Flex Device Interface

Protocol for field configuration of Flex Dynamic Recording drives, drives with media that can change recording method and capacity, with extensions to ZAC and ZBC

Revision 0; December 20, 2017

Editor: Timothy Feldman
Seagate Technology
389 Disc Drive
Longmont, CO 80503
USA
+1 720 684 2780
timothy.r.feldman@seagate.com

Table of Contents

1. Revision History **6**

2. Scope and Goals **7**

2.1. Goals 7

2.2. Scope 8

2.3. Style 8

2.4. Definitions 8

3. Flex Device Model **9**

3.1. Flex Logical Block Provisioning 9

3.1.1. Conversion between CMR and SMR 9

3.1.1.1. Determining the Space Modified by a Conversion 10

3.2. Zone Model 10

3.2.1. FMR zones 10

3.2.1.1. CMR space and FMR zone provisioning 10

3.2.1.2. Writing in FMR zones 11

3.2.1.2.1. Filling an FMR zone 11

3.2.1.3. Reading in FMR zones 12

3.2.1.4. Substitute data pattern for FMR zones 12

3.2.1.5. FMR zone state machine 12

3.2.1.5.1. Empty state 13

3.2.1.5.2. Implicitly Opened state 15

3.2.1.5.3. Full state 17

3.2.1.5.4. Offline state 18

3.2.1.6. FMR zone attributes 19

3.2.2. SMR zones 19

3.2.2.1. SMR space and SMR zone provisioning 19

3.2.2.2. SMR zone states and attributes 20

3.3. Operations 21

3.3.1. FMR zone operations 21

3.3.2. Flex Convert and Flex Query operations 22

3.3.2.1. Flex Convert operation overview 22

3.3.2.2. Flex Query operation overview 22

3.3.2.3. Extents for Flex Convert and Flex Query operations with All = false 22

3.3.2.4. Operation syntax 23

3.3.2.5. Operation prerequisites 24

3.3.2.6. Operation results 25

3.4. Other Operations 28

3.4.1. Operations on whole device or logical unit 28

3.4.2. Reset Write Pointer operations 29

4. ATA Additions and Modifications for Flex 30

4.1. New feature set 30

4.1.1. Flex feature set 30

4.1.1.1. Basic zoned device model 31

4.1.1.2. Zone attributes and types 31

4.2. New Commands 31

4.2.1. FLEX CONVERT NON-QUEUED, FLEX CONVERT QUEUED, FLEX QUERY NON-QUEUED and FLEX QUERY QUEUED commands 31

4.2.1.1. Feature set 31

4.2.1.2. Description 31

4.2.1.3. Inputs 32

4.2.1.4. Flex Convert and Flex Query operation invocation 34

4.2.1.5. Normal outputs 34

4.2.1.6. Error outputs 34

4.2.1.7. Input from the device to the host data structure 34

4.2.1.7.1. Extent descriptor format 36

4.2.2. Flex Control subcommand 37

4.3. New Logs 38

4.3.1. Flex Capabilities log 39

4.4. Extensions to ACS-4, SATA 3.3 and ZAC 40

4.4.1. Extensions to definitions, abbreviations, and conventions 41

4.4.2. Extensions to feature set definitions 42

4.4.3. Extensions to command descriptions 53

4.4.3.1. RECEIVE FPDMA QUEUED subcommands 53

4.4.3.1.1. SATA RECEIVE FPDMA QUEUED subcommands 53

4.4.3.1.2. ATA RECEIVE FPDMA QUEUED subcommands 54

4.4.3.2. Extensions to ZAC management out commands 54

4.4.3.3. Extensions to CLOSE ZONE EXT 55

4.4.3.4. Extensions to FINISH ZONE EXT 55

4.4.3.5. Extensions to OPEN ZONE EXT 55

4.4.3.6. Extensions to REPORT ZONES EXT 56

4.4.3.6.1. Support for a compact form of zone descriptors 56

4.4.3.7. Extensions to RESET WRITE POINTER EXT 56

4.4.4. Extensions to log definitions 58

4.4.4.1. Extensions to the Capacity log page 58

4.4.4.2. Extensions to the Supported Capabilities log page 59

5. SCSI Additions and Modifications for Flex 60

5.1. New Commands 60

5.1.1. FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) commands 60

5.1.1.1. Flex Convert and Flex Query operation invocation 63

5.1.1.2. Parameter data 63

5.2. New Parameters 66

5.2.1. Flex Control mode page 66

5.2.2. Flex Capabilities VPD page 68

5.3. Device Type Reporting 69

5.3.1. PERIPHERAL DEVICE TYPE field 69

5.3.2. ZONED field 69

5.3.3. Supported capabilities 69

5.4. Extensions to SBC-4 and ZBC 69

5.4.1. Extensions to definitions, symbols, abbreviations, and conventions 69

5.4.2. Extensions to zoned block device model 70

5.4.3. Extensions to commands 70

5.4.3.1. Extensions to REPORT ZONES 70

5.4.3.1.1. Support for a compact form of zone descriptors 70

5.4.3.1.2. Support for listing FMR zones and SMRmax 70

5.4.3.2. Extensions to RESET WRITE POINTER 71

5.4.3.3. Extensions to READ CAPACITY (16) 71

5.4.3.4. Extensions to zone management commands 72

5.4.4. Extensions to parameters for zoned block devices 72

5.4.4.1. Extensions to mode parameters 72

5.4.4.2. Extensions to vital product data (VPD) parameters 73

5.4.4.2.1. Extensions to Block Device Characteristics VPD page 73

6. SCSI to ATA Translation for Flex 75

6.1. FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) commands 75

6.1.1. Commands overview 75

6.1.2. Command processing 76

6.1.3. Parameter data 77

6.2. Flex Control Mode Page 79

6.3. Flex Capabilities VPD Page 80

6.4. Extensions to SAT-4 81

6.4.1. Extensions to parameters for SAT implementations 81

6.4.1.1. Extensions to vital product data parameters 81

6.4.1.1.1. Extensions to Block Device Characteristics VPD Page 81

7. Examples 82

7.1. Example Flex Drive 82

7.1.1. Example zone layout 82

7.1.2. Unprovisioned media at a seam 83

7.1.3. Example drive configurations 84

7.2. Example Conversions 85

7.2.1. Converting the first FMR zone to offline 85

7.2.2. Converting the second SMR zone to online 86

7.2.3. Converting an SMR zone to online without an FMR zone going offline 87

7.2.4. Converting the last SMR zone to online 88

7.2.5. Converting the highest possible 100 GiB of SMR space to online while leaving the last FMR zone online 88

7.2.6. Converting the last two SMR zones to CMR 91

7.2.7. Converting the next-to-last two SMR Zones 2 and 3 to CMR 91

7.3. Example Use of ATA PASS-THROUGH (16) 92

8. Appendix A: Capabilities Reporting 94

1. Revision History

Revision 0; December 20, 2017

2. Scope and Goals

There is a drive concept that has a mix of recording methods: partly non-shingled or conventional magnetic recording (CMR), and partly shingled magnetic recording (SMR). This was suggested in Google's white paper for FAST 2016, Disks for Data Centers, which refers to it as a hybrid drive.

One expression of a mixed recording method drive is to use different recording methods on different surfaces. A more compelling approach is to assign the recording method by radius. Flex allows the recording method to be assigned by zone, and thus is conducive to the latter approach.

Note that the term "hybrid drive" is overloaded (see "Hybrid drive" on Wikipedia) and this feature is extensible to dynamic recording methods beyond just CMR and SMR, and so the term Flex is coined for this feature.

2.1. Goals

This proposal is a protocol for field configuration of a block device that supports two recording methods. Specific consideration is given to CMR and SMR as the two recording methods and is written with that bias; however, the protocol can be used for other combinations of recording methods such as Flash in SLC and MLC mode. It is also extensible to other advances such as more than two recording methods, and media that dynamically becomes able to be provisioned.

This proposal includes the following qualities.

- Conversion events are explicit steps that are instigated by the host.
- The resulting amount of capacity has an *a priori* known resulting amount of CMR and SMR capacity.
- CMR and SMR media are accessed in separated LBA spaces, both spaces made up of zones.
- Logical block data retention is well defined and no logical block data are lost as a side effect of the conversion process.
    - Space to be taken offline must be part of an `Empty` zone.
    - Capacity that comes online is initially in the `Empty` state.
    - The onus for initializing writes ahead of random write access to CMR space is on the host so that the host can initialize with useful data. This avoids autonomous initialization with fill data that takes a long time. In many use cases autonomous initialization is useless as it is immediately followed with user data ingress from the host.
- Conversion and Query operation complexity is small and completion time can be short. The conversion command completion time should be on the order of 100 milliseconds to 1 second.
- After conversion to a given configuration, existing applications compatible with that configuration can use a Flex drive without any knowledge of Flex and without modification. A drive with 100% of its CMR capacity online and in the `Full` state can be used as a conventional drive. Similarly, a drive can be configured identically to other Host Managed devices, and with the zones in the CMR space in the `Full` state the drive can be used by unmodified Host Managed applications.
- Conversion and states are compatible with other disk drive features.

- Straightforward extension to recording methods other than CMR and SMR, and to more than two recording methods is possible.

2.2. Scope

This proposal is agnostic to the specific underlying recording methods and the physical constraints of surfaces and radial locations, but is inspired by CMR and SMR. The ability for the host to define the recording method in this Flex protocol is directly enabled by a new write pointer zone type, the `FMR` type which in turn is inspired by the existing write pointer zone types.

2.3. Style

Formal names that are used in the device interface -- field names, opcodes, enumerations, et cetera -- are formatted in a `monospace` font. In pseudocode, multiple-word, mixed-case formal names are written without embedded spaces.

[Editor's notes are in square brackets and green font.]

2.4. Definitions

extent   A consecutive set of zones is called an extent. An FMR extent is a consecutive set of FMR zones. An SMR extent is a consecutive set of SMR zones.

FMR zones   The zones that make up the CMR space are referred to as FMR zones. The ZAC and ZBC standards formally define conventional zones. Since Flex needs a different zone type, "conventional" and "CMR" are often used synonymously, and to avoid confusion, zones in CMR space are called FMR zones.

SMR zones   The sequential write preferred and sequential write required zones that make up the SMR space are referred to as SMR zones.

seam   A seam is a boundary between differing recording methods. Each seam has a corresponding pair of zones, one of which is online and the other offline, in each address space.

3. Flex Device Model

Flex drives allow conversion between recording methods in the field.

This section describes the Flex device model including how the zoned block device model of Zoned-device ATA Commands (ZAC) and Zoned Block Command (ZBC) is leveraged.

3.1. Flex Logical Block Provisioning

Flex devices support CMR and SMR recording methods exposed as CMR space and SMR space, respectively. The two spaces share physical media; thus, for a given portion of media, one space is provisioned and online and the other space is not provisioned and offline. The CMR space is organized as a set FMR zones and the SMR space as a set of SMR zones. Zones are the atomic unit of provisioning; that is, for each zone, that zone is either provisioned and online or not provisioned and offline. Both sets of zones are always extant in the full zone descriptor list. Zones boundaries are immutable.

There are two separate, non-overlapping logical block address spaces, one for the CMR space and one for SMR. Best practice is to have CMR space accessed at lower LBAs and SMR at upper LBAs. There may or may not be a reserved LBA range between the two spaces.

The Flex logical block provisioning model is different than the existing models, whether the normal full provisioning, thin provisioning, or data preserving depopulation. All zones that are online have media provisioned. To change offline zones to online requires that media provisioning changes. These conversions are requested by the host. At any given moment there are zones that are offline that may later change to online, and online zones that may later change to offline.

3.1.1. Conversion between CMR and SMR

A conversion is made up of the following sequence.

1. The host selects an extent to go offline, or an extent to come online. The extent can be in CMR or SMR space.
2. The host can determine the corresponding extent by using the `Flex Query` operation. This operation returns the extents that would be modified by this specified hypothetical conversion: the extents that would go offline and come online.
3. The host reads from the extent to go offline as necessary to copy valid data to another location.
4. The host explicitly discards or trims the data in the extent to go offline by issuing a `Reset Write Pointer` operation to this extent.
5. The host issues a command to invoke the conversion operation, `Flex Convert`, specifying either the extent to go offline or come online.

As a consequence of this sequence:

- The extent that goes offline changed to the `Empty` state and then to the `Offline` state, and no media is provisioned for these zones.

- The extent that comes online changed from the `Offline` state to the `Empty` state, and media is provisioned for these zones.

Having newly provisioned space come online in the `Empty` state frees the device from initializing the media, resulting in an efficient conversion process. The host is in full control of the data written to these locations, for instance so it can write user data instead of an initialization pattern to CMR space. Additionally, the host can determine when to perform those writes to minimize impact on other operations.

For conversion from SMR to CMR, next steps include filling the FMR extent that came online so that it becomes randomly writable.

3.1.1.1. Determining the Space Modified by a Conversion

A host can determine the extent that will be modified ahead of an actual conversion by issuing a command that invokes the `Flex Query` operation. The command returns a list of descriptors of the extents that would change state due to a specified hypothetical conversion based on the state of the drive when the query is processed.

3.2. Zone Model

Extensions to the zone model in ZAC and ZBC include:

- `FMR` as a new zone type that is distinct from the conventional (non-write pointer) zone type.
- Device type with one or more FMR zones and one or more SMR zones.
- The `Flex Convert` and `Flex Query` operations.

This specification requires that zones in the `Offline` state can be converted and the Flex Convert operation is the only way for a zone to transition into and out of the `Offline` state. Any other feature that uses an offline state either requires a unique state, or the use of the `Offline` state has to resolve all issues with the Flex feature.

3.2.1. FMR zones

The FMR zone type is a write pointer zone type. This section specifies the model, states and conditions for the FMR zone type.

3.2.1.1. CMR space and FMR zone provisioning

Flex drives have a set of FMR zones in which each LBA in CMR space is a member of exactly one FMR zone. FMR zones are provisioned during device manufacturing or by a conversion from SMR to CMR.

Best practice is for all FMR zones to be the same size; 256 MiB is the industry *de facto* standard for the zone size.

CMR space spans LBA `0` to `FMRmax`
where `FMRmax+1` is the capacity in logical blocks for the 100% CMR configuration.

Note that some markets may require the 100% CMR capacity to conform to SFF-8447.

3.2.1.2. Writing in FMR zones

FMR zones embrace a few fundamental goals for the behavior of write operations.

1. FMR zones in the `Full` state behave like conventional space: there are no alignment constraints.
2. FMR zones in the `Empty` or `Implicitly Opened` state behave like conventional space below the write pointer, and like a Sequential Write Required zone at and above the write pointer.
    a. Writes may not start above an FMR zone's write pointer.
    b. Writes that end at or above an FMR zone's write pointer must end at a physical sector boundary or write the last LBA of this zone.
    c. Writes below an FMR zone's write pointer do not have alignment constraints.
3. Writes may cross an FMR zone boundary if they do not violate any other constraint.

For example, a write to an `Empty` zone may:

- start at this zone's write pointer and end at a physical sector boundary; or
- start at this zone's write pointer and write the last LBA of this zone.

