LEGAL NOTICE:

This NVM Express revision 1.3 technical proposal is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this NVM Express revision 1.3 technical proposal subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **"© 2007 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

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

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**TP 4014 SANITIZE Enhancements**
Change Date	**November 26, 2018**
Builds on Specification	**NVM Express 1.3b**

Technical Proposal Author(s)

Name	Company
John Geldman	Toshiba
Paul Suhler	Micron

The SANITIZE feature in NVMe has brought in a new feature into the ecosystem, the **No Deallocate After Sanitize** bit was intended to provide the ability to verify that a SANITIZE has completed successfully and completely. This feature is new to the industry and some aspects have been discovered after ratification. This is currently a bit in a command dword, without a support bit.

This bit is defined to not allow deallocation after sanitize, so that the 'raw' contents could 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 (an implementation feature) that the hardware does not allow incorrect data to be retrieved. This feature is incompatible with this function.

It has been suggested that such devices perform an overwrite to enable auditing, however such an operation destroys the forensic results and provides no assurance that the sanitize operation did anything (when not on the audit path). Note that reading deallocated LBA's does provide assurance that some state in the device has changed.

Some SSD customers require audit capabilities while some do not. Most seem to require some mode of Sanitize. This proposal is intended to enable devices to identify which features they support and to define the behavior if the feature is invoked when not supported.

Revision History

6/20/2017	Initial document creation
7/13/2017	TPAR number added
7/28/2017	Changed problem statement and proposed solution to closely match how SCSI handles this issue.
8/3/2017	More changes after email exchange with David Black, Curtis Ballard, Jim Hatfield, John Geldman
8/3/2017	Additional changes after NVMe meeting
8/3/2017	Major changes after meeting to support simplified TPAR with no errors
8/31/17	Major changes to re-explain problem statement and move implementation thoughts to after TPAR acceptance
5/5/18	Complete relatively coherent rewrite (proposal 2.0)
5/24/18	Changes from subteam discussion: Set Features changes removed the related bit from SANICAP specified scope as subsystem Changed organization of new bits in SANICAP field

	Changed NDCMAS to two bit field Added Set Features of new identifier to privileged actions Added to 8.15
5/29/18	Changed new feature identifier name to Sanitize Added missing text in No-Deallocate Error Or Warning Config in (major change to review) Backed out a change to sanitize command completion in 8.15 (return to black text)
5/31/18	Rewrote No-Deallocate Error Or Warning Config (for clarity) Added Sanitize Status bit 9: No-Deallocate After Sanitize setting Editorial cleanups
6/18/18	Slight reorg to get changes in order (set features Identifier table was out of sequence) Added Asynchronous Event: Sanitize Operation Completed With Warning event Revised Sanitize Config added entry to Sanitize Config Get Features reference table changed Sanitize Config to non-persistent added requirement to default value of NDRMC if Sanitize Config is not saveable) Added Estimated Time For Overwrite With No-Deallocate Media Modification field Sanitize Status removed NDAS in the Sanitize Status (that reported the bit's value in the sanitize command that requested the most recent sanitize operation, as the command's word 10 value is already provided in the log)
6/18/18(ab)	Fixed editorial (capitalization, corrected references) issues and set scope of the new feature to the NvM subsystem.
6/19/18	Changes resulting from subgroup discussion Many clarifications and re-abbreviations Clarified intent to issue only one async event per sanitize completion Clarified requirement of support of new async event (required when feature identifier is supported) Changed intent of the new Sanitize Status (Successfully completed with deallocation) to be only set when deallocation occurs after no-deallocate was requested
6/21/18	Updated language describing new Asynchronous Event Inverted the '10b' selection Removed the remaining "NDRMC" reference (except in history)
6/22/18	Incorporated editorial comments from Judy Block's comments Four commented areas to discuss in next meeting.
6/27/18	Changes made in SubGroup Meeting Sanitize Operation Complete Asynchronous Events references clarified to include associated additional media modification pass (if existing) Sanitize Complete clarified to include associated additional media modification (if existing) Changes made to ensure additional media modification continuation specified for additional media modification operation (if existing) Text added that a NODMMAS value of 00b shall only be allowed in controllers compliant to 1.3. Change not made: did not define Sanitize Config with persistent property
7/9/18	Rewrote multiple areas to describe "associated additional media modification"
7/31/18	Accepted editorial comments from Phase 2 Ballot close vote. Inserted new comments on phase 3 comments
8/28/18	Editorial updates of incomplete references (xxx and with …)
8/29/18	Editorial updates: Comments relating to older versions removed (old 'new text' comments are no longer relevant) 'Sanitize Operation Completed With Unexpected Deallocation' is now consistently described as an asynchronous event <global> Fixed Overwrite Support bit description to be about Overwrite <Figure 109> Clarified inclusion of the extra time in both time estimates and progress reporting <NODMMAS> Effort made to match table/figures numbers to 1.3b

	Editorial language cleanup in Annex
10/03/18	Renamed TP as Sanitize Enhancements (no changes beyond this line!)
11/08/18	Integration: Added Sanitize Config to I/O Controller – Feature Support Formatted final sentence in proposal as an “and” list
11/26/2018	Ratified

Description of Specification Changes

Modify Section 5.2 as shown:

5.2 Asynchronous Event Request command

…

The following event types are defined:

