) command.

Retrieval of a UTF-8 encoded Admin Label will be accomplished by defining a new Extended Attribute that could be included in a Extended Discovery Log Page Entry.

Revision History

Revision Date	Change Description
2022.03.21	Initial draft
2022.03.22	Incorporated edits from the FMDS meeting on 3/22/2022
2022.04.26	Incorporated edits from the FMDS meeting on 4/26/2022
2022.05.09	Incorporated portions of TP4145 that define UTF-8 strings
2022.05.13	Added section 1.4.NEW to provide usage information.
2022.05.19	Incorporated changes from the TWG meeting on 2022.05.19
2022.06.30	Updated document to reflect phase 3 status.
2022.07.07	UTF-8 text updates, including correction to RFC 3629 reference, allowing null termination of Host Metadata UTF-8 and updating NVMe-MI definitions of UTF-8 string and ASCII string to match the Base spec. Noted issue on use of "shall" in ASCII and UTF-8 string definitions.
2022.07.08	Updates from discussion with Austin Bolen, including removing all mention of 8-bit ASCII usage in the NVMe-MI spec (that's material inherited from IPMI that is best left as-is).
2022.07.12	Updates from FMDS discussion – use "i.e.," for most parenthesized text in string definitions, minor edits to 4.x material on UTF-8 string processing, proposed resolution (in comments) to use of "shall" in string definitions. Reset change tracking for review outside of FMDS.
2022.07.21	Minor edits from Technical WG discussion, including use of Unicode normalization as a better example than conversion to lower case.
2022.07.22	Edits from Austin Bolen to clarify what happens where in UTF-8 string processing and generalize MI requirements to all of MI (not just VPD).
2022.07.28	Clean version for member review
2022.11.28	Integrated
2022.12.06	Updated "UTF-8" to be nonbreaking hyphen

Description for Changes Document for TP8010a

- Modified existing usage of “symbolic name” to “admin label ASCII”
- Added a new definition for “admin label UTF-8”
- Added a new Extended Attribute Type (EXATYPE) for “admin label UTF-8”
- Added a new Extended Attribute Length (EXATLEN) for “admin label UTF-8”
- Added a new Extended Attribute Value (EXATVAL) for “admin label UTF-8”

Description for Changes Document for NVMe Base 2.0b

New Feature:

- Define UTF-8 string format and usage.

Description for Changes Document for NVMe-MI 1.2b

New Feature:

- Define UTF-8 string format and usage.

Markup Conventions:

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

Description of Specification Changes for NVMe 2.0b Base specification

1.4.2 Numerical Descriptions

...

Some parameters are defined as an ASCII string. ASCII strings shall contain only code values (i.e., byte values or octet values) 20h through 7Eh. For the string “Copyright”, the character “C” is the first byte, the character “o” is the second byte, etc. ~~The ASCII strings is are~~ left justified. ~~and If padding is necessary, then the string~~ shall be padded with spaces (i.e., ASCII character 20h) to the right ~~if necessary unless the string is specified as null-terminated.~~

Some parameters are defined as a UTF-8 string. UTF-8 strings shall contain only byte values (i.e., octet values) 20h through 7Eh, 80h through BFh, and C2h through F4h (refer to sections 1 to 3 of RFC 3629). For the string “Copyright”, the character “C” is the first byte, the character “o” is the second byte, etc. UTF-8 strings are left justified. If padding is necessary, then the string shall be padded with spaces (i.e., ASCII character 20h, Unicode character U+0020) to the right unless the string is specified as null-terminated.

If padding is necessary for a field that contains a null-terminated string then the field should be padded with nulls (i.e., ASCII character 00h, Unicode character U+0000) to the right of the string.

1.5. NEW admin label

An admin label is an administratively configured ASCII or UTF-8 string (refer to section 1.4.2) that may be used to help identify specific NVMe entities (i.e., Hosts, NVM subsystems and namespaces). An admin label is capable of describing the entity's physical location, DNS name or other information.

1.8 References

...

RFC 3629, ~~Alis Technologies~~, F. Yergeau, "UTF-8, a transformation format of ISO 10646", November 2003. Available from <https://www.ietf.org/rfc.html>.

...

4.5 NVMe Qualified Names

...

NVMe hosts, controllers and NVM subsystems ~~process compare~~ (e.g., ~~compare~~ for equality) UTF-8 NVMe Qualified Names ~~used by NVMe as binary strings~~ without any text processing or text comparison logic that is specific to the Unicode character set or locale ~~(e.g., case folding or conversion to lower case, Unicode normalization)~~ (refer to section 4.x). ~~Any such text processing:~~

- ~~a) may occur as part of entry of NVMe Qualified Names into NVMe hosts and NVM subsystems; and~~
- ~~b) should not occur as part of receiving NVMe Qualified Names via an NVMe connection, as shown in Figure 136.~~

