

This document is intended to help the reader understand changes in the NVM Express™ Base Specification revision 1.4 .

One category of changes is new features that are optional capabilities. These features may be implemented by a revision 1.4 compatible device based on market need.

Another category is changes from past behavior. One set of changes here includes extensions of past functionality. Another set of changes is incompatible changes from past behavior, or new mandatory requirements for how a feature shall be implemented (where there may have been multiple reasonable interpretations of previous specification language).

The changes included in NVMe revision 1.4 were developed in Technical Proposals and ECNs. The documents are available at:

- <https://nvmexpress.org/wp-content/uploads/NVM-Express-1.3-Ratified-TPs.zip>
- <https://nvmexpress.org/wp-content/uploads/NVM-Express-1.3-Ratified-ECN.zip>

## New Features

This section describes new features in NVMe revision 1.4.

- Persistent Memory Region (optional)
  - Defines a Persistent Memory Region (PMR), which is a PCIe® memory region whose contents persist across power cycles, resets, and disabling of the PMR region.
  - References:
    - NVMe revision 1.4 section 3.1, 4.8, 5.21 & 8.15
    - Technical Proposal 4000a, 4032
- IO Determinism (optional)
  - Defines a capability, Predictable Latency Mode, that enables a well-behaved host to achieve a deterministic read latency.
  - References:
    - NVMe revision 1.4 sections 5.2, 5.14, 5.15, 5.21 & 8.18
    - Technical Proposal 4003c
- Asymmetric Namespace Access (optional)
  - Defines capabilities that enable reporting of Asymmetric Namespace Access characteristics to the host.
  - References:
    - NVMe revision 1.4 section 4.6, 5.14, 5.15, 5.19, 5.20, 5.21, 6.1, 8.12, 8.20 & 8.21
    - Technical Proposal 4004b, 4028a

- Namespace Write Protect (optional)
  - Defines enhancement that controls write protection on a per namespace basis. This feature may be used to prevent modification of the specified namespace.
  - References:
    - NVMe revision 1.4 section 5.14, 5.15, 5.21, 8.10 & 8.19
    - Technical Proposal 4005c
  
- Persistent Event Log (optional)
  - Defines a mechanism for a persistent event log in NVMe subsystems. The log is intended to persistently capture significant events for use by software/system vendors that are not the NVMe subsystem manufacturer.
  - Persistent Event Log Events Defined:
    - SMART/Health Log Snapshot
    - Firmware Commit
    - Timestamp Change
    - Power-on or Reset
    - NVM Subsystem Hardware Error
    - Change Namespace
    - Format NVM Start & Completion
    - Sanitize Start & Completion
    - Set Feature
    - Telemetry Log Created
    - Thermal Excursion
    - Vendor Specific
    - TCG Reserved Event
  - References:
    - NVMe revision 1.4 section 5.14, 5.15
    - Technical Proposal 4007a, 4042a
  
- Rebuild Assist (optional)
  - Defines a new Get LBA Status capability that identifies the potentially unrecoverable LBAs to the host. This status may be used to determine what LBAs on a device should be recovered by the host from another location and re-written.
  - References:
    - NVMe revision 1.4 section 5.14, 5.15, 5.21, 5.27, 8.22
    - Technical Proposal 4016
  
- NVM Sets (optional)
  - Defines NVM Sets as a collection of NVM that is separate (logically and potentially physically) from NVM in other NVM Sets.
  - References:
    - NVMe revision 1.4 section 4.9, 5.14, 5.15, 5.20, 5.21 & 8.12
    - Technical Proposal 4018b

