Offline Logical DePop

April 26, 2016

Revision 1

Toshiba

Technical Editors:

Joe Breher
Western Digital
joe.breher@hgst.com
+1 (478) 227-3437

Jim Hatfield
389 Disc Drive
Longmont, CO 80503
720-684-2120
James.C.Hatfield@Seagate.com

Mark Carlson
Principal Engineer, Industry Standards
Toshiba
mark.carlson@taec.toshiba.com
303-720-6139

Document Status

Revision History		
Rev	**Date**	**Description**
0	2016 Apr 19	Initial revision
1	2016 Apr 26	Incorporated comments from plenary meeting of 2016 Apr 19

I – Introduction

A significant performance problem exists in storage systems where data is spread across multiple drives when one of the drives takes an inordinate amount of time to respond compared to the others. The host software may already have sufficient information to respond to the users request due to erasure coding or replication but the request ends up taking as long as the slowest drive. Know as a "long tail" of performance, many use cases will then mark these slow drives as "failed" and remove them from service, requiring replacement.

Offline Logical DePop is meant to address this use case and allow returning a reduced capacity drive to service with the slow physical elements of the drive removed from the logical address space as a result. With many software defined storage solutions, this reformatted, empty drive can now be rewritten over time with new data. This proposal is not intended to meet the use case of a drive used in a RAID stripe set where rebuilding of the drive could be accelerated by retaining the data on the physical elements that were not depopulated.

Many storage devices are implemented employing multiple ~~identical~~ subunits, each of which provide some amount of storage resources. The entire capacity of the storage device is the sum of the capacity of each of these subunits (i.e. physical storage elements). Such devices may experience failure of some subset of these physical elements. The failure of some subset of physical elements may not fundamentally preclude operation of the remainder of the device.

This proposal defines application layer constructs for the management of such a device's ability to operate in a reduced-capacity manner, as an alternative to total device replacement. Among the constructs comprising this management mechanism are:

a) a signal from device to host that a physical element may be degraded;
b) a means of the host querying the status of all physical elements within the device; and
c) a command by which a host may specify that the device shall 'offline' a specified physical storage element.

Upon being directed by the host to 'offline' a physical element, the device:

1) makes the physical storage element ineligible for storage of user data;
2) reduces the reported capacity of the device to reflect the storage within those physical storage elements still valid; ~~and~~
3) ~~reformats the device to this new lowered capacity~~may rewrite any accessible LBAs; and
4) reports a new lowered capacity.

II – Scope

This proposal is written against ACS-4 revision 12.

III – Change marking conventions

Unless otherwise indicated additions are shown in blue, deletions in red strikethrough, and comments in green.

IV – Changes to ACS-4

3 Definitions, abbreviations, and conventions

Editor's Note 1: This clause contains items that are new to ACS. As such, edit markers denote text changes since rev 0.

3.1 Definitions

3.1.90 depopulate
~~to render a physical storage element invalid for the purpose of storing user data~~to reduce the usable capacity of the media, by the quantity of valid physical sectors associated with a specified physical storage element or physical storage subelement

3.1.91 physical element
subcomponent of a physical entity that implements an ATA device

3.1.92 physical storage element
physical element that provides non-volatile storage for an associated group of logical blocks (see 4.16)

3.1.93 physical subelement
physical element that is a proper subset of a physical element

3.1.94 physical storage subelement
physical element that is a proper subset of a physical storage subelement

4 Feature set definitions

4.1 Overview

4.1.1 Feature set summary

Table 1 lists the feature sets in alphabetical order and shows whether a feature set is mandatory, optional, or prohibited for ATA devices.