- Error event: Indicates a general error that is not associated with a specific command. To clear this event, host software reads the Error Information log (refer to section 5.14.1.1) using the Get Log Page command with the Retain Asynchronous Event field cleared to '0'.
- SMART / Health Status event: Indicates a SMART or health status event. To clear this event, host software reads the SMART / Health Information log (refer to section 5.14.1.2) using Get Log Page with the Retain Asynchronous Event field cleared to '0'. The SMART / Health conditions that trigger asynchronous events may be configured in the Asynchronous Event Configuration feature using the Set Features command (~~see~~ refer to section 5.21).
- I/O Command Set events: Events that are defined by an I/O command set:
    - NVM Command Set Events:
        - Reservation Log Page Available event: Indicates that one or more Reservation Notification log pages (refer to section 5.14.1.9.1) are available. To clear this event, host software reads the Reservation Notification log page using the Get Log Page command with the Retain Asynchronous Event ~~field~~ bit cleared to '0';
        - Sanitize Operation Completed event: Indicates that a sanitize operation has completed (including any associated additional media modification, refer to the No-Deallocate Modifies Media After Sanitize field in Figure 109 <Identify Controller Data Structure>) without unexpected deallocation of all LBAs (refer to section 5.21.1.TBD) and status is available in the Sanitize Status log page (refer to section 5.14.1.9.2). To clear this event, host software reads the Sanitize Status log page using the Get Log Page command with the Retain Asynchronous Event ~~field~~ bit cleared to '0'; and
        - Sanitize Operation Completed With Unexpected Deallocation asynchronous event: Indicates that a sanitize operation has completed with unexpected deallocation of all LBAs (refer to section 5.21.1.TBD) and status is available in the Sanitize Status log page (refer to section 5.14.1.9.2). To clear this event, host software reads the Sanitize Status log page using the Get Log Page command with the Retain Asynchronous Event bit cleared to '0';

  and
- Vendor Specific event: Indicates a vendor specific event. To clear this event, host software reads the indicated vendor specific log page using Get Log Page command with the Retain Asynchronous Event ~~field~~ bit cleared to '0'.

The Sanitize Operation Completed With Unexpected Deallocation asynchronous event shall be supported if the controller supports the Sanitize Config feature (refer to section 5.21.1.TBD).

Asynchronous events are reported due to a new entry being added to a log page (e.g., Error Information log) or a status update (e.g., status in the SMART / Health log). A status change may be permanent (e.g., the media has become read only) or transient (e.g., the temperature exceeded a threshold for a period of time). Host software should modify the event threshold or mask the event for transient and permanent status changes before issuing another Asynchronous Event Request command to avoid repeated reporting of asynchronous events.

If the controller needs to report an event and there are no outstanding Asynchronous Event Request commands, the controller should send a single notification of that Asynchronous Event Type when an Asynchronous Event Request command is received. If a Get Log Page command clears the event prior to receiving the Asynchronous Event Request command or if a power off condition occurs, then a notification is not sent.

5.2.1 Command Completion

A completion queue entry is posted to the Admin Completion Queue if there is an asynchronous event to report to the host. Command specific status values associated with Asynchronous Event Request are defined in Figure 45.

Figure 45: Status Code – Command Specific Status Values

Value	Description
5h	**Asynchronous Event Request Limit Exceeded:** The number of concurrently outstanding Asynchronous Event Request commands has been exceeded.

Dword 0 of the completion queue entry contains information about the asynchronous event. The definition of Dword 0 of the completion queue entry is in Figure 46.

Figure 46: Asynchronous Event Request – Completion Queue Entry Dword 0

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:24</td><td>Reserved</td></tr>
<tr><td>23:16</td><td>Log Page Identifier: Indicates the log page associated with the asynchronous event. This log page needs to be read by the host to clear the event.</td></tr>
<tr><td>15:08</td><td>Asynchronous Event Information: Refer to Figure 138, Figure 139, Figure 140, and Figure 141 for detailed information regarding the asynchronous event.</td></tr>
<tr><td>07:03</td><td>Reserved</td></tr>
<tr><td>02:00</td><td>Asynchronous Event Type: Indicates the type of the asynchronous event. More specific information on the event is provided in the Asynchronous Event Information field.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0h</td><td>Error status</td></tr>
<tr><td>1h</td><td>SMART / Health status</td></tr>
<tr><td>2h</td><td>Notice</td></tr>
<tr><td>3h to 5h</td><td>Reserved</td></tr>
<tr><td>6h</td><td>I/O Command Set specific status</td></tr>
<tr><td>7h</td><td>Vendor specific</td></tr>
</table>
</td></tr>
</table>

The information in either Figure 47, Figure 48, or Figure 50 is returned in the Asynchronous Event Information field, depending on the Asynchronous Event Type.

…

Figure 50: Asynchronous Event Information – NVM Command Set Specific Status

Value	Description
0h	**Reservation Log Page Available:** Indicates that one or more Reservation Notification log pages (refer to section 5.14.1.13.1) have been added to the Reservation Notification log.
1h	**Sanitize Operation Completed:** Indicates that a sanitize operation has completed (including any associated additional media modification, refer to the No-Deallocate Modifies Media After Sanitize field in Figure 109 <Identify Controller Data Structure>) without unexpected deallocation of all LBAs (refer to section 5.21.1.TBD) and status is available in the Sanitize Status log page (refer to section 5.14.1.13.2).
2h	**Sanitize Operation Completed With Unexpected Deallocation:** Indicates that a sanitize operation for which No-Deallocate After Sanitize (refer to Figure 174 <Sanitize – Command Dword 10>) was requested has completed with the unexpected deallocation of all LBAs (refer to section 5.21.1.TBD) and status is available in the Sanitize Status log page (refer to section 5.14.1.9.2).
3h~~2h~~ to FFh	Reserved

Added reference to Sanitize Config Feature Identifier in Get Features, section 5.13

5.13 Get Features Command

…

Figure 84 describes the Feature Identifiers whose attributes may be retrieved using Get Features. The definition of the attributes returned and associated format is specified in the section indicated.