~~Upon entry (e.g., at point 1 in Figure 136, described as "input" in RFC4122), NVMe host software may process an NVMe Qualified Name (e.g., for conversion to lower case based on the Unicode locale). Upon entry (e.g., at point 3 in Figure 136, described as "input" in RFC4122), a controller may process an NVMe Qualified Name (e.g., for conversion to lower case based on the Unicode locale). Upon receipt by the host (e.g., at point 2 in Figure 136) of an NVMe Qualified Name from the controller, no text process (e.g., no case folding) should occur. Upon receipt by the controller (e.g., at point 4 in Figure 136) of an NVMe Qualified Name from the host, no text processing (e.g., no case folding) should occur.~~

Figure 136: NQN Processing

<Text deleted above along with Figure 136 have been moved to new 4.x section below and generalized to UTF-8 strings. Editor to renumber Figures accordingly.>

4.x UTF-8 String Processing

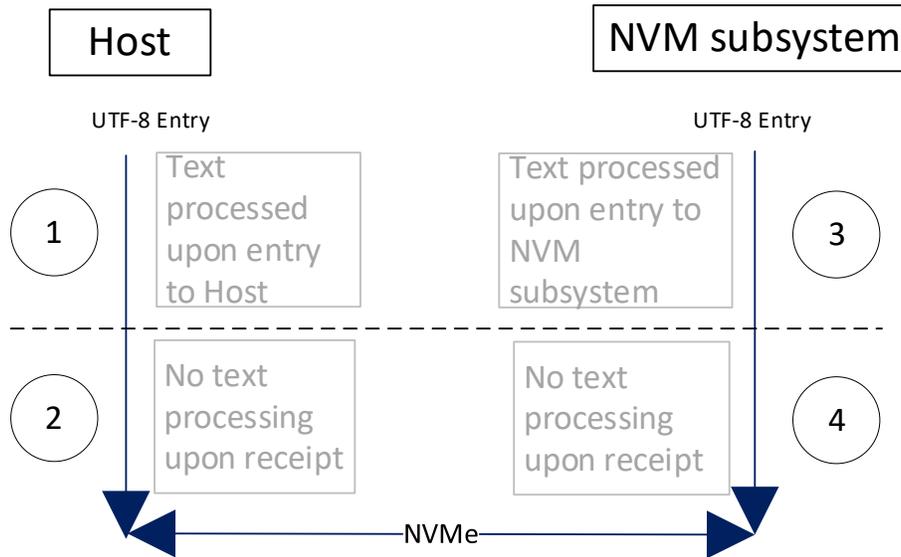
NVMe hosts, controllers and NVM subsystems process (e.g., store, compare) UTF-8 strings used by NVMe as binary strings without any text processing or text comparison logic that is specific to the Unicode character set or locale (e.g., check for byte values not used by UTF-8, Unicode normalization, etc.). Any such text processing:

- a) may occur as part of entry of UTF-8 strings into NVMe hosts and NVM subsystems as shown at points 1 and 3 in Figure nnn; and
- b) should not occur as part of receiving UTF-8 strings via NVMe, as shown at points 2 and 4 in Figure nnn.

Upon entry into the NVMe host (e.g., via a configuration interface at point 1 in Figure nnn, described as "input" in RFC4122), NVMe host software may process a UTF-8 string (e.g., perform Unicode normalization). Upon entry into the NVM subsystem (e.g., via a configuration interface at point 3 in Figure

nnn, described as “input” in RFC4122), a controller may process a UTF-8 string (e.g., perform Unicode normalization). Upon receipt by the host (e.g., at point 2 in Figure nnn) of a UTF-8 string from the controller, text processing (e.g., Unicode normalization) should not occur. Upon receipt by the controller (e.g., at point 4 in Figure nnn) of a UTF-8 string from the host, text processing (e.g., Unicode normalization) should not occur.

Figure nnn: UTF-8 Input Processing



Description of Specification Changes for TP8010a

1 Introduction

...

1.5 Definitions

...

1.5.NEW6 ~~symbolic-name~~ admin label ASCII

An ASCII string. Refer to section 1.4.2 for ASCII string requirements. Refer to section 1.5.NEW for admin label usage.

1.5.NEW7 admin label UTF-8

A UTF-8 string. Refer to section 1.4.2 for UTF-8 string requirements. Refer to section 1.5.NEW for admin label usage.

5.NEW1 Discovery Information Management command

...