For example, a write to an `Implicitly Opened` zone may:

- start below this zone's write pointer and end below this zone's write pointer;
- start below this zone's write pointer and end at or above this zone's write pointer at a physical sector boundary;
- start below this zone's write pointer and write the last LBA of this zone;
- start at this zone's write pointer and end above this zone's write pointer at a physical sector boundary; or
- start at this zone's write pointer and write the last LBA of this zone.

For a write command:

```
    if any LBAs specified by this command are in an FRM zone
        if (the lowest LBA that is written in this zone is greater than this zone's write pointer)
            then the write command is rejected
        else if (the highest LBA that is written in this zone is greater than or equal to this zone's
                write pointer) and (the highest LBA that is written in this zone is not equal
                to the last logical sector within a physical sector) and (the highest LBA
                that is written in this zone is not the last LBA of this zone) then
            the write command is rejected
        else
            the write command should succeed
            if the highest LBA of this zone is written
                then this zone transitions to the Full state and this zone's write pointer
                        becomes invalid
            else
                this zone's write pointer is incremented by the number of logical blocks
                        written in this zone
```

3.2.1.2.1. Filling an FMR zone

When an FMR zone transitions to the `Empty` state, either due to a conversion or `Reset Write Pointer` operation, this zone is not randomly writable and the user data becomes inaccessible for read operations with URFZ = 0 or changes to all zeros for read operations with URFZ = 1. To become populated with meaningful data and to be randomly writable the host has to initialize the sectors of this zone using write operations. The host has the prerogative of using an fill pattern of its choice, or filling the zone with meaningful data. When an FMR zone is partially initialized and is in the Implicitly Opened state written logical blocks can be fetched by a random read and updated by a random write with the same performance as accesses to a full FMR zone; that is, fetches and updates of written data do not have to be held off until the zone is full. Similarly, if the data are being copied from another zone on this drive or any other drive, the source can immediately be retired in favor of the data in this zone as updates can go to this zone instead of to the previous source; thus, issues of coherence between the previous source and this zone can be avoided.

After all sectors have been written, the zone is fully initialized, transitions to the Full state and supports random reads and writes as with a conventional zone.

Initializing writes or writes at the write pointer can be issued to FMR zones in any zone order; that is, when multiple FMR zones are online and in the Empty state or Implicitly Opened state, these zones do not have to be filled or written in a predetermined order. This means that different FMR zones can be allocated to different asynchronous processes without any need to synchronize the filling of the different FMR zones.

3.2.1.3. Reading in FMR zones

A read to an LBA below the write pointer in an FMR zone should succeed. A read to an LBA at or above the write pointer fails if the value of the URFZ bit is zero, or returns the device's substitute data pattern if the value of the URFZ bit is one.

```
For a read command:
    if any LBAs specified by this command are in an FMR zone
        if (the highest LBA that is read in this zone is greater than or equal to this zone's write
                pointer) and (URFZ = 0) then
            the read command is rejected
        else
            the read command should succeed
```

3.2.1.4. Substitute data pattern for FMR zones

For FMR zones, the substitute data pattern is the same as the substitute data pattern for Sequential Write Required zones (see ZAC and ZBC).

3.2.1.5. FMR zone state machine

Each FMR zone is in either:

- online and in either the
  - Empty state,
  - Full state, or
  - Implicitly Opened state;

or

- offline and in the Offline state.

FMR zones do not support the following states.

- The Closed and Explicitly Opened states are not supported.

The following figure shows a simplified state diagram for FMR zones including state transitions due to conversions, Reset Write Pointer operations, and writes.

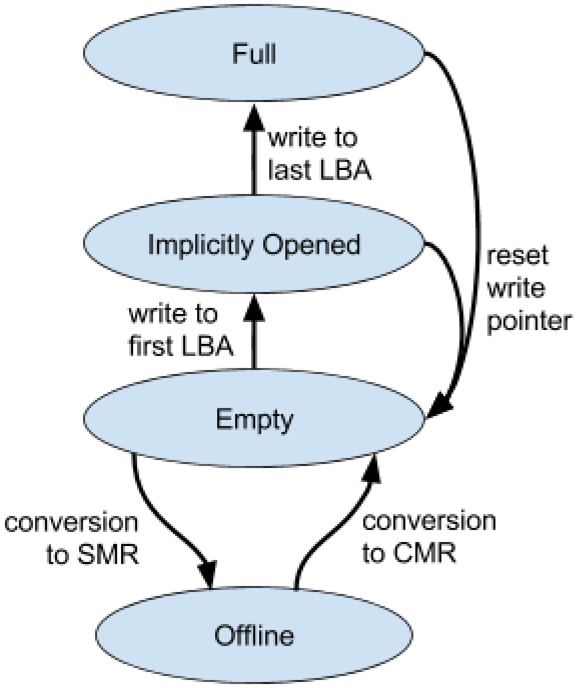


Figure 1: FMR Zone State Diagram

FMR zone states do not change over a power cycle or reset. This is in contrast to sequential write preferred and sequential write required zones that change from the Explicitly Opened state or Implicitly Opened state to the Closed state.

3.2.1.5.1. Empty state

An FMR zone in the Empty state is online, provisioned and empty. There is no limit to the number of empty FMR zones; all FMR zones can be concurrently in the Empty state.

Writes that start above the write pointer of a zone in the Empty state are disallowed so that devices can apply this simple rule for servicing reads, and be relieved of having to track the initialization state of each logical block individually.

The Empty state for an FMR zone indicates:

- This zone has processed a Reset Write Pointer operation or a conversion to CMR.
- This zone has not processed a write to its first LBA since transitioning to the Empty state.
- This zone has not processed a conversion to SMR since transitioning to the Empty state.

For an FMR zone in the Empty state:

- This zone is online and provisioned.
- This zone has a valid write pointer value that is equal to this zone’s start LBA.
- For a read operation to an LBA in this zone:

  ```
  if URFZ = 0 then
      the read is rejected
  else
      the read should succeed using the substitute data pattern for FMR zones
  ```

- For a write operation to an LBA in this zone:

  ```
  if the lowest LBA written in this zone is not the lowest LBA of this zone
      the write is rejected
  else if (the highest LBA that is written in this zone is not equal to the last logical sector
          within a physical sector) and (the highest LBA that is written in this zone
          is not the last LBA of this zone) then
      the write command is rejected
  else
      the write should succeed
  if the last LBA of the zone is written then
      this zone’s write pointer becomes invalid
      this zone transitions to the Full state
  else
      this zone’s write pointer becomes equal to the LBA following the last LBA written
      this zone transitions to the Implicitly Opened state
  ```

- A Reset Write Pointer operation to this zone should succeed.
- A Flex Convert operation in the CMR to SMR direction to this zone with all prerequisites met should cause this zone to transition to the Offline state.

To transition into the Empty state:

- A Flex Convert operation in the SMR to CMR direction on an FMR zone with all prerequisites met causes the zone to transition to the Empty state.
- A successful Reset Write Pointer operation on an FMR zone causes the zone to transition to the Empty state.

To transition out of the Empty state:

- A Flex Convert operation in the CMR to SMR direction on an FMR zone with all prerequisites met causes the FMR zone to transition to the Offline state.
- A successful write operation to an FMR zone in the Empty state causes the zone to transition to the Implicitly Opened state.

The following figure shows the read and write rules of an FMR zone in the Empty state.

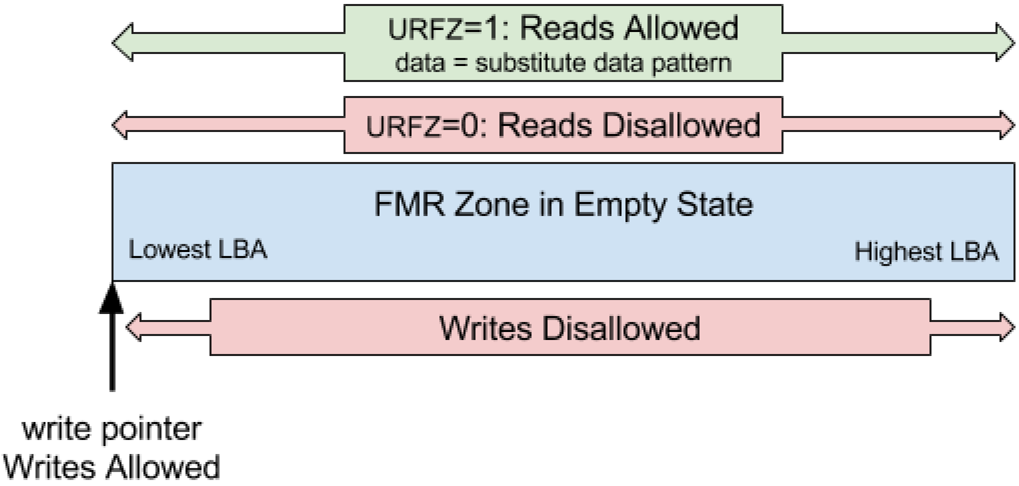


Figure 2: Read and write rules for an FMR zone in the Empty state

3.2.1.5.2. Implicitly Opened state

An FMR zone in the `Implicitly Opened` state is online, provisioned, open and partially filled. There is no limit to the number of open FMR zones; all FMR zones can be concurrently in the `Implicitly Opened` state.

Writes that start above this zone's write pointer are disallowed so that devices can apply this simple rule for servicing reads, and be relieved of having to track the initialization state of each logical block individually.

The `Implicitly Opened` state for an FMR zone indicates:

- This zone has processed a write to the first LBA of this zone subsequent to the last time this zone was in the `Empty` state.
- This zone has not processed a write to the last LBA of this zone since transitioning to the `Implicitly Opened` state.
- This zone has not processed a `Reset Write Pointer` operation since transitioning to the `Implicitly Opened` state.

For an FMR zone in the `Implicitly Opened` state:

- This zone is online and provisioned.
- This zone has a valid write pointer that is equal to the LBA following the largest LBA of the zone written since it transitioned into this state.
- For a read operation to an LBA in this zone:

```
if the highest LBA read in this zone is below this zone's write pointer then
    the read should succeed
else
    if URFZ = 0 then
        the read is rejected
    else
        the read should succeed using the substitute data pattern for FMR zones
                for LBAs at or above the write pointer
```

- For a write operation to an LBA in this zone:

```
        if the lowest LBA written is not at or below this zone's write pointer
                the write is rejected
        else if (the highest LBA that is written in this zone is greater than or equal to this zone's
                        write pointer) and (the highest LBA that is written in this zone is not equal
                        to the last logical sector within a physical sector) and (the highest LBA
                        that is written in this zone is not the last LBA of this zone) then
                the write command is rejected
        else
                the write should succeed
        if the last LBA of the zone is written then
                this zone's write pointer becomes invalid
                this zone transitions to the Full state
        else
                this zone's write pointer becomes equal to the LBA following the last LBA written
```

- A `Reset Write Pointer` operation to this zone should succeed and causes this zone to transition to the `Empty` state.
- This zone may not be converted to offline.

To transition into the `Implicitly Opened` state:

- A successful write operation to an FMR zone in the `Empty` state causes the zone to transition to the `Implicitly Opened` state.

To transition out of the `Implicitly Opened` state:

- A successful write operation to the ending LBA of an FMR zone in the `Implicitly Opened` state will cause the zone to transition to the `Full` state.
- A successful `Reset Write Pointer` operation on an FMR zone in the `Implicitly Opened` state causes this zone to transition to the `Empty` state.

The following figure shows the read and write rules of an FMR zone in the `Implictly Opened` state.

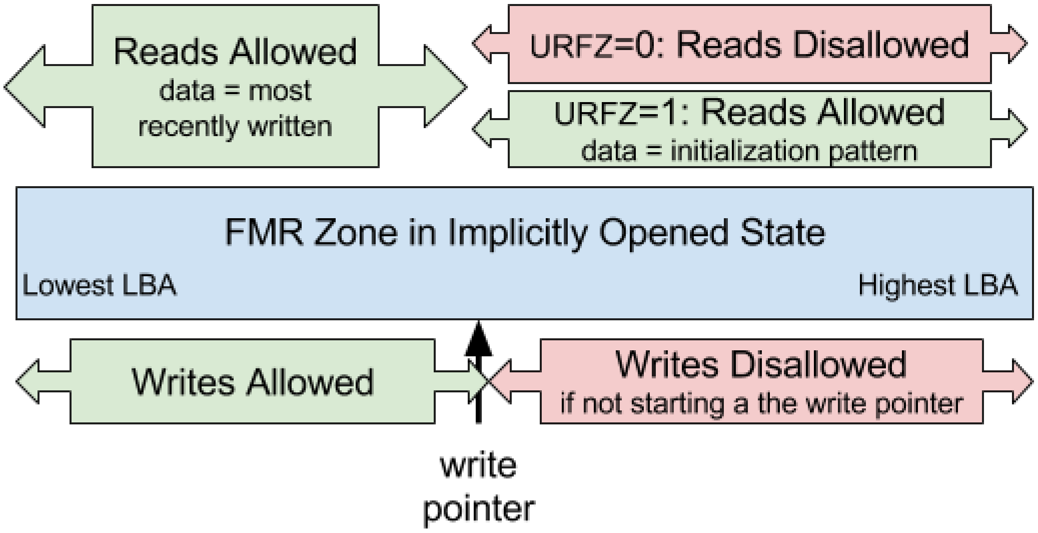

Reads Allowed
data = most recently written
URFZ=0: Reads Disallowed
URFZ=1: Reads Allowed
data = initialization pattern
FMR Zone in Implicitly Opened State
Lowest LBA
Highest LBA
Writes Allowed
Writes Disallowed
if not starting a the write pointer
write pointer

Figure 3: Read and write rules for an FMR zone in the Implicitly Opened state

3.2.1.5.3. Full state

An FMR zone in the Full state is online, provisioned and full. There is no limit to the number of full FMR zones; all FMR zones can be concurrently in the Full state.

An FMR zone in the Full state is similar to a conventional zone with the addition of support for Flex Convert, Flex Query and Reset Write Pointer operations.

The Full state for an FMR zone indicates that this zone:

- This zone has processed a successful write to the ending LBA of this zone.
- This zone has not processed a successful Reset Write Pointer operation since transitioning to the Full state.

For an FMR zone in the Full state:

- This zone is online and provisioned.
- This zone does not have a valid write pointer.
- A read operation to an LBA in this zone should succeed.
- A write operation to an LBA in this zone should succeed.
- A Reset Write Pointer operation to this zone should succeed and causes this zone to change to the Empty state.
- This zone cannot be converted to offline by a Flex Convert operation.

To transition into the Full state:

- A successful write operation to the ending LBA of an FMR zone in the Implicitly Opened state causes this zone to transition to the Full state.

To transition out of the Full state:

- A successful Reset Write Pointer operation on an FMR zone in the Full state causes this zone to transition to the Empty state.

The following figure shows the read and write rules of an FMR zone in the Full state.