Figure 84: Get Features – Feature Identifiers

Description	Section Defining Format of Attributes Returned
Sanitize Config	Section 5.21.1.TBD
NVM Command Set Specific	

Modify 5.21.1 Set Features – Features Identify Figure 128 as shown:

Figure 1: Set Features – Feature Identifiers

Feature Identifier	O/M[6]	Persistent Across Power Cycle and Reset[2]	Uses Memory Buffer for Attributes	Description
17h	O	Yes	No	Sanitize Config

Figure 1: Set Features – Feature Identifiers

Feature Identifier	O/M[6]	Persistent Across Power Cycle and Reset[2]	Uses Memory Buffer for Attributes	Description
NOTES: 1. The behavior of a controller in response to an inactive namespace ID to a vendor specific Feature Identifier is vendor specific. 2. This column is only valid if the feature is not saveable (refer to section 7.8). If the feature is saveable, then this column is not used and any feature may be configured to be saved across power cycles and reset. 3. The controller does not save settings for the Host Memory Buffer feature across power states and reset events, however, host software may restore the previous values. Refer to section 8.9. 4. The feature does not use a memory buffer for Set Features, but it does use a memory buffer for Get Features. Refer to section 8.9. 5. The feature is mandatory for NVMe over PCIe. This feature is not supported for NVMe over Fabrics. 6. O/M: O = Optional, M = Mandatory.				

Added Sanitize Config to I/O Controller – Feature Support section 7.1.1.

Figure 404: I/O Controller – Feature Support

Feature Name	Feature Support Requirements [1]
Sanitize Config	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. The feature is mandatory for NVMe over PCIe. This feature is not supported for NVMe over Fabrics.	

Added section on Sanitize Config Feature Identifier to Set Features section 5.21

5.21.1.TBD Sanitize Config (Feature Identifier 17h), (Optional)

This Feature controls behavior of the Sanitize command and sanitize operations. The scope of this Feature is the NVM subsystem.

The attributes are indicated in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure New2 are returned in Dword 0 of the completion queue entry for that command.

If Dword 0 bit 0 of the completion entry of the Get Features command with SEL=011b (i.e., Supported Capabilities) for this Feature Identifier is cleared to ‘0’, (i.e., not saveable, refer to section 7.8), then the default value of the NODRM attribute shall be cleared to ‘0’ (i.e., No-Deallocate Error Response Mode).

If the capabilities of the Sanitize Config Feature Identifier are both changeable and saveable (refer to section 7.8), then the host is able to configure this Feature when initially provisioning a device.

Figure New405: Sanitize Config – Command Dword 11

Bit	Description
31:01	Reserved
00	**No-Deallocate Response Mode (NODRM):** If the No-Deallocate Inhibited bit in the Sanitize Capabilities field of the Identify Controller data structure (refer to Figure 109) is set to '1', then the NODRM bit defines the response of the controller to a Sanitize command processed with the No Deallocate After Sanitize bit (refer to Figure 174 <Sanitize – Command Dword 10>) set to '1'. If the NODRM bit is set to '1' (i.e., No-Deallocate Warning Response Mode), then the controller shall process such Sanitize commands, and if the resulting sanitize operation is completed successfully, then bits 2:0 of the Sanitize Status field in the Sanitize Status log page shall be set to 100b (refer to Figure 104 <Get Log Page – Sanitize Status Log>). If the NODRM bit is cleared to '0' (i.e., No-Deallocate Error Response Mode), then the controller shall abort such Sanitize commands with a status of Invalid Field in Command. If the No-Deallocate Inhibited bit in the Sanitize Capabilities field of the Identify Controller data structure (refer to Figure 109) is cleared to '0', then this bit has no effect.

Modify Section 5.24 as shown:

5.24 Sanitize command – NVM Command Set Specific

The Sanitize command is used to start a sanitize operation or to recover from a previously failed sanitize operation. The sanitize operation types that may be supported are Block Erase, Crypto Erase, and Overwrite. All sanitize operations are processed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation). Refer to section 8.15 for details on the sanitize operation.

When a sanitize operation starts on any controller, all controllers in the NVM subsystem:

- Shall clear any outstanding Sanitize Operation Completed asynchronous event or Sanitize Operation Completed With Unexpected Deallocation asynchronous event;
- Shall update the Sanitize Status log (refer to section 5.14.1.13.2);
- Shall abort any command (submitted or in progress) not allowed during a sanitize operation with a status of Sanitize In Progress (refer to section 8.15.1);
- Should suspend power management activities; and
- Shall release stream identifiers for any open streams.

While a sanitize operation is in progress, all controllers in the NVM subsystem shall abort any command not allowed during a sanitize operation with a status of Sanitize In Progress (refer to section 8.15.1) and the Persistent Memory Region shall behave as described in section 8.15.1.

After a sanitize operation fails, all controllers in the NVM subsystem shall abort any command not allowed during a sanitize operation with a status of Sanitize Failed (refer to section 8.15.1) and the Persistent Memory Region shall behave as described in section 8.15.1 until a subsequent sanitize operation is started or successful recovery from the failed sanitize operation occurs.

If the most recent failed sanitize operation was started in unrestricted completion mode (i.e. the AUSE bit was set to '1' in the Sanitize command), failure recovery requires the host to issue a subsequent Sanitize command in restricted or unrestricted completion mode or to issue a subsequent Sanitize command with the Exit Failure Mode action.

If the most recent failed sanitize operation was started in restricted completion mode (i.e. the AUSE bit was cleared to '0' in the Sanitize command), failure recovery requires the host to issue a subsequent

Sanitize command in restricted completion mode. In the case of a sanitize operation failure in restricted completion mode, before starting another sanitize operation:

- any subsequent Sanitize command issued with the Exit Failure Mode action shall be aborted with a status of Sanitize Failed; and
- any Sanitize command issued in unrestricted completion mode shall be aborted with a status of Sanitize Failed.

