TRUSTED® COMPUTING GROUP

SPECIFICATION

TCG Storage Feature Set: Configurable Locking for NVMe Namespaces and SCSI LUNs

Version 1.02
Revision 1.14
September 15, 2021

Contact: admin@trustedcomputinggroup.org

PUBLIC REVIEW

WORK IN PROGRESS

This document is an intermediate draft for comment only and is subject to change without notice. Readers should not design products based on this document.

DISCLAIMERS, NOTICES, AND LICENSE TERMS

CHANGE HISTORY

REVISION	DATE	DESCRIPTION
V1.02 / D1.01	March 24, 2021	• Based on V1.01, D1.11 Public Review. • Changed “Informative Content” to “Informative Comment” • Added SUM_C field in the Configurable Namespace Locking Feature Descriptor • Added AssignToSUMRange parameter to the Assign Method • Updated behavior for Assign when AssignToSUMRange is not specified or is set to FALSE • Reorganized sections 4.6.2.1 and 4.6.2.2 to separate rules based on RangeStartLengthPolicy equal to 0 or 1. • Added rules related to RangeStartLengthPolicy equal to 1 (sections 4.6.2.1.2 and 4.6.2.2.2).
V1.02 / D1.02	April 5, 2021	• Addressed feedback from initial review. • Added Informative Content indicating a SD may have some ranges activated in SUM and some ranges not activated in SUM. • Clarified case where AssignToSUMRange is not specified or set to FALSE to handle case where SUM is not supported by SD. • Added requirement that if SUM_C is set to one then the SUM must be supported. • Cleaned up section 4.6.2.1.2 according to feedback from initial review
V1.02 / D1.03	April 21, 2021	• Fixed KeepNamespaceGlobalRangeKey section heading level in the Deassign Parameter Descriptions section • Fixed success case for SCSI in section 3.1.1.1.3.1 and success case in section 3.1.1.1.3.2 to match success case rule for NVMe when AssignToSUMRange is not specified or is set to False. This change is intended to make it more clear what the device should do when the AssignToSUMRange parameter is not specified or set to FALSE and the device does or does not support Single User Mode. • Added proposed text to make Format NVM handling more clear • Extended Format NVM interactions to explicitly indicate that Locking SP security states shall not be modified as a result of Format NVM command. • Updated Copyright year
V1.02 / D1.04	May 3, 2021	• Addressed all feedback from last review. • Clarified the error conditions for when the AssignToSUMRange is set to TRUE but SUM is not supported. • Removed old FormatNVM statements. • Cleaned up the range selection statements in section 3.1.1.1.3.1 and 3.1.1.1.3.2 to handle cases where SUM is supported and not supported.
V1.02 / D1.05	May 17, 2021	• Added informative content to section 3.1.1.1.3.1 to separate out the positive case for NVMe and SCSI interfaces • Modified the informative content of sections 3.1.1.1.3.1 and 3.1.1.1.3.2 to clarify that the items 1 and 2 are based on device’s support of SUM and that the device only needs to implement one or the other
V1.02 / D1.06	June 1, 2021	• Addressed the comments from the previous specification v1.01 which were not addressed at publication time • Updated section 1.3.5 to be similar to Opal 2.02 • Updated “Begin Informative Comment” statement to “Start of Informative Comment” to be similar to Opal 2.02 • Added explanation for the pseudo-code snippets listed in sections 3.1.1.1 and 3.1.1.2.
V1.02 / D1.07	June 15, 2021	• Clarified section 2.3.3 to simplify the NVMe FormatNVM interactions with namespaces where Namespace Non-global locking objects may or may not span the entire namespace
V1.02 / D1.08	June 21, 2021	• Added SD, SP, and TPer definitions to section 1.7 • Updated section 2.3.3 to specify the handling of all types of namespace global or non-global configurations
V1.02 / D1.09	June 29, 2021	• Cleaned up typo errors in section 4.6.2.2.2 • Added a rule to 4.6.2.2.2 clarifying the deassignment of a namespace GRLO when the global range locking object is activated in SUM and KeepNamespaceGlobalRangeKey is set to True. • Updated the reference for TCG SIIS to point to the latest revision • Cleaned up formatting for lists in section 4.6.2.2.2 to align to documents list convention in 1.3.6

V1.02 / D1.10	July 13, 2021	• Updated internal section references • Updates section 4.6.2.1.2 to differentiate when a TPer selects a Single User Mode Range or a Non-Single User Mode Range. • Updates section 3.1.1.1.3.4 to fix rule that allows the Assign method to succeed when the TPer is in Global Locking Object Multiple Namespaces Mode • Added Feature Set Minor Version Number field to the Configurable Namespace Locking Feature Descriptor • Updated the Version field in the Configurable Namespace Locking Feature Descriptor to be called Feature Descriptor Version Number and updated it to value 0x2 • Removed Opal SSC within the title of the feature set.
V1.02 / D1.11	July 27, 2021	• Removed unnecessary linefeeds which were breaking sentences • Cleaned formatting errors to align to Font conventions • Cleaned up capitalization of Feature Set • Cleaned up capitalization of Non-Global Locking object
V1.02 / D1.12	August 10, 2021	• Clarified Informative Comment section of 3.1.2 on the relationship of KeepGlobalRangeKey parameter with Namespace Global Range Locking objects. • Clarified the Revert and RevertSP methods behavior related to the Unused Key Count to make them work properly with both NVMe and SCSI interfaces. • Clarified the definition for Namespace Global Range Locking objects and Non-Global Range Locking objects to address both NVMe and SCSI interfaces • Cleaned up formatting errors to align to Font conventions
V1.02 / D1.13	September 14, 2021	• Addressing TC comments
V1.02 / D1.14	September 15, 2021	• Updated the Version field in Section 4.2.2.5 to align to the wording of the Configurable Namespace Locking Feature Descriptor Feature Descriptor Version Number

CONTENTS

1 Introduction

1.1 Document Purpose

The Storage Workgroup specifications provide a comprehensive architecture for Storage Devices under policy control as determined by the trusted platform host, the capabilities of the Storage Device to conform to the policies of the trusted platform, and the lifecycle state of the Storage Device as a Trusted Peripheral.

1.2 Scope and Intended Audience

This specification defines Configurable Locking for NVMe Namespaces and SCSI LUNs for the Opal Security Subsystem Class (SSC). Any Storage Device that claims Configurable Locking for NVMe Namespaces and SCSI LUNs compatibility SHALL conform to this specification.

The intended audience for this specification is both trusted Storage Device manufacturers and developers that want to use these Storage Devices in their systems.

1.3 Conventions

1.3.1 Document Precedence

In the event of conflict between this specification and other documents, the precedence for requirements is:

1. This specification;
2. TCG Storage Security Subsystem Class: Opal [4];
3. TCG Storage Interface Interactions Specification [3];
4. TCG Storage Architecture Core Specification [2]; and
5. TCG Single User Mode Feature Set [6].

1.3.2 Key Words

Key words are used to signify requirements.

The Key Words “SHALL”, “SHALL NOT”, “SHOULD,” and “MAY” are used in this document. These words are a subset of the RFC 2119 key words used by TCG, and have been chosen since they map to key words used in T10/T13 specifications. These key words are to be interpreted as described in [1].

In addition to the above key words, the following are also used in this document to describe the requirements of particular features, including tables, methods, and usages thereof.

- Mandatory (M): When a feature is Mandatory, the feature SHALL be implemented. A Compliance test SHALL validate that the feature is operational.
- Optional (O): When a feature is Optional, the feature MAY be implemented. If implemented, a Compliance test SHALL validate that the feature is operational.
- Excluded (X): When a feature is Excluded, the feature SHALL NOT be implemented. A Compliance test SHALL validate that the feature is not operational.
- Not Required (N) When a feature is Not Required, the feature MAY be implemented. No Compliance test is required.

1.3.3 Fonts

Names of methods and SP tables are in Courier New font (e.g., the `Set` method, the `Locking` table). This convention does not apply to method and table names appearing in headings or captions.

1.3.4 Statement Type

Please note a very important distinction between different sections of text throughout this document. There are two distinctive kinds of text: informative comment and normative statements. Because most of the text in this

specification will be of the kind normative statements, the authors have informally defined it as the default and, as such, have specifically called out text of the kind informative comment. They have done this by flagging the beginning and end of each informative comment and highlighting its text in gray. This means that unless text is specifically marked as of the kind informative comment, it can be considered a kind of normative statements.

EXAMPLE:

Start of Informative Comment

This is the first paragraph of 1-n paragraphs containing text of the kind informative comment ...

This is the second paragraph of text of the kind informative comment ...

This is the nth paragraph of text of the kind informative comment ...

To understand the TCG specification the user must read the specification. (This use of MUST does not require any action).

End of Informative Comment

1.3.5 Legend

The legend in Table 1 defines the SP table cell color coding, with the RGB values for the shading of each cell indicated in parentheses. This color coding is informative only. The table cell content is normative.

Table 1 - SP Table Legend

Table Cell Legend	R-W	Value	Access Control	Comment
Arial-Narrow (230, 230, 230)	Read-only	Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set specified	Fixed	• Cell content is Read-Only. • Access control is fixed. • Value is specified by the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set
Arial Narrow bold-under (230, 230, 230)	Read-only	VU	Fixed	• Cell content is Read-Only. • Access Control is fixed. • Values are Vendor Unique (VU). A minimum or maximum value may be specified.

Table Cell Legend	R-W	Value	Access Control	Comment
Arial-Narrow (0, 0, 0)	Not Defined	(N)	Not Defined	• Cell content is (N). • Access control is not defined. • Any text in table cell is informative only. • A `Get` MAY omit this column from the method response.
Arial Narrow bold-under (179, 179, 179)	Write	Preconfigured, user personalizable	Preconfigured, user personalizable	• Cell content is writable. • Access control is personalizable • `Get` Access Control is not described by this color coding
Arial-Narrow (179, 179, 179)	Write	Preconfigured, user personalizable	Fixed	• Cell content is writable. • Access control is fixed. • `Get` Access Control is not described by this color coding

1.3.6 Lists

1.3.6.1 Lists overview

Lists are associated with an introductory paragraph or phrase, and are numbered relative to that paragraph or phrase (i.e., all lists begin with an a) or 1) entry).

Each item in a list is preceded by an identification with the style of the identification being determined by whether the list is intended to be an ordered list or an unordered list.

If the item in a list is not a complete sentence, the first word in the item is not capitalized. If the item in a list is a complete sentence, the first word in the item is capitalized.

Each item in a list ends with a semicolon, except the last item, which ends in a period. The next to the last entry in the list ends with a semicolon followed by an “and” or an “or” (i.e., “…; and”, or “…; or”). The “and” is used if all the items in the list are required. The “or” is used if only one or more items in the list are required.

1.3.6.2 Unordered lists

An unordered list is one in which the order of the listed items is unimportant (i.e., it does not matter where in the list an item occurs as all items have equal importance). Each list item shall start with a lower case letter followed by a close parenthesis. If it is necessary to subdivide a list item further with an additional unordered list (i.e., have a nested unordered list), then the nested unordered list shall be indented and each item in the nested unordered list shall start with an upper case letter followed by a close parenthesis.

The following is an example of an unordered list with a nested unordered list:

EXAMPLE - The following are the items for the assembly:
a) a box containing:
    A) a bolt;

B) a nut; and
C) a washer;
b) a screwdriver; and
c) a wrench.

1.3.6.3 Ordered lists

An ordered list is one in which the order of the listed items is important (i.e., item n is required before item n+1). Each listed item starts with a Western-Arabic numeral followed by a close parenthesis. If it is necessary to subdivide a list item further with an additional unordered list (i.e., have a nested unordered list), then the nested unordered list shall be indented and each item in the nested unordered list shall start with an upper case letter followed by a close parenthesis.

The following is an example of an ordered list with a nested unordered list:

EXAMPLE - The following are the instructions for the assembly:
1) remove the contents from the box;
2) assemble the item;
A) use a screwdriver to tighten the screws; and
B) use a wrench to tighten the bolts;
and
3) take a break.