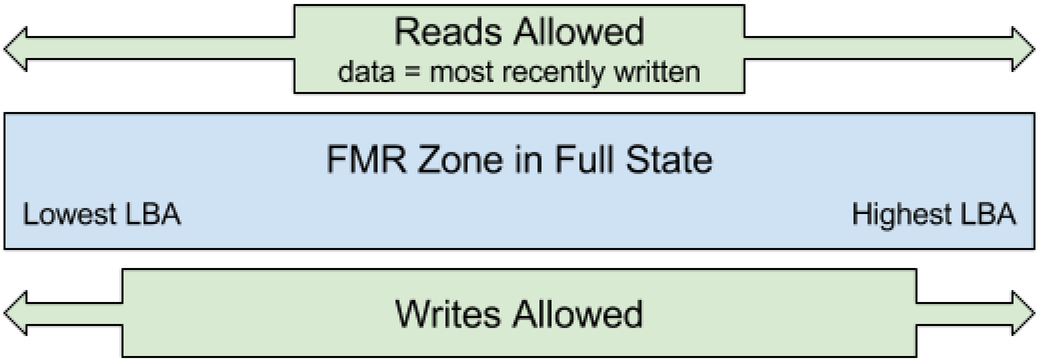


Figure 4: Read and write rules for an FMR zone in the Full state

3.2.1.5.4. Offline state

An FMR zone in the `Offline` state is offline and not provisioned. There is no limit to the number of offline FMR zones; all FMR zones can be concurrently in the `Offline` state.

The `Offline` state for an FMR zone indicates:

- This zone has processed a conversion to SMR.
- This zone has not processed a subsequent conversion to FMR.

For an FMR zone in the `Offline` state:

- This zone is offline and not provisioned.
- A read operation to an LBA in this zone is rejected.
- A write operation to an LBA in this zone is rejected.
- A `Reset Write Pointer` operation to this zone is rejected.
- This zone can transition to the `Empty` state due to a `Flex Convert` operation unless prohibited by reasons outside the scope of this protocol.

To transition into the `Offline` state:

- A `Flex Convert` operation in the CMR to SMR direction on an FMR zone with all prerequisites met causes the FMR zone to transition to the `Offline` state.
- Operations outside the scope of this protocol may cause an FMR zone to transition to the `Offline` state.

To transition out of the `Offline` state:

- A `Flex Convert` operation in the SMR to CMR direction on an FMR zone with all prerequisites met causes the zone to transition to the `Empty` state.

The following figure shows the read and write rules of an FMR zone in the `Offline` state.

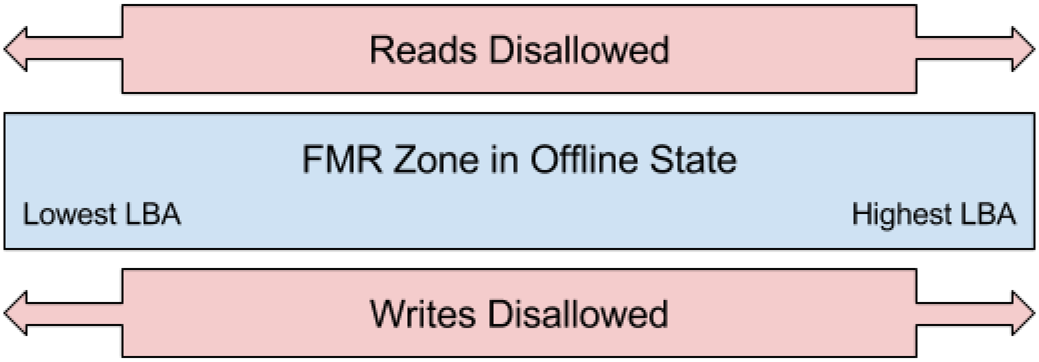


Figure 5: Read and write rules for an FMR zone in the Offline state

3.2.1.6. FMR zone attributes

FMR zones support the following attributes.

- `Zone Type`, `Zone Condition` and `WPointer` zone attributes are supported by FMR zones.

FMR zones do not support the following attributes.

- `RWPRecommended` and `Non-Sequential Write Resources Active` zone attributes are not supported by FMR zones.

3.2.2. SMR zones

SMR zones describe the LBA space used to access media using the SMR recording method. SMR zones can be either sequential write preferred zones or sequential write required zones. Refer to T10 ZBC and T13 ZAC standards for the specification of these two zone types.

The standards for sequential write preferred and sequential write required zones are recording method agnostic. In this document they are described as being provisioned by SMR media.

3.2.2.1. SMR space and SMR zone provisioning

Flex drives have a set of SMR zones in which each SMR LBA is a member of exactly one SMR zone. SMR zones are provisioned during device manufacturing or by a conversion from CMR to SMR. Best practice is for all SMR zones to be the same size; 256 MiB is the industry *de facto* standard for the zone size.

SMR zones span LBA `SMRmin` to `SMRmax` where:
`SMRmin` is larger than `FMRmax`; and
`SMRmax+1-SMRmin` is the capacity for the 100% SMR configuration.

Best practice is for `SMRmin` to be an integer multiple of the SMR zone size; that is, the starting LBA of SMR zones should be zone-size aligned.

[Editor’s note: This best practice may result in `SMRmin>FMRmax+1` and, thus, LBAs between `FMRmax` and `SMRmin` that are not part of any zone. For example, a device with a 100% CMR capacity that is not larger than 8 TB and compliant with SFF-8447 has a value of `FMRmax`+1 that is not an integer multiple of 256 MiB, and, with 256-MiB SMR zones, `SMRmin` would be larger than `FMRmax+1`.]

Best practice is for `SMRmin` to be the smallest integer power of 2 that is larger than the 100% SMR capacity so that a single bit in every LBA indicates whether the logical block is in CMR or SMR space.

[Editor's note: This best practice may result in `SMRmin>FMRmax+1` and, thus, LBAs between `FMRmax` and `SMRmin` that are not part of any zone.]

3.2.2.2. SMR zone states and attributes

SMR zones are as defined by ZAC and ZBC. These standards fully defines their states and attributes.

Each SMR zone is either:

- online and in either the
  - `Closed` state,
  - `Empty` state,
  - `Explicitly Opened` state,
  - `Implicitly Opened` state,
  - `Full` state; or
- offline and in the `Offline` state.

The following figure shows a simplified state diagram for SMR zones including state transitions due to conversions, `Reset Write Pointer` operations and writes. The `Closed` and `Explicitly Opened` states are not included in this simplified diagram.

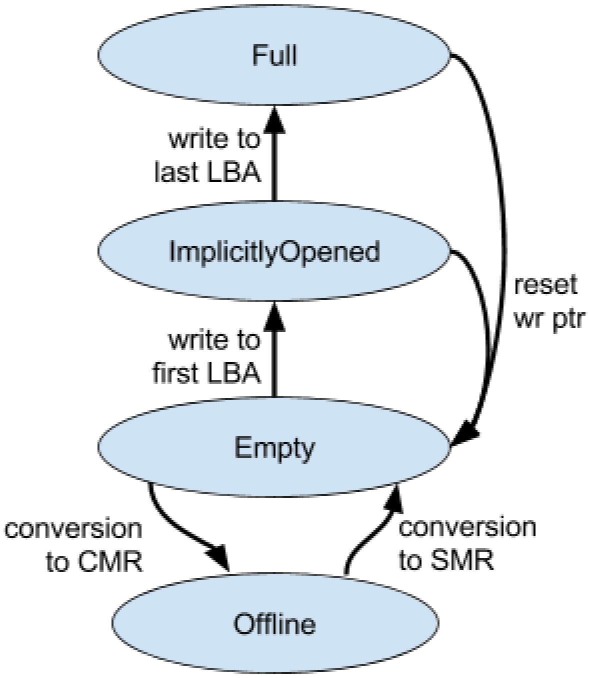


Figure 6: SMR Zone State Diagram

Online SMR zones typically cycle through the `Empty` state, `Explicitly Opened` or `Implicitly Opened` state, and `Full` state as zones are allocated, filled, returned to a free pool, and re-allocated. Some systems also use the `Closed` state.

As a result of conversion from CMR to SMR, SMR zones that come online transition from the `Offline` state to the `Empty` state.

3.3. Operations

3.3.1. FMR zone operations

FMR zones support the following operations.

- `Reset Write Pointer` operations are supported.
- `Flex Query` and `Flex Convert` operations are supported.

FMR zones do not support the following operations.

- `Open`, `Close` and `Finish` operations are not supported.

3.3.2. Flex Convert and Flex Query operations

3.3.2.1. Flex Convert operation overview

The `Flex Convert` operation causes an actual conversion between recording methods if the prerequisites are met. The operation has input parameters that specify an extent for conversion. The operation returns data that indicate the results of the actual or hypothetical conversion.

The `Flex Convert` operation is atomic. For the zones modified by the specified conversion, either all of or none of these zones change state.

`Flex Convert` operations depend on the state of the drive and may change the state of the drive, so these operations are serviced in the order received with respect to other `Flex Convert` and `Flex Query` operations, and with respect to overlapping read and write operations. As an example, if a `Flex Convert` operation is received that will change a zone to offline and read and write operations are subsequently issued to that zone, then the `Flex Convert` operation is serviced first and the read and write operations subsequently fail because they address offline space.

The `Flex Convert` operation is invoked by the ATA commands `FLEX CONVERT NON-QUEUED` and `FLEX CONVERT QUEUED` (see 4.2.1), and by the SCSI commands `FLEX CONVERT (16)` and `FLEX CONVERT (32)` (see 5.1.1).

3.3.2.2. Flex Query operation overview

The `Flex Query` operation exposes the results of a hypothetical conversion if a `Flex Convert` operation with the same parameters were issued given the current state of the drive. The operation has input parameters that specify a hypothetical conversion. The operation returns data that indicate the results of the hypothetical conversion.

`Flex Query` operations do not change the state of the drive, but the returned data do depend on the current state of the drive, so these operations shall be serviced in the order received with respect to other `Flex Convert` and `Flex Query` operations.

The `Flex Query` operation is invoked by the ATA commands `FLEX QUERY NON-QUEUED` and `FLEX QUERY QUEUED` (see 4.2.1), and by the SCSI commands `FLEX QUERY (16)` and `FLEX QUERY (32)` (see 5.1.1).

3.3.2.3. Extents for Flex Convert and Flex Query operations with All = false

Each `Flex Convert` and `Flex Query` operation with the operation parameter `All = false` has a “from extent” that is the extent to actually or hypothetically go offline, and a “to extent” that is the extent to actually or hypothetically come online. One extent is specified and the other extent is implied.

The `Starting Zone Locator` and `Number Of Zones` parameters specify one extent. The specified extent is in the space in which the `Starting Zone Locator` resides. The implied extent is in the other space.

Table 1 -- Roles of Flex Convert and Flex Query Extents

Location of the `Starting Zone Locator` parameter	Value of the `Direction` operation parameter	Role of the specified extent	Role of the implied extent
CMR space	`From Cmr To Smr`	“From extent” that goes offline	“To extent” that comes online
SMR space	`From Cmr To Smr`	“To extent” that comes online	“From extent” that goes offline

If the implied extent is the “from extent” then its size is as few zones as possible; that is, a minimum number of zones go offline and a minimum amount of media is unprovisioned.

In some cases the implied extent may be a zero-sized extent such as when the implied extent is the “from extent” and there is media available to bring the specified “to extent” online without unprovisioning any zones. Conversely, if the implied extent is the “to extent” then it is a many zones as possible; that is, a maximum number of zones come online and a maximum amount of media is provisioned.

For a `Flex Convert` operation, the implied extent can be discovered

- before the actual conversion occurs by issuing a `Flex Query` operation and examining the returned data of this operation;
- after the actual conversion occurs by examining the returned data from the `Flex Convert` operation; or
- after the actual conversion occurs by examining the new zone states returned by a subsequent `Report Zones` operation.

The specified extent is not modified to include a larger number of zones even if there is media available to bring more zones online. For example, if the specified extent is in SMR space and `Direction = From Cmr To Smr`, the implied FMR extent might represent media that allows additional SMR zones to be in the SMR extent. In this case the SMR extent is not expanded.

If `Direction` = `From Cmr To Smr` then the FMR extent actually or hypothetically goes offline and the SMR extent actually or hypothetically comes online. Conversely, if `Direction` = `From Smr To Cmr` then the SMR extent actually or hypothetically goes offline and the FMR extent actually or hypothetically comes online.

3.3.2.4. Operation syntax

Syntax:
```
FlexConvert(Direction, All, StartingZoneLocator, NumberOfZones,
        SmrZoneType)
FlexQuery(Direction, All, StartingZoneLocator, NumberOfZones,
        SmrZoneType)
```
where

`Direction` specifies the direction of the conversion.

If `Direction = From Cmr To Smr`

- the direction is from the CMR to SMR recording method

- Smr Zone Type specifies the SMR zone type

If Direction = From Smr To Cmr

- the direction is from the SMR to CMR recording method
- Smr Zone Type is ignored

All specifies the source of the specified extent.

If All = false

- Starting Zone Locator specifies the lowest LBA of the first zone of the specified extent.
- Number Of Zones specifies the number of zones of the specified extent

If All = true

- All of the online zones in the "from" direction are the specified extents
- Starting Zone Locator is ignored
- Number Of Zones is ignored

Smr Zone Type specifies the type of the SMR zones.

Smr Zone Type = Sequential Write Preferred specifies that the SMR zones that come online are Sequential Write Preferred zones.

Smr Zone Type = Sequential Write Required specifies that the SMR zones that come online are Sequential Write Required zones.

[Editor's note: The All operation parameter indicates that the conversion is for transitioning any offline zones of the selected recording method to online. One use case is in repurposing a drive in which the drive needs to be converted to 100% SMR. In this case the All operation parameter allows this conversion in a single Flex Convert operation even from configurations with multiple seams.]

3.3.2.5. Operation prerequisites

If All = false

- Starting Zone Locator is equal to the lowest LBA of a zone.
- Number Of Zones is greater than zero.
- The specified extent does not include more than one zone type, and does not include LBAs that are not mapped to any zone.
- The zones in the "from extent" are in the Empty state state.
- The zones in the "to extent" are in the Offline state.
- If TCG is supported, the zones in both extents are part of TCG Band 0 and that band is not locked.

If All = true

- At least one zone in the "from" space is online.
- All online zones in the "from" space are in the Empty state.
- If TCG is supported, the online zones in the "from" space and the offline zones in the "to" space are part of TCG Band 0 and that band is not locked.

Table 2 -- Zone State Prerequisites for Flex Convert and Flex Query Operations

`Direction`	`All`	CMR space	SMR space
`From Cmr To Smr`	`false`	All zones in the FMR extent are in the `Empty` state	All zones in the SMR extent are in the `Offline` state
`From Cmr To Smr`	`true`	At least one zone in the CMR space is online, and all online FMR zones are in the `Empty` state	No zone state prerequisites for the SMR space
`From Smr To Cmr`	`false`	All zones in the FMR extent are in the `Offline` state	All zones in the SMR extent are in the `Empty` state
`From Smr To Cmr`	`true`	No zone state prerequisites for the SMR space	At least one zone in the SMR space is online, and all online SMR zones are in the `Empty` state

3.3.2.6. Operation results

The `Flex Convert` and `Flex Query` operations return good status.