- Read Recovery Levels (optional)
  - Defines Read Recovery Levels as a configurable attribute that balances the completion time for read commands and the amount of error recovery applied to those read commands.
  - References:
    - NVMe revision 1.4 section 5.15, 5.21 & 8.16
    - Technical Proposal 4018b
- Endurance Groups (optional)
  - Defines a mechanism that enables endurance management to occur within a single NVM Set or across a collection of NVM Sets.
  - References:
    - NVMe revision 1.4 section 5.14, 5.15, 5.21 & 8.17
    - Technical Proposal 4018b, 4050
- Traffic Based Keep Alive (optional)
  - Defines a mode where a host and controller reset the Keep Alive Timer in the presence of Admin commands or I/O commands. Admin or I/O command processing indicates proper health of a host, controller and transport connectivity.
  - References:
    - NVMe revision 1.4 section 5.15, 5.16 & 7.12
    - Technical Proposal 4024
- IO Performance and Endurance Hints (optional)
  - Defines performance and endurance hints enabling the controller to indicate granularities and alignments that are preferred for Write and Deallocate operations.
  - References:
    - NVMe revision 1.4 section 5.15 & 8.25
    - Technical Proposal 4025
- UUIDs for Vendor Specific Information (optional)
  - Defines a mechanism for commands to select between vendor specific definitions of the information sent or received by that command. The organization which defines the vendor specific definition is identified by a specific UUID.
  - References:
    - NVMe revision 1.4 section 5.11, 5.13, 5.14, 5.15, 5.21 & 8.24
    - Technical Proposal 4027
- Verify Command (optional)
  - Adds a Verify command to check the integrity of stored data and metadata. The Verify command is roughly equivalent to a Read command that discards the data and metadata after reading. The important behavior

of Verify is that errors are reported if the data or metadata cannot be read without expending the time and resources required to return the data and metadata to the host.

- References:
  - NVMe revision 1.4 section 4.6, 5.14, 5.15, 5.23, 5.27, 6.7, 6.14 & 8.19
  - Technical Proposal 4030
- Administrative Controller (optional)
  - Defines an Administrative Controller as a controller that supports functionality intended to provide host management capabilities, such as the NVMe-MI in-band mechanism. The primary benefit of an administrative controller in PCIe is that it has a unique Programming Interface allowing a special NVMe driver to be loaded.
  - References:
    - NVMe revision 1.4 section 1.2, 1.6, 2.1, 3.1, 5.15, 5.19, & 7.1
    - Technical Proposal 4039a
- Submission Queue Association (optional)
  - Defines a method for a host to associate a given NVM Set and a specific I/O Submission Queue.
  - References:
    - NVMe revision 1.4 section 5.4, 5.15 & 8.23
    - Technical Proposal 4045

## Feature Enhancements

This section describes optional feature enhancements which are changes from past NVMe behavior.

- Host Memory Buffer Enhancements
  - Allow a controller to indicate limitations to the host in order to help the host in allocation of host memory as some controllers are unable to support complex host memory layouts.
    - Maximum number of descriptor entries
    - Minimum size of host memory allocation for a specific descriptor
  - References:
    - NVMe revision 1.4 section 5.15 & 8.9
    - Technical Proposal 4002
- Namespace Granularity
  - Defines a mechanism for the controller to indicate to the host the allocation granularities for namespaces.
  - References:
    - NVMe revision 1.4 section 5.15 & 8.12

- Technical Proposal 4006
- Transport SGL
  - Defines a new SGL Descriptor type to indicate that the NVMe Transport handles data transfer using Transport specific buffering.
  - References:
    - NVMe revision 1.4 section 4.2, 4.4 & 5.15
    - Technical Proposal 4008
- Sanitize Enhancements
  - Defines a mechanism to support “No Deallocate After Sanitize.” This enhancement indicates that deallocation is not allowed after a Sanitize command so that ‘raw’ contents may be audited. After a Crypto Erase or Block Erase operation, ECC and Protection bytes are typically not correct. Many data paths have certified (e.g. FIPS 140) behavior that the hardware does not allow incorrect data to be retrieved. This feature is not compatible with that functionality.
  - **Additional requirement / incompatible change** in section 5.15.2.2. The settings of the No Deallocate Inhibited (NDI) bit, the No Deallocate Response Mode (NODRM) field, and the No Deallocate After Sanitize bit in a Sanitize command may result in
    - the sanitize operation completing with deallocation (with No Deallocate set to ‘1’); or
    - the Sanitize command being aborted with the status “Invalid Field in Command”.
  - References:
    - NVMe revision 1.4 section 5.2, 5.14, 5.15, 5.21, 5.24, 8.15 & Annex A
    - Technical Proposal 4014
- Shared Stream Write
  - Defines a mechanism to enable write streams to be shared across multiple hosts.
  - References:
    - NVMe revision 1.4 section 9.3
    - Technical Proposal 4031a
- Enhanced Command Retry
  - Defines enhancements to the command retry capability:
    - Command Retry Delays with support for 3 different delay values or no delay
    - Error codes to indicate a command should be retried without providing further information
    - Indications to the host of support for the enhanced capabilities
  - References:

- NVMe revision 1.4 section 4.6, 5.15 & 5.21
  - Technical Proposal 4033
- Controller Memory Buffer Extensions
  - Defines a mechanism that relaxes the restrictions on the host usage of the Controller Memory Buffer.
  - References:
    - NVMe revision 1.4 section 3.1 & 4.7
    - Technical Proposal 4051
- Multiple applicable Error Status condition (optional)
  - Enhanced the condition when multiple error statuses are applicable to allow the device to select which of the applicable errors to return.
  - New requirement / incompatible change in section 4.6.1.2:
    - “Unless otherwise specified, if multiple status codes apply, then the controller selects the status code that is returned.”
  - References:
    - NVMe revision 1.4 section 4.6
    - ECN002

## Required Changes

This section describes mandatory behavior changes required to comply with NVM Express Base Specification 1.4.

- Sanitize during Device Self-Test (**mandatory**)
  - Adds requirement that all controllers in the NVM Subsystem shall abort device self-test operations in progress when a sanitize operation starts on any controller.
  - New requirement / incompatible change in section 5.24:
    - “When a sanitize operation starts on any controller, all controllers in the NVM subsystem shall abort device self-test operations in progress.”
  - References:
    - NVMe revision 1.4 section 5.24 & 8.11
    - Technical Proposal 4022
- Clarifications regarding various usages of NSID value FFFFFFFFh (**mandatory**)
  - NVM I/O Command clarifications:
    - Defines the behavior of a Flush command when the NSID is set to FFFFFFFFh.
    - Defines all NVM I/O Commands, except optionally Flush, as not supporting the NSID set to FFFFFFFFh.
  - Feature Value clarifications:

- The previous Feature Value text (section 7.8) contained conflicting requirements for the value of the NSID field for features that applied to the controller. Clarified that the NSID set to 0h or set to FFFFFFFFh are both permitted for controller feature values.
  - The result of some Get Feature or Set Feature commands with an NSID value of 0h or FFFFFFFFh for a namespace specific feature were not specified; the missing cases are now specified as resulting in an error.
- Admin command clarifications:
  - The command description implied that the NSID could be set to FFFFFFFFh for any Admin command. The specification now clarifies the list of Admin commands for which an NSID of FFFFFFFFh applies.
- Reservation command clarifications:
  - Reservation commands with the NSID set to FFFFFFFFh require reservation permissions on all namespaces impacted by that command. If a reservation is present on any of the impacted namespaces that does not grant permission, then the command fails with Reservation Conflict status. This was previously undefined.
- New requirement / incompatible change in section 4.2, 5 & 6:
  - Namespace Identifier (NSID): “The value FFFFFFFFh in this field is a broadcast value (refer to section 6.1), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 139, Figure 140, and Figure 346 for commands that support the use of the value FFFFFFFFh in this field.”
  - Figures 139 & 140 are updated with a column that specifies the supported NSID values for all Admin Commands.
  - Figure 346 adds Notes specifying NVM Command Set command support for the NSID field value of FFFFFFFFh.
- New requirement / incompatible change in sections 5.15 & 6.8:
  - Volatile Write Cache (VWC) – “Support for the NSID field set to FFFFFFFFh is not indicated. Only controllers compliant with versions 1.3 and earlier of the specification shall be allowed to return this value.”
- New requirement / incompatible change in sections 5.21.1.3:
  - “A Set Features command with the Feature Identifier set to 03h and the NSID field set to FFFFFFFFh shall be aborted with a status of Invalid Field in Command.”
- New requirement / incompatible change in section 7.8:
  - “For feature values that apply to the controller:

1. a) if the NSID field is cleared to 0h or set to FFFFFFFFh, then:

- the Set Features command shall set the specified feature value for the controller; and
- the Get Features command shall return the current setting of the requested feature value for the controller;

and

6. b) if the NSID field is set to a valid namespace identifier (refer to section 6.1), then:
  - the Set Features command shall fail with a status code of Feature Not Namespace Specific; and
  - the Get Features command shall return the current setting of the requested feature value for the controller.”
- **New requirement / incompatible change** in section 8.8:
  - “If a host submits a command with the NSID set to FFFFFFFFh in the presence of a reservation on any of the namespaces impacted by that command and that host lacks sufficient rights on all the impacted namespaces, then the command is aborted by the controller with a status of Reservation Conflict.”
- References:
  - NVMe revision 1.4 section 4.2, 5, 5.15, 6, 6.8, 7.8 & 8.8
  - Technical Proposal 4035, ECN002, ECN004, ECN006
- **Controller Memory Buffer and Persistent Memory Region Enhancements (if CMB is supported – mandatory)**
  - Fixes the potential for DMA misrouting with Controller Memory Buffer (CMB) and Persistent Memory Region (PMR).
  - New requirement / incompatible change in section 3.1.1:
    - “Controller Memory Buffer Supported (CMBS): If set to ‘1’, this bit indicates that the controller supports the Controller Memory Buffer, and that addresses supplied by the host are permitted to reference the Controller Memory Buffer only if the host has enabled the Controller Memory Buffer’s controller memory space. If the controller supports the Controller Memory Buffer, this bit shall be set to ‘1’.”
  - **Additional requirements / incompatible changes** if CMB is supported in sections 3.1.11, 3.1.12, 3.1.16 & 3.1.17:
    - CMBLOC – Controller Memory Buffer Location – “If the controller supports the Controller Memory Buffer and CMBMSC.CRE is cleared to ‘0’, this register shall be cleared to 0h.”
    - CMBSZ – Controller Memory Buffer Size – “If the controller does not support the Controller Memory Buffer feature or if the controller supports the Controller Memory Buffer (CAP.CMBS) and



CMBMSC.CRE is cleared to '0', then this register shall be cleared to 0h."

- CMBMSC – Controller Memory Buffer Memory Space Control – “This register specifies how the controller references the Controller Memory Buffer with host-supplied addresses. If the controller supports the Controller Memory Buffer (CAP.CMBS), this register is mandatory. Otherwise, this register is reserved.”
- CMBSTS – Controller Memory Buffer Status – “This register indicates the status of the Controller Memory Buffer. If the controller supports the Controller Memory Buffer (CAP.CMBS), this register is mandatory. Otherwise, this register is reserved.”
- References:
  - NVMe revision 1.4 section 3.1, 4.7, 4.8 & 7.3
  - Technical Proposal 4054
- Reservation Notification Log maximum pages reached (**if Reservation Notification Log is supported – mandatory**)
  - The behavior of the Reservation Notification Log has changed for the condition when the maximum number of log pages has been reached. Additionally, the definition of the zero Log Page Count has been modified, where the 0h value now means that the log page is empty.
  - New requirement / incompatible change in sections 5.14.1.16.1:
    - “Log Page Count: This is a 64-bit incrementing Reservation Notification log page count, indicating a unique identifier (modulo 64 bit) for this notification. The count starts at 0h following a Controller Level Reset and is incremented for every event that causes a reservation notification regardless of whether that notification is added to the queue. If the value of this field is FFFFFFFF\_FFFFFFFFh, then the field is set to 1h when incremented (i.e. rolls over to 1h) and a new log page is created.

If there are no Reservation Notification log pages to return (i.e., the queue of reservation log pages is empty), then this field shall return the value 0h. Subsequent reservation notifications continue incrementing this unique identifier from the last non-zero value (i.e., the value that identified the previous Reservation Notification log page). A value of 0h indicates the log page is empty.”

- References:
  - NVMe revision 1.4 section 5.14
  - ECN001
- Controller ID for the Registered Controller data structure (**if a dynamic controller – mandatory**)