1.3.7 Numbering

A binary number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included between characters in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately preceded by "0x". Underscores or spaces may be included between characters in hexadecimal number representations to increase readability or delineate field boundaries (e.g., 0xFD8C FA23 or 0xFD8C_FA23). Hexadecimal numbers are in Courier New font.

A decimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 25). This standard uses the following conventions for representing decimal numbers:

a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
b) the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space; and
c) the thousands separator is used in both the integer portion and the fraction portion of a number.

A decimal number represented in this standard with an overline over one or more digits following the decimal point is a number where the overlined digits are infinitely repeating (e.g., $666.\overline{6}$ means 666.666 666… or 666 2/3, and $12.\overline{142857}$ means 12.142 857 142 857… or 12 1/7).

1.3.8 Bit conventions

Name (n:m), where n is greater than m, denotes a set of bits (e.g., Feature (7:0)). n:m where n is greater than m denotes a bit range in a table.

1.3.9 Number range convention

p..q, where p is less than q, represents a range of numbers (e.g., words 100..103 represents words 100, 101, 102, and 103).

1.4 Document References

[1] IETF RFC 2119, 1997, “Key words for use in RFCs to Indicate Requirement Levels”

[2] Trusted Computing Group (TCG), “TCG Storage Architecture Core Specification”, Version 2.01

[3] Trusted Computing Group (TCG), “Storage Interface Interactions Specification”, Version 1.09

[4] Trusted Computing Group (TCG), “TCG Storage Security Subsystem Class: Opal”, Version 2.01

[5] NVM Express, Inc., “NVM Express”, Revision 1.4

[6] Trusted Computing Group (TCG), “TCG Storage Opal SSC Feature Set: Single User Mode”, Version 1.00

[7] T10 INCITS, “SCSI Architecture Model - 6 (SAM-6)”, Revision 05

1.5 Dependencies on Other Feature Sets

This feature set does not depend upon any other feature sets.

1.6 Interactions with Other Feature Sets

This feature set defines the interactions with the Single User Mode Feature Set (see [6]) .

1.7 Definitions of Terms

Term	Definition
Globally-Associated Namespace	any namespace or LUN that is not associated with a Namespace Global Range Locking object, and is thus associated with the Global Range Locking object
LO	Locking object
LUN	Logical Unit Number
Namespace Global Range Locking object	For the NVMe interface: A Locking object with a NamespaceID column value not equal to zero or `0xFFFF_FFFF` and a NamespaceGlobalRange column value of TRUE For the SCSI interface: A Locking object with a NamespaceID column value not equal to `0xFFFF_FFFF` and a NamespaceGlobalRange column value of TRUE
Namespace Non-Global Range Locking object	For the NVMe interface: A Locking object with a NamespaceID column value not equal to zero or `0xFFFF_FFFF` and a NamespaceGlobalRange column value of FALSE For the SCSI interface: A Locking object with a NamespaceID column value not equal to `0xFFFF_FFFF` and a NamespaceGlobalRange column value of FALSE
Namespace/LUN	Namespace or LUN
Non-Global Range Locking object	a Locking object other than the Global Range Locking object (Namespace Global Range Locking objects and Namespace Non-Global Range Locking objects are examples of Non-Global Range Locking objects)
NS	Namespace

Term	Definition
OFS	Original Factory State
Read Locked	the state of a Locking object with a ReadLockEnabled column value of TRUE and a ReadLocked column value of TRUE
Read Unlocked	the state of a Locking object with a ReadLockEnabled column value of FALSE or a ReadLocked column value of FALSE
SD	Storage Device
SP	Security Provider
TPer	The Trusted Peripheral is the subset of a Storage Device for which TCG manages security
Unused Key Count	the number of media encryption key resources which are not in use and are available for use (see section 4.2.1.9)
Write Locked	the state of a Locking object with a WriteLockEnabled column value of TRUE and a WriteLocked column value of TRUE
Write Unlocked	the state of a Locking object with a WriteLockEnabled column value of FALSE or a WriteLocked column value of FALSE

2 Namespaces/LUNs Overview

2.1 Namespace/LUN Operations Overview

The Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set provides a means for a Locking SP to support separate locking management of different namespaces/LUNs, as well as of LBA ranges within a namespace/LUN.

This specification defines the following Locking object usages:

a) A Namespace Global Range Locking object is the first Locking object to be uniquely associated with a namespace/LUN; and

b) A Namespace Non-Global Range Locking object is a Locking object associated with an LBA range within a namespace/LUN.

This specification also defines the relationships of those Locking objects to LBA ranges, the interactions with the Global Range Locking object, and interactions with interface commands.

This specification defines two new methods, `Assign` (see 3.1.1.1) and `Deassign` (see 3.1.1.2). These methods associate Locking objects with namespaces/LUNs and, optionally, with LBA ranges in namespaces/LUNs.

This specification adds two columns to the `Locking` table, NamespaceID (see 3.2.2.1.1) and NamespaceGlobalRange (see 3.2.2.1.2). The column values determine with which namespace/LUN a Locking object is associated and whether a Locking object is a Namespace Global Range Locking object or a Namespace Non-Global Range Locking object.

Each logical block is associated with exactly one Locking object.

A Globally-Associated Namespace is a namespace/LUN that is associated with the Global Range Locking object.

If the Storage Device supports the NVMe interface, then the Locking SP is in one of three modes:

a) Global LO / Multiple NS: The TPer contains one or more namespaces, all of which are associated many-to-one with the Global Range Locking object;

b) Multiple LO / Single NS: The TPer contains exactly one namespace, which is associated one-to-one with the Global Range Locking object. LBA ranges in the namespace MAY be associated one-to-one with Non-Global Range Locking objects, by invoking the `Set` method; and

c) Multiple LO / Multiple NS: The TPer contains one or more namespaces, each of which is associated either many-to-one with the Global Range Locking object, or using the `Assign` method one-to-one with a Namespace Global Range Locking object. Each LBA range of a namespace MAY be associated one-to-one with a Namespace Non-Global Range Locking object.

If the Storage Device supports the SCSI interface, then the Locking SP is in one of two modes:

a) Global LO / Multiple NS: The device contains one or more LUNs, all of which are associated many-to-one with the Global Range Locking object; and

b) Multiple LO / Multiple NS: The device contains one or more LUNs, each of which is associated either many-to-one with the Global Range Locking object, or using the `Assign` method one-to-one with a Namespace Global Range Locking object. Each LBA range of a LUN MAY be associated one-to-one with a Namespace Non-Global Range Locking object.

These modes are shown in Figure 1 and Figure 2.

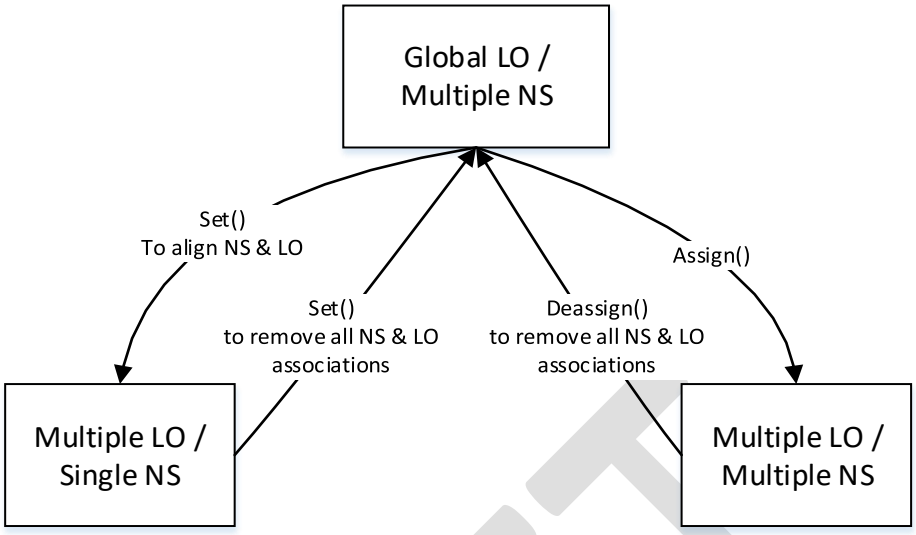

Figure 1 - Locking SP Modes (NVMe)

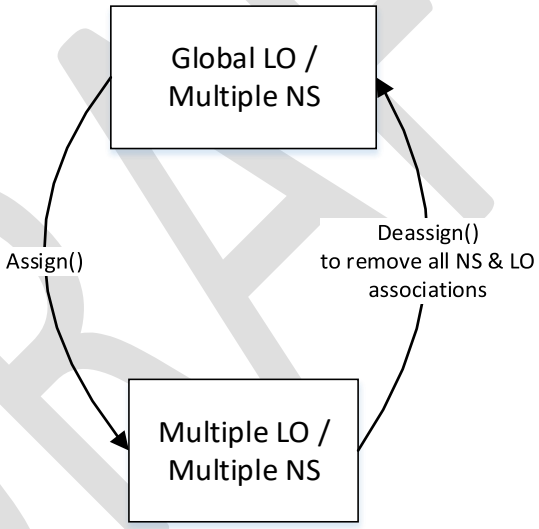

Figure 2 - Locking SP Modes (SCSI)

The uses of the `Set` method defined in [3] perform the transitions between “Global LO / Multiple NS mode” and “Multiple LO / Single NS mode” as shown above (although the terms “Global LO / Multiple NS mode” and “Multiple LO / Single NS mode” are not used in [3]). Transitions between the Global LO / Multiple NS mode and the Multiple LO / Multiple NS mode are specified in this specification. Transitions directly between the Multiple LO / Single NS mode and the Multiple LO / Multiple NS mode are not allowed.

If the Storage Device supports the NVMe interface and exactly one namespace exists in the TPer, and if no Non-Global Range Locking objects have been configured, then the TPer is in the Global LO / Multiple NS mode.

If the Storage Device supports the NVMe interface and exactly one namespace exists in the TPer, and if at least one Non-Global Range Locking object has been configured using the `Set` method, then the TPer is in the Multiple LO / Single NS mode.

In Multiple LO / Single NS mode, requirements are specified in [3] and the only section of this specification that applies is Level 0 Discovery (see 4.2).

In Multiple LO / Multiple NS mode, the requirements in [3] for single namespaces and multiple namespaces are overridden by the requirements in this specification.

2.2 Interactions with the Global Range Locking Object

Start of Informative Comment

The following rules specify the errors reported for commands that would violate the Read Locked or Write Locked state of a Globally-Associated namespace.

End of Informative Comment

If the Global Range Locking object is Read Locked, then any command that reads user data or metadata in a Globally-Associated Namespace SHALL fail with a status of Data Protection Error. See [3] for more information.

If the Global Range Locking object is Write Locked, then any command that modifies user data or metadata in a Globally-Associated Namespace SHALL fail with a status of Data Protection Error. See [3] for more information.

Start of Informative Comment

The following rules specify operations on the media encryption keys of Globally-Associated Namespaces.

End of Informative Comment

The TPer SHALL support a minimum of one media encryption key per namespace. In this case, the K_AES_* object referenced by the ActiveKey column value of the Global Range Locking object SHALL be a collective representation of all the media encryption keys in use for Globally-Associated Namespaces. Any method that modifies the Key column of the K_AES_* object indicated by the ActiveKey column of the Global Range Locking object SHALL be applied individually to each of the keys represented by that K_AES_* object. Successful execution of any method that results in the cryptographic erase of the Global Range Locking object SHALL result in the cryptographic erase of all Globally-Associated Namespaces.

2.3 NVMe Interactions

Start of Informative Comment

The following section describes interactions between the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set and the NVMe interface.

End of Informative Comment

2.3.1 Examples of Interactions with SIIS Single Namespace and Multiple Namespace Specifications

Start of Informative Comment

Table 2 illustrates a Locking SP in which all namespaces are implicitly associated with the Global Range Locking object.

Table 2 - Example: Global LO / Multiple NS

Locking Object	Namespace
Global	NS1 NS2 ... NSn

Table 3 illustrates a Locking SP with only a single namespace, but where invocations of the `Set` method have associated eight LBA ranges of the namespace with Non-Global Range Locking objects. The number of each LBA

range indicates the order in which that range was associated with a Locking object, e.g., NS1 LBA Range 1 was associated with LO1 first, NS1 LBA Range 2 was associated with LO2 second, etc. See [3].