State changes:

For `Flex Convert` and all prerequisites are met:

If `Direction=FromCmrToSmr:`

the zones in the FMR extent are in the `Offline` state and media previously provisioned for these zones are no longer provisioned to CMR space; and

the zones in the SMR extent are online, in the `Empty` state, and media is provisioned for these zones.

If `Direction=FromSmrToCmr:`

the zones in the SMR extent are in the `Offline` state and media previously provisioned for these zones are no longer provisioned to SMR space; and

the zones in the FMR extent are online, in the `Empty` state, and media is provisioned for these zones.

The states and logical block data of all other zones are unchanged.

For `Flex Query`, and `Flex Convert` and not all prerequisites are met:

The states and logical block data of all zones are not changed.

Returned data:

The following is a set of status bits. Every bit is set according to its definition; that is multiple bits are set when multiple conditions are met. Status bits apply to both `Flex Convert` and `Flex Query` operations. Their applicability to `Flex Query` is to provide a method for the host to discover if a subsequent `Flex Convert` would not actually have a conversion occur due to the prerequisites.

`SZLBV` (Starting Zone Locator Boundary Violation) indicates `Starting Zone Locator` is not the lowest LBA of a zone and `All = false`. `Extent Descriptors` are not returned if this bit equals true.

`NOZIZ` (Number Of Zones Is Zero) indicates `Number Of Zones` is zero and `All = false`. `Extent Descriptors` are not returned if this bit equals true.

`SEHMZT` (Specified Extent Has Multiple Zone Types) indicates the extent specified by `Starting Zone Locator` and `Number Of Zones` includes more than one zone type and `All = false`. `Extent Descriptors` are not returned if this bit equals true.

`FZSV` (FMR Zone State Violation) indicates an FMR zone is not in a valid state for conversion.

If `Direction = From Cmr To Smr` and `All = false`, then

`FZSV = true` indicates a zone in the FMR extent is in a state other than the `Empty` state.

If `Direction = From Cmr To Smr` and `All = true`, then

`FZSV = true` indicates

no zone in the CMR space is in the `Empty` state; or

some zone in the CMR space is in the `Implicitly Opened` state or the `Full` state.

If `Direction = From Smr To Cmr` and `All = false`, then

`FZSV = true` indicates a zone in the FMR extent is in a state other than the `Offline` state.

`FZTBV` (FMR Zone TCG Band Violation) indicates, for `Flex Convert`, a zone in the FMR extent is not in `TCG Band 0`.

`FZTBZL` (FMR Zone TCG Band Zero Locked) indicates, for `Flex Convert`, a zone in the FMR extent is in `TCG Band 0` and that band is locked.

`SZSV` (SMR Zone State Violation) indicates an SMR zone is not in a valid state for conversion.

If `Direction = From Smr To Cmr` and `All = false`, then

`SZSV = true` indicates a zone in the SMR extent is in a state other than the `Empty` state.

If `Direction = From Smr To Cmr` and `All = true`, then

`SZSV = true` indicates

no zone in the SMR space is in the `Empty` state; or

some zone in the SMR space is in the `Closed` state, the `Explicitly Opened` state, the `Implicitly Opened` state or the `Full` state .

If `Direction = From Cmr To Smr` and `All = false`, then

`SZSV = true` indicates a zone in the SMR extent is in a state other than the `Offline` state.

SZTBV (SMR Zone TCG Band Violation) indicates, for Flex Convert, a zone in the SMR extent is not in TCG Band 0.

SZTBZL (SMR Zone TCG Band Zero Locked) indicates, for Flex Convert, a zone in the SMR extent is in TCG Band 0 and that band is locked.

Converted indicates that a conversion actually occurred.

Direction is the same as Direction in the request.

All is the same as All in the request. Extent Descriptors are not returned if this bit equals true.

Number Of Zones is the same as Number Of Zones in the request.

Smr Zone Type is the same as Smr Zone Type in the request.

[Editor's notes:

The Direction, All, Number Of Zones and Smr Zone Type fields are included as copies of the request parameters for the convenience of the host. This information along with the extent descriptors that describe extents that went offline may be sufficient for the response processing to understand what the request was without requiring additional data such as a request handle.

One request parameter that is not echoed in the returned data is Starting Zone Locator. This is a large field, up to 8 bytes in the 32-byte SCSI CDBs, but there is reserved space to add an 8-byte aligned field without moving any other field.

If the Starting Zone Locator field is allowed to specify any LBA of the starting zone and is not required to specify the first LBA of a zone, returning the first LBA of the starting zone may be very valuable.

Other convenient values include the number of zones that change from online to offline, and the number that change from offline to online. This could be particularly useful with All = true.]

Extent Descriptors is a list of descriptors that indicate the extents that were or would be modified by the hypothetical conversion. Extent Descriptors are returned in all cases other than those explicitly specified as not returning them. That is, an operation that returns SZLBV, NOZIZ, SEHMZT and All set to false returns Extent Descriptors, and if Converted is set to false then the Extent Descriptors describe a hypothetical conversion.

There is a separate Extent Descriptor for each extent that differs by type or condition; thus, there may be more than one Extent Descriptor for a given space. There are one or more descriptors that indicate the new zone conditions of the zones in the specified extent, and zero one or more descriptors that indicate the new zone conditions of the the corresponding extent.

Each Extent Descriptor has the following fields.

`Zone Type` indicates the type of all of the zones in the extent.

`Zone Condition` indicates the condition of all of the zones in the extent. This is the condition that is or would be the resulting condition of the zones for the actual or hypothetical conversion; that is, `Offline` for zones in the space that did or would go offline, and `Empty` for zones in the space that did or would come online.

`Starting Zone Locator` indicates the lowest LBA of the first zone of the extent.

`Number Of Zones` indicates the size of the extent in zones.

[Editor’s note: The `Flex Convert` and `Flex Query` operations return a list of extents, and not just a single extent, for multiple reasons.

A) All of the modified extents are listed, so there is typically at least two extents: one going offline and one coming online.
B) Some zones may stay offline due to other reasons, so the space coming online may or may not be described by more than one extent.
C) Some recording methods not described in this document may have more than one LBA range affected by a conversion.]

3.4. Other Operations

3.4.1. Operations on whole device or logical unit

The following operations (commands) operate on all online zones.

- Format (SCSI `FORMAT UNIT`)

3.4.2. Reset Write Pointer operations

Reset Write Pointer commands should succeed if all invocations of the Reset Write Pointer function succeeds. The Reset Write Pointer function behaves as described in the following table.

Table 3 -- Reset Write Pointer function behavior

	Single zone specified by the command	Zone range specified by the command	Command specifies all zones (ALL=1)
SMR zone(s), online and not write-locked	The zone transitions to the Empty state	The zones transition to the Empty state	The zone(s) transition to the Empty state
FMR zone(s), online and not write-locked	The zone transitions to the Empty state	The zones transition to the Empty state	The Reset Write Pointer function is not performed on FMR zones
Offline zone(s)	Command fails	The Reset Write Pointer function is not performed on offline zones	The Reset Write Pointer function is not performed on offline zones
Write-locked zone(s)	Command fails	The Reset Write Pointer function is not performed on write-locked zones	The Reset Write Pointer function is not performed on write-locked zones
Atomicity and monotonicity	The zone is either unchanged, or transitions to the Empty state	The range is processed in order and each zone that is processed transitions to the Empty state. If a zone cannot be processed successfully, no subsequent zones are processed and the command fails.	Each zone is either unchanged or transitions to the Empty state. Processing order is not specified.
Ordering with respect to TCG operations	The zone is processed at a single intermediate TCG state	All zones are processed at the same intermediate TCG state	All zones are processed at the same intermediate TCG state

4. ATA Additions and Modifications for Flex

This section specifies the ATA syntax for Flex including modifications to existing text in ACS-4 r20, SATA 3.3, and ZAC as well as new commands, data structures and enumerations. Additional changes may be needed. In particular, all reference to `Empty`, `Implicitly Opened` and `Full` states in ZAC may need to be updated to describe the behavior of FMR zones.

Section numbers are disambiguated by being prefixed with the respective document name.

4.1. New feature set

4.1.1. Flex feature set

This section specifies the Flex feature set.

Devices that support the Flex feature set shall support all feature sets that are defined as mandatory by ACS-4, shall not support feature sets that are defined as prohibited by ACS-4, and may support feature sets that are defined as optional by ACS-4.

[Editor’s note: There is an opportunity to make the Flex feature set more constrained by specifying which features sets shall and which shall not be supported as with the Host Managed Zones feature set.]

Devices that support the Flex feature set:

a) shall report the ATA device signature (see ACS-4);
b) shall support at least one FMR zone (see 3.2.1);
c) shall support at least one SMR zone (see 3.2.2);
d) shall support the `Zoned Device Information` page (see ZAC) in the `IDENTIFY DEVICE` data log (see ACS-4);
e) shall support the `Flex Capabilities` log (see 4.3.1);
f) shall set the `ZONED` field (see ACS-4 9.10.5.12.1) to 00b (not reported) in:
    i) the Register FIS transmitted by the device as part of the `COMRESET` protocol; and
    ii) the `Supported Capabilities` page of the `IDENTIFY DEVICE` data log (see ACS-4);
g) shall set the `ZAC MINOR REVISION` field (see ZAC 6.2.6.1) to FFFFh (not reported) in the `Zoned Device Information` page of the `IDENTIFY DEVICE` data log (see ZAC 6.2.2);
h) shall set the FLEX SUPP bit to one in the the `Supported Capabilities` page of the `IDENTIFY DEVICE` data log (see 4.4.4.2);
i) shall not support zone transitions to the Offline state as the result of the device detecting a media failure (see ZAC 4.6.3.4 and 4.7.4);
j) shall not set the RWP Recommended zone attribute to one (see ZAC 4.5.5);
k) shall not set the RESET bit to one in a zone descriptor (see ZAC 5.2.7.6.6);
l) shall support the Sense Data Reporting feature set (see ACS-4);
m) shall support the `FLEX CONVERT NON-QUEUED` and `FLEX QUERY NON-QUEUE`D commands;
n) shall support the `Flex Control` subcommand of the `SET FEATURES` command (see 4.2.2);

o) shall support the non-NCQ command format for the zone management commands defined as mandatory in ZAC; and
p) if the NCQ feature set (see ACS-4) is supported, then:
    i) shall support the FLEX CONVERT QUEUED and FLEX QUERY QUEUED commands;
    ii) shall support the NCQ command encapsulations for the zone management commands defined as mandatory in ZAC;
    iii) shall support NCQ autosense (i.e., set the NCQ AUTOSENSE SUPPORTED bit to one in the Serial ATA page of th IDENTIFY DEVICE data log (see ACS-4));
    iv) shall support bytes 512..1 023 of the NCQ Command Error log (see ACS-4); and
    v) shall support the use of the NCQ command encapsulations of GPL feature set commands (see ACS-4) to read and write:
        1) the SCT Command/Status log (see ACS-4); and
        2) the SCT Data Transfer log (see ACS-4).

[Editor's note: Flex devices mimic the device signature of drive managed and legacy (non-ZAC) devices for backward compatibility.]

4.1.1.1. Basic zoned device model

See the extensions to ZAC (see 4.4.2) for the basic zoned device model.

4.1.1.2. Zone attributes and types

See the extensions to ZAC (see 4.4.2) for the zone attributes and types.

4.2. New Commands

4.2.1. FLEX CONVERT NON-QUEUED, FLEX CONVERT QUEUED, FLEX QUERY NON-QUEUED and FLEX QUERY QUEUED commands

This section specifies the ATA syntax of the commands that invoke the Flex Convert and Flex Query operations.

4.2.1.1. Feature set

These 48-bit commands are for devices that support the Flex feature set (see 4.1.1).

4.2.1.2. Description

The FLEX CONVERT NON-QUEUED and FLEX CONVERT QUEUED commands request a Flex Convert operation. The FLEX QUERY NON-QUEUED and FLEX QUERY QUEUED commands request a Flex Query operation.

The Smr Zone Type operation parameter is not transported by these commands. If DIRECTION is set to FROM CMR TO SMR then the FSSZT device parameter is used. If the FSSZT device parameter can be changed (that is, FSWP is set to one and FSWR is set to one) then the application should set the desired value of FSSZT by using the SET FEATURES subcommand Flex Control before issuing these commands.

The Number Of Zones operation parameter may or may not be transported by these commands. If NOZSRC is set to zero then the FSNOZ device parameter is used and the application should set the desired value of FSNOZ by using the SET FEATURES subcommand Flex Control before issuing these commands.

4.2.1.3. Inputs

The command inputs transport the Flex operation request parameters. See the following tables for the command inputs for these commands.

Table 4 -- FLEX CONVERT NON-QUEUED and FLEX QUERY NON-QUEUED command inputs

Field	Description
FEATURE	Bit Description 15:14 Reserved 13 ALL 12:3 Reserved 2 NOZSRC 1:0 DIRECTION
COUNT	RETURN PAGE COUNT field
LBA	STARTING ZONE LOCATOR field
AUXILIARY	AUXNOZ field
COMMAND	shall be set to C1h for FLEX CONVERT NON-QUEUED or C2h for FLEX QUERY NON-QUEUED

Table 5 -- FLEX CONVERT QUEUED and FLEX QUERY QUEUED command inputs

Field	**Description**
FEATURE	RETURN PAGE COUNT field
COUNT	Bit Direction 15:14 PRIO 13 ALL 12:8 SUBCOMMAND field = shall be set to 1Eh for FLEX CONVERT QUEUED or 1Fh for FLEX QUERY QUEUED 7:3 NCQ TAG 2 NOZSRC 1:0 DIRECTION
LBA	STARTING ZONE LOCATOR field
AUXILIARY	AUXNOZ field

COMMAND	65h

[Editor's notes:

- RETURN PAGE COUNT is the typical use of the FEATURE field in RECEIVE FPDMA QUEUED.
- The bits used by ALL and DIRECTION are reserved in RECEIVE FPDMA QUEUED in ACS-4 (see ACS-4 7.31.3.1).
- STARTING ZONE LOCATOR is the typical use of the LBA field in RECEIVE FPDMA QUEUED.
- AUXNOZ is not a typical use of the AUXILIARY field in RECEIVE FPDMA QUEUED.]

The device shall return the number of 512-byte pages specified in the RETURN PAGE COUNT field. Pad bytes are appended as needed to meet this requirement. Pad bytes shall have a value of 00h.

Unless otherwise specified, if the RETURN PAGE COUNT field specifies fewer 512-byte pages than the device has available to be returned, the device:

a) shall truncate the returned data to the specified number of 512-byte pages; and
b) shall not modify any of the returned data as a result of the truncation.

If the RETURN PAGE COUNT field specifies more 512-byte pages than the device has available to be returned, then 512 bytes of all zeros shall be returned for those pages.

The value 0000h is reserved in the RETURN PAGE COUNT field.

The PRIO field is defined in ACS-4 4.14.2.

The ALL bit specifies the All operation parameter of the operation request. The ALL bit is ignored if COMMAND equals C2h (FLEX QUERY NON-QUEUED).

The NCQ AG field is defined in ACS-4 7.16.3.3.