- There is a new requirement that the Controller ID in the Registered Controller data structure in Figure 392 must be FFFFh if it is a dynamic controller that is not associated with a host.
- **New requirement / incompatible change** in section 6.13:
  - Controller ID: “If the controller is a dynamic controller (refer to the NVMe over Fabrics specification) that is not associated with a host, then the Controller ID field shall be set to FFFFh.”
- References:
  - NVMe revision 1.4 section 6.13
  - ECN001
- **SMART Critical Composite Temperature Time definition change (mandatory)**
  - The SMART Critical Composite Temperature Time was modified to cover greater than or equal to the threshold.
  - **New requirement / incompatible change** in section 5.14.1.2:
    - Critical Composite Temperature Time: “Contains the amount of time in minutes that the controller is operational and the Composite Temperature is greater than or equal to the Critical Composite Temperature Threshold (CCTEMP ) field in the Identify Controller data structure in Figure 247”
  - References:
    - NVMe revision 1.4 section 5.14
    - ECN002
- **Reservation Report command conflicts (mandatory)**
  - Reservation Report commands that conflict with the existing Host ID state now report a specific error.
  - **New requirement / incompatible change** in section 6.13:
    - “If a 64-bit Host Identifier has been specified and the Extended Data Structure bit is set to ‘1’ in Command Dword 11, then the controller shall abort the command with the status code of Host Identifier Inconsistent Format. If a 128-bit Host Identifier has been specified and the Extended Data Structure bit is cleared to ‘0’ in Command Dword 11, then the controller shall abort the command with the status code of Host Identifier Inconsistent Format.”
  - References:
    - NVMe revision 1.4 section 6.13
    - ECN003
- **Abort command completion (mandatory)**
  - An explicit requirement was added to ensure completion of an Abort command with specific completion queue entry (CQE) contents in the case that the Abort command did not successfully abort the specified command.
  - **New requirement / incompatible change** in section 5.1.1:

- “Upon completion of the Abort command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the Abort command and indicating whether the command to abort was aborted. Dword 0 of the completion queue entry indicates whether the command to abort was aborted.

If the command to abort was successfully aborted, then a completion queue entry for the aborted command shall be posted to the appropriate Admin or I/O Completion Queue with a status of Command Abort Requested before the completion queue entry for the Abort command is posted to the Admin Completion Queue, and bit 0 of Dword 0 shall be cleared to ‘0’ in the completion queue entry for the Abort command. If the command to abort was not aborted for any reason, then bit 0 of Dword 0 shall be set to ‘1’ in the completion queue entry for the Abort command.”

- References:
  - NVMe revision 1.4 section 5.1
  - ECN003
- LBA Range feature results with multiple Set Feature commands processed (**mandatory**)
  - The LBA Range feature was updated to specify that subsequent Set Features commands replace the settings of any earlier command.
  - **New requirement / incompatible change** in section 5.21.1.3:
    - “This feature indicates the type and attributes of LBA ranges that are part of the specified namespace. If multiple Set Features commands for this feature are processed, then only information from the most recent successful command is retained (i.e., subsequent commands replace information provided by previous commands).”
  - References:
    - NVMe revision 1.4 section 5.21
    - ECN004, ECN006
- Number of Queues feature undefined error (**mandatory**)
  - The Number of Queues feature required it be issued before queue creation, but did not specify the error if the host violated this requirement. The error is now specified.
  - **New requirement / incompatible change** in section 5.21.1.7:
    - “If a Set Features command is issued for this feature after creation of any I/O Submission and/or I/O Completion Queues, then the Set Features command shall fail with status code of Command Sequence Error.”
  - References:
    - NVMe revision 1.4 section 5.21
    - ECN004

- Host Memory Buffer error case clarifications (**mandatory**)
  - The Host Memory Buffer (HMB) feature was enhanced to specify the error associated with attempting to enable the HMB when already enabled. The enhancement includes specifying that disabling HMB that is already disabled is not an error condition.
  - **New requirement / incompatible change** in section 5.21.1.13:
    - “If the host memory buffer is enabled, then a Set Features command to enable the host memory buffer (i.e., the EHM bit (refer to Figure 291) set to ‘1’) shall fail with a status code of Command Sequence Error.”
    - “If the host memory buffer is not enabled, then a Set Features command to disable the host memory buffer (i.e., the EHM bit (refer to Figure 291) cleared to ‘0’) shall succeed without taking any action.”
  - References:
    - NVMe revision 1.4 section 5.21
    - ECN004
  
- Identify command for Allocated Namespace ID list – Return value ordering clarification (**mandatory**)
  - The Identify command with a CNS value of 10h now explicitly states the ordering requirements of the returned NSID values.
  - **New requirement / incompatible change** in section 5.15.2.6:
    - “A list of up to 1,024 namespace IDs is returned to the host containing allocated NSIDs in increasing order that are greater than the value specified in the Namespace Identifier (NSID) field of the Identify command.”
  - References:
    - NVMe revision 1.4 section 5.15
    - ECN005
  