Table 3 - Example: Multiple LO / Single NS

Locking Object	Namespace	Range within Namespace
Global	NS1	NS1 Global
LO1	NS1	NS1 LBA Range 1
LO2	NS1	NS1 LBA Range 2
LO3	NS1	NS1 LBA Range 8
LO4	NS1	NS1 LBA Range 4
LO5	NS1	NS1 LBA Range 6
LO6	NS1	NS1 LBA Range 5
LO7	NS1	NS1 LBA Range 7
LO8	NS1	NS1 LBA Range 3

Table 4 illustrates a Locking SP in which invocations of the `Assign` method have associated each of four namespaces with a different Non-Global Range Locking object. Three other namespaces remain implicitly associated with the Global Range Locking object.

Table 4 - Example: Multiple LO / Multiple NS

Locking Object	Namespace	Range within Namespace
Global	NS1	NS1 Global
	NS3	NS3 Global
	NS5	NS5 Global
LO1	NS6	NS6 Global
LO2	NS6	NS6 LBA Range 1
LO3	NS2	NS2 Global
LO4	NS4	NS4 Global
LO5	NS7	NS7 Global
LO6	NS7	NS7 LBA Range 2
LO7	NS7	NS7 LBA Range 1

The association of a namespace and its LBA ranges with one or more Locking objects is shown in Figure 3. When a namespace is first created, it is associated by default with the Global Range Locking object. When the `Assign` method is first invoked on a namespace, it selects a Non-Global Range Locking object in the `Locking` table with a NamespaceID column value of zero (i.e., is not associated with a namespace), configures the new Locking object as a Namespace Global Range Locking object, and associates the namespace with the new Locking object.

When the `Assign` method is invoked on a namespace which is associated with a Namespace Global Range Locking object, it selects a Non-Global Range Locking object in the `Locking` table with a NamespaceID column value of zero, configures the new Locking object as a Namespace Non-Global Range Locking object, and associates the

specified LBA range of that namespace with the new Locking object. Multiple LBA ranges within a namespace may be associated with different Locking objects. LBA ranges that have not been associated with Namespace Non-Global Range Locking objects are by default associated with the Namespace Global Range Locking object.

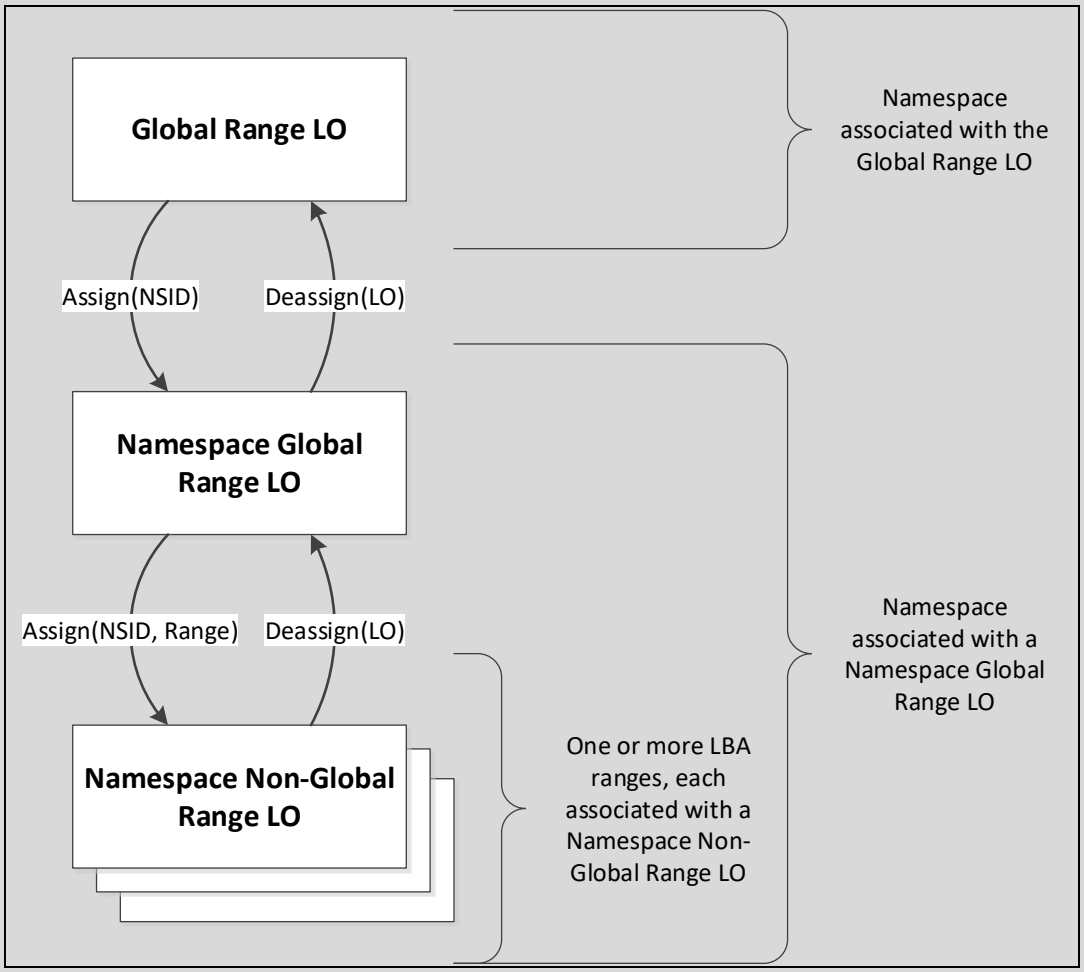

Figure 3 - Flows in Namespace Associations with Locking Objects

Invocation of the `Deassign` method on a Namespace Non-Global Range Locking object returns the LBA range associated with the Namespace Global Range Locking object. When there are no Namespace Non-Global Range Locking objects associated with a namespace, then the `Deassign` method may be invoked on the Namespace Global Range Locking object. This causes the namespace (and implicitly all its LBA ranges) to be associated with the Global Range Locking object.

End of Informative Comment

2.3.2 Interactions with the Namespace Management Command

The Namespace Management command (see [5]) MAY be supported on a TPer that supports the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set.

Start of Informative Comment

The following rule specifies normal operation of namespace creation.

End of Informative Comment

If:

a) the Select (SEL) field of the command is Create;
b) the Global Range Locking object is Read Unlocked;
c) the Global Range Locking object is Write Unlocked; and
d) the Unused Key Count is greater than or equal to one,

then:

1) the Namespace Management command SHALL be processed as defined in [5]; and
2) if the Namespace Management command succeeds in any life cycle state of the SPs, then:
    a. the Unused Key Count SHALL be decremented by one; and
    b. the new namespace SHALL be associated with an unused media encryption key.

Start of Informative Comment

The following rule prevents creation of a namespace if there is no media encryption key resource available.

End of Informative Comment

If:

a) the Select (SEL) field of the command is Create;
b) the Global Range Locking object is Read Unlocked;
c) the Global Range Locking object is Write Unlocked;
d) the command would otherwise succeed; and
e) the Unused Key Count is zero,

then:

a) the Namespace Management command SHALL fail with a status of Operation Denied; and
b) the Unused Key Count SHALL NOT be changed.

Start of Informative Comment

The following rule prevents creation of a namespace if the Global Range Locking Object is Read Locked or Write Locked.

End of Informative Comment

If:

a) the Select (SEL) field of the command is Create;
b) the Unused Key Count is greater than or equal to one;
c) the command would otherwise succeed; and
d) the Global Range Locking object is:
    a. Read Locked; or
    b. Write Locked,

then:

a) the Namespace Management command SHALL fail with a status of Operation Denied; and

b) the Unused Key Count SHALL NOT be changed.

Start of Informative Comment

The following rule specifies normal operation of namespace deletion. If the Namespace Identifier (NSID) field of the command indicates all namespaces (i.e., `0xFFFF_FFFF`), then all namespaces must be associated with the Global Range Locking object.

End of Informative Comment

If:

a) the Select (SEL) field of the command is Delete;

b) the Global Range Locking object is Read Unlocked;

c) the Global Range Locking object is Write Unlocked; and

d) the Namespace Identifier (NSID) field of the command does not specify any namespace associated with a Namespace Global Range Locking object;

then:

1. the Namespace Management command SHALL be processed as defined in [5]; and
2. if the Namespace Management command succeeds in any life cycle state of the SPs, then:
    a. the Unused Key Count SHALL be incremented by the number of namespaces that were deleted; and
    b. the media encryption key associated with any deleted namespace SHALL be eradicated.

Start of Informative Comment

The following rule prevents deletion of a namespace that is not associated with the Global Range Locking object.

End of Informative Comment

If:

a) the Select (SEL) field of the command is Delete;

b) the Global Range Locking object is Read Unlocked;

c) the Global Range Locking object is Write Unlocked:

d) the command would otherwise succeed; and

e) the Namespace Identifier (NSID) field of the command specifies one or more namespaces associated with a Namespace Global Range Locking object;

then the Namespace Management command SHALL fail with a status of Operation Denied.

Start of Informative Comment

The following rule prevents deletion of a namespace associated with the Global Range Locking object if the Global Range Locking object is Read Locked or Write Locked.

End of Informative Comment

If:

a) the Select (SEL) field of the command is Delete;

b) the Namespace Identifier (NSID) field of the command does not specify any namespace associated with a Namespace Global Range Locking object;

c) the command would otherwise succeed; and

d) the Global Range Locking object is:

   a. Read Locked; or

   b. Write Locked,

then the Namespace Management command SHALL fail with a status of Operation Denied.

2.3.3 Interactions with the Format NVM Command

Start of Informative Comment

The Format NVM command specifies that either one namespace or all namespaces (i.e., namespace ID `0xFFFF_FFFF`) are to be formatted.

End of Informative Comment

If the Format NVM command is received by the storage device and all logical blocks in every targeted Namespace are associated with a Locking object that is in a Write Unlocked state, then the Format NVM command SHALL be processed as specified in [5] and SHALL NOT alter the Locking SP configuration, including PIN values, access control settings, etc. See section 2.1 for more details on LBA and logical block association to Locking objects.

If the Format NVM command is received by the storage device and any logical block in any targeted Namespace is associated with a Locking object that is in a Write Locked state, then the Format NVM command SHALL fail with a status of Invalid Security State. See section 2.1 for more details on LBA and logical block association to Locking objects.

2.4 SCSI Interactions

Start of Informative Comment

The following section describes interactions between the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set and the SCSI interface.

End of Informative Comment

2.4.1 Interactions with the SCSI LUNs

Start of Informative Comment

The following rules specify configuration requirements for SCSI LUNs (see [7] for details).

End of Informative Comment

The Storage Device SHALL support a minimum of 2 LUNs.

The Storage Device SHALL NOT support LUN ID values that cannot be independently addressed using only the first four bytes of the LUN ID as described in section 3.2.2.1.1.

The Storage Device SHALL NOT support a LUN ID that maps to a NamespaceID column value of `0xFFFF_FFFF` as described in section 3.2.2.1.1.

2.4.2 Interactions with the SCSI Commands that affect multiple LUNs

Start of Informative Comment

SCSI commands may affect one or more LUNs on the Storage Device (see [7] for details).

The following rules specify the behavior when a command affects multiple LUNs.

End of Informative Comment

If:

a) the command affects two or more LUNs;

b) the command is a Read Command (see [3]); and

c) one or more of the Locking objects associated with any of the affected LUNs is Read Locked,

then the command SHALL behave as defined in the SCSI command interactions with the Locking SP table (see [3]).

If:

a) the command affects two or more LUNs;

b) the command is a Write Command (see [3]); and

c) one or more of the Locking objects associated with any of the affected LUNs is Write Locked,

then the command SHALL behave as defined in the SCSI command interactions with the Locking SP table (see [3]).

3 SSC Specific Functionality

This section specifies the additional SSC-specific functionality (not contained in [2] or [4]) required to support the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set.

3.1 Methods

This section defines new methods and modifications to existing methods required for this feature set.

3.1.1 New Methods

This section defines the new methods that are required to support this feature set.