Table 1 — Feature set summary

Feature set	ATA devices
48-bit Address feature set (see 4.3)	O
Accessible Max Address Configuration feature set (see 4.4)	O
Advanced Power Management (APM) feature set (see 4.5)	O
CompactFlash Association (CFA) feature set (see 4.6)	O
Device Statistics Notifications (DSN) feature set (see 4.7)	O
Extended Power Conditions (EPC) feature set (see 4.8)	O
Free-fall Control feature set (see 4.9)	O
General feature set (see 4.2)	M
General Purpose Logging (GPL) feature set (see 4.10)	M
Long Logical Sector (LLS) feature set (see 4.11)	O
Long Physical Sector (LPS) feature set (see 4.12)	O
Native Command Queuing (NCQ) feature set (see 4.13)	O
Offline Logical DePop feature set (see 4.x)	O
PACKET feature set (see ACS-3)	P
Power Management feature set (see 4.14)	M
Power-Up In Standby (PUIS) feature set (see 4.15)	O
Rebuild Assist feature set (see 4.16)	O
Sanitize Device feature set (see 4.17)	O
SATA Hardware Feature Control feature set (see 4.22)	O
Security feature set (see 4.18)	O
Self-Monitoring, Analysis, and Reporting Technology (SMART) feature set (see 4.19)	O
Sense Data Reporting feature set (see 4.20)	O
Software Settings Preservation (SSP) feature set (see 4.21)	O
Streaming feature set (see 4.23)	O
Trusted Computing feature set (see 4.24)	O
Write-Read-Verify feature set (see 4.25)	O
Key: M – Mandatory, O – Optional, P – Prohibited	

4.2 Offline Logical DePop

4.2.1 Overview

Editor's Note 2: This entire subclause is new, and edit markers denote text changes since rev 0.

Offline Logical DePop provides a mechanism for an application client to improve some aspect of device performance (e.g., latency) by means of making a specified physical storage element an invalid location of LBA mapping resources.

Block device implementations may contain a number of physical storage elements. The media in such a device may consist of some number of these physical storage elements. These physical storage elements may contain physical storage subelements. Each of these elements (i.e., physical storage element or physical storage subelement):

a) is associated with some number of physical sectors; and
b) may have a health status independent of the other elements in the device.

In some cases, the health status of a given element may become degraded. Such degradation may affect the overall performance of the device as seen by the application client.

An application client may specify that a physical storage element or a physical storage subelement be depopulated by means of the Offline Logical DePop ~~as specified in this subclause~~feature set.

Devices that support this feature set shall support the:

a) General Purpose Logging feature set (see 4.10); and
b) Sense Data Reporting feature set (see 4.20).

A device that supports Offline Logical DePop ~~as specified in this subclause~~shall:

a) ~~shall~~ set the OFFLDP SUPPORTED bit to one (see 9.11.5.2.x);
b) ~~shall~~ support the OFFLDP ENABLED bit (see 9.11.6.2.x);
c) ~~shall~~ support the Physical Element Status Input log page (see 9.x);
d) ~~shall~~ support the LOGICAL DEPOP command (see clause 5.x); and
e) ~~shall~~ support the ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command (see clause 7.99.3).

A physical storage element or physical storage subelement that has been depopulated contains no usable physical sectors (e.g., LBA mapping resources). The depopulation ~~consequently~~ reduces the usable capacity of the media, by the quantity of valid physical sectors that were associated with the physical storage element or physical storage subelement before the depopulation.

~~All user data in any logical blocks associated with a physical storage element or physical storage subelement prior to that physical storage element or physical storage subelement being depopulated shall become permanently irretrievable. After a physical storage element or physical storage subelement is depopulated, the device shall not allow access to the associated media.~~

The Offline Logical DePop feature set as described in this subclause uses the ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command.

An ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command specifies that the device shall reduce the capacity of the media before returning status for that command. The capacity to which the media shall be truncated is the capacity at the time of command receipt minus the capacity associated with the physical storage element or physical storage subelement being depopulated. This lowered capacity may be subject to vendor unique rounding.

An ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command may result in a format operation or other reinitialization of all data on the media.