The Sanitize Capabilities field in the Identify Controller data structure indicates:

a) the sanitize operation types supported;
b) whether setting No-Deallocate After Sanitize bit (i.e. Sanitize command Dword 10 bit 9) causes media to be modified after a successful sanitize operation completes; and
c) whether the controller inhibits the functionality of the No-Dellocation After Sanitize bit in the Sanitize command.

If an unsupported sanitize operation type is selected by a Sanitize command then the controller shall abort the command with a status of Invalid Field in Command.

If any Persistent Memory Region is enabled in an NVM subsystem, then the controller shall abort any Sanitize command with a status of Sanitize Prohibited While Persistent Memory Region is Enabled. A sanitize operation is prohibited while the Persistent Memory Region is enabled.

If a firmware activation is pending, then the controller shall abort any Sanitize command with a status of Firmware Activation Requires NVM Subsystem Reset. Activation of new firmware is prohibited during a sanitize operation (refer to section 8.15.1).

Support for Sanitize commands in a Controller Memory Buffer (i.e., submitted to an Admin Submission Queue in a Controller Memory Buffer or specifying an Admin Completion Queue in a Controller Memory Buffer) is implementation specific. If an implementation does not support Sanitize commands in a Controller Memory Buffer and a controller's Admin Submission Queue or Admin Completion Queue is in the Controller Memory Buffer, then the controller shall abort all Sanitize commands with a status of Command Not Supported for Queue in CMB.

All sanitize operations (Block Erase, Crypto Erase, Overwrite) are performed in the background (i.e., Sanitize command completion does not indicate sanitize operation completion). If a sanitize operation is started, then the controller shall complete the Sanitize command with a status of Successful Completion. If the controller completes a Sanitize command with any status other than Successful Completion, then the controller:

- shall not start the sanitize operation for that command;
- shall not modify the Sanitize Status log page; and
- shall not alter any user data.

The Sanitize command uses Command Dword 10 and Command Dword 11. All other command specific fields are reserved.

Modify Figure 109 as shown:

Figure 109: Identify – Identify Controller Data Structure

<table>
<tr>
<td>331:328</td>
<td>O</td>
<td>
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 is reserved. Refer to section 8.15.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:30</td><td>
No-Deallocate Modifies Media After Sanitize (NODMMAS): This field indicates if media is additionally modified by the NVMe controller after a sanitize operation successfully completes that had been started a Sanitize command with the No-Deallocate After Sanitize bit set to ‘1’.

The work required for the associated additional media modification is included both in the estimated time for each sanitize operation and in the Sanitize Progress field (refer to Figure 104 <Sanitize Status Log>)
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>Additional media modification after sanitize operation completes successfully is not defined. Only controllers compliant with versions 1.3 and earlier of the specification shall be allowed to return this value.</td></tr>
<tr><td>01b</td><td>Media is not additionally modified by the NVMe controller after sanitize operation completes successfully.</td></tr>
<tr><td>10b</td><td>Media is additionally modified by the NVMe controller after sanitize operation completes successfully. The Sanitize Operation Completed event does not occur until the additional media modification associated with this field has completed.</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table>
If bits 2:0 of the SANICAP field are cleared to 000b, then the controller shall clear this field to 00b.
</td></tr>
<tr><td>29</td><td>
No-Deallocate Inhibited (NDI)

If set to ‘1’ and the No-Deallocate Response Mode bit is set to ‘1’, then the controller deallocates after the sanitize operation even if the No-Deallocate After Sanitize bit is set to ‘1’ in a Sanitize command.

If:
a) set to ‘1’;
b) the No-Deallocate Response Mode bit (refer to Figure New2) is cleared to ‘0’; and
c) the No-Deallocate After Sanitize bit is set to ‘1’ in a Sanitize command,

then the controller aborts the Sanitize command with a status of Invalid Field in Command.

If cleared to ‘0’, then the controller supports the No-Deallocate After Sanitize bit in a Sanitize command.

If bits 2:0 of the SANICAP field are cleared to 000b, then the controller shall clear this bit to ‘0’.
</td></tr>
<tr><td>28:03</td><td>Reserved</td></tr>
</table>
</td>
</tr>
</table>

2	**Overwrite Support (OWS)**: If set to ‘1’, then the controller supports the Overwrite sanitize operation. If cleared to ‘0’, then the controller does not support the Overwrite sanitize operation.
1	**Block Erase Support (BES)**: If set to ‘1’, then the controller supports the Block Erase sanitize operation. If cleared to ‘0’, then the controller does not support the Block Erase sanitize operation.
0	**Crypto Erase Support (CES)**: If set to ‘1’, then the controller supports the Crypto Erase sanitize operation. If cleared to ‘0’, then the controller does not support the Crypto Erase sanitize operation.

~~Bits 31:3 are reserved.~~

~~Bit 2 if set to ‘1’ then the controller supports the Overwrite sanitize operation. If cleared to ‘0’ then the controller does not support the Overwrite sanitize operation.~~

~~Bit 1 if set to ‘1’ then the controller supports the Block Erase sanitize operation. If cleared to ‘0’ then the controller does not support the Block Erase sanitize operation.~~

~~Bit 0 if set to ‘1’ then the controller supports the Crypto Erase sanitize operation. If cleared to ‘0’ then the controller does not support the Crypto Erase sanitize operation.~~

Modify Figure 174 as shown:

Figure 174: Sanitize – Command Dword 10

Bit	Description
31:10	Reserved
09	**No Deallocate After Sanitize**: If set to ‘1’ and the No-Deallocate Inhibited bit (refer to Figure 109 <Identify Controller Data Structure>) is cleared to ‘0’, then the controller shall not deallocate any logical blocks as a result of successfully completing the sanitize operation. If: a) cleared to ‘0’; or b) set to ‘1’ and the No-Deallocate Inhibited bit is set to ‘1’, then the controller should deallocate logical blocks as a result of ~~successfullly~~ successfully completing the sanitize operation. This bit shall be ignored if the Sanitize Action field is set to 001b (i.e., Exit Failure Mode).