- Sanitize Capabilities value when Sanitize command is not supported (**mandatory**)
  - The Sanitize Capabilities (SANICAP) field is now required to be defined and cleared to 0h when the Sanitize command is not supported.
  - **New requirement / incompatible change** in section 5.15.2.2:
    - Sanitize Capabilities (SANICAP): “This field indicates attributes for sanitize operations. If the Sanitize command is supported, then this field shall be non-zero. If the Sanitize command is not supported, then this field shall be cleared to 0h. Refer to section 8.15.”
  - References:
    - NVMe revision 1.4 section 5.15
    - ECN005
  
- Command Aborted due to Missing Fused Command (**if fused commands are supported – mandatory**)

- Clarifies definition for when the Command Aborted due to Missing Fused Command status must be returned.
- **New requirement / incompatible change** in sections 4.6:
  - Command Aborted due to Missing Fused Command – “The fused command was aborted due to the adjacent submission queue entry not containing a fused command that is the other command in a supported fused operation (refer to section 6.2)”
- References:
  - NVMe revision 1.4 section 4.6
  - ECN006
- **Unsupported Fused Operation error clarification (mandatory)**
  - Specified the error status code for commands that request an unsupported fused operation.
  - **New requirement / incompatible change** in section 4.12:
    - “If the FUSE field is set to a non-zero value and the controller does not support the requested fused operation, then the controller should abort the command with a status of Invalid Field in Command.”
  - References:
    - NVMe revision 1.4 section 4.12
    - ECN006
- **Clarify inconsistent temperature thresholds (mandatory)**
  - For consistency throughout the specification, all upper bound temperature thresholds are specified as “greater than or equal to.”
  - For consistency throughout the specification, all lower bound temperature thresholds are specified as “less than or equal to.”
  - **New requirement / incompatible change** in section 5.2.1 & 5.14.1.2:
    - Asynchronous Event Information – SMART / Health Status – Temperature Threshold:

“A temperature is greater than or equal to an over temperature threshold or less than or equal to an under temperature threshold (refer to section 5.21.1.4).”

- Get Log Page – SMART / Health Status – Critical Warning Bit 1:

“If set to ‘1’, then a temperature is:

1. greater than or equal to an over temperature threshold; or
2. less than or equal to an under temperature threshold,

(refer to section 5.21.1.4).”

- References:

- NVMe revision 1.4 section 5.2 & 5.14
  - ECN006
- Get Log Page with invalid LPOL and LPOU fields (**mandatory**)
  - Specified the error requirements for handling a Get Log Page command with invalid Log Page Offset Lower (LPOL) and Log Page Offset Upper (LPOU) fields.
  - **New requirement / incompatible change** in section 5.14:
    - “If the host specifies an offset (i.e., LPOL and LPOU) that is greater than the size of the log page requested (e.g., a log page containing 100 bytes is requested starting at offset 200), then the controller shall abort the command with a status of Invalid Field in Command.”
  - References:
    - NVMe revision 1.4 section 5.14
    - ECN006
- Idle Power and Sanitize interaction clarification (**mandatory**)
  - Enhanced the Idle Power (IDLP) field description to specify that the device is not considered to be idle if a sanitize operation is in process.
  - **New requirement / incompatible change** in section 5.15.2.2:
    - Idle Power (IDLP): “This field indicates the typical power consumed by the NVM subsystem over 30 seconds in this power state when idle (i.e., there are no pending commands, register accesses, background processes, sanitize operation, nor device self-test operations).”
  - References:
    - NVMe revision 1.4 section 5.15
    - ECN006
- Timestamp Enhancements (**mandatory**)
  - Descriptions of the operations of the Sync bit and Origin field in the Timestamp data structure were enhanced to include additional cases.
  - **New requirement / incompatible change** in section 5.21.1.14:
    - “The Timestamp feature enables the host to set a timestamp value in the controller. A controller indicates support for the Timestamp feature through the Optional NVM Command Support (ONCS) field in the Identify Controller data structure. The Timestamp value (refer to Figure 300) in a Set Features command sets a timestamp value in the controller. After the current value for this feature is set, the controller updates that value as time passes. A Get Features command that requests the current value reports the timestamp value in the controller at the time the Get Features command is processed (e.g., the value set with a Set Features command for the current value plus the elapsed time since being set).”