4.2.2 interactions

Editor's Note 3: We may need to define interactions with other commands and activities such as SET ACCESSSIBLE MAX ADDRESS, security feature set (abort this in Locked, executable in Unlocked, Disabled, Frozen?), background activities, TCG Opal

4.2.2.1 interactions with resets

4.2.2.2 interactions with other commands and logs

4.2.2.3 interactions with caches

7.99 LOGICAL DEPOP – TBDh, Non-Data

Editor's Note 4: This entire subclause is new, and edit markers denote text changes since rev 0.

7.99.1 Introduction

7.99.1.1 Feature Set

This 48-bit command is for devices the support the Offline Logical DePop feature set or the Online Logical DePop feature set.

7.99.1.2 Description

The LOGICAL DEPOP command is used by the host to manage the inventory of physical storage elements with the device.

7.99.1.3 Inputs

See table 2 for the LOGICAL DEPOP command inputs.

Table 2 — LOGICAL DEPOP command inputs

Field	Description
FEATURE	15:8 Reserved 7:0 LOGICAL DEPOP SUBCOMMAND field – See 7.99.2
COUNT	Subcommand specific
LBA	Subcommand specific
DEVICE	**Bit Description** 7 Obsolete 6 N/A 5 Obsolete 4 Transport Dependent – See 6.2.11 3:0 Reserved
COMMAND	7:0 TBD

7.99.1.4 Normal Outputs

See table 296.

7.99.1.5 Error Outputs

The ABORT bit shall be set to one if any subcommand input value is not supported or is invalid. See table 306.

7.99.2 LOGICAL DEPOP subcommands

The LOGICAL DEPOP SUBCOMMAND field (see table 2) specifies the LOGICAL DEPOP ~~subcommand~~command to be processed using the codes shown in table 3.

Table 3 — LOGICAL DEPOP command subcommand codes

Code	Description
00h	Reserved
01h	~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL (see 7.99.3)
02h..FFh	Reserved

7.99.3 ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command

7.99.3.1 Feature Set

This 48-bit command is for devices that support the Offline Logical DePop feature set (see 4.x).

7.99.3.2 Description

The ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command specifies a physical storage element to be depopulated. An ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command may be issued for each physical storage element that is to be removed from the current operating configuration.

7.99.3.3 Inputs

7.99.4 Overview

See table 4 for the ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command inputs.

Table 4 — ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command inputs

Field	Description
FEATURE	~~0001h~~ **Bit Description** 15:8 Reserved 7:0 01h
COUNT	**Bit Description** 15:1 Reserved 0 SUB bit – See 7.99.4.1
LBA	**Bit Description** 47:24 Reserved 23:16 PHYSICAL SUBELEMENT field – See 7.99.4.2 15:0 PHYSICAL ELEMENT field – See 7.99.4.3
DEVICE	**Bit Description** 7:5 Obsolete 4 Transport Dependent – See 6.2.11 3:0 Reserved
COMMAND	7:0 TBD

7.99.4.1 SUB bit

A SUB bit set to zero specifies that the element to be depopulated is a physical element. A SUB bit set to one specifies that the element to be depopulated is a physical subelement.

7.99.4.2 PHYSICAL SUBELEMENT field

If the SUB bit is set to one, the PHYSICAL SUBELEMENT field specifies the element to be depopulated. If the SUB bit is set to zero, the PHYSICAL SUBELEMENT field shall be ignored.

7.99.4.3 PHYSICAL ELEMENT field

If the SUB bit is set to zero, then the PHYSICAL ELEMENT field specifies the element to be depopulated. If the SUB bit is set to one, the PHYSICAL ELEMENT field specifies the physical storage element containing the physical subelement to be depopulated.

7.99.5 Normal Outputs

See .

7.99.6 Error Outputs

The ABORT bit shall be set to one if any of the following are true:

a) the OFFLDP ENABLED bit is not set to 1b;
b) the PHYSICAL ELEMENT field and the PHYSICAL SUBELEMENT together specify an element not supported by the device; or
c) tbd.