3.1.1.1 Assign (M)

The `Assign` method is a Locking Template-specific method.

Start of Informative Comment

The `Assign` method selects a Non-Global Range Locking object in the `Locking` table that has a NamespaceID column value of zero or `0xFFFF_FFFF` (i.e., that is not associated with a namespace/LUN), and sets multiple column values in a single operation. The `Assign` method returns the UID column value of the selected Locking object and the NamespaceGlobalRange column value.

The `Assign` method is invoked upon a Locking SP that is in either the Global LO / Multiple NS mode or the Multiple LO / Multiple NS mode (see Figure 1).

End of Informative Comment

The following pseudo-code is the signature of the `Assign` method (see [2] for more information).

```
LockingTableUID.Assign [
     NamespaceID : bytes_4,
     RangeStart     = uinteger_8,
     RangeLength  = uinteger_8,
     AssignToSUMRange = boolean]
=>
[ UID                    : uidref,
  NamespaceGlobalRange : boolean ]
```

Method UID: 00 00 00 06 00 00 08 04

3.1.1.1.1 Parameter Descriptions

3.1.1.1.1.1 NamespaceID

The NamespaceID parameter specifies the value to which the NamespaceID column of the Locking object (see 3.2.2.1.1) SHALL be set.

3.1.1.1.1.2 RangeStart

The RangeStart parameter (if present) specifies the value to which the RangeStart column of the Locking object SHALL be set.

3.1.1.1.1.3 RangeLength

The RangeLength parameter (if present) specifies the value to which the RangeLength column of the Locking object SHALL be set.

3.1.1.1.1.4 AssignToSUMRange

Start of Informative Comment

Per [6], some locking ranges may be activated in SUM and others may not. The AssignToSUMRange parameter allows the host to tell the SD that the namespace should be assigned to a locking range activated in Single User Mode or not.

End of Informative Comment

The AssignToSUMRange parameter (if present) specifies that the SD should assign the namespace to a Locking object that is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table. See section 4.6.2.1

3.1.1.1.2 Returned Value Descriptions

3.1.1.1.2.1 UID

The UID value is the UID column value of the selected Locking object.

3.1.1.1.2.2 NamespaceGlobalRange

The NamespaceGlobalRange value is the NamespaceGlobalRange column value of the selected Locking object (see 3.2.2.1.2).

3.1.1.1.3 Assign Method Operation

The operation of the `Assign` method depends in part upon whether a Locking object is associated with the namespace/LUN specified by the method and whether the Storage Device supports Namespace Non-Global Range Locking objects:

a) If the `Locking` table contains no Locking objects associated with the specified namespace, then `Assign` configures the Namespace Global Range Locking object for that namespace; and

b) If the `Locking` table contains a Namespace Global Range Locking object associated with the specified namespace and the RANGE_C bit is set to one in the Configurable Namespace Locking Feature Descriptor (see 4.2.1), then `Assign` configures a Namespace Non-Global Range Locking object for that namespace. If the RangeLength parameter is set to zero, then no logical blocks are associated with that Locking object. Subsequently, the `Set` method MAY be invoked to associate logical blocks with that Locking object.

In order to prevent invalid configurations of the Locking SP, the following uses of `Assign` are not allowed:

a) configuring a Namespace Non-Global Range Locking object in a Storage Device that does not support Non-Global Range Locking objects; and

b) configuring a Namespace Non-Global Range Locking object for a range of a namespace that overlaps a range specified by another Namespace Non-Global Range Locking object for the same namespace.

The invocation of the `Assign` method (see 3.1.1.1) on a Globally-Associated Namespace associates a Namespace Global Range Locking object with that namespace/LUN. If `Assign` is invoked again on that namespace, then it associates a Namespace Non-Global Range Locking object with a (possibly zero-length) range of LBAs within that namespace/LUN.

If a namespace/LUN has LBAs that are not associated with any of the Namespace Non-Global Range Locking objects for that namespace/LUN, then those LBAs are associated with the Namespace Global Range Locking object.

A Namespace Non-Global Range Locking object with a RangeLength column value of zero is considered to not overlap with any other LBA range, regardless of the RangeStart column value.

3.1.1.1.3.1 Assigning a Namespace Global Range Locking object

Start of Informative Comment

The following rule specifies successful operation of the `Assign` method when assigning a Namespace Global Range Locking object on storage devices which support the NVMe interface.

Note: The behavior of the Assign method as described in items 1 and 2 below depends on the SD's support for Single User Mode Feature Set (see [6]) . A SD would not need to implement both items 1 and 2, but instead one or the other.

End of Informative Comment

If:

a) the Storage Device supports the NVMe interface

b) the NamespaceID parameter:

   a. is not equal to zero;

   b. is not equal to 0xFFFF_FFFF; and

   c. specifies a value that is not equal to the NamespaceID column value in any Locking object;

c) the RangeStart parameter, if present, is set to zero;

d) the RangeLength parameter, if present, is set to zero;

e) the Global Range Locking object is Write Unlocked;

f) the Global Range Locking object is Read Unlocked; and

g) the AssignToSUMRange parameter is not specified or is set to FALSE,

then the Assign method SHALL:

1. If the SD supports the Single User Mode Feature Set (see [6]), select a Non-Global Range Locking object in the Locking table with:
   a. a NamespaceID column value of zero; and
   b. a UID column value that is not included in the single_user_ranges list in the SingleUserModeRanges column of the LockingInfo table;
2. If the SD does not support the Single User Mode Feature Set (see [6]), select a Non-Global Range Locking object in the Locking table with a NamespaceID column value of zero.
3. set the NamespaceID column value to the value of the NamespaceID parameter;
4. set the RangeStart column value to zero;
5. set the RangeLength column value to zero;
6. set the NamespaceGlobalRange column value to TRUE;
7. keep the current Unused Key Count;
8. transfer control of the media encryption key associated with the namespace to the K_AES_* object indicated by the ActiveKey column value of the selected Locking object; and
9. return:
   a) the UID of the selected Locking object;
   b) the NamespaceGlobalRange column value; and
   c) a status code of SUCCESS.

Start of Informative Comment

The following rule specifies successful operation of the Assign method when assigning a Namespace Global Range Locking object on storage devices which support the SCSI interface.

Note: The behavior of the `Assign` method as described in items 1 and 2 below depends on the SD's support for Single User Mode Feature Set. A SD would not need to implement both items 1 and 2, but instead one or the other.

End of Informative Comment

If:

a) the Storage Device supports the SCSI interface;

b) the NamespaceID parameter:

   a. is not equal to `0xFFFF_FFFF`; and

   b. specifies a value that is not equal to the NamespaceID column value in any Locking object;

c) the RangeStart parameter, if present, is set to zero;

d) the RangeLength parameter, if present, is set to zero;

e) the Global Range Locking object is Write Unlocked;

f) the Global Range Locking object is Read Unlocked; and

g) the AssignToSUMRange parameter is not specified or is set to FALSE,

then the `Assign` method SHALL:

1. If the SD supports the Single User Mode Feature Set (see [6]), select a Non-Global Range Locking object in the `Locking` table with:
   a. a NamespaceID column value of zero; and
   b. a UID column value that is not included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table;
2. If the SD does not support the Single User Mode Feature Set (see [6]), select a Non-Global Range Locking object in the `Locking` table with a NamespaceID column value of zero. set the NamespaceID column value to the value of the NamespaceID parameter;
3. set the RangeStart column value to zero;
4. set the RangeLength column value to zero;
5. set the NamespaceGlobalRange column value to TRUE;
6. keep the current Unused Key Count;
7. transfer control of the media encryption key associated with the LUN to the K_AES_* object indicated by the ActiveKey column value of the selected Locking object; and
8. return:
   a. the UID of the selected Locking object;
   b. the NamespaceGlobalRange column value; and
   c. a status code of SUCCESS.

Assigning a Namespace Global Range Locking object SHALL NOT cause a cryptographic erase of that namespace/LUN.

Start of Informative Comment

The following rule prevents assignment of a Namespace Global Range Locking object if the Global Range Locking object is Write Locked or Read Locked.

End of Informative Comment

If:

a) the Storage Device supports the NVMe interface

b) the NamespaceID parameter:

   a. is not equal to zero;

   b. is not equal to `0xFFFF_FFFF`; and

   c. specifies a value that is not equal to the NamespaceID column value in any Locking object;

   and

c) the Global Range Locking object is Write Locked or Read Locked,

then the `Assign` method SHALL fail with a status of FAIL.

If:

a) the Storage Device supports the SCSI interface

b) the NamespaceID parameter:

   a. is not equal to `0xFFFF_FFFF`; and

   b. specifies a value that is not equal to the NamespaceID column value in any Locking object;

   and

c) the Global Range Locking object is Write Locked or Read Locked,

then the `Assign` method SHALL fail with a status of FAIL.

Start of Informative Comment

The following rule prevents assignment of a Namespace Global Range Locking object that specifies an LBA range.

End of Informative Comment

If:

a) the Storage Device is supports the NVMe interface

b) the NamespaceID parameter:

   a. is not equal to zero;

   b. is not equal to `0xFFFF_FFFF`; and

   c. specifies a value which is not equal to the NamespaceID column value in any Locking object;

   and

c) other parameters are set as follows:

   a. the RangeStart parameter is set to a nonzero value; or

   b. the RangeLength parameter is set to a nonzero value,

then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

If:

a) the Storage Device supports the SCSI interface

b) the NamespaceID parameter:

   a. is not equal to 0xFFFF_FFFF; and

   b. specifies a value that is not equal to the NamespaceID column value in any Locking object;

   and

c) other parameters are set as follows:

   a. he RangeStart parameter is set to a nonzero value; or

   b. the RangeLength parameter is set to a nonzero value,

then the Assign method SHALL fail with a status of INVALID_PARAMETER.

3.1.1.1.3.2 Assigning a Namespace Non-Global Range Locking object

Start of Informative Comment

The following rule specifies successful operation of the Assign method when assigning a Namespace Non-Global Range Locking object on storage devices which support the NVMe interface.

Note: The behavior of the Assign method as described in items 1 and 2 below depends on the SD's support for Single User Mode Feature Set (see [6]). A SD would not need to implement both items 1 and 2, but instead one or the other

End of Informative Comment

If:

a) the Storage Device reports a value of one in the Range Capable (Range_C) field of the Configurable Namespace Locking Feature Descriptor (see 4.2.1);

b) the NamespaceID parameter specifies a value that is equal to the NamespaceID column value in a Namespace Global Range Locking object;

c) the RangeStart parameter and the RangeLength parameter, if present, specify an LBA range in which no logical blocks are assigned to any Namespace Non-Global Range Locking object having a NamespaceID column value equal to the NamespaceID parameter;

d) the Maximum Ranges Per Namespace field of the Configurable Namespace Locking Feature Descriptor (see 4.2.1) specifies:

   a. a value of 0xFFFF_FFFF; or

   b. a value less than 0xFFFF_FFFF and that value is greater than the number of Namespace Non-Global Range Locking objects that have a NamespaceID column value equal to the NamespaceID parameter;

e) the Unused Key Count is greater than or equal to one; and

f) the AssignToSUMRange parameter is not specified or is set to FALSE,

then the Assign method SHALL:

1. If the SD supports the Single User Mode Feature Set (see [6]), select a Non-Global Range Locking object in the Locking table with:

   a. a NamespaceID column value of zero or 0xFFFF_FFFF;

b. a UID column value that is not included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table;

2. If the SD does not support the Single User Mode Feature Set (see [6]), select a Non-Global Range Locking object in the `Locking` table with a NamespaceID column value of zero or `0xFFFF_FFFF`;
3. set the NamespaceID column value to the value of the NamespaceID parameter;
4. set the RangeStart column value to the specified value or to zero if not specified;
5. set the RangeLength column value to the specified value or to zero if not specified;
6. set the NamespaceGlobalRange column value to FALSE;
7. decrement the Unused Key Count by one; and
8. return:
    a. the UID of the selected Locking object;
    b. the NamespaceGlobalRange column value; and
    c. a status code of SUCCESS.

Start of Informative Comment

The following rule prevents assignment of a Namespace Non-Global Range Locking object if the Range Capable bit is set to 0 in the Level 0 Discovery response data.