The NOZSRC (Number Of Zones Source) indicates the source of the Number Of Zones operation parameter or the operation invocation. If NOZSRC is set to zero then the setting of the FSNOZ device parameter specifies the Number Of Zones operation parameter and the AUXNOZ field is ignored. If NOZSRC is set to one then the AUXNOZ (Auxillary Number Of Zones) field specifies the Number Of Zones operation parameter.

The DIRECTION field (see the following table) specifies the Direction operation parameter of the operation request.

Table 6 -- DIRECTION field

Code	Direction
0h	FROM CMR TO SMR
1h	FROM SMR TO CMR
2h..3h	Reserved

The STARTING ZONE LOCATOR field specifies the Starting Zone Locator operation parameter of the operation request.

Operation parameters All, Direction, Starting Zone Locator and Number Of Zones are defined in clause 3.3.2.4.

4.2.1.4. Flex Convert and Flex Query operation invocation

The Flex Convert and Flex Query operations are invoked by these commands as follows.

```
if NOZSRC = 0
      NumberOfZones=FSNOZ
else
      NumberOfZones=AUXNOZ
if COMMAND = C1h or (COMMAND = 65h and SUBCOMMAND = 1Eh)
      // FLEX CONVERT NON-QUEUED or FLEX CONVERT QUEUED
      FlexConvert(Direction=DIRECTION, All=ALL,
      StartingZoneLocator=STARTING ZONE LOCATOR, NumberOfZones,
      SmrZoneType=FSSZT)
else
      // FLEX QUERY NON-QUEUED or FLEX QUERY QUEUED
      FlexQuery(Direction=DIRECTION, All=ALL,
      StartingZoneLocator=STARTING ZONE LOCATOR, NumberOfZones,
      SmrZoneType=FSSZT)
```

4.2.1.5. Normal outputs

See ACS-4.

4.2.1.6. Error outputs

See ACS-4.

[Editor’s note: These commands are aborted for scenarios specified in ACS-4; otherwise, they return with good status even if there are status bits set in the data returned by the Flex Convert or Flex Query operation.]

4.2.1.7. Input from the device to the host data structure

The input data from the device to the host transports the returned data of the operation. The format of the data input from the device to the host (see the following table) includes an extent descriptor list that contains the first through the last extent descriptor.

Table 7 -- FLEX CONVERT NON-QUEUED and FLEX QUERY NON-QUEUED input from device to host

Offset	Type	Description
0..3	DWord	EXTENT LIST LENGTH field
4..5	Word	Bit Meaning 15 Converted 14:9 Reserved 8 SZTBZL 7 SZTBV 6 SZSV 5 FZTBZL 4 FZTBV 3 FZSV 2 SEHMZT 1 NOZIZ 0 SZLBV
6..7	Word	Bit Meaning 15:14 Reserved 13 All bit 12:10 Reserved 9:8 Direction 7:0 Smr Zone Type
8..11	DWord	Bit Meaning 31:28 Reserved 27:0 Number Of Zones
12..31	Bytes	Reserved
32		Extent descriptor (first)
		Extent descriptor
		...
		Extent descriptor (last)
		Padding

The EXTENT LIST LENGTH field contains the length in bytes of the extent descriptors list; that is, Extent descriptor (first) through Extent descriptor (last in the table above). The extent list length is the number of extents reported multiplied by 16. This value may indicate bytes that are not returned due to truncation based on the value of the RETURN PAGE COUNT field.

The SZLBV bit indicates the SZLBV bit of the operation returned data.

The NOZIZ bit indicates the NOZIZ bit of the operation returned data.

The SEHMZT bit indicates the SEHMZT bit of the operation returned data.

The FZSV bit indicates the FZSV bit of the operation returned data.

The FZTBV bit indicates the FZTBV bit of the operation returned data.

The FZTBZL bit indicates the FZTBZL bit of the operation returned data.

The SZSV bit indicates the SZSV bit of the operation returned data.

The SZTBV bit indicates the SZTBV bit of the operation returned data.

The SZTBZL bit indicates the SZTBZL bit of the operation returned data.

The Converted bit indicates the Converted bit of the operation returned data.

The Smr Zone Type field indicates the Smr Zone Type field of the operation returned data.

The ALL bit indicates the All bit of the operation returned data.

The DIRECTION field indicates the Direction field of the operation returned data.

The Number Of Zones field indicates the Number Of Zones field of the operation returned data.

Operation returned data parameters are defined in clause 3.3.2.6.

4.2.1.7.1. Extent descriptor format

The extent descriptor list contains extent descriptors of the operation returned data. The descriptors shall be sorted in ascending order based on the STARTING ZONE LOCATOR of each extent. Each extent descriptor (see the following table) describes one extent.

Table 8 -- Extent descriptor format

Offset	**Type**	**Description**
0	Byte	Bit Meaning 7:4 Reserved 3:0 ZONE TYPE field
1	Byte	Bit Meaning 7:4 ZONE CONDITION field 3:0 Reserved
2..3	Bytes	Reserved
4..7	DWord	NUMBER OF ZONES field
8..15	QWord	STARTING ZONE LOCATOR field

The ZONE TYPE field indicates the Zone Type of this extent descriptor. The encoding of the ZONE TYPE field is defined in ZAC 5.2.7.6.6.2.

The ZONE CONDITION field indicates the Zone Condition of this extent descriptor. The encoding of the ZONE CONDITION field is defined in ZAC 5.2.7.6.6.3.

The NUMBER OF ZONES field indicates the Number Of Zones of this extent descriptor.

The STARTING ZONE LOCATOR field indicates the Starting Zone Locator of this extent descriptor.

Extent descriptor parameters Number Of Zones and Starting Zone Locator are defined in clause 3.3.2.6.

4.2.2. Flex Control subcommand

SET FEATURES subcommand code DCh configures the behavior of Flex operations. The table below describes the command inputs. This is a vendor specific extension to SET FEATURES (see ACS-4 7.43).

[Editor's Note: SET FEATURES command subcommand code DCh is vendor specific (see ACS-4 7.43.2).]

If the FLEX SUPP (see 4.4.4.2) bit is cleared to zero, the device shall return command aborted. The device shall return command aborted if the FLEX SUPP bit is set to one and either:

a) the FSWP bit is cleared to zero and the FSSZT field specifies SEQUENTIAL WRITE PREFERRED; or
b) the FSWR bit is cleared to zero and the FSSZT field specifies SEQUENTIAL WRITE REQUIRED.

Table 9 -- SET FEATURES command, Flex Control subcommand inputs

Field	Description
FEATURE	7:0 DCh
COUNT	FSSZT field
LBA	FSNOZ field

The FSSZT (Flex Subsequent SMR Zone Type) field specifies the setting for the FSSZT device parameter. The FSSZT device parameter applies to SMR zones only and is used by Flex Convert and Flex Query operations when the value of Direction = From Cmr To Smr. A value of Unchanged specifies that the setting shall not change. The setting of the FSSZT device parameter shall not be affected by a power-on reset, a hardware reset, or a software reset. The following table describes the FSSZT field.

Table 10 -- FSSZT field

Code	Description
00h	Unchanged
01h	Reserved
02h	SEQUENTIAL WRITE REQUIRED
03h	SEQUENTIAL WRITE PREFERRED
04h..FFh	Reserved

The FSSNOZ (Flex Subsequent Number Of Zones) field specifies the setting for the FSSNOZ device parameter. The setting of the FSSNOZ device parameter shall not be affected by a power-on reset, a hardware reset, or a software reset.

Note to entry: SET FEATURES is a 28-bit command and, thus, the FSSNOZ field is a 28-bit field.

4.3. New Logs

This section defines extensions to the ATA log definitions (see ACS-4 clause 9). Logs are accessible via READ LOG EXT, READ LOG DMA EXT, SMART READ LOG, SMART WRITE LOG, WRITE LOG EXT and WRITE LOG DMA EXT.

4.3.1. Flex Capabilities log

The Flex Capabilities log (see the table below) is log address DDh. The Flex Capabilities log is mandatory for Flex devices and reports device configuration information.

[Editor's Note: The log address, DDh, for the Flex Capabilities log is a device vendor specific value (see ACS-4 9.1).]

Table 11 -- Flex Capabilities log

Offset	Type	Description
0..7	QWord	Flex Capabilities log information header Bit Description 63 Shall be set to one 62..16 Reserved 15:0 Log revision number. Shall be set to 000h.
8..15	QWord	Flex Capabilities Bit Description 63 Shall be set to one 62..16 Reserved 15..8 FLEX VERSION 4..7 Reserved 3 FSWP 2 FSWR 1 URFZ 0 Reserved
16..23	QWord	Reserved
24..31	QWord	FMRmax
32..39	QWord	SMRmin
40..47	QWord	SMRmax
48..55	QWord	Flex Current Settings Bit Description 63..56 FSSZT 55..28 Reserved 27..0 FSNOZ
56..511	Bytes	Reserved

An URFZ (Unrestricted Read in FMR Zones) bit cleared to zero indicates that the device does not support reading unwritten logical sectors in FMR zones. A URFZ bit set to one indicates that the device supports reading unwritten logical sectors in FMR zones.

A FSWR (Flex Sequential Write Required) bit set to one indicates that the device supports SMR zones of the sequential write required SMR zone type.

A FSWP (Flex Sequential Write Preferred) bit set to one indicates that the device supports SMR zones of the sequential write preferred SMR zone type.

The FLEX VERSION field indicates the version of the Flex Protocol which this device implements. Devices that implement the version defined in this document shall set this field to 00h.

[Editor's note: It is intended for the FLEX VERSION field to be set to 10h for devices that implement the formally approved revision of this document, and that unique value get assigned to a larger value for subsequent major releases and revisions.]

The FMRmax field is defined in the FMR zone model.

The SMRmin and SMRmax fields are defined in the SMR zone model.

The FSSZT (Flex Subsequent SMR Zone Type) field indicates the setting of the FSSZT device parameter. Values for this field are defined in the following table.

Table 12 -- FSSZT field

FSSZT field	Zone Type
00h..01h	Reserved
02h	Sequential Write Required
03h	Sequential Write Preferred
04h..0Fh	Reserved

This setting can be changed by the SET FEATURES command Flex Control subcommand.

The Flex Subsequent Number Of Zone (FSNOZ) field indicates the setting of the FSNOZ device parameter. This setting can be changed by the SET FEATURES command Flex Control subcommand.

4.4. Extensions to ACS-4, SATA 3.3 and ZAC

In this section indented items indicate changes to existing contents of ACS-4 r20, SATA 3.3, and ZAC. The section and table numbers are from the respective standard and are disambiguated by being prefixed with the respective document name, and references to other clauses in this document are disambiguated by being prefixed with FDI.

Existing text is in grey, new or changed text is in blue, deleted text in ~~red strikethrough~~ and ellipses show omitted unchanged content.

4.4.1. Extensions to definitions, abbreviations, and conventions

ACS-4 3 Definitions, abbreviations, and conventions
ACS-4 3.1 Definitions
…
ACS-4 3.1.3 accessible max address

maximum LBA that, if FLEX SUPP is zero is accessible by read commands and write commands and if FLEX SUPP is one is accessible by read and write commands in online CMR space, that return command completion without error

…

ACS-4 3.1.99 user data area

area of the media that, if FLEX SUPP is zero is addressable from LBA 0 to the native max address if the Accessible Max Address Configuration feature set is supported, or LBA 0 to the maximum value defined in table 5 if the Accessible Max Address Configuration feature set is not supported, and if FLEX SUPP is one is provisioned for online zones

...

ZAC 3 Definitions, abbreviations, and conventions

ZAC 3.1 Definitions

…

3.1.58 write pointer
pointer to a logical sector in a write pointer zone where the next write command in that zone should start

Note 1 to entry: See ZAC 4.6.3.1.

[Editor’s note: The write pointer of an FMR zone is the largest logical sector where the next write command in that zone should start, but the existing definition may be good enough.]

3.1.59 write pointer zone

zone that has an associated write pointer

Note 1 to entry: See ZAC 4.6.3.

[Editor’s note: FMR zones have a write pointer and, thus, are write pointer zones.]

...

4.4.2. Extensions to feature set definitions

This section specifies extensions to the ZAC feature set definitions.

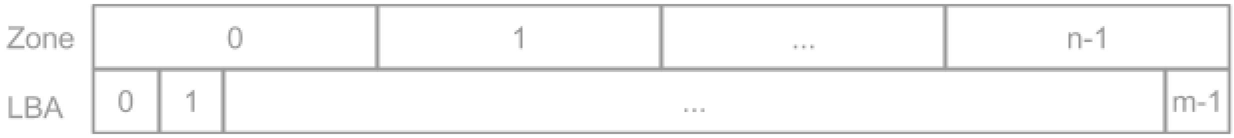
ZAC 4 Feature set definitions
ZAC 4.1 Introduction

Based on the basic zoned device model (see ZAC 4.2), the following feature sets are defined by this standard:

a) Host Aware Zones feature set (see ZAC 4.3); ~~and~~
b) Host Managed Zones feature set (see ZAC 4.4); and
c) Flex feature set (see FDI 4.1.1).

ZAC 4.2 Basic zoned device model

Zoned devices are accessed using LBAs. The LBAs are divided into ranges called zones. The entire capacity of a zoned device is organized into a set of contiguous, non-overlapping zones. Figure ZAC 3 shows a zoned device with n zones and m LBAs where LBA 0 is the lowest LBA of zone 0 and LBA m–1 is the highest LBA of zone n–1.



ZAC Figure 3 -- Zones in a zoned device

Each zone is one of the following types:

a) a Conventional zone (see ZAC 4.6.2); or
b) a write pointer zone (see ZAC 4.6.3) that is:
A) a Sequential Write Preferred zone (see ZAC 4.6.3.2); ~~or~~
B) a Sequential Write Required zone (see ZAC 4.6.3.3); or
C) an FMR zone (see FDI 3.2.1).

The REPORT ZONES EXT command (see ZAC 5.2.7) returns the zone type for each zone.

ZAC 4.3 Host Aware Zones feature set

...

ZAC 4.4 Host Managed Zones feature set

...

ZAC 4.x Flex feature set

See FDI 4.1.1.

ZAC 4.5 Zone attributes

ZAC 4.5.1 Summary

...

ZAC 4.5.2 Zone Type zone attribute

Zone Type indicates (see ZAC table 3) the type of zone.

ZAC Table 3 -- Zone Type zone attribute

Zone Type	Reference
CONVENTIONAL	ZAC 4.6.2
FMR	FDI 3.2.1
SEQUENTIAL WRITE PREFERRED	ZAC 4.6.3.2
SEQUENTIAL WRITE REQUIRED	ZAC 4.6.3.3

The relationships between Zone Type and other zone attributes are shown in table 4.