If the ABORT bit is set to one, then the command shall not cause any physical elements to be depopulated. See table 309 for the definition of Error Outputs.

Editor's Note 5: Reformatting the drive takes time so how about adding an option bit to allow whether reformat is done in captive (i.e., before command completion) or immediate (i.e., after command completion). For example, if immediate mode, ~~Offline DePop and Reformat~~DESTRUCTIVE ELEMENT REMOVAL ~~subcommand~~command can complete quickly but the actual reformat can continue after command completion as a background activity. The host may query by REQUEST SENSE DATA EXT for the progress of reformat. (We can use COUNT field to report progress). For offline mode we want to abort any commands but REQUEST SENSE DATA EXT similar to sanitize operation and SANITIZE STATUS EXT. A comment was rendered that the term 'immediate' may be problematic. I use it here in the sense that command may complete before the action specified by the command is executed.

Editor's Note 6: Need to add GET LBA MAPPING command

---

9 Log Definitions

9.99 Physical Element Status log (Log Address tbd)

Editor's Note 7: This entire subclause is new, and edit markers denote text changes since rev 0.

9.99.1 Overview

The Physical Element Status log provides information about the health of physical elements within the device.

9.99.2 Contents of the Physical Element Status log

Table 5 defines the format of the Pending Defects log for page 0. Table 6 defines the format of all subsequent pages of the log. The size (i.e., number of pages) of the Pending Defects log is indicated in the General Purpose Directory log (see 9.2).

Table 5 — Physical Element Status log (page 0)

Offset	Type	Description
0..3	DWord	NUMBER OF LOG DESCRIPTORS field (see 9.99.3)
4..7		Reserved
8..15	Bytes	Physical Element Status Log descriptor 0 (see 9.99.4)
16..23	Bytes	Physical Element Status Log descriptor 1
…		…
504..511	Bytes	Physical Element Status Log descriptor 63

Table 6 — Physical Element Status log (page 1..n)

Offset	Type	Description
0..7	Bytes	Physical Element Status Log descriptor 64 + ((log page number–1) × 64)
8..15	Bytes	Physical Element Status Log descriptor 65 + ((log page number–1) × 64)
…		…
504..511	Bytes	Physical Element Status Log descriptor 127 + ((log page number–1) × 64)

The Physical Element Status log shall contain a Physical Element Status descriptor for every physical element within the device (i.e., depopulation operations do not remove Physical Element Status log descriptors from the Physical Element Status log).

The Physical Element Status descriptors shall be sorted with PHYSICAL ELEMENT being the most significant sort order and the PHYSICAL SUBELEMENT being the next significant sort order.

Editor's Note 8: The following comment was received: *no. make this like the current device status log. start at byte offset 0 and spans multiple pages, with padding to end of the page after final non-zero data*. Note however that this was cloned from the Pending Defects log. I'll change if it is the consensus. A subsequent comment: "*I don't understand the intent behind the original comment. I do like where you are going. However, I do not think that you need to sort by PHYS ELEM / SUBPHYS ELEM here. You do need some method of indicating that some or all of a sub-element has failed so that the host can decide to prune the whole element, or just a list of sub-elements.*"

Editor's Note 9: The following comment was received: *can descriptors in the middle be zeros and skipped over ? or does the first all-zero descriptor mark the end of data ?* Note that adopting the Pending Defects log template skirts these questions - though it should be acknowledged that the Pending Defects log also does not define padding for unused descriptors at the end of a page. See below for the LBA Status log resolution of this issue. A subsequent comment: "*I think that the first occurrence of a NULL descriptor should define the end of it all. Not only should there be no zeroed descriptors in the middle, I do not think the log should be listed in any order. This means that the device can simply append more descriptors as it sees fit. Yes, that does fit into my comments for Note 5.*"

If the last Physical Element Status log page contains less than 512 bytes of valid Physical Element Status descriptors (i.e., nonzero value in the NUMBER OF LOGICAL BLOCKS field), then the remaining Physical Element Status descriptors in that Physical Element Status log page shall be padded with zero filled Physical Element Status descriptors

9.99.3 NUMBER OF LOG DESCRIPTORS field

The NUMBER OF LOG DESCRIPTORS field indicates the number of Physical Element Status descriptors in the Physical Element Status log. If the value of the NUMBER OF LOG DESCRIPTORS field is greater than or equal to tbd, then:

a) tbd.