End of Informative Comment

If:

a) the Storage Device reports a value of zero in the Range Capable (Range_C) field of the Namespace Feature Descriptor; and

b) the `Assign` method specifies a NamespaceID parameter value that is equal to the NamespaceID column value in a Namespace Global Range Locking object,

then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

Start of Informative Comment

The following rule prevents assignment of a Namespace Non-Global Range Locking object with an LBA range that overlaps the LBA range in another Namespace Non-Global Range Locking object for the same namespace/LUN. A zero-length LBA range (i.e., the RangeLength parameter is zero) does not overlap any LBA range, regardless of the value of the RangeStart parameter.

End of Informative Comment

If the `Assign` method specifies:

a) a NamespaceID parameter equal to the NamespaceID column value in any Namespace Non-Global Range Locking object; and

b) an LBA range in which one or more logical blocks are assigned to that Namespace Non-Global Range Locking object,

then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

Start of Informative Comment

The following rule prevents assignment of a Namespace Non-Global Range Locking object if there are no resources to store a new media encryption key.

End of Informative Comment

If:

a) the Storage Device reports a value of one in the Range Capable (Range_C) field of the Configurable Namespace Locking Feature Descriptor (see 4.2.1);

b) the NamespaceID parameter specifies a value that is equal to the NamespaceID column value in a Namespace Global Range Locking object;

c) the RangeStart parameter and the RangeLength parameter, if present, specify an LBA range in which no logical blocks are assigned to any Namespace Non-Global Range Locking object having a NamespaceID column value equal to the NamespaceID parameter; and

d) the Unused Key Count is equal to zero,

then the `Assign` method SHALL fail with a status of FAIL.

Start of Informative Comment

The following rule prevents assignment of more Namespace Non-Global Range Locking objects to a namespace/LUN than are indicated by the Maximum Ranges Per Namespace field of the Namespace Feature Descriptor.

End of Informative Comment

If:

a) the Maximum Ranges Per Namespace field of the Configurable Namespace Locking Feature Descriptor (see 4.2.1) specifies a value less than `0xFFFF_FFFF`;

b) the Unused Key Count is greater than zero; and

c) the value of the Maximum Ranges Per Namespace field is equal to the number of Namespace Non-Global Range Locking objects that have a NamespaceID column value equal to the NamespaceID parameter,

then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

3.1.1.1.3.3 General requirements

Start of Informative Comment

The following rule prevents assignment of a Namespace Global Range Locking object or a Namespace Non-Global Range Locking object for a non-existent namespace/LUN. This rule implicitly prevents setting the NamespaceID column value to `0x0000_0000` or to `0xFFFF_FFFF` for Storage Devices that support the NVMe interface and `0xFFFF_FFFF` for Storage Devices that support the SCSI interface (see 3.2.2.1.1).

End of Informative Comment

If the Storage Device supports the NVMe interface and the NamespaceID parameter specifies a value that is not an allocated namespace identifier in the NVM subsystem [5], then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

If the Storage Device supports the SCSI interface and the NamespaceID parameter specifies a value of `0xFFFF_FFFF` or a LUN that does not exist, then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

3.1.1.1.3.4 Interaction with the namespace interactions specified in SIIS

The namespace interactions specified in [3] include the Global LO / Multiple NS mode and the Multiple LO / Single NS mode shown in Figure 1.

Start of Informative Comment

The following rule describes the condition in which the Locking SP is in the Global LO / Multiple NS mode and is thus allowed to transition to the Multiple LO / Multiple NS mode. This condition is not the only condition in which the `Assign` method may be invoked.

End of Informative Comment

If:

a) the Storage Device supports the NVMe interface;

b) the TPer contains one or more namespaces; and

c) all Non-Global Range Locking objects have a RangeStart column value equal to zero and a RangeLength column value equal to zero,

then the `Assign` method MAY succeed with a status of SUCCESS.

Start of Informative Comment

The following rule prevents successful operation of the `Assign` method if the Locking SP is in the Multiple LO / Single NS mode.

End of Informative Comment

If the `Locking` table contains any Locking object with a NamespaceID column value of zero or `0xFFFF_FFFF`; and

a) the Storage Device supports the NVMe interface;

b) the value of the RangeStart column is not equal to zero; or

c) the value of the RangeLength column is not equal to zero,

then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

If the `Locking` table contains any Locking object with a NamespaceID column value of `0xFFFF_FFFF`; and

a) the Storage Device supports the SCSI interface;

b) the value of the RangeStart column is not equal to zero; or

c) the value of the RangeLength column is not equal to zero,

then the `Assign` method SHALL fail with a status of INVALID_PARAMETER.

3.1.1.1.3.5 Insufficient Unassigned Locking Objects

Start of Informative Comment

The following rule prevents successful operation of the `Assign` method when there are no Namespaces or LUNs available for assignment.

End of Informative Comment

If the Storage Device supports the NVMe interface and the `Locking` table contains no Non-Global Range Locking object with the NamespaceID column value equal to zero or `0xFFFF_FFFF`, then the `Assign` method SHALL fail with a status of INSUFFICIENT_ROWS (see [2]).

If the Storage Device supports the SCSI interface and the `Locking` table contains no Non-Global Range Locking object with the NamespaceID column value equal to `0xFFFF_FFFF`, then the `Assign` method SHALL fail with a status of INSUFFICIENT_ROWS (see [2]).

3.1.1.2 Deassign (M)

The `Deassign` method is a Locking Template-specific method.

Start of Informative Comment

The `Deassign` method removes a Locking object's association with a namespace/LUN or namespace/LUN LBA range and resets multiple column values in a single operation.

The `Deassign` method is invoked upon a Locking SP which is in the Multiple LO / Multiple NS mode (see Figure 1).

End of Informative Comment

The following pseudo-code is the signature of the `Deassign` method (see [2] for more information).

```
LockingTableUID.Deassign[
    UID                          : uidref,
    KeepNamespaceGlobalRangeKey = boolean ]
=>
[ ]
```

Method UID: 00 00 00 06 00 00 08 05

3.1.1.2.1 Parameter Descriptions

3.1.1.2.1.1 UID

The UID parameter specifies the Locking object that is to be deassigned.

3.1.1.2.1.2 KeepNamespaceGlobalRangeKey

The KeepNamespaceGlobalRangeKey parameter specifies whether the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value is eradicated when a Namespace Global Range Locking object is deassigned.

A TRUE value for the KeepNamespaceGlobalRangeKey parameter is allowed only when the `Deassign` method is called on a Namespace Global Range Locking object.

3.1.1.2.2 Deassign Method Operation

Start of Informative Comment

The `Deassign` method is used to remove the association between an LBA range and a Namespace Non-Global Range Locking object or between a namespace/LUN and a Namespace Global Range Locking object. Logical blocks associated with the deassigned Locking object are then associated with another Locking object:

a) Deassigning a Namespace Non-Global Range Locking object:
   a. associates the logical blocks in the LBA range indicated by the Locking object with the Namespace Global Range Locking object associated with the namespace/LUN; and
   b. cryptographically erases the LBA range indicated by the Locking object with the Namespace Non-Global Range Locking object associated with the Namespace/LUN

b) Deassigning a Namespace Global Range Locking object:
   a. associates the logical blocks in the namespace/LUN with the Global Range Locking object; and
   b. cryptographically erases the LBA range indicated by the Locking object with the Namespace Global Range Locking object associated with the Namespace/LUN if specified by the KeepNamespaceGlobalRangeKey parameter.

The Deassign method is subject to constraints that prevent invalid configurations of the Locking SP. The following uses of Deassign are not allowed:

a) Deassigning a Namespace Global Range Locking object when there exist one or more Namespace Non-Global Range Locking objects associated with that namespace/LUN; and

b) Deassigning a Namespace Non-Global Range Locking object and retaining the media encryption key.

The constraint in a) above requires that the Deassign method be invoked successfully on all of the Namespace Non-Global Range Locking objects associated with the namespace/LUN before the Deassign method is invoked on the Namespace Global Range Locking object associated with the namespace/LUN.

It is required that the Namespace Global Range Locking object be deassigned from a namespace/LUN before the Namespace Management command is invoked to delete the namespace/LUN. For the details of using the Namespace Management command for this purpose, see 2.3.2 and [3].

End of Informative Comment

If the Deassign method succeeds, then:

1. the method SHALL set all column values in the selected Locking object to original factory values; and
2. if the Locking object is a Namespace Global Range Locking object, the method SHALL process the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value as specified by the KeepNamespaceGlobalRangeKey parameter.

Note that upon successful completion of the Deassign method, the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value is not in use for any logical blocks.

Start of Informative Comment

The following rule specifies successful operation of the Deassign method on a Namespace Non-Global Range Locking object.

End of Informative Comment

If:

a) the Storage Device supports the NVMe interface;

b) the Locking object indicated by the UID parameter:

   a. is Read Unlocked and Write Unlocked;

   b. has the NamespaceGlobalRange column value set to FALSE; and

   c. has the NamespaceID column value set to a value other than zero or 0xFFFF_FFFF;

   and

c) the KeepNamespaceGlobalRangeKey parameter is FALSE,

then the Deassign method SHALL:

1. eradicate the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value;
2. increment the Unused Key Count by one; and
3. return a status code of SUCCESS.

If:

a) the Storage Device supports the SCSI interface;

b) the Locking object indicated by the UID parameter:

a. is Read Unlocked and Write Unlocked;
b. has the NamespaceGlobalRange column value set to FALSE; and
c. has the NamespaceID column value set to a value other than `0xFFFF_FFFF`;

and

c) the KeepNamespaceGlobalRangeKey parameter is FALSE,

then the `Deassign` method SHALL:

1. eradicate the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value;
2. increment the Unused Key Count by one; and
3. return a status code of SUCCESS.

Start of Informative Comment

The following rule prevents successful operation of the `Deassign` method on a Namespace Non-Global Range Locking object with the KeepNamespaceGlobalRangeKey parameter set to TRUE.

End of Informative Comment

If:

a) the Locking object indicated by the UID parameter has the NamespaceGlobalRange column value set to FALSE; and
b) the KeepNamespaceGlobalRangeKey parameter is TRUE,

then the `Deassign` method SHALL fail with a status of INVALID_PARAMETER.

Start of Informative Comment

The following rule prevents deassignment of a Namespace Non-Global Range Locking object when that Locking object is Read Locked or Write Locked.

End of Informative Comment

If the Locking object indicated by the UID parameter:

a) has the NamespaceGlobalRange column value set to FALSE; and
b) is Read Locked or Write Locked,

then the `Deassign` method SHALL fail with a status of FAIL.

Start of Informative Comment

The following rule specifies successful operation of the `Deassign` method on a Namespace Global Range Locking object when the KeepNamespaceGlobalRangeKey parameter is set to TRUE.

End of Informative Comment

If:

1. the Global Range Locking object is Read Unlocked and Write Unlocked;
2. the Locking object indicated by the UID parameter:
   a. has the NamespaceGlobalRange column value set to TRUE;
   b. is Read Unlocked and Write Unlocked; and

c. has a NamespaceID column value that is not equal to the NamespaceID column value in any other Locking object in the Locking SP;

and

3. the KeepNamespaceGlobalRangeKey parameter is set to TRUE,

then the `Deassign` method:

1. SHALL NOT perform a cryptographic erase of the LBAs in that namespace/LUN;
2. SHALL NOT change the Unused Key Count;
3. SHALL transfer control of the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value of the Namespace Global Range Locking object to the K_AES_* object indicated by the ActiveKey column value of Global Range Locking object;
4. SHALL change the value of the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value of the Namespace Global Range Locking object; and
5. SHALL return a status code of SUCCESS.

Start of Informative Comment

The following rule specifies successful operation of the `Deassign` method on a Namespace Global Range Locking object when the KeepNamespaceGlobalRangeKey parameter is set to FALSE.