ZAC Table 4 -- Relationships between zone attributes

Zone Type	Other zone attributes			
	Non-Sequential Write Resources Active	RWP Recommended	WPointer	Zone Condition
CONVENTIONAL	false (a)	false (a)	invalid (a)	see ZAC 4.5.3
FMR	false (c)	false (c)	see ZAC 4.5.4 and ZAC table 7	
SEQUENTIAL WRITE PREFERRED	See ZAC 4.5.6	see ZAC 4.5.5		
SEQUENTIAL WRITE REQUIRED	false (b)			

(a) The zone attributes values for a zone with Zone Type CONVENTIONAL are required to be compatible with devices that conform to ACS-4.
(b) Zone resources for non-sequential writes are not allocated for a zone in which all non-sequential write commands return command completion with an error.
(c) The zone attributes values for a zone with Zone Type FMR with a zone condition of FULL are required to be compatible with devices that conform to ACS-4.

ZAC 4.5.3 Zone Condition zone attribute

The Zone Condition (see ZAC table 5) is determined by the Zone Type (see ZAC 4.5.2) and the Zone Condition state machine state (see ZAC 4.6.3.4).

ZAC Table 5 -- Zone Condition zone attribute

Zone Type (a)	Zone Condition state machine state	Reference	Zone Condition
CONVENTIONAL	Outside the scope of this standard	ZAC 4.6.2	NOT WRITE POINTER
FMR, SWP or SWR	ZC1: Empty state	ZAC 4.6.3.4.2	EMPTY
FMR, SWP or SWR	ZC2: Implicit_Open state	ZAC 4.6.3.4.3	IMPLICITLY OPENED
SWP or SWR	ZC3: Explicit_Open state	ZAC 4.6.3.4.4	EXPLICITLY OPENED
SWP or SWR	ZC4: Closed state	ZAC 4.6.3.4.5	CLOSED
FMR, SWP or SWR	ZC5: Full state	ZAC 4.6.3.4.6	FULL
SWP or SWR	ZC6: Read_Only state	ZAC 4.6.3.4 7	READ ONLY
FMR, SWP or SWR	ZC7: Offline state	ZAC 4.6.3.4.8	OFFLINE
(a) SWP stands for SEQUENTIAL WRITE PREFERRED and SWR stands for SEQUENTIAL WRITE REQUIRED			

ZAC 4.5.4 WPointer zone attribute

...

ZAC 4.5.5 RWP Recommended zone attribute

For write pointer zones (see ZAC 4.6.3), if RWP Recommended is:

a) true, then the device has determined (e.g., through the detection of insufficient zone resources) that the processing of a RESET WRITE POINTER EXT command (see ZAC 5.2.8) specifying this zone is recommended; or

b) false, then the device has no recommendation for or against the processing of a RESET WRITE POINTER EXT command specifying this zone.

RWP Recommended shall be set to false if:

a) the Zone Type is CONVENTIONAL or FMR; or

b) the Zone Condition is EMPTY, READ ONLY, or OFFLINE.

...

ZAC 4.5.6 Non-Sequential Write Resources Active zone attribute

…

The Non-Sequential Write Resources Active shall be set to false if:

a) the Zone Type is CONVENTIONAL, FMR or SEQUENTIAL WRITE REQUIRED;
b) the Zone Condition is EMPTY, READ ONLY, or OFFLINE; or
c) the device has not processed a non-sequential write command in a zone since the last time the Zone Condition was EMPTY.

ZAC 4.6 Zone types

ZAC 4.6.1 Overview

In a zoned block device, each zone:

a) has the zone attributes defined in ZAC 4.5; and
b) is one of the following zone types:
    A) a Conventional zone (see ZAC 4.6.2); or
    B) a write pointer zone (see ZAC 4.6.3) that is either:
        a) a Sequential Write Preferred zone (see ZAC 4.6.3.2); ~~or~~
        b) a Sequential Write Required zone (see ZAC 4.6.3.3); or
        c) an FMR zone (see FDI 3.2.1).

ZAC 4.6.2 Conventional zones

...

ZAC 4.6.3 Write pointer zones

ZAC 4.6.3.1 Features common to all write pointer zones

…

ZAC 4.6.3.x FMR zones

See FDI 3.2.1.

ZAC 4.6.3.2 Sequential Write Preferred zones

…

ZAC 4.6.3.3 Sequential Write Required zones

...

ZAC 4.6.3.4 Zone Condition state machine

ZAC 4.6.3.4.1 Overview

There is one Zone Condition state machine for each write pointer zone (see ZAC 4.6.3). The Zone Condition state machine (see ZAC figure 9) controls the operational characteristics of each write pointer zone. This state machine consists of the following states:

a) ZC1: Empty state (see ZAC 4.6.3.4.2);
b) ZC2: Implicit_Open state (see ZAC 4.6.3.4.3);
c) ZC3: Explicit_Open state (see ZAC 4.6.3.4.4);
d) ZC4: Closed state (see ZAC 4.6.3.4.5);
e) ZC5: Full state (see ZAC 4.6.3.4.6);
f) ZC6: Read_Only state (see ZAC 4.6.3.4.7); and
g) ZC7: Offline state (see ZAC 4.6.3.4.8).

Except as specified in ZAC 4.7, the device shall maintain the zone state after processing all resets except a power-on reset (see ACS-4).

After a power-on reset, the initial state for each write pointer zone shall be:

a) the ZC1: Empty state for a zone in which:
    A) the write pointer is valid and indicates the lowest LBA in the zone; and
    B) Non-Sequential Write Resources Active is false;
x) the ZC2: Implicit_Open state for a zone:
    A) in which the zone type is FMR; and
    B) in Zone Condition IMPLICITLY OPENED before the power-on reset;
b) the ZC4: Closed state for a zone in which:
    A) the zone type is not FMR; and
    B) either
        A) the write pointer is valid and indicates an LBA that is not the lowest LBA in the zone; or
        B) Non-Sequential Write Resources Active is true;
c) the ZC5: Full state for a zone in Zone Condition FULL before the power-on reset;
d) the ZC6: Read_Only state for a zone in Zone Condition READ ONLY before the power-on reset; and
e) the ZC7: Offline state for a zone in Zone Condition OFFLINE before the power-on reset.

On completion of a sanitize operation (see ZAC 4.7.6), for each write pointer zone whose Zone Condition is not changed as a result of processing the sanitize operation, that zone's state shall be:

a) the ZC1: Empty state for a zone in which:
    A) the write pointer is valid and indicates the lowest LBA in the zone; and
    B) Non-Sequential Write Resources Active is false;
x) the ZC2: Implicit_Open state for a zone in which
    A) the zone type is FMR
    B) the the write pointer is valid and indicates an LBA that is not the lowest LBA in the zone;
b) the ZC4: Closed state for a zone in which:
    A) the zone type is not FMR; and
    B) either

A) the write pointer is valid and indicates an LBA that is not the lowest LBA in the zone; or

B) Non-Sequential Write Resources Active is true;

c) the ZC5: Full state for a zone in Zone Condition FULL before the sanitize operation;

d) the ZC6: Read_Only state for a zone in Zone Condition READ ONLY before the sanitize operation; and

e) the ZC7: Offline state for a zone in Zone Condition OFFLINE before the sanitize operation.

…

The characteristics and zone attributes associated with the state of a zone are summarized in ZAC table 7.

ZAC Table 7 -- Characteristics and attributes associated with zone state

State	Zone characteristics and attributes			
	Write pointer valid (a)	Open zone resources in use	Accessible for	
			Reads	Writes
ZC1: Empty state	Yes	No	See (b)	See (c)
ZC2: Implicit_Open state	Yes	Yes	Yes	Yes
ZC3: Explicit_Open state	Yes	Yes	Yes	Yes
ZC4: Closed state	Yes	No	Yes	See (c)
ZC5: Full state	No	No	Yes	See (d)
ZC6: Read_Only state	No	No	Yes	No
ZC7: Offline state	No	No	No	No

(a) A valid write pointer (i.e., Yes) indicates a specific LBA in the zone as described in ZAC 4.6.3.1. An invalid write pointer (i.e., No) provides no information.

(b) This zone is accessible for reads if:
a) the zone type is Sequential Write Preferred (see ZAC 4.6.3.2); ~~or~~
b) the zone type is Sequential Write Required (see ZAC 4.6.3.3) and the URSWRZ bit is set to one (see ZAC 6.2.2.2); or
c) the zone type is FMR (see FDI 3.2.1) and the URFZ bit is set to one (see FDI 3.2.1) .

(c) This zone is accessible for writes if the Manage Open Zone Resources function (see ZAC 4.6.3.4.9) is able to complete with success and the zone transitions to the ZC2: Implicit_Open state.

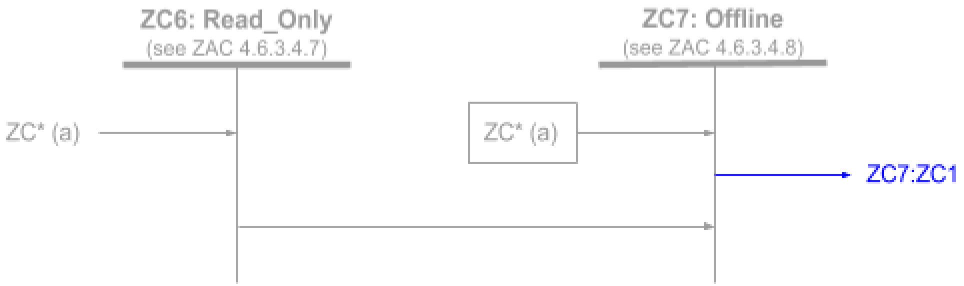
(d) This zone is accessible for writes if:
a) the zone type is Sequential Write Preferred (see ZAC 4.6.3.2); or
b) the zone type is FMR (see FDI 3.2.1).

This state machine shall maintain the OZR Available state machine variable to indicate the value returned by a Manage Open Zone Resources function (see ZAC 4.6.3.4.9).

ZAC Figure 9 shows the Zone Condition state machine.

…

[Editor's note: The ZC7:ZC1 arc into ZC1: Empty in part 1 of 2 of ZAC figure 9 is not shown in this markup.]



(a) The ZC1: Empty state, the ZC2: Implicit_Open state, the ZC3: Explicit_Open state, the ZC4: Closed state, and the ZC5: Full state include a transition to this state.

ZAC Figure 9 -- Zone Condition state machine (part 2 of 2)

ZAC 4.6.3.4.2 ZC1: Empty state

For a zone in this state:
…

g) if the device processes a read command, then the device shall process the read command as described in:

A) ZAC 4.6.3.2.2 for Sequential Write Preferred zones; ~~or~~
B) ZAC 4.6.3.3.2 for Sequential Write Required zones; or
C) FDI 3.2.1 for FMR zones.

Transition ZC1:ZC7: The zone may transition from the ZC1: Empty state to the ZC7: Offline state (see ZAC 4.6.3.4.8):

a) for the Host Aware feature set and Host Managed feature set,
a) as the result of the device detecting a media failure (see ZAC 4.7.4); or
b) for reasons outside the scope of this standard; and
b) for the Flex feature set, as a result of a Flex Convert operation.

ZAC 4.6.3.4.3 ZC2: Implicit_Open state

For a zone in this state:
…

f) if the device processes a write command, then the device shall process the write command as described in:
   A) ZAC 4.6.3.2.1 for Sequential Write Preferred zones; ~~or~~
   B) ZAC 4.6.3.3.1 for Sequential Write Required zones; or
   C) FDI 3.2.1 for FMR zones;

and

g) if the device processes a read command, then the device shall process the read command as described in:
   A) ZAC 4.6.3.2.2 for Sequential Write Preferred zones; ~~or~~
   B) ZAC 4.6.3.3.2 for Sequential Write Required zones; or
   C) FDI 3.2.1 for FMR zones.

…

Transition ZC2:ZC5:

….

For an FMR zone, the zone shall transition from the ZC2: Implicit_Open state to the ZC5: Full state (see ZAC 4.6.3.4.6) after successful completion of a write command with the starting LBA equal to or less than the write pointer that writes the highest LBA in the zone.

...

Transition ZC2:ZC7: ~~The~~ For the Host Aware feature set and Host Managed feature set, the zone may transition from the ZC2: Implicit_Open state to the ZC7: Offline state (see ZAC 4.6.3.4.8):

a) as the result of the device detecting a media failure (see ZAC 4.7.4); or

b) for reasons outside the scope of this standard.

For the Flex feature set, this transition shall not occur.

ZAC 4.6.3.4.4 ZC3: Explicit_Open state

…

ZAC 4.6.3.4.5 ZC4: Closed state

…

ZAC 4.6.3.4.6 ZC5: Full state

For a zone in this state:

...

h) if the device processes a read command, then the device shall process the read command as described in:
   A) 4.6.3.2.2 for Sequential Write Preferred zones; ~~or~~
   B) 4.6.3.3.2 for Sequential Write Required zones; or
   C) FDI 3.2.1 for FMR zones.

...

Transition ZC5:ZC7: ~~The~~ For the Host Aware feature set and Host Managed feature set, the zone may transition from the ZC5: Full state to the ZC7: Offline state (see ZAC 4.6.3.4.8):

a) as the result of the device detecting a media failure (see ZAC 4.7.4); or
b) for reasons outside the scope of this standard.

For the Flex feature set, the zone may transition from the ZC5: Full state to the ZC7: Offline state as the result of a Flex Convert operation (see FDI 3.3.2).

ZAC 4.6.3.4.7 ZC6: Read_Only state

…

ZAC 4.6.3.4.8 ZC7: Offline state

…

For the Host Aware feature set and Host Managed feature set, ~~Transitions~~ transitions out of this state are outside the scope of this standard.

Transition ZC5:ZC1: For the Flex feature set, the zone may transition from the ZC7: Offline state to the ZC1: Empty state (see ZAC 4.6.3.4.2) as the result of a Flex Convert operation (see FDI 3.3.2).

ZAC 4.6.3.4.9 Manage Open Zone Resources function

ZAC 4.6.3.4.9.1 Overview

…

ZAC 4.6.3.4.9.x Processing an FMR zone

For an FMR zone:

a) the device shall not process a Close Zone function (see ZAC 4.6.3.4.11); and
b) the function shall return SUCCESS.

ZAC 4.6.3.4.9.2 Processing a sequential write preferred zone

…

ZAC 4.7 Additional features for the Flex feature set, Host Aware Zones feature set and the Host Managed Zones feature set

ZAC 4.7.1 Introduction

Support for the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) and the Host Managed Zones feature set (see ZAC 4.4) modifies features defined in ACS-4.

ZAC 4.7.2 Zoned device internal resource management

A device that supports the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4) requires internal resources (e.g., persistent zone resources) to maintain each zone. Insufficient resources may result in degraded functionality (e.g., reduced performance, increased power consumption, or increased reporting of write errors).

...

ZAC 4.7.6 Interactions with the Sanitize Device feature set

If a device that supports the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4) also supports the Sanitize Device feature set (see ACS-4), then the additional requirements described in this subclause apply.

...

An OVERWRITE EXT command that completes without an error modifies the substitute data pattern (see ZAC 4.6.3.2.3 and ZAC 4.6.3.3.3).

[Editor's note: Sanitize Device feature set commands do not invoke a `Flex Convert` operation; that is, offline zones stay offline and online zones stay only.

...