There shall be no unused Physical Element Status descriptors (see 9.99.4) included in the range specified by the NUMBER OF LOG DESCRIPTORS field.

The number of Physical Element Status descriptors in the Physical Element Status log is vendor specific.

9.99.4 Physical Element Status descriptor format

Each Physical Element Status descriptor indicates the health status associated with a physical element or physical subelement. Table 7 defines the format of each Physical Element Status descriptor.

Table 7 — Physical Element Status descriptor format

Offset	Type	Description
0..1	Word	PHYSICAL ELEMENT field
2	Byte	PHYSICAL SUBELEMENT field
3	Byte	PHYSICAL ELEMENT TYPE field
4..6	Bytes	Reserved
7	Byte	PHYSICAL ELEMENT HEALTH field

The PHYSICAL ELEMENT field contains the index of the physical element associated with this Physical Element Status descriptor. Physical elements shall be identified by an index number ranging from zero to one less than the number of physical elements within the device.

The PHYSICAL SUBELEMENT field contains the index of the physical subelement within a physical element associated with this Physical Element Status descriptor. Physical subelements shall be identified by an index number ranging from zero to one less than the number of physical subelements within the enclosing physical element. If this physical element does not support any physical subelements, the physical subelement shall be set to zero.

Editor's Note 10: use cases (classes) of physical subelement might include: number of coupled heads (TDMR); or identifier of area of disk impacted by scratch. We need to address the field size for these and any other use cases.
Note that the idea that physical element might represent anything other than a subset of the media may be controversial.

Editor's Note 11: Are these fields appropriately sized? Subsequent comment: "*These fields are way too small. Each should be a 64 bit value for a few years of future-proofing.*"

Editor's Note 12: Need to define mapping between Rebuild Assist physical element bits and the physical element field

The PHYSICAL ELEMENT TYPE field indicates the type of the physical element associated with this Physical Element Status descriptor, as described in table 8.

Table 8 — physical element type

code	description
00h	not reported
01h	device
02h	head
03h	surface
04h	die
05h	channel
06h	armature
~~06h~~07h	spindle
~~07h~~08h - 7Fh	reserved
80h - FFh	vendor specific

The PHYSICAL ELEMENT HEALTH field provides an indication of the health of the physical element or physical subelement associated with this Physical Element Status descriptor, as described by table 9.

Editor's Note 13: The following text was recommended for removal, but I'm not sure the intent is made clear without it. "This value represents a rough normalized value, in relation to the vendor specific performance limit.The value is specified in proportion to a percentage of the manufacturer's specification limit. A value of 64h (i.e., 100%) indicates the limit of the manufacturer's specification. No scale is defined above or below this limit."

Table 9 — physical element health

code	description
00h	unspecified
01h to 63h[a]	within ~~(better than)~~ manufacturer's specification limit
64h	~~within but~~ at manufacturer's specification limit
65h to FEh [a.]	~~beyond (worse than)~~outside manufacturer's specification limit
FFh	depopulated

a. the device may implement a subset of values in a vendor-specific manner

Editor's Note 14: This allows go (e.g. 01h) / no-go (e.g., FEh) implementations

Editor's Note 15: Need to add IDENTIFY log bits for support & enable. Apr plenary had a wide-ranging discussion as to whether the Enable bit could be sticky across resets. This discussion spilled over into sticky enable for Sense Data Reporting.