End of Informative Comment

If:

a) the Global Range Locking object is Read Unlocked and Write Unlocked;

b) the Locking object indicated by the UID parameter:

   a. has the NamespaceGlobalRange column value set to TRUE;

   b. is Read Unlocked and Write Unlocked; and

   c. has a NamespaceID column value that is not equal to the NamespaceID column value in any other Locking object in the Locking SP;

   and

c) the KeepNamespaceGlobalRangeKey parameter is set to FALSE,

then the `Deassign` method:

1. SHALL eradicate the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value;
2. SHALL fill in the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value of the Namespace Global Range Locking object with new key material;
3. SHALL transfer control of the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value of the Namespace Global Range Locking object to the K_AES_* object indicated by the ActiveKey column value of Global Range Locking object;
4. SHALL fill in the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value of the Namespace Global Range Locking object with new key material;
5. SHALL NOT change the Unused Key Count; and
6. SHALL return a status code of SUCCESS.

Start of Informative Comment

The following rule prevents deassignment of a Namespace Global Range Locking object when a Namespace Non-Global Range Locking object is associated with that namespace/LUN.

End of Informative Comment

If the Locking object indicated by the UID parameter:

a) has the NamespaceGlobalRange column value set to TRUE; and

b) has a NamespaceID column value equal to the NamespaceID column value in another assigned Locking object in the Locking SP,

then the `Deassign` method SHALL fail with a status of INVALID_PARAMETER.

Start of Informative Comment

The following rule prevents deassigment of a Namespace Global Range Locking object when that Locking object or the Global Range Locking object is Read Locked or Write Locked.

End of Informative Comment

If:

a) the Locking object indicated by the UID parameter:

   a. has the NamespaceGlobalRange column value set to TRUE; and

   b. is Read Locked or Write Locked;

   or

b) the Global Range Locking object is Read Locked or Write Locked,

then the `Deassign` method SHALL fail with a status of FAIL.

Start of Informative Comment

The following rule specifies the error to report when the `Deassign` method is invoked on a non-existent or unassigned Namespace Global Range Locking object. The rule also specifies the error to report when the `Deassign` method is invoked on the Global Range Locking object.

End of Informative Comment

If the Storage Device supports the NVMe interface and the Locking object indicated by the UID parameter:

a) does not exist; or

b) has a NamespaceID column value of zero or `0xFFFF_FFFF`,

then the `Deassign` method SHALL fail with a status of INVALID_PARAMETER.

If the Storage Device supports the SCSI interface and the Locking object indicated by the UID parameter:

a) does not exist; or

b) has a NamespaceID column value of `0xFFFF_FFFF`,

then the `Deassign` method SHALL fail with a status of INVALID_PARAMETER.

3.1.2 Modified Methods

This feature set modifies the following methods:

a) `Set`;

b) `Revert`; and

c) `RevertSP`.

3.1.2.1 Set

Start of Informative Comment

The `Set` method is subject to constraints that prevent invalid configurations of the Locking SP. The following uses of Set are not allowed, based on access control settings:

a) Changing which namespace/LUN is associated with a Locking object (i.e., changing the NamespaceID column value);

b) Changing a Namespace Global Range Locking object into a Namespace Non-Global Range Locking object (i.e., changing the NamespaceGlobalRange column value);

c) Changing a Namespace Non-Global Range Locking object into a Namespace Global Range Locking object (i.e., changing the NamespaceGlobalRange column value); and

d) Assigning more than one Namespace Global Range Locking object for the same namespace/LUN.

Because the constraints described in this section apply to Namespace Global Range Locking objects and Namespace Non-Global Range Locking objects, they are meaningful only when the Locking SP is in the Multiple LO / Multiple NS mode (see 2.1).

The constraints described in this section are enforced by the access control list for the Locking SP, which does not permit the `Set` method to modify either the NamespaceID column value or NamespaceGlobalRange column value.

The following rule prevents the modification of a Namespace Non-Global Range Locking object to cause its LBA range to overlap the LBA range of another Namespace Non-Global Range Locking object.

End of Informative Comment

If the `Set` method is invoked on a Namespace Non-Global Range Locking object and specifies an LBA range in which one or more logical blocks are associated with a different Non-Global Range Locking object specifying the same NamespaceID column value, then the `Set` method SHALL fail with a status of INVALID_PARAMETER.

Start of Informative Comment

The following rule prevents the modification of the RangeStart and RangeLength columns on Namespace Global Range Locking objects.

End of Informative Comment

If the `Set` method is invoked on a Namespace Global Range and specifies values for the RangeStart or RangeLength columns, then the `Set` method SHALL fail with a status of INVALID_PARAMETER.

3.1.2.1.1 Interaction with the namespace management model specified in SIIS

Start of Informative Comment

The namespace management model specified in SIIS applies when the Locking SP is in either the Global LO / Multiple NS mode or the Multiple LO / Single NS mode (see 2.1), i.e., the `Locking` table does not contain a Namespace Global Range Locking object or a Namespace Non-Global Range Locking object. The Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set functionality is disallowed when the Locking SP is in the Global LO / Multiple NS mode.

End of Informative Comment

If the Storage Device supports the NVMe interface and any Locking object has a non-zero NamespaceID column value and the `Set` method is invoked on any Locking object with a NamespaceID column value of zero, other than the Global Range Locking object, then the `Set` method SHALL fail with a status of INVALID_PARAMETER.

3.1.2.2 Revert

Upon a successful invocation of the Revert method that results in reverting the Locking SP, the method SHALL increment the Unused Key Count by the number of Namespace Non-Global Range Locking objects when the method was invoked.

The Unused Key Count SHALL NOT be otherwise affected by the Revert method invocation, i.e. the Unused Key Count should not be returned to its OFS values.

3.1.2.3 RevertSP

Start of Informative Comment

If a namespace is associated with a Namespace Global Range Locking object when the RevertSP method is invoked, then the media encryption key of that namespace is eradicated, regardless of whether the KeepGlobalRangeKey parameter is set to TRUE or FALSE.

If the invoker of the RevertSP method wishes to keep the keys associated with Namespace Global Range Locking objects, the Deassign method should be invoked on those Namespace Global Range Locking objects prior to invoking the RevertSP method.

If the Deassign method is successfully invoked on a Namespace Global Range Locking object before the RevertSP method is invoked, then the media encryption key of that namespace/LUN would be associated with the Global Range Locking object at time of RevertSP method invocation.

End of Informative Comment

Upon successful invocation of the RevertSP method, the method SHALL increment the Unused Key Count by the number of Namespace Non-Global Range Locking objects when the method was invoked.

If the RevertSP method is invoked with the KeepGlobalRangeKey parameter set to TRUE, then the TPer SHALL:

a) continue to use the media encryption key for each namespace/LUN that was associated with the Global Range Locking object; and

b) eradicate the media encryption key associated with the K_AES_* object indicated by the ActiveKey column value of each Non-Global Locking object.

The Unused Key Count SHALL NOT be otherwise affected by the RevertSP method invocation, i.e. the Unused Key Count should not be returned to its OFS values.

3.2 Tables

This section defines new tables and modifications to existing tables required for this feature set.

3.2.1 New Tables

There are no new tables defined by this feature set.

3.2.2 Modified Tables

This feature set modifies the following tables:

a) Locking.

3.2.2.1 Locking SP

This feature set modifies the Locking table by adding the following columns (see Table 5), in addition to those defined in [2]:

Table 5 - Locking SP – Locking Table Columns

Column Number	Column Name	IsUnique	Column Type
0x14	NamespaceID		bytes_4
0x15	NamespaceGlobalRange		boolean

The behavior of the Global Range Locking object is modified (see 4.4.1.1.1).

3.2.2.1.1 NamespaceID (M)

Start of Informative Comment

The NamespaceID column name was defined prior to SCSI LUNs being included in this feature set. The name was retained to maintain backward compatibility.

End of Informative Comment

The NamespaceID column value indicates which namespace/LUN is associated with this Locking object.

If the Storage Device supports the NVMe interface, the NamespaceID column value of the Global Range Locking object SHALL be set to 0x0000_0000 or `0xFFFF_FFFF`.

If the Storage Device supports the SCSI interface, the NamespaceID column value of the Global Range Locking object SHALL be set to `0xFFFF_FFFF`.

If the Storage Device supports the SCSI interface, the Storage Device SHALL map LUN ID values (eight bytes) to NamespaceID column values (4 bytes) by taking the value of the first four bytes (i.e., 0-3) of the LUN ID.

3.2.2.1.2 NamespaceGlobalRange (M)

Start of Informative Comment

The NamespaceGlobalRange column name was defined prior to SCSI LUNs being included in this feature set. The name was kept to maintain backward compatibility.

End of Informative Comment

The NamespaceGlobalRange column value indicates whether the Locking object is associated with:

a) a designated LBA range in the namespace/LUN; or

b) all logical blocks in the namespace/LUN that are not associated with any other Locking object.

If the NamespaceGlobalRange column value is FALSE, then the Locking object is associated with the LBA range indicated by the RangeStart and RangeLength column values, in the namespace/LUN indicated by the NamespaceID column value.

If the NamespaceGlobalRange column value is TRUE, then the Locking object is associated with all logical blocks in the namespace/LUN that are not associated with any other Locking object.

The NamespaceGlobalRange column value of the Global Range Locking object SHALL be ignored,

3.2.2.2 Access Control (M)

The Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set modifies the `AccessControl` table by adding and modifying the following rows defined in [4]:

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
ACE																
		00 00 00 08 00 03 80 02	ACE_Locking_Namespace_IdtoGlbRng	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 80 02	ACE_Locking_Namespace_IdtoGlbRng	Get		ACE_ACE_Get_All				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
Locking																
		00 00 08 02 00 03 00 01	Locking_Range1	Get		ACE_Locking_Range1_Get_RangeStartToActiveKey, ACE_Anybody_Get_CommonName,, ACE_Locking_Namespace_IDtoGlbRng				ACE_Anybody						
		00 03 00 00 (+NN NN)	Locking_RangeNNNN	Get		ACE_Locking_RangeNNNN_Get_RangeStartToActiveKey, ACE_Anybody_Get_CommonName, ACE_Locking_Namespace_IDtoGlbRng				ACE_Anybody						

3.2.2.3 ACE (M)

The Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set modifies the ACE Table by adding the following rows, in addition to those defined in [4]:

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
Locking					

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	00 00 00 08 00 03 80 02	"ACE_Locking_Namespace_IdtoGlbRng"		Admins	NamespaceID, NamespaceGlobalRange

3.3 Types

This section defines new types and modifications to existing types required for this feature set.

3.3.1 New Types

There are no new types defined by this feature set.

3.3.2 Modified Types

There are no types modified by this feature set.

4 Feature Set Requirements

This section defines the Mandatory (M) and Optional (O) requirements for the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set.

4.1 Requirements Overview

The Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set consists of namespace/LUN specific capabilities that MAY be implemented in a TPer. A Host discovers the namespace/LUN specific capabilities and properties of a TPer by examining the Configurable Namespace Locking Feature Descriptor returned in Level 0 Discovery.

4.2 Level 0 Discovery

A Storage Device implementing the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set SHALL:

a) return the Configurable Namespace Locking Feature Descriptor as defined in 4.2.1; and

b) support the Level 0 Discovery response requirements defined in [4].

4.2.1 Configurable Namespace Locking Feature Descriptor (Feature Code = 0x0403) (M)

The Configurable Namespace Locking Feature Descriptor SHALL be returned when the Storage Device supports the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set. The contents of the feature descriptor are defined in Table 6.

Table 6 - Level 0 Discovery – Configurable Namespace Locking Feature Descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1				Feature Code				(LSB)
2	Feature Descriptor Version Number				Feature Set Minor Version Number			
3	Length							
4	Range_C	Range_P	SUM_C	Reserved				
5 – 7	Reserved							
8	(MSB)							
…				Maximum Key Count				
11								(LSB)
12	(MSB)							
…				Unused Key Count				
15								(LSB)
16	(MSB)							
…				Maximum Ranges Per Namespace				
19								(LSB)

4.2.1.1 Feature Code

The Feature Code field value SHALL be set to `0x0403`.

4.2.1.2 Feature Descriptor Version Number

The Feature Descriptor Version Number field SHALL be set to `0x2`.

4.2.1.3 Feature Set Minor Version Number

The Feature Set Minor Version Number represents the minor version of the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set supported by the Storage Device.

The Feature Set Minor Version Number field SHALL be set to a value as specified in Table 7.