ZAC 4.7.8 Interactions with the SECURITY ERASE UNIT command

If a device supports the Flex feature set (see FDI 4.1.1) or the Host Aware feature set (see ZAC 4.3) and the device processes a SECURITY ERASE UNIT command (see ACS-4), then prior to returning command completion without error:

- a) the device shall set Non-Sequential Write Resources Active to false for all write pointer zones (see ZAC 4.6.3); and
- b) if the ZAC SECURITY OPTION bit (see ACS-4) is:
  - A) cleared to zero, then the device shall cause the Zone Condition to become EMPTY for all write pointer zones; or
  - B) set to one, then the device shall cause the Zone Condition to become FULL for all write pointer zones.

[Editor's notes:

FMR zones have a write pointer and, thus, are write pointer zones and the ZAC SECURITY OPTION bit applies the the resulting Zone Condition of FMR zones.

The SECURITY ERASE UNIT command does not invoke a `Flex Convert` operation; that is, offline zones stay offline and online zones stay only.]

...

4.4.3. Extensions to command descriptions

4.4.3.1. RECEIVE FPDMA QUEUED subcommands

4.4.3.1.1. SATA RECEIVE FPDMA QUEUED subcommands

This section specifies extensions to RECEIVE FPDMA QUEUED in SATA 3.3, adding new subcommands for Flex commands.

SATA 13.6.7.5 RECEIVE FPDMA QUEUED subcommands

Subcommands for the RECEIVE FPDMA QUEUED commands are contained within the COUNT field (12:8). The allowed values are defined in SATA Table 111.

SATA Table 111 - Subcommands for RECEIVE FPDMA QUEUED

Value	Description	Reference
00h	Reserved	
01h	READ LOG DMA EXT subcommand	SATA 13.6.7.6
02h	ZAC MANAGEMENT IN subcommand	SATA 13.6.7.7
03h..1Dh	Reserved	
1Eh	FLEX CONVERT QUEUED subcommand	FDI 6.1.2
1Fh	FLEX QUERY QUEUED subcommand	FDI 6.1.2

[Editor's Note: These RECEIVE FPDMA QUEUED command subcommand codes used by FLEX CONVERT QUEUED and FLEX QUERY QUEUED are reserved. An alternative might be to use a command opcode that is reserved for SATA: 62h, 66h, or 67h (see ACS-4 Annex A).]

4.4.3.1.2. ATA RECEIVE FPDMA QUEUED subcommands

This section specifies extensions to `RECEIVE FPDMA QUEUED` in ACS-4, adding new subcommands for Flex commands.

> ACS-4 7.31 RECEIVE FPDMA QUEUED - 65h, DMA Queued
> …
> ACS-4 7.31.3 Inputs
> …
> ACS-4 7.31.3.2 Subcommand
>
> ACS-4 table 92 defines the `RECEIVE FPDMA QUEUED` subcommands.

ACS-4 Table 92 - `RECEIVE FPDMA QUEUED` Subcommands

Subcommand	Description	Processing order requirements	Reference	
			Queued command	Non-queued command
00h	Reserved			
01h	`READ LOG DMA EXT`	sequential	ACS-4 7.24.6	ACS-4 7.24
02h	`ZAC Management In`	none	ZAC	ZAC
03h..1Dh	Reserved			
1Eh	`FLEX CONVERT QUEUED`	sequential	FDI 4.2.1	n/a
1Fh	`FLEX QUERY QUEUED`	sequential	FDI 4.2.1	n/a

4.4.3.2. Extensions to ZAC management out commands

Flex drives shall support new extensions to the following existing commands allow a zone range to be specified.

- `CLOSE ZONES EXT`
- `FINISH ZONES EXT`
- `OPEN ZONES EXT`
- `RESET WRITE POINTERS EXT`

Refer to the approved proposals: T13 documents f17138r2, f17139r2 and f17140r2.

[Editor's Note: One significant use case is using this form of `RESET WRITE POINTERS EXT` to request a `Reset Write Pointer` operation for an extent in a single command as part of a conversion. This is in contrast to hundreds of commands that each specify a single zone as part of conversion of extents on the order of 100 GB.]

4.4.3.3. Extensions to CLOSE ZONE EXT

ZAC 5.2.4 CLOSE ZONE EXT command – 9Fh/01h, Non-Data

ZAC 5.2.4.1 Feature Set

This 48-bit command is for devices that support the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4).

…

ZAC 5.2.4.3.3 ZONE ID field

The ZONE ID field specifies the lowest LBA of the ~~write pointer~~ sequential write preferred or sequential write required zone for which the device shall process the command specified by the ZM_ACTION field (see ZAC 5.2.2.3.3) and the COMMAND field.

If the ZONE ID field does not specify the lowest LBA of the ~~write pointer~~ sequential write preferred or sequential write required zone, the device shall:

a) return command completion with an error; and
b) set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN CDB (see ACS-4 and SPC-5).

...

4.4.3.4. Extensions to FINISH ZONE EXT

ZAC 5.2.5 FINISH ZONE EXT command – 9Fh/02h, Non-Data

ZAC 5.2.4.1 Feature Set

This 48-bit command is for devices that support the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4).

…

4.4.3.5. Extensions to OPEN ZONE EXT

ZAC 5.2.6 OPEN ZONE EXT command – 9Fh/03h, Non-Data

ZAC 5.2.4.1 Feature Set

This 48-bit command is for devices that support the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4).

…

4.4.3.6. Extensions to REPORT ZONES EXT

4.4.3.6.1. Support for a compact form of zone descriptors

Refer to T13 document f17114 for an extension to the `REPORT ZONES EXT` command for a new zone descriptor format.

[Editor's note: This proposal has not yet been approved by T13.]

ZAC 5.2.7 REPORT ZONES EXT command -- 4Ah/00h, DMA

ZAC 5.2.7.1 Feature set

This 48-bit command is for devices that support the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4).

...

ZAC 5.2.7.6 Inputs From the Device to the Host Data Structure

ZAC 5.2.7.6.1 Overview

…

ZAC 5.2.7.6.4 MAXIMUM LBA field

The MAXIMUM LBA field indicates the LBA of the last logical sector on the device, including all logical sectors in all zones. If `FLEX SUPP` is set to one, this field specifies `SMRmax` as defined in the SMR zone model.

…

ZAC 5.2.7.6.6 Zone descriptor format

…

ZAC 5.2.7.6.6.2 Zone Type field

…

ZAC Table 28 - `Zone Type` field

Code	`Zone Type`
...	...
4h	`FMR`
5h..Fh	Reserved

…

4.4.3.7. Extensions to RESET WRITE POINTER EXT

ZAC 5.2.8 RESET WRITE POINTER EXT command -- 9Fh/04h, DMA

ZAC 5.2.4.1 Feature Set

This 48-bit command is for devices that support the Flex feature set (see FDI 4.1.1), the Host Aware Zones feature set (see ZAC 4.3) or the Host Managed Zones feature set (see ZAC 4.4).

…

ZAC 5.2.8.3 Inputs

ZAC 5.2.8.3.1 Overview

See table 30 for the RESET WRITE POINTER EXT command inputs.

ZAC Table 30 -- RESET WRITE POINTER EXT command inputs

Field	Description
...	...
LBA	ZONE ID field -- See ZAC 5.2.4.3.3
...	...

ZAC 5.2.8.3.2 RESET ALL bit

If the RESET ALL bit is set to one, the device shall ignore the contents of the ZONE ID field and process a Reset Write Pointer function (see ZAC 4.6.3.4.13) for each sequential write preferred zone and sequential write required zone with a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or FULL.
…

ZAC 5.2.8.3.x ZONE ID field

The ZONE ID field specifies the lowest LBA of the write pointer zone for which the device shall process the command specified by the ZM_ACTION field (see ZAC 5.2.2.3.3) and the COMMAND field.

If the ZONE ID field does not specify the lowest LBA of the write pointer zone, the device shall:

a) return command completion with an error; and

b) set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN CDB (see ACS-4 and SPC-5).

...

[Editor’s note: this change does not change existing ZAC-compliant behavior, but does specify that the RESET ALL bit does not affect FMR zones. The motivation is to allow a Flex drive to be configured with some FMR zones and some SMR zones, and then be used by a Host Aware or Host Managed application and particularly have that application able to issue “Reset All” with the same result as a Host Aware or Host Managed device.]

4.4.4. Extensions to log definitions

4.4.4.1. Extensions to the Capacity log page

This section specifies extensions to the Capacity data log page (page 02h of IDENTIFY DEVICE data log (Log Address 30h)). This extension allows Flex drives to be used in their 100% CMR line configuration in environments that have no support for this protocol.

[Editor's Notes:

The ACCESSIBLE CAPACITY field is extended to report FMRmax+1. This also affects the value reported in IDENTIFY DEVICE data word 60..61 (see ACS-4 7.13.6.22) and the normal outputs of GET NATIVE MAX ADDRESS EXT (see ACS-4 7.2.2).

Also, the Flex Capabilities log has SMRmin and SMRmax as well as FMRmax. From these values the 100% CMR capacity and 100% SMR capacity can be directly calculated, at least if there are no non-Flex reasons that zones are offline.

Another value that could be of interest is the current online capacity. A field could be added for this in the Flex Capabilities page. Another possibility is to define a Reporting Options code for REPORT ZONES EXT that specifies online zones are to be returned, thus allowing a calculation of online capacity based on the calculated number of online zones.]

> ACS-4 9.10 IDENTIFY DEVICE data log (Log Address 30h)
>
> …
>
> ACS-4 9.10.4 Capacity (page 02h)
>
> …
>
> ACS-4 9.10.4.2 ACCESSIBLE CAPACITY field
>
> The ACCESSIBLE CAPACITY field (see ACS-4 table 244) is a mandatory field which:
>
> a) if the value of the FLEX SUPP bit is zero, contains a value that is one greater than the maximum LBA in user accessible space; and
> b) if the value of the FLEX SUPP bit is one, contains a value that is one greater than the maximum LBA in CMR space.
>
> The maximum value that shall be placed in the ACCESSIBLE CAPACITY field is FFFF_FFFF_FFFFh. The contents of the ACCESSIBLE CAPACITY field may be affected by commands in the Accessible Max Address Configuration feature set (see ACS-4 4.4) and the Storage Element Depopulation feature set (see ACS-4 4.24).

4.4.4.2. Extensions to the Supported Capabilities log page

This section specifies extensions to the `Supported Capabilities` data log page (page 03h of `IDENTIFY DEVICE` data log (Log Address 30h)). This extension adds the `FLEX SUPP` bit to the `Zoned Capabilities` field. This extension allows Flex drives to be directly discovered through a page supported by ATA devices.

ACS-4 9.10 IDENTIFY DEVICE data log (Log Address 30h)
…
ACS-4 9.10.5 Supported Capabilities (page 03h)
The Supported Capabilities log page (see ACS-4 table 245) provides a mechanism for the device to report support for feature sets, features, commands and other device capabilities.

ACS-4 Table 245 -- Supported Capabilities

Offset	Type	Content
...	...	...
104..111	QWord	Zoned Capabilities Bit Description 63 Contents of the QWord are valid ~~62:2 Reserved~~ 62:9 Reserved 8 FLEX SUPP bit (see ACS-4 9.10.5.12.x) 7:2 Reserved 1:0 ZONED field (see ACS-4 9.10.5.12.1)

…
ACS-4 9.10.5.12 `Zoned Capabilities`

ACS-r 9.10.5.12.x `FLEX SUPP` bit

A `FLEX SUPP` (Flex Supported) bit (see ACS-4 table 245) set to one indicates that the device supports the Flex feature set. A `FLEX SUPP` bit (see ACS-4 table 245) set to zero indicates that the device does not support the Flex feature set.

5. SCSI Additions and Modifications for Flex

This section specifies the SCSI syntax for Flex including modifications to existing text in SPC-5, SBC-4 r13 and ZBC as well as new commands, data structures and enumerations. Additional changes may be needed. In particular, all reference to `Empty`, `Implicitly Opened` and `Full` states in ZBC may need to be updated to describe the behavior of FMR zones.

5.1. New Commands

5.1.1. FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) commands

The `FLEX CONVERT (16)`, `FLEX CONVERT (32)`, `FLEX QUERY(16)` and `FLEX QUERY (32)` commands (see the following tables) request the device server to perform a Flex operation and transports some of the operation request parameters. The `FLEX CONVERT (16)` and `FLEX CONVERT (32)` commands request the device server to perform a `Flex Convert` operation. The `FLEX QUERY (16)` and `FLEX QUERY (32)` commands request the device server to perform a `Flex Query` operation.

The `Smr Zone Type` operation parameter is not transported by these commands. If `DIRECTION` is set to `FROM CMR TO SMR` then the `FSSZT` device parameter is used. If the `FSSZT` device parameter can be changed (that is, `FSWP` is set to one and `FSWR` is set to one) then the application should use the `MODE SELECT` command to set the `FSSZT` field in the `Flex Control` mode page before issuing these commands.

The `Number Of Zones` operation parameter is not transported by the `FLEX CONVERT(16)` and `FLEX QUERY (16)`; instead, the `FSNOZ` device parameter is used. The application should use the `MODE SELECT` command to set the `FSNOZ` field in the `Flex Control` mode page before issuing these commands.

Table 13 -- FLEX CONVERT (16) and FLEX QUERY (16) commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (D5h for FLEX CONVERT (16) and D6h for FLEX QUERY (16))							
1	ALL	DIRECTION		Reserved				
2 … 9	STARTING ZONE LOCATOR							
10 … 13	ALLOCATION LENGTH							
14	Reserved							
15	CONTROL							

Table 14 -- FLEX CONVERT (32) and FLEX QUERY (32) commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2 … 6	Reserved							
7	ADDITIONAL CDB LENGTH (18h)							
8 9	SERVICE ACTION (F800h for FLEX CONVERT, F801h for FLEX QUERY)							
10	ALL	Reserved					DIRECTION	
11	Reserved							
12 … 19	STARTING ZONE LOCATOR							
20 … 23	NUMBER OF ZONES							
24 … 27	Reserved							
28 … 31	ALLOCATION LENGTH							

The OPERATION CODE, ADDITIONAL CDB LENGTH and SERVICE ACTION field are defined in SPC-5 and shall be set to the values shown in the tables above.

[Editor's Note: The OPERATION CODE values for FLEX CONVERT (16) and FLEX QUERY (16) and the SERVICE ACTION values for FLEX CONVERT (32) and FLEX QUERY (32) are in a vendor specific range.]

The CONTROL byte is defined in SAM-5.

The ALL bit specifies the All parameter of the operation request.

The DIRECTION field (see the following table) specifies the Direction parameter of the operation request.

Table 15 -- DIRECTION field

Code	Direction
0h	FROM CMR TO SMR
1h	FROM SMR TO CMR
2h..3h	Reserved

The STARTING ZONE LOCATOR field specifies the Starting Zone Locator parameter of the operation request.

The NUMBER OF ZONES field specifies the Number Of Zones operation parameter of the operation request for FLEX CONVERT (32) and FLEX QUERY (32). commands.

The ALLOCATION LENGTH field is defined in SPC-5.

Operation input parameters are defined in clause 3.3.2.4.