Note: If the Timestamp feature supports a saveable value and the host sets a saveable value, then the timestamp value restored after a subsequent power on or reset event is the value that was saved (refer to section 7.8). As a result, the timestamp may appear to move backwards in time.

The accuracy of Timestamp values after initialization may be affected by vendor specific factors, such as whether the controller continuously counts after the timestamp is initialized, or whether the controller stops counting during certain intervals (e.g., non-operational power states). If the controller stops counting during such intervals, then the Synch bit in the Timestamp – Data Structure for Get Features (refer to Figure 301) shall be set to ‘1’.

If the controller maintains (i.e., continues to update) the timestamp value across any type of Controller Level Reset (e.g., across a Controller Reset), then the controller shall also preserve the Timestamp Origin field (refer to Figure 301) across that type of Controller Level Reset.”

- References:
  - NVMe revision 1.4 section 5.21
  - ECN006
- Error definition for Format command overlap with other commands (**mandatory**)
  - Specified the status code to be returned when interactions between overlapping Format commands and other commands allowed for the Format command to fail and the other overlapping commands to fail under certain conditions.
  - **New requirement / incompatible change** in section 5.23:
    - “The Format NVM command may fail if there are outstanding I/O commands to the namespace specified to be formatted. I/O commands for a namespace that has a Format NVM command in progress may be aborted and if aborted, the controller should return a status code of Format in Progress.”
  - References:
    - NVMe revision 1.4 section 5.23
    - ECN006
- Security Receive and Security Send specified undefined failure status code (**mandatory**)
  - The Security Receive command and Security Send command specified a failure status code that does not exist (Invalid Parameter). The correct error code was specified.
  - **New requirement / incompatible change** in section 5.25 & 5.26:
    - Security Protocol (SECP): “This field specifies the security protocol as defined in SFSC. The controller shall abort the command with

- status of Invalid Field in Command if an unsupported value of the Security Protocol is specified.”
  - References:
    - NVMe revision 1.4 section 5.25 & 5.26
    - ECN006
- Reservation Register command interaction clarifications (**mandatory**)
  - Specified the Reservation Register command did not explain interactions between the CPTPL field and the saveable Reservation Persistence Feature.
  - **New requirement / incompatible change** in section 6.11:
    - Change Persist Through Power Loss State (CPTPL): “This field allows the Persist Through Power Loss (PTPL) state associated with the namespace to be modified as a side effect of processing this command. If a saveable value is supported for the Reservation Persistence Feature (refer to section 5.21.1.28), then any change to the PTPL state as a result of processing this command shall be applied to both the current value and the saveable value of that feature.”
  - References:
    - NVMe revision 1.4 section 6.11
    - ECN006
- Define Preempting error when the Current Reservation Key doesn’t match (**mandatory**)
  - Preempting did not specify the error if the Current Reservation Key (CRKEY) did not match. The error is now specified.
  - **New requirement / incompatible change** in section 8.8.7:
    - “The CRKEY value shall match that used by the registrant to register with the namespace. If the CRKEY value does not match, then the command is aborted with status Reservation Conflict.”
  - References:
    - NVMe revision 1.4 section 8.8
    - ECN006
- Clarified Device Self-Test termination on Format scenario (**mandatory**)
  - Clarified the definition of “the same” when terminating a Device Self-Test (DST) operation as a result of processing a Format command with “the same” NSID value. The clarification includes several specific cases where the DST operation is terminated in this situation.
  - **New requirement / incompatible change** in section 8.11:
    - A short device self-test operation “shall be aborted by a Format NVM command as described in Figure 475”
    - Added Figure 475: Format NVM command Aborting a Device Self-Test Operation to clarify example scenarios.



- References:
  - NVMe revision 1.4 section 8.11
  - ECN006
- Release of Stream Resources clarification (**mandatory**)
  - Releasing of Stream Resources when a namespace is deleted was not described completely. Stream Resources and Stream Identifiers are both released when a namespace is deleted.
  - **New requirement / incompatible change** in section 9.3:
    - “If the host deletes a namespace, then all stream resources and all stream identifiers for that namespace shall be released.”
  - References:
    - NVMe revision 1.4 section 9.3
    - ECN006