Modify Figure 104 as shown:

Figure 104: Get Log Page – Sanitize Status Log

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>01:00</td><td>Sanitize Progress (SPROG): This field indicates the fraction complete of the sanitize operation. The value is a numerator of the fraction complete that has 65,536 (10000h) as its denominator. This value shall be set to FFFFh if the SSTAT field is not set to 010b. If a sanitize operation has been started by a Sanitize command with the No-Deallocate After Sanitize bit set to ‘1’ (refer to section 5.24 <Sanitize Command>) and if NODMMAS field in the Identify Controller data structure is set to 10b (refer to Figure 109 <Identify Controller Data Structure>), then the fraction reported shall include the time related to the additional media modification.</td></tr>
<tr><td>03:02</td><td>Sanitize Status (SSTAT): This field indicates the status associated with the most recent sanitize operation.
Bits 15:9 are reserved.
Bit 8 (Global Data Erased): If ~~if~~ set to ‘1’, then no namespace logical block in the NVM subsystem has been written to and no Persistent Memory Region in the NVM subsystem has been enabled:
• since being manufactured and the NVM subsystem has never been sanitized; or
• since the most recent successful sanitize operation.
If cleared to ‘0’, then a namespace logical block in the NVM subsystem has been written to or a Persistent Memory Region in the NVM subsystem has been enabled:
a) since being manufactured and the NVM subsystem has never been sanitized; or
b) since the most recent successful sanitize operation of the NVM subsystem.
Bits 7:3 contains the number of completed passes if the most recent sanitize operation was an Overwrite. This field shall be cleared to 00000b if the most recent sanitize operation was not an Overwrite.
Bits 2:0 contains the status of the most recent sanitize operation as shown below.

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>The NVM subsystem has never been sanitized.</td></tr>
<tr><td>001b</td><td>The most recent sanitize operation completed successfully, including any additional media modification (refer to the No-Deallocate Modifies Media After Sanitize field in Figure 109 <Identify Controller Data Structure>).</td></tr>
<tr><td>010b</td><td>A sanitize operation is currently in progress.</td></tr>
<tr><td>011b</td><td>The most recent sanitize operation failed.</td></tr>
<tr><td>100b</td><td>The most recent sanitize operation for which No-Deallocate After Sanitize (refer to section 5.24) was requested has completed successfully with deallocation of all LBAs (refer to section 5.21.1.TBD).</td></tr>
<tr><td>~~100b~~101b to 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>07:04</td><td>Sanitize Command Dword 10 Information (SCDW10): This field contains the value of the Command Dword 10 field of the Sanitize command that started the sanitize operation whose status is reported in the SSTAT field. Refer to Figure 282.</td></tr>
<tr><td>11:08</td><td>Estimated Time For Overwrite: This field indicates the number of seconds required to complete an Overwrite sanitize operation with 16 passes in the background (refer to section 5.24) when the No-Deallocate Modifies Media After Sanitize field (refer to Figure 109 <Identify Controller Data Structure>) is not set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</td></tr>
<tr><td>15:12</td><td>Estimated Time For Block Erase: This field indicates the number of seconds required to complete a Block Erase sanitize operation in the background (refer to section 5.24) when the No-Deallocate Modifies Media After Sanitize field (refer to Figure 109 <Identify Controller Data Structure>) is not set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</td></tr>
</table>

Figure 104: Get Log Page – Sanitize Status Log

Bytes	Description
19:16	**Estimated Time For Crypto Erase:** This field indicates the number of seconds required to complete a Crypto Erase sanitize operation in the background (refer to section 5.24) when the No-Deallocate Modifies Media After Sanitize field (refer to Figure 109 <Identify Controller Data Structure>) is not set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.
23:20	**Estimated Time For Overwrite With No-Deallocate Media Modification:** This field indicates the number of seconds required to complete an Overwrite sanitize operation and the associated additional media modification after the Overwrite sanitize operation in the background (refer to section 5.24) when: a) the No-Deallocate bit was set to '1' in the Sanitize command that requested the Overwrite sanitize operation; and b) the No-Deallocate Modifies Media After Sanitize field (refer to Figure 109 <Identify Controller Data Structure>) is set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.
27:24	**Estimated Time For Block Erase With No-Deallocate Media Modification:** This field indicates the number of seconds required to complete a Block Erase sanitize operation and the associated additional media modification after the Block Erase sanitize operation in the background (refer to section 5.24) when: a) the No-Deallocate bit was set to '1' in the Sanitize command that requested the Block Erase sanitize operation; and b) the No-Deallocate Modifies Media After Sanitize field (refer to Figure 109 <Identify Controller Data Structure>) is set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.
31:28	**Estimated Time For Crypto Erase With No-Deallocate Media Modification:** This field indicates the number of seconds required to complete a Crypto Erase sanitize operation and the associated additional media modification after the Crypto Erase sanitize operation in the background (refer to section 5.24) when: a) the No-Deallocate bit was set to '1' in the Sanitize command that requested the Crypto Erase sanitize operation; and b) the No-Deallocate Modifies Media After Sanitize field (refer to Figure 109 <Identify Controller Data Structure>) is set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.
511:~~20~~32	Reserved

Modify Section 7.14 as shown:

7.14 Privileged Actions

Privileged actions are actions (e.g., command, register write) that affect or have the potential to affect the state of the entire NVM subsystem and not only the controller and/or namespace with which the action is associated.

Admin commands that are privileged include Namespace Management, Namespace Attachment, Virtualization Management, Format NVM, Set Features with Feature Identifier 17h (i.e., Sanitize Config, refer to section 5.1.21.TBD), and Sanitize. A privileged register action is NVM subsystem reset. Vendor specific commands and registers may also be privileged.