Table 7 - Feature Set Minor Versions

Feature Set Minor Version Number	Standard Referenced
0x00	Configurable Namespace Locking v1.00
0x01	Configurable Locking for NVMe Namespaces and SCSI LUNs v1.01
0x02	Configurable Locking for NVMe Namespaces and SCSI LUNs v1.02
All others	Reserved

4.2.1.4 Length

The Length field indicates the number of bytes in the descriptor following byte 3. The value of the Length field SHALL be set to `0x10`.

4.2.1.5 Range_C

The Range Capable (Range_C) field is set to one to indicate that the Storage Device supports Namespace Non-Global Range Locking objects. The Range_C field is set to zero to indicate that the Storage Device does not support Namespace Non-Global Range Locking objects.

If the Range_C field is set to one, then the Storage Device SHALL support Namespace Level 0 Discovery as defined in 4.2.2.

If the Range_C field is set to zero, then the Storage Device SHOULD support Namespace Level 0 Discovery as defined in 4.2.2.

4.2.1.6 Range_P

The Range Present (Range_P) field is set to one to indicate that the `Locking` table contains one or more Namespace Non-Global Range Locking objects. The Range_P field is set to zero to indicate that the `Locking` table does not contain any Namespace Non-Global Range Locking objects.

4.2.1.7 SUM_C

The SUM Capable (SUM_C) field indicates whether the SD supports assigning namespaces to ranges that are activated in Single User Mode.

If the SUM_C field is set to one, then the SD supports the AssignToSUMRange parameter in the `Assign` method (see section 3.1.1.1.1.4) to support assigning namespaces to ranges that are activated in Single User Mode (see section 4.6.2.1.2).

If the SUM_C field is set to one, then the SD SHALL support the Single User Mode Feature Set defined in [6].

If the SUM_C field is cleared to zero, then the SD does not support the AssignToSUMRange parameter in the `Assign` method and does not support assigning namespaces to ranges that are activated in Single User Mode.

4.2.1.8 Maximum Key Count

The Maximum Key Count field indicates the maximum number of media encryption keys the Storage Device supports concurrently.

The value of the Maximum Key Count field SHALL be set during the Storage Device manufacturing process and be greater than or equal to the Number of Namespaces (see [5]). The value of the Maximum Key Count field MAY be less than the Number of Namespaces multiplied by the value of the Maximum Ranges Per Namespace field.

4.2.1.9 Unused Key Count

The Unused Key Count field indicates how many media encryption keys are unused and are thus available for use.

If the Locking SP is in the Multiple LO / Single NS mode, the value of the Maximum Key Count field and the value of the Unused Key Count field are related according to the following equation:

```
Unused Key Count  =  Maximum Key Count
                       - (  number of existing namespaces/LUNs in the TPer (i.e., one)
                          +  number of configured Non-Global Range Locking objects
                         )
```

If the Locking SP is in the Global LO / Multiple NS mode or in the Multiple LO / Multiple NS mode, the value of the Maximum Key Count field and the value of the Unused Key Count field are related according to the following equation:

```
Unused Key Count  =  Maximum Key Count
                       - ( number of existing namespaces/LUNs in the TPer
                          + number of configured Namespace Non-Global Range Locking objects
                         )
```

4.2.1.10 Maximum Ranges Per Namespace

The Maximum Ranges Per Namespace field indicates the maximum number of Namespace Non-Global Range Locking objects that can be configured for each namespace/LUN. If the Maximum Ranges Per Namespace field is set to a value less than `0xFFFF_FFFF`, then that value is the maximum number of Namespace Non-Global Range Locking objects that are able to be assigned to any namespace/LUN.

If the Maximum Ranges Per Namespace field is set to `0xFFFF_FFFF`, then a limit set by this field SHALL NOT apply.

If the Range_C field is set to one, then the Maximum Ranges Per Namespace field SHALL be set to a nonzero value.

If the Range_C field is set to zero, then the Maximum Ranges Per Namespace field SHALL be set to zero.

4.2.2 Namespace Level 0 Discovery

4.2.2.1 Overview

The Namespace Level 0 Discovery command provides a host with basic information about TPer capabilities both current and potential, for a specific Namespace.

4.2.2.2 IF-SEND Command

There is no IF-SEND command defined for Namespace Level 0 Discovery, so if IF-SEND is invoked with a Protocol ID of `0x01` and a ComID of `0x0002`, then the TPer SHALL:

1. transfer all of the data from the host;
2. discard the data; and
3. return 'good' status to the host.

4.2.2.3 IF-RECV Command

Namespace Level 0 Discovery is invoked by sending an IF-RECV command with:

Protocol ID = `0x01`
ComID = `0x0002`
Transfer Length = maximum length of the Namespace Level 0 Discovery response data that the host elects to receive.
NamespaceID = identifier for a namespace/LUN.

The Namespace Level 0 Discovery IF-RECV command MAY be processed at any time, without regard to sessions or prior authentication.

If the Transfer Length parameter is less than the size of the Namespace Level 0 Discovery response data that is available, then the TPer SHALL return the requested amount of data, even if it is truncated.

If the Transfer Length parameter is greater than the size of the Namespace Level 0 Discovery response data, then the device SHALL pad according to the rules specified in the transport.

If the NamespaceID parameter [3] specifies:

a) an allocated namespace/LUN identifier (i.e., a value that corresponds to an existing namespace/LUN), then the TPer SHALL return Namespace Level 0 Discovery Response Data containing feature descriptors corresponding to that namespace/LUN;

b) a value of 0xFFFF_FFFF, then the TPer SHALL return the Namespace Level 0 Discovery header and zero feature descriptors; and

c) any other value, then the TPer SHALL fail the command with a status of Other Invalid Command Parameter.

The Namespace Level 0 Discovery response data (see Table 8) consists of a header field and zero or more variable length feature descriptors. A TPer SHALL NOT include feature descriptors for namespace/LUN features that it does not implement. The data is not packetized.

Table 8 - Namespace Level 0 Discovery Response Data Format

Bit / Byte	7	6	5	4	3	2	1	0
0 – 47	Namespace Level 0 Discovery Header (see Table 9)							
48 – n	Feature Descriptor(s) (see 4.2.2.4)							

The Namespace Level 0 Discovery Header format is defined in Table 9.

Table 9 - Namespace Level 0 Discovery Header Format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				Length of Parameter Data				
3								(LSB)
4	(MSB)							
				Data Structure Revision				
7								(LSB)
8								
				Reserved				
47								

Length of Parameter Data

The Length of the Parameter Data field indicates the total number of bytes that are valid in the Namespace Level 0 Discovery header and all of the feature descriptors returned, not including this field. If no feature descriptors are returned, then this field SHALL be set to 0x0000_002C.

4.2.2.3.1 Data Structure Revision

The Data Structure Revision field describes the format of the Namespace Level 0 Discovery header returned. The value SHALL be 0x0000_0001.

4.2.2.4 Namespace Level 0 Discovery Feature Descriptors

The Namespace Feature Descriptors SHALL be returned in the Namespace Level 0 Discovery response data in order of increasing namespace/LUN feature code values. Namespace features that are not implemented SHALL NOT be returned.

Table 10 contains the feature code for Namespace Level 0 Discovery.

Table 10 - Namespace Level 0 Discovery Feature Codes

Feature Code	Feature Name	Description
0x0405	Namespace Geometry Reporting Feature	See 4.2.2.5

4.2.2.5 Namespace Geometry Reporting Feature (Feature Code = 0x0405)

The Namespace Geometry Reporting Feature (see Table 11) indicates the logical block and physical block geometry supported within the namespace/LUN indicated by the NamespaceID parameter of the IF-RECV command. The Namespace Geometry Reporting Feature Descriptor feature MAY be returned in the Namespace Level 0 Discovery response. See [2] for additional information.

Table 11 - Level 0 Discovery – Namespace Geometry Reporting Feature Descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1	Feature Code							(LSB)
2	Feature Descriptor Version Number				Reserved			
3	Length							
4	Reserved							Align
5	(MSB)							
…	Reserved							
11								(LSB)
12	(MSB)							
…	LogicalBlockSize							
15								(LSB)
16	(MSB)							
…	AlignmentGranularity							
23								(LSB)
24	(MSB)							
…	LowestAlignedLBA							
31								(LSB)

4.2.2.5.1 Feature Code

The Feature Code field SHALL be set to 0x0405.

4.2.2.5.2 Feature Descriptor Version Number

The Feature Descriptor Version Number field SHALL be set to 0x1.

4.2.2.5.3 Length

The Length field indicates the number of bytes in the descriptor following byte 3. The value SHALL be set to 0x1C.

4.2.2.5.4 Align

The Align field indicates whether the TPer requires ranges in the specified namespace/LUN to be aligned. If Align is TRUE then the TPer requires ranges in the specified Namespace to be aligned. If Align is FALSE, then the TPer does not require ranges to be aligned.

4.2.2.5.5 LogicalBlockSize

The LogicalBlockSize field indicates the number of bytes in a logical block for the specified Namespace.

4.2.2.5.6 AlignmentGranularity

The AlignmentGranularity field indicates the number of logical blocks in a group, for alignment purposes within the specified Namespace. (For details about access granularity see [4].)

4.2.2.5.7 LowestAlignedLBA

The LowestAlignedLBA field indicates the lowest logical block address that is located at the beginning of an alignment granularity group for the specified Namespace. (For details about access granularity see [4].)

4.3 Admin SP

This feature set modifies the behavior of the `Revert` method (see 3.1.2.2).

4.4 Locking SP

A Storage Device implementing this feature set SHALL support the additions to the Locking SP specified in this section, in addition to the Locking SP requirements defined in [4].

This feature set modifies the behavior of the `RevertSP` method (see 3.1.2.3).

4.4.1 Tables

4.4.1.1 Locking table (M)

4.4.1.1.1 Global Range (M)

The Global Range Locking object SHALL apply to all namespace/LUNs that are not associated with any other Locking object.

If the Storage Device supports the NVMe interface, the NamespaceID column value of the Global Range Locking object SHALL be set to `0x0000_0000` or `0xFFFF_FFFF`.

If the Storage Device supports the SCSI interface, the NamespaceID column value of the Global Range Locking object SHALL be set to `0xFFFF_FFFF`.

The NamespaceGlobalRange column value of the Global Range Locking object SHALL be set to TRUE.

4.4.1.1.2 Preconfiguration

In addition to the requirements in [4], the MethodID table preconfiguration SHALL be modified as specified in Table 12:

Table 12 - Locking SP – MethodID Table Preconfiguration

UID	Name	CommonName	TemplateID
00 00 00 06 00 00 08 04	“Assign”		
00 00 00 06 00 00 08 05	“Deassign”		

In addition to the requirements in [4], the `AccessControl` table SHALL be preconfigured as specified in Table 13:

Table 13 - Locking SP – AccessControl Table Preconfiguration

Table Association – informative only	UID	InvokingID	InvokingID Name – informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
SP																
		00 00 08 02 00 00 00 00	LockingTableUID	Assign		ACE_Assign				ACE_Anybody						
		00 00 08 02 00 00 00 00	LockingTableUID	Deassign		ACE_Deassign				ACE_Anybody						
ACE																
		00 00 00 08 00 03 F9 01	ACE_Assign	Get		ACE_ACE_Get_All				ACE_Anybody						
		00 00 00 08 00 03 F9 01	ACE_Assign	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association – informative only	UID	InvokingID	InvokingID Name – informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 F9 02	ACE_Deassign	Get		ACE_ACE_Get_All				ACE_Anybody						
		00 00 00 08 00 03 F9 02	ACE_Deassign	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

In addition to the requirements in [4], the ACE Table SHALL be preconfigured as specified in Table 14:

Table 14 - Locking SP – ACE Table Preconfiguration

Table Association – informative only	UID	Name	CommonName	BooleanExpr	Columns
ACE					
	00 00 00 08 00 03 F9 01	“ACE_Assign”		Admins	All
	00 00 00 08 00 03 F9 02	“ACE_Deassign”		Admins	All

In addition to the requirements in [4], the added columns in the Locking table SHALL be preconfigured as specified in Table 15.

*LT1 = indirectly writeable using the Assign method and the Deassign method.