Figure NEW.12: Extended Attribute

Bytes	Description																					
01:00	<p>Extended Attribute Type (EXATTYPE): This field specifies the type of extended attribute for the extended attribute contained in the extended discovery information entry.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>O/M ¹</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Reserved</td> </tr> <tr> <td>1h</td> <td>HM</td> <td>Host Identifier</td> </tr> <tr> <td>2h</td> <td>O</td> <td>Symbolic Name Admin label ASCII</td> </tr> <tr> <td>3h</td> <td>O</td> <td>Admin label UTF-8</td> </tr> <tr> <td>4h4h to FFFFh</td> <td></td> <td>Reserved</td> </tr> <tr> <td>FF00h to FFFFh</td> <td>O</td> <td>Vendor Specific</td> </tr> </tbody> </table>	Value	O/M ¹	Definition	0h		Reserved	1h	HM	Host Identifier	2h	O	Symbolic Name Admin label ASCII	3h	O	Admin label UTF-8	4h4h to FFFFh		Reserved	FF00h to FFFFh	O	Vendor Specific
Value	O/M ¹	Definition																				
0h		Reserved																				
1h	HM	Host Identifier																				
2h	O	Symbolic Name Admin label ASCII																				
3h	O	Admin label UTF-8																				
4h4h to FFFFh		Reserved																				
FF00h to FFFFh	O	Vendor Specific																				
03:02	<p>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.</p> <table border="1"> <thead> <tr> <th>Extended Attribute Type</th> <th>Length</th> </tr> </thead> <tbody> <tr> <td>Host Identifier</td> <td>16 bytes</td> </tr> <tr> <td>Admin label ASCII</td> <td rowspan="2">4 to 256 bytes</td> </tr> <tr> <td>Admin label UTF-8</td> </tr> </tbody> </table>	Extended Attribute Type	Length	Host Identifier	16 bytes	Admin label ASCII	4 to 256 bytes	Admin label UTF-8														
Extended Attribute Type	Length																					
Host Identifier	16 bytes																					
Admin label ASCII	4 to 256 bytes																					
Admin label UTF-8																						
(EXATLEN - 1) + 4:04	<p>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.</p> <table border="1"> <thead> <tr> <th>Extended Attribute Type</th> <th>Value</th> </tr> </thead> <tbody> <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>Admin label ASCII</td> <td>This value specifies the symbolic name admin label of the host or NVM subsystem encoded in ASCII. The symbolic name admin label is used to identify a host or an NVM subsystem.</td> </tr> <tr> <td>Admin label UTF-8</td> <td>This value specifies the admin label of the host or NVM subsystem encoded in UTF-8. The admin label is used to identify a host or an NVM subsystem.</td> </tr> </tbody> </table>	Extended Attribute Type	Value	Host Identifier	This value specifies the Host Identifier of the host, as defined in section 5.27.1.25.	Admin label ASCII	This value specifies the symbolic name admin label of the host or NVM subsystem encoded in ASCII. The symbolic name admin label is used to identify a host or an NVM subsystem.	Admin label UTF-8	This value specifies the admin label of the host or NVM subsystem encoded in UTF-8. The admin label is used to identify a host or an NVM subsystem.													
Extended Attribute Type	Value																					
Host Identifier	This value specifies the Host Identifier of the host, as defined in section 5.27.1.25.																					
Admin label ASCII	This value specifies the symbolic name admin label of the host or NVM subsystem encoded in ASCII. The symbolic name admin label is used to identify a host or an NVM subsystem.																					
Admin label UTF-8	This value specifies the admin label of the host or NVM subsystem encoded in UTF-8. The admin label is used to identify a host or an NVM subsystem.																					

Figure NEW.12: Extended Attribute

Bytes	Description
NOTES:	
1. O/M definition: O = Optional, M = Mandatory, HM = Mandatory for hosts and prohibited for NVM subsystems.	

Description of Specification Changes for NVMe-MI 1.2b specification

1.7 Conventions

...

~~Some parameters are defined as a string of ASCII or UTF-8 characters. ASCII strings shall contain only code values 20h to 7Eh. UTF-8 is backwards compatible with ASCII encoding and supports additional characters with variable length encoding. For the string "Copyright", the character "C" is the least significant byte, the character "o" is the second byte, etc. The string is left justified and shall be padded with spaces (ASCII character 20h) to the right if necessary.~~

Some parameters are defined as an ASCII string. ASCII strings shall contain only code values (i.e., byte values or octet values) 20h through 7Eh. For the string "Copyright", the character "C" is the first byte, the character "o" is the second byte, etc. ASCII strings are left justified. If padding is necessary, then the string shall be padded with spaces (i.e., ASCII character 20h) to the right unless the string is specified as null-terminated.

Some parameters are defined as a UTF-8 string. UTF-8 strings shall contain only byte values (i.e., octet values) 20h through 7Eh, 80h through BFh, and C2h through F4h (refer to sections 1 to 3 of RFC 3629). For the string "Copyright", the character "C" is the first byte, the character "o" is the second byte, etc. UTF-8 strings are left justified. If padding is necessary, then the string shall be padded with spaces (i.e., ASCII character 20h, Unicode character U+0020) to the right unless the string is specified as null-terminated.

If padding is necessary for a field that contains a null-terminated string then the field should be padded with nulls (i.e., ASCII character 00h, Unicode character U+0000) to the right of the string.

For any ASCII string or UTF-8 string received from a Requester, a Responder shall treat the string as a binary string (e.g., it shall not perform any text processing that is specific to the character set or locale such as checks for byte values not used by UTF-8, Unicode normalization, etc.).

...

1.11 References

...

PCI-SIG PCI Express SFF-8639 Module Specification, Revision 3.0, Version 1.0. Available from <https://www.pcisig.com>.

RFC 3629, F. Yergeau, "UTF-8, a transformation format of ISO 10646", November 2003. Available from <https://www.ietf.org/rfc.html>.

SNIA Native NVMe-oF™ Drive Specification, Version 1.0.1. Available from <https://www.snia.org>.

...