Modify Section 8.15 as shown:

8.15 Sanitize Operations (Optional)

A sanitize operation alters all user data in the NVM subsystem such that recovery of any previous user data from any cache, the non-volatile media, or any Controller Memory Buffer is not possible. It is implementation specific whether Submission Queues and Completion Queues within a Controller Memory Buffer are altered by a sanitize operation; all other data stored in all Controller Memory Buffers is altered by a sanitize operation. If a portion of the user data was not altered and the sanitize operation completed successfully, then the NVM subsystem shall ensure permanent inaccessibility of that portion of the user data for any future use within the NVM subsystem (e.g., retrieval from NVM media, caches, or any Controller Memory Buffer) and permanent inaccessibility of that portion of the user data via any interface to the NVM subsystem, including management interfaces such as NVMe-MI.

The scope of a sanitize operation is all locations in the NVM subsystem that are able to contain user data, including caches and unallocated or deallocated areas of the media. Sanitize operations do not affect the Replay Protected Memory Block, boot partitions, or other media and caches that do not contain user data. A sanitize operation also may alter log pages as necessary (e.g., to prevent derivation of user data from log page information). Once a sanitize operation is started~~,~~ it cannot be aborted and continues after a Controller Level Reset including across power cycles. Refer to Annex A for further information about sanitize operations.

The Sanitize command (refer to section 5.24) is used to start a sanitize operation or to recover from a previously failed sanitize operation. All sanitize operations are performed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation). The completion of a sanitize operation is indicated in the Sanitize Status log page, and with either the Sanitize Operation Completed asynchronous event or the Sanitize Operation Completed With Unexpected Deallocation asynchronous event (if an Asynchronous Event Request Command is outstanding).

The Sanitize Capabilities field of the Identify Controller data structure indicates the sanitize operation types supported and controller attributes specific to sanitize operations.

The sanitize operation types are:

- The Block Erase sanitize operation alters user data with a low-level block erase method that is specific to the media for all locations on the media within the NVM subsystem in which user data may be stored.
- The Crypto Erase sanitize operation alters user data by changing the media encryption keys for all locations on the media within the NVM subsystem in which user data may be stored.
- The Overwrite sanitize operation alters user data by writing a fixed data pattern or related patterns to all locations on the media within the NVM subsystem in which user data may be stored one or more times. Figure 281 defines the data pattern or patterns that are written.

Controller attributes specific to sanitize operations include:

- The No-Deallocate Modifies Media After Sanitize (NODMMAS) field which indicates if media is modified by the controller after a sanitize operation successfully completes that had been requested with No-Deallocate After Sanitize set to ‘1’ in the Sanitize command that started the sanitize operation; and
- No-Deallocate Inhibited (NDI) bit which indicates if the controller supports the No-Deallocate After Sanitize bit in the Sanitize Command.

The NODMMAS field in the Identify Controller data structure (refer to Figure 109), specifies that if a Sanitize command includes No-Deallocate After Sanitize set to ‘1’ and NODMMAS is set to 10b, then a sanitize operation has an associated additional media modification operation. This additional media modification operation acts upon the results of the requested sanitize operation with the purpose of making all LBA contents readable. Refer to Annex A for further information about sanitize operations and interactions with integrity circuits.

This additional media modification shall complete before the NVM subsystem:

a) reports sanitize completion by Asynchronous Event (refer to section 5.2); and
b) reports sanitize completion in the Sanitize Status log (refer to section 5.14.1.9.2 <Sanitize Status>).

The Overwrite sanitize operation is media specific and may not be appropriate for all media types. For example, if the media is NAND, multiple pass overwrite operations may have an adverse effect on media endurance.

Figure 281: Sanitize Operations – Overwrite Mechanism

OIPBP[1]	Overwrite Pass Count[1]	Overwrite Pass Number	Logical Block Data and Non-PI Metadata[2]	Protection Information[3]
'0'	All	All	Overwrite Pattern[1]	FFFFFFFF_FFFFFFFFh
'1'	Even	First	Inversion of Overwrite Pattern[1]	00000000_00000000h
		Subsequent	Inversion of Overwrite Pattern[1] from previous pass (i.e., each bit XORed with '1')	
'1'	Odd	First	Overwrite Pattern[1]	FFFFFFFF_FFFFFFFFh
		Subsequent	Inversion of Overwrite Pattern[1] from previous pass (i.e., each bit XORed with '1')	

NOTES:
1. Parameters are specified in Command Dword 10 and Command Dword 11 of the corresponding Sanitize command that started the Overwrite operation. The Overwrite Invert Pattern Between Passes (OIPBP) field is defined in Command Dword 10. The Overwrite Pass Count is defined in Command Dword 10. The Overwrite Pattern is defined in Command Dword 11. Refer to section 5.24.
2. If metadata other than Protection Infomation is present.
3. If Protection Information is present within the metadata.

To start a sanitize operation, the host submits a Sanitize command specifying one of the sanitize operation types (i.e., Block Erase, Overwrite, or Crypto Erase). The host sets command parameters, including the Allow Unrestricted Sanitize Exit bit and the No Deallocate After Sanitize bit, to the desired values. After validating the Sanitize command parameters, the controller starts the sanitize operation in the background, updates the Sanitize Status log page and then completes the Sanitize command with Successful Completion status. If the sanitize operation is to be followed by an associated additional media modification operation (refer to NODMMAS in Figure 109), then the associated additional media modification operation shall be completed before the controller reports sanitize operation complete. If a Sanitize command is completed with any status other than Successful Completion, then the controller shall not start the sanitize operation and shall not update the Sanitize Status log page. The controller ignores Critical Warning(s) in the SMART / Health Information log page (e.g., read only mode) and attempts to complete the sanitize operation requested. While a sanitize operation is in progress, all controllers shall abort any commands not listed in Figure 283 with a status of Sanitize In Progress (refer to section 8.15.1).