Table 15 - Locking SP – Locking Table Preconfiguration

UID	Name	NamespaceID	NamespaceGlobalRange
00 00 08 02 00 00 00 01	“Locking_GlobalRange”	0x0000_0000	**T**
00 00 08 02 00 03 00 01	“Locking_Range1”	0x0000_0000 *LT1	**F** *LT1
00 00 08 02 00 03 NN NN	“Locking_RangeNNNN”	0x0000_0000 *LT1	**F** *LT1

4.5 Additional SPs

This feature set requires no additional SPs.

4.6 Single User Mode Feature Set Interactions

4.6.1 Overview

The Single User Mode Feature Set (see [6]) MAY be supported on a TPer that supports the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set.

This section describes the interactions when the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set and the Single User Mode Feature Set (see [6]) are both supported and present (enabled or disabled).

User authorities that are Single User Mode Locking object owners SHALL NOT be permitted to be added to ACE_Assign and ACE_Deassign.

4.6.2 Modified Methods

This section defines modifications to methods that are required to support the Single User Mode Feature Set (see [6]) in addition to the Configurable Locking for NVMe Namespaces and SCSI LUNs Feature Set.

4.6.2.1 Assign

Assign method interactions with Single User Mode vary based on the column value of the RangeStartLengthPolicy column in the LockingInfo table.

4.6.2.1.1 Assign interactions when RangeStartLengthPolicy is 0

Start of Informative Comment

The following rule prevents assignment of a Namespace Global Range Locking object if the Global Locking object is under exclusive control of a user authority.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the NamespaceID parameter specifies a value that is not equal to the NamespaceID column value of any Locking object;

c) the RangeStartLengthPolicy column value of the LockingInfo table is 0; and

d) the Global Range Locking object is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table,

then the `Assign` method SHALL fail with a status of NOT_AUTHORIZED.

Start of Informative Comment

The following rule prevents assignment of a Namespace Non-Global Range Locking object if the Namespace Global Locking object is under exclusive control of a user authority.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the `LockingInfo` table is 0; and

c) the NamespaceID parameter specifies a value that is equal to the NamespaceID column value in a Namespace Global Range Locking object that is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table,

then the `Assign` method SHALL fail with a status of NOT_AUTHORIZED.

4.6.2.1.2 Assign interactions when RangeStartLengthPolicy is 1

Start of Informative Comment

The following rule defines the assignment of a Namespace Global Range Locking object when the Global Range Locking object is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table and the AssignToSUMRange is not specified or is set to FALSE.

Compared to the normal Assign operation, assigning a namespace associated with the Global Range when the Global Range is activated in Single User Mode should result in erasure of data in the namespace as part of the Assign operation.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the NamespaceID parameter specifies a value that is not equal to the NamespaceID column value of any Locking object;

c) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1;

d) the UID of the Global Range Locking object is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table; and

e) the AssignToSUMRange parameter is not specified or is set to FALSE,

then the `Assign` method SHALL succeed as defined in section 3.1.1.1.3.1 with the following exceptions:

The SD shall:

1. eradicate the media encryption key associated with the namespace; and
2. transfer control of the media encryption key associated with the namespace to the K_AES_* object indicated by the ActiveKey column value of the selected Locking object.

Start of Informative Comment

The following rule defines the assignment of a Namespace Global Range Locking object that is activated in Single User Mode using AssignToSUMRange.

Compared to the normal Assign operation, assigning a namespace to a Namespace Global Range Locking object that is activated in Single User Mode should result in erasure of data in the namespace as part of the Assign operation.
End of Informative Comment

If:

a) the method would otherwise succeed;

b) the NamespaceID parameter specifies a value that is not equal to the NamespaceID column value of any Locking object;

c) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1; and

d) the AssignToSUMRange parameter is set to TRUE,

then the `Assign` method SHALL succeed as defined in section 3.1.1.1.3.1 with the following exceptions:

The SD shall:

1. Select a Non-Global Range Locking object that:
   a. has a NamespaceID column value of zero; and
   b. has a UID that is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table.
2. eradicate the media encryption key associated with the namespace; and
3. transfer control of the media encryption key associated with the namespace to the K_AES_* object indicated by the ActiveKey column value of the selected Locking object.

Start of Informative Comment

The following rule defines the assignment of a Namespace Non-Global Range Locking object when the Namespace Global Range Locking object associated with the namespace is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table and the AssignToSUMRange is not specified or is set to FALSE.

Assigning a namespace associated with a Namespace Global Range Locking object activated in Single User Mode should result in erasure of data in the LBA range specified in the `Assign` method.

[2] states:
“If the RangeStart/RangeLength column values are returned to their previous values, it is not guaranteed that the original data is preserved on the logical blocks whose control transitioned from one LBA range to another”.

This rule relies on this behavior to erase data when assigning Namespace Non-Global Range Locking object.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the NamespaceID parameter specifies a value that is not equal to the NamespaceID column value of any Locking object;

c) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1;

d) The Namespace Global Range Locking object associated with the namespace is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table; and

f) the AssignToSUMRange parameter is not specified or is set to FALSE,

then the `Assign` method SHALL succeed as defined in section 3.1.1.1.3.2.

Start of Informative Comment

The following rule defines the assignment of a Namespace Non-Global Range Locking object that is activated in Single User Mode using AssignToSUMRange.

Assigning a namespace to a Namespace Non-Global Range Locking object activated in Single User Mode should result in erasure of data in the LBA range specified in the `Assign` method.

[2] states:

"If the RangeStart/RangeLength column values are returned to their previous values, it is not guaranteed that the original data is preserved on the logical blocks whose control transitioned from one LBA range to another".

This rule relies on this behavior to erase data when assigning Namespace Non-Global Range Locking object that is activated in Single User Mode.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the NamespaceID parameter specifies a value that is not equal to the NamespaceID column value of any Locking object;

c) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1; and

d) the AssignToSUMRange paramater is set to TRUE,

then the `Assign` method SHALL succeed as defined in section 3.1.1.1.3.2 with the following exceptions:

The SD shall:

1. Select a Non-Global Range Locking object that:
   a. has a NamespaceID column value of zero; and
   b. has a UID that is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table.

Start of Informative Comment

The following rule defines the error condition when the `Assign` method is invoked with the AssignToSUMRange parameter set to TRUE and SUM is not supported.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the SD does not support the Single User Mode Feature Set (see [6]); and

c) the AssignToSUMRange parameter is set to TRUE,

then the Assign method SHALL fail with a status of INVALID_PARAMETER.

4.6.2.2 Deassign

Deassign method interactions with Single User Mode vary based on the column value of the RangeStartLengthPolicy column in the LockingInfo table.

4.6.2.2.1 Deassign interactions when RangeStartLengthPolicy is 0

Start of Informative Comment

The following rule prevents deassignment of a Locking object if that Locking object is under exclusive control of a user authority.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the LockingInfo table is 0; and

c) the value of the UID parameter is included in the single_user_ranges list in the SingleUserModeRanges column of the LockingInfo table,

then the Deassign method SHALL fail with a status of NOT_AUTHORIZED.

Start of Informative Comment

The following rule prevents deassignment of a Namespace Non-Global Range Locking object if the corresponding Namespace Global Range Locking object is under exclusive control of a user authority.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the LockingInfo table is 0;

c) the Locking object indicated by the UID parameter is a Namespace Non-Global Range Locking object; and

d) the NamespaceID column value of this Namespace Non-Global Range Locking object is equal to the NamespaceID column value in a Namespace Global Range Locking object that is included in the single_user_ranges list in the SingleUserModeRanges column of the LockingInfo table,

then the Deassign method SHALL fail with a status of NOT_AUTHORIZED.

Start of Informative Comment

The following rule prevents deassignment of a Namespace Global Range Locking object if the Global Range Locking object is under exclusive control of a user authority.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the LockingInfo table is 0;

c) the UID of the Global Range Locking object is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table; and

d) the Locking object indicated by the UID parameter is a Namespace Global Range Locking object,

then the `Deassign` method SHALL fail with a status of NOT_AUTHORIZED.

4.6.2.2.2 Deassign interactions when RangeStartLengthPolicy is 1

Start of Informative Comment

The following rule defines deassignment of a Namespace Global Range Locking object that is activated in Single User Mode and KeepNamespaceGlobalRangeKey is not specified or set to FALSE.

Deassigning a Namespace Global Range Locking object with KeepNamespaceGlobalRangeKey not specified or set to FALSE results in erasure of the data associated with the Namespace Global Locking object.

[2] states:
“If the RangeStart/RangeLength column values are returned to their previous values, it is not guaranteed that the original data is preserved on the logical blocks whose control transitioned from one LBA range to another”.

This rule relies on this behavior to erase data when Deassigning a Namespace Global Range Locking object that is activated in Single User Mode.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1;

c) the Locking object indicated by the UID parameter is a Namespace Global Range Locking object;

d) the value of the UID parameter of the `Deassign` method is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table; and

e) the KeepNamespaceGlobalRangeKey parameter is:

   a. Not specified; or

   b. Set to FALSE,

then the `Deassign` method SHALL succeed as defined in section 3.1.1.2.2.

Start of Informative Comment

The following rule defines deassignment of a Namespace Non-Global Range Locking object that is activated in Single User Mode.

Deassigning a Namespace Non-Global Range Locking object is similar to de-personalizing a non-global locking range which results in erasure of the data associated with the non-global locking range.

[2] states:
“If the RangeStart/RangeLength column values are returned to their previous values, it is not guaranteed that the original data is preserved on the logical blocks whose control transitioned from one LBA range to another”.

This rule relies on this behavior to erase data when Deassigning a Namespace Non-Global Range Locking object that is activated in Single User Mode.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1;

c) the Locking object indicated by the UID parameter is a Namespace Non-Global Range Locking object; and

d) the value of the UID parameter of the `Deassign` method is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table,

then the `Deassign` method SHALL succeed as defined in section 3.1.1.2.2.

Start of Informative Comment

The following rule prevents deassignment of a Namespace Global Range Locking object that is activated in Single User Mode when KeepNamespaceGlobalRangeKey is TRUE.

The objective is to prevent other authorities from being able to transition ownership of the namespace associated with the Single User Mode back to the global range without erasing data.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1;

c) the Locking object indicated by the UID parameter is a Namespace Global Range Locking object;

d) the value of the UID parameter of the `Deassign` method is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table; and

e) the KeepNamespaceGlobalRangeKey parameter is TRUE,

then the `Deassign` method SHALL fail with a status of NOT_AUTHORIZED.

Start of Informative Comment

The following rule prevents deassignment of a Namespace Global Range Locking object when the Global Range Locking object is activated in Single User Mode and KeepNamespaceGlobalRangeKey is TRUE.

The objective is to prevent other authorities from being able to transition ownership of the namespace to the global range activated in Single User Mode without erasing data.

End of Informative Comment

If:

a) the method would otherwise succeed;

b) the RangeStartLengthPolicy column value of the `LockingInfo` table is 1;

c) the Locking object indicated by the UID parameter is a Namespace Global Range Locking object;

d) the UID of the Global Range Locking object is included in the single_user_ranges list in the SingleUserModeRanges column of the `LockingInfo` table; and

e) the KeepNamespaceGlobalRangeKey parameter is TRUE,

then the `Deassign` method SHALL fail with a status of NOT_AUTHORIZED.

4.6.2.3 Reactivate Method

Upon successful invocation of the `Reactivate` method:

a) The NamespaceID and NamespaceGlobalRange column values SHALL remain at their current values.

b) User authorities that are owners of Single User Mode Locking objects SHALL be removed from the BooleanExpr column values of the ACE_Assign and the ACE_Deassign ACEs.

4.6.2.4 Set Method

Start of Informative Comment

The following rule prevents the addition of Single User Mode Locking object owners to the ACE_Assign or ACE_Deassign ACEs

End of Informative Comment

If:

a) the `Set` method is invoked on the ACE_Assign or the ACE_Deassign ACEs;

b) the BooleanExpr column is included in the `Set` method parameters; and

c) the value of the BooleanExpr column parameter includes one or more User authorities that are Single User Mode Locking object owners,

then the TPer SHALL fail the `Set` method invocation with status INVALID_PARAMETER.