The user data values that result from a successful sanitize operation are specified in Figure 282. If the controller deallocates user data after successful completion of a sanitize operation, then values read from deallocated logical blocks are described in section 6.7.1.1. The host may specify that sanitized logical blocks not be deallocated by setting the No Deallocate After Sanitize bit to '1' in the Sanitize command.

Figure 282: Sanitize Operations – User Data Values

Sanitize Operation	Logical Blocks	Non-PI Metadata[1]	Protection Information[2]
Block Erase	Vendor specific value	Vendor specific value	Vendor specific value
Crypto Erase	Indeterminate	Indeterminate	Indeterminate
Overwrite	Refer to Figure 281	Refer to Figure 281	Refer to Figure 281

NOTES:
1. If metadata other than Protection Information is present.
2. If Protection Information is present within the metadata.

The Sanitize Status log page (refer to section 5.14.1.9.2) contains estimated times for sanitize operations and a consistent snapshot of information about the most recently started sanitize operation, including whether a sanitize operation is in progress, the sanitize operation parameters and the status of the most recent sanitize operation. The controller shall report sanitize operation in progress if either a sanitize operation is in progress or an associated additional media modification operation is in progress. If a sanitize operation is not in progress, then the Global Data Erased bit in the log page indicates whether the NVM subsystem may contain any user data (i.e., has not been written to since the most recent successful sanitize operation).

The Sanitize Status log page shall be updated as described:

- Initialize before any controller in the NVM subsystem is ready.
- Update before a Sanitize command that starts a sanitize operation is completed (i.e., prior to the completion queue entry being posted for the Sanitize command).
- Update when a sanitize operation is complete (e.g., immediately prior to the completion queue entry being posted for the Sanitize Operation Completed asynchronous event or for the Sanitize Operation Completed With Unexpected Deallocation asynchronous event).

The Sanitize Status log page should be updated periodically during a sanitize operation to make progress information available to hosts.
During a sanitize operation, the host may periodically examine the Sanitize Status log page to check for progress, however, the host should limit this polling (e.g., to at most once every several minutes) to avoid interfering with the progress of the sanitize operation itself.
On completion of a sanitize operation:

- If the sanitize operation is successful, then the Global Data Erased bit shall be set to '1'.
- The Sanitize Status log page is updated.
- The controller to which the Sanitize command was submitted completes an Asynchronous Event Request command (if one is outstanding) with the following information:
  - The Log Page Identifier field is set to 81h (i.e., Sanitize Status).
  - The Asynchronous Event Information field is set to Sanitize Operation Completed or to Sanitize Operation Completed With Unexpected Deallocation asynchronous event (refer to section 5.2).
  - The Asynchronous Event Type field is set to 6h (i.e., I/O Command Set specific status).
- All controllers in the NVM subsystem may resume any power management that was suspended when the sanitize operation started.

The host should read the Sanitize Status log page upon completion of a sanitize operation (which clears the asynchronous event, if one was generated).
If a sanitize operation fails, all controllers in the NVM subsystem shall abort any command not allowed during a sanitize operation with a status of Sanitize Failed (refer to section 8.15.1) until a subsequent sanitize operation is started or successful recovery from the failed sanitize operation occurs. A subsequent successful sanitize operation or the Exit Failure Mode action may be used to recover from a failed sanitize operation. Refer to section 5.24 for recovery details.
If the Sanitize command is supported, then the NVM subsystem and all controllers shall:

- Support the Sanitize Status log page;
- Support the Sanitize Operation Completed asynchronous event ~~and enable the event by default~~;
- Support the Sanitize Operation Completed With Unexpected Deallocation asynchronous event, if the Sanitize Config feature is supported;
- Support the Exit Failure Mode action for a Sanitize command;
- Support at least one of the following sanitize operation types: Block Erase, Overwrite, or Crypto Erase; and
- Indicate support for all supported sanitize operation types in the Sanitize Capabilities field in the Identify Controller data structure.

The Sanitize Config Feature Identifier (refer to section 5.21.1.TBD) contains the No-Deallocate Response Mode bit that specifies the response of the controller to a Sanitize command processed with the No Deallocate After Sanitize bit (refer to Figure 174 <Sanitize – Command Dword 10>) set to '1' if the No-Deallocate Inhibited bit in the Sanitize Capabilities field of the Identify Data Structure (refer to Figure 109) is set to '1'. In the No-Deallocate Error Response Mode, the controller aborts such Sanitize commands with a status of Invalid Field in Command. In the No-Deallocate Warning Response Mode, the controller processes such Sanitize commands, and if a resulting sanitize operation is completed successfully, then bits 2:0 of the Sanitize Status field in the Sanitize Status log page are set to 100b (refer to Figure 104 <Get Log Page – Sanitize Status Log >).

Added section 8.15.1 for reference (no intended changes)

8.15.1 Command Restrictions

While performing a sanitize operation and while a failed sanitize operation has occurred but successful recovery from that failure has not occurred, all enabled controllers and namespaces in the NVM subsystem are restricted to performing only a limited set of actions.

While a sanitize operation is in progress:

- All controllers in the NVM subsystem shall only process the Admin commands listed in Figure 283 subject to the additional restrictions stated in that figure;
- All I/O Commands shall be aborted with a status of Sanitize In Progress; and
- Any command or command option that is not explicitly permitted in Figure 283 shall be aborted with a status of Sanitize in Progress if fetched by any controller in the NVM subsystem.

While a failed sanitize operation has occurred, a subsequent sanitize operation has not started and successful recovery from the failed sanitize operation has not occurred:

- All controllers in the NVM subsystem shall only process the Sanitize command (refer to section 5.24) and the Admin commands listed in Figure 283 subject to the additional restrictions noted in that figure;
- All I/O Commands are shall be aborted with a status of Sanitize Failed;
- The Sanitize command is permitted with action restrictions (refer to section 5.24); and
- Aside from the Sanitize command, any other command or command option that is not explicitly permitted in Figure 283 shall be aborted with a status of Sanitize Failed if fetched by any controller in the NVM subsystem.

Figure 406: Sanitize Operations – Admin Commands Allowed

<table>
<tr><th>Admin Command</th><th>Additional Restrictions</th></tr>
<tr><td>Abort</td><td></td></tr>
<tr><td>Asynchronous Event Request</td><td></td></tr>
<tr><td>Create I/O Completion Queue</td><td></td></tr>
<tr><td>Create I/O Submission Queue</td><td></td></tr>
<tr><td>Delete I/O Completion Queue</td><td></td></tr>
<tr><td>Delete I/O Submission Queue</td><td></td></tr>
<tr><td>Get Features</td><td></td></tr>
<tr><td>Get Log Page</td><td>The log pages allowed are listed below.

<table><tr><th>Log Pages</th><th>Additional Restrictions</th></tr><tr><td>Error Information</td><td>Return zeros in the LBA field.</td></tr><tr><td>SMART / Health Information</td><td></td></tr><tr><td>Changed Namespace List</td><td></td></tr><tr><td>Reservation Notification</td><td></td></tr><tr><td>Sanitize Status</td><td></td></tr><tr><td>Asymmetric Namespace Access</td><td></td></tr></table></td></tr>
<tr><td>Identify</td><td></td></tr>
<tr><td>Keep Alive</td><td></td></tr>
<tr><td>NVMe-MI Receive</td><td rowspan="2">Prohibited unless explicitly allowed in the NVM Express Management Interface Specification.</td></tr>
<tr><td>NVMe-MI Send</td></tr>
<tr><td>Set Features</td><td>Namespace Write Protection Config Feature is not allowed.</td></tr>
<tr><td>Opcode 7Fh</td><td>The Fabric Commands allowed are listed below. Refer to the NVMe over Fabrics specification.

<table><tr><th>Fabrics Commands</th><th>Additional Restrictions</th></tr><tr><td>Property Set</td><td></td></tr><tr><td>Connect</td><td></td></tr><tr><td>Property Get</td><td></td></tr><tr><td>Authentication Send</td><td></td></tr><tr><td>Authentication Receive</td><td></td></tr><tr><td>Vendor Specific</td><td>Commands are allowed that do not affect or retrieve user data.</td></tr></table></td></tr>
<tr><td>Vendor Specific</td><td>Commands are allowed that do not affect or retrieve user data.</td></tr>
</table>

Insert appendix as shown:

Annex A Sanitize Operation Considerations (Informative)

Annex A.1 Overview

The Sanitize command initiates a sanitize operation that makes all user data previously written to the device inaccessible. To do this a Sanitize command is provided over the device's physical interface that cause the controller to process the requested operation. The actual result of the operation is very difficult to prove as complete. This annex provides some context and considerations for understanding the result of the operation and the practical limitations for auditing the result of the sanitize operation.

Annex A.2 Hidden Storage (Overprovisioning)

Sanitize operations affect all physical storage that is able to hold user data. Many NVMe SSDs contain more physical storage than is addressable through the interface (overprovisioning), which is used for vendor specific purposes that may include providing increasing endurance, improving performance, and providing extra blocks to allow retiring bad or worn-out storage without affecting capacity. This excess capacity as well as any retired storage are not accessible through the interface. Vendor specific innovative use of this extra capacity supports advantages to the end user, but the lack of observability makes it difficult to ensure that all storage within the device has been affected. Only the accessible storage can be audited for the results of a sanitization operation.

Annex A.3 Integrity checks and No-Deallocate After Sanitize

Another issue is availability of the data returned through the interface. Some of the sanitize operations (e.g., Block Erase) affect the physical devices in such a way that directly reading the accessible storage may trigger internal integrity checks resulting in error responses instead of returning the contents of the storage. Other sanitize operations (e.g., Crypto Erase) may scramble the internal vendor specific internal format of the data also resulting in error responses instead of returning the contents of the storage.

Some devices compensate for these issues by performing an additional internal write operation on all storage that can be allocated for user data. However, this has the side effect of potentially significant additional wear on the device as well as the side effect of obscuring the results of the initial sanitize operation (i.e., the writes forensically destroy the ability to audit the result of the initial sanitize operation). Given this side effect, process audits of sanitize behavior only prove effective results when the No-Deallocate After Sanitize bit is set the same way (e.g., set to '1') for both process audits and the individual device audits.

The Sanitize command introduced in NVM Express Revision 1.3 included a mechanism to specify that sanitized addressable storage not be deallocated, thereby allowing observations of the results of the sanitization operation. However, some architectures and products with integrity checking circuitry interact with this capability in such a way as to defeat the sanitize result observability purpose. A technical proposal has been developed that applies to that NVM Express revision. This ratified technical proposal included extended information about the sanitization capabilities of devices, a new asynchronous event, and configuration of the response to No-Deallocate After Sanitize requests. This new material is intended to both support new systems that understand the new capabilities, as well to help manage legacy systems that do not understand the new capabilities without losing the ability to sanitize as requested.

Annex A.4 Bad Block and Vendor Specific NAND Use

Another audit capability that is not supported by NVM Express is checking that any blocks that could not be sanitized (e.g., bad physical blocks) have been removed from the pool of storage that can be used as addressable storage.

An approach that is performed under some circumstances is removing the storage components from the NVM Express device after a sanitize operation and reading the contents in laboratory conditions. However this approach also has multiple difficulties. When physical storage devices are removed from a NVM Express device, much context is lost. This includes:

a) any encoding for zero's/one's balance;
b) identification of which components contain device firmware or other non-data information; and
c) which blocks have been retired and cannot be sanitized.