5.1.1.1. Flex Convert and Flex Query operation invocation

The Flex Convert and Flex Query operations are invoked by this command as follows.

```
if OPERATION CODE = D5h or OPERATION CODE = D56h
      NumberOfZones=FSNOZ
else
      NumberOfZones=NUMBER OF ZONES
if (OPERATION CODE = D5h) or (OPERATION CODE = 7Fh and SERVICE ACTION =
   F800h)    // FLEX CONVERT (16) or FLEX CONVERT (32)
      FlexConvert(Direction=DIRECTION, All=ALL,
         StartingZoneLocator=STARTING ZONE LOCATOR, NumberOfZones,
         SmrZoneType=FSSZT)
else    // FLEX QUERY (16) or FLEX QUERY (32)
      FlexQuery(Direction=DIRECTION, All=ALL,
         StartingZoneLocator=STARTING ZONE LOCATOR, NumberOfZones,
         SmrZoneType=FSSZT)
```

5.1.1.2. Parameter data

The FLEX CONVERT and FLEX QUERY parameter data are defined in the following table, and transports the returned data of the operation.

Table 16 -- FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) parameter data

Bit Byte	7	6	5	4	3	2	1	0
0 … 3	EXTENT LIST LENGTH (n-31)							
4	SZTBV	SZSV	FZTBZL	FZTBV	FZSV	SEHMZT	NOZIZ	SZLBV
5	Converted	Reserved						SZTBZL
6	Smr Zone Type							
7	Reserved		All	Reserved			Direction	
8	Reserved				Number Of Zones			
… 11								
12 … 31	Reserved							
32 … 47	Extent descriptor [first]							
	...							
n-15 … n	Extent descriptor [last]							

The EXTENT LIST LENGTH field shall contain the length in bytes of the extent descriptors list; that is, Extent descriptor (first) through Extent descriptor (last in the table above) or n-31. The extent list length is the number of extents reported multiplied by 16. This value may indicate bytes that are not returned due to truncation based on the value of the ALLOCATION LENGTH field.

The SZLBV bit indicates the SZLBV bit of the operation returned data.

The NOZIZ bit indicates the NOZIZ bit of the operation returned data.

The SEHMZT bit indicates the SEHMZT bit of the operation returned data.

The FZSV bit indicates the FZSV bit of the operation returned data.

The FZTBV bit indicates the FZTBV bit of the operation returned data.

The FZTBZL bit indicates the FZTBZL bit of the operation returned data.

The SZSV bit indicates the SZSV bit of the operation returned data.

The SZTBV bit indicates the SZTBV bit of the operation returned data.

The SZTBZL bit indicates the SZTBZL bit of the operation returned data.

The Converted bit indicates the Converted bit of the operation returned data.

The Smr Zone Type field indicates the Smr Zone Type field of the operation returned data.

The DIRECTION field is the Direction field of the operation returned data.

The ALL bit is the All bit of the operation returned data.

The Number Of Zones field indicates the Number Of Zones field of the operation returned data.

Operation return data parameters are defined in clause 3.3.2.6.

The extent descriptors list contains extent descriptors of the operation returned data. Each extent descriptor (see the table below) contains the description of a single extent. The descriptors shall be sorted in ascending order based on the STARTING ZONE LOCATOR field of each extent descriptor.

Table 17 -- FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) extent descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			
2 3	Reserved							
4 … 7	NUMBER OF ZONES							
8 … 15	STARTING ZONE LOCATOR							

The ZONE TYPE field indicates the Zone Type of this extent descriptor.

The ZONE CONDITION field indicates the Zone Condition of this extent descriptor.

The NUMBER OF ZONES field indicates the Number Of Zones of this extent descriptor.

The START ZONE LOCATOR field indicates the Starting Zone Locator of this extent descriptor.

Extent descriptor parameters Number Of Zones and Starting Zone Locator are defined in clause 3.3.2.6.

[Editor’s note: These commands may return status other than GOOD based on reasons outside the scope of this specification. Other specifications that may specify other status values include SPC, SBC and ZBC; otherwise, they return with good success status even if there are status error bits set in the data returned by the Flex Convert or Flex Query operation.]

5.2. New Parameters

[Editor’s note: There are additional considerations for mode parameter block descriptors that need to be specified.]

5.2.1. Flex Control mode page

The Flex Control mode page (see the table below) provides controls of Flex operations.

If:

a) the FSWP bit (see 5.2.2) is cleared to zero and the FSSZT field specifies SEQUENTIAL WRITE PREFERRED; or

b) the FSWR bit (see 5.2.2) is cleared to zero and the FSSZT field specifies SEQUENTIAL WRITE REQUIRED

then the device server shall terminate the MODE SELECT command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 18 -- Flex Control mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (3Dh)					
1	SUBPAGE CODE (08h)							
2 3	PAGE LENGTH (01FDh)							
4	Reserved				FSNOZ			
… 7	FSNOZ							
8	FSSZT							
9 … 511	Reserved							

The PS bit is described in SPC-5.

The SPF bit, PAGE CODE field, SUBPAGE CODE field, and PAGE LENGTH field are described in SPC-5 and shall be set as shown in the table above for the Flex Control mode page.

[Editor's Note: The PAGE CODE and SUBPAGE CODE values are in vendor specific ranges.]

The FSSZT (Flex Subsequent SMR Zone Type) field specifies the setting for the FSSZT device parameter. The FSSZT device parameter applies to SMR zones only and is used by Flex Convert and Flex Query operations when Direction = From Cmr To Smr. The setting shall persist across soft reset, hard reset and power cycles. The following table describes the FSSZT field.

Table 19 -- FSSZT field

Code	Description
0h..1h	Reserved
2h	SEQUENTIAL WRITE REQUIRED
3h	SEQUENTIAL WRITE PREFERRED
All others	Reserved

The FSNOZ (Flex Subsequent Number Of Zones) field specifies the setting for the FSNOZ device parameter. The setting of the FSNOZ device parameter shall not be affected by a power-on reset, a hardware reset, or a software reset.

[Editor's Note: The FSNOZ field is 28 bits to match the size of the FSNOZ fields in ATA SET FEATURES, Flex Control subcommand and ATA Flex Capabilities log. Matching the sizes simplifies SCSI to ATA translation.]

5.2.2. Flex Capabilities VPD page

The Flex Capabilities VPD page, page E1h, (see the following table) indicates the capabilities of the Flex devices.

Table 20 -- Flex Capabilities VPD page

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (00E1h)							
2 3	PAGE LENGTH (003Ch)							
4 5	Page revision number							
6 7	Reserved							
8	Reserved				FSWP	FSWR	URFZ	Reserved
9	FLEX VERSION							
10 … 23	Reserved							
24 … 31	FMRmax							
32 … 39	SMRmin							
40 … 47	SMRmax							
48 … 63	Reserved							

An URFZ (Unrestricted Read in FMR Zones) bit cleared to zero indicates that the device does not support reading unwritten logical sectors in FMR zones. A URFZ bit set to one indicates that the device supports reading unwritten logical sectors in FMR zones.

A FSWP (Flex Sequential Write Preferred) bit set to one indicates that the device supports SMR zones of the sequential write preferred SMR zone type.

A FSWR (Flex Sequential Write Required) bit set to one indicates that the device supports SMR zones of the sequential write required SMR zone type.

The FLEX VERSION field indicates the version of the Flex Protocol which this device implements. Devices that implement the version defined in this document shall set this field to 00h.

The FMRmax field is defined in the FMR zone model.

The SMRmin and SMRmax fields are defined in the SMR zone model.

5.3. Device Type Reporting

5.3.1. PERIPHERAL DEVICE TYPE field

The value of the PERIPHERAL DEVICE TYPE field for a Flex drive is 00h, direct access block device.

[Editor's Note: This value of the PERIPHERAL DEVICE TYPE field is required for Flex drives that are 100% CMR to be used by hosts that are not aware of the Flex feature set.]

5.3.2. ZONED field

Flex devices shall set the ZONED field to 00b (not reported) as with drive managed and legacy (non-ZAC) devices. This applies to the Block Device Characteristics VPD page.

[Editor's Note: This value of the ZONED field is required for Flex drives that are 100% CMR to be used by hosts that are not aware of the Flex feature set.]

5.3.3. Supported capabilities

Flex drives set the FLEX SUPP bit to one, and set the FLEX VERSION field to the appropriate value.

5.4. Extensions to SBC-4 and ZBC

In this section indented items indicate changes to existing contents of SBC-4 r16 and ZBC. The section and table numbers are from the respective standard and are disambiguated by being prefixed with the respective document name, and references to other clauses in this document are disambiguated by being prefixed with FDI.

Existing text is in grey, new or changed text is in blue, and ellipses show omitted unchanged content.

5.4.1. Extensions to definitions, symbols, abbreviations, and conventions

Extensions to SBC-4 and ZBC definitions, symbols, abbreviations, and conventions are not yet specified.

[Editor’s note: this is work to be done.]

5.4.2. Extensions to zoned block device model

Extensions to the ZBC zoned block device model are not yet specified.

[Editor’s note: this is work to be done.]

5.4.3. Extensions to commands

The new Flex commands are implemented as commands for direct access devices.

> SBC-4 5.1 Commands for direct access block devices overview
>
> The command for direct access block devices are listed in SBC-4 table 35.

SBC-4 Table 35 - Commands for direct access block devices

Command	Operation Code	Type	LBACT	Reference
...	...	...	...	...
EXTENDED COPY	83h/01h	O	n/a	SPC-5
FLEX CONVERT (16)	D5h	O	n/a	FDI 5.1.1
FLEX CONVERT (32)	7Fh/F800h	O	n/a	FDI 5.1.1
FLEX QUERY (16)	D6h	O	n/a	FDI 5.1.1
FLEX QUERY (32)	7Fh/F801h	O	n/a	FDI 5.1.1
FINISH ZONE	94h/02h	X	n/a	ZBC
...	...	...	...	...

...

5.4.3.1. Extensions to REPORT ZONES

5.4.3.1.1. Support for a compact form of zone descriptors

A T10 proposal 17-125 defines extensions to the `REPORT ZONES` command for a new zone descriptor format.

[Editor’s note: This proposal has not yet been approved by T10.]

5.4.3.1.2. Support for listing FMR zones and SMRmax

> ZBC 5.6 `REPORT ZONES` command
>
> ...

ZBC 5.6.2 REPORT ZONES parameter data

...

The MAXIMUM LBA field contains the LBA of the last logical block on the logical unit. If FLEX SUPP is set to one, this field specifies SMRmax as defined in the zone model.

…

The ZONE TYPE field indicates the Zone Type (see ZBC 4.3.2) of [the] zone as described in ZBC table 25.

ZBC Table 25 - Zone Type field

Code	Zone Type
...	...
4h	FMR
5h..Fh	Reserved

…

5.4.3.2. Extensions to RESET WRITE POINTER

ZBC 5.7 RESET WRITE POINTER command

...

If the ALL bit is set to zero, then the ZONE ID field specifies the lowest LBA of the write pointer zone on which the device server shall perform a reset write pointer operation. If the ALL bit is set to one, then the device server shall ignore the ZONE ID field.

An ALL bit set to one specifies that the device server shall perform a reset write pointer operation (see ZBC 4.4.3.2.5) on each sequential write preferred zone and sequential write required zone with a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, FULL, or CLOSED.

...

[Editor’s note: this change does not change existing ZBC-compliant behavior, but does specify that the ALL bit does not affect FMR zones. The motivation is to allow a Flex drive to be configured with some FMR zones and some SMR zones, and then be used by a Host Aware or Host Managed application and particularly have that application able to issue “Reset All” with the same result as a Host Aware or Host Managed device.]

5.4.3.3. Extensions to READ CAPACITY (16)

[Editor’s Notes:

The READ CAPACITY (16) fields are extended to report FMRmax.

Also, the Flex Capabilities VPD page has SMRmin and SMRmax as well as FMRmax. From these values the 100% CMR capacity and 100% SMR capacity can be directly calculated, at least if there are no non-Flex reasons that zones are offline.

Another value that could be of interest is the current online capacity. A field could be added for this in the Flex Capabilities VPD page. Another possibility is to define a Reporting Options code for REPORT ZONES that specifies online zones are to be returned, thus allowing a calculation of online capacity based on the calculated number of online zones.]

This section specifies extensions to the READ CAPACITY (16) command. This extension allows Flex drives to be used in their 100% CMR line configuration in environments that have no support for this protocol.

> ZBC 5.5 READ CAPACITY (16) command
> ZBC 5.5.1 READ CAPACITY (16) command overview
>
> The READ CAPACITY (16) command is defined in SBC-4.
>
> For a zoned block device and a Flex device, the READ CAPACITY (16) parameter data is defined in ZBC 5.5.2
> …

5.4.3.4. Extensions to zone management commands

Flex drives shall support new extensions to the following existing commands allow a zone range to be specified.

- CLOSE ZONES
- FINISH ZONES
- OPEN ZONES
- RESET WRITE POINTERS

Refer to T10 proposal 17-124.

[Editor's Note: One significant use case is using this form of RESET WRITE POINTERS to request a Reset Write Pointer operation for an extent in a single command as part of a conversion. This is in contrast to hundreds of commands that each specify a single zone as part of conversion of extents on the order of 100 GB.]

5.4.4. Extensions to parameters for zoned block devices

5.4.4.1. Extensions to mode parameters

> ZBC 6.3 Mode parameters
>
> The mode pages and their corresponding page codes and subpage codes for Flex devices are defined in ZBC table x, SPC-5 and SBC-4.

ZBC Table x -- Mode page codes and subpage codes for Flex devices

Mode page name	Page code	Subpage code	Reference
Flex Capabilities	3Dh	08h	FDI 5.2.1

The mode pages and their corresponding page codes and subpage codes for host aware zoned block devices are defined in SPC-5 and SBC-4.

...

5.4.4.2. Extensions to vital product data (VPD) parameters

5.4.4.2.1. Extensions to Block Device Characteristics VPD page

ZBC 6.4 Vital product data (VPD) parameters

ZBC 6.4.1 VPD parameters overview

The VPD pages and their corresponding page codes for Flex devices are defined in ZBC table 31, SPC-5, and SBC-4.

The VPD pages and their corresponding page codes for host aware zoned block devices are defined in ZBC table 31, SPC-5, and SBC-4.

The VPD pages and their corresponding page codes for host managed zoned block devices are defined in ZBC table 31 and in SPC-5.

ZBC Table 31 — VPD page codes for zoned block devices

VPD page name	Page code (a)	Reference	Host managed zoned block device support requirements (b)	Host aware zoned block device support requirements (b)	Flex device support requirements (b)
ATA Information	89h	SAT-4	See SAT-4	See SAT-4	See SAT-4
Block Device Characteristics	B1h	SBC-4	mandatory	mandatory	mandatory
Block Device Characteristics Extension	B5h	SBC-4	optional	optional	optional
Block Limits	B0h	SBC-4	mandatory	mandatory	mandatory
Flex Capabilities	E1h	FDI 5.2.2			mandatory
Supported Block Lengths And Protection Types	B4h	SBC-4	optional	optional	optional
Zoned Block Device Characteristics	B6h	ZBC 6.4.2	mandatory	mandatory	mandatory

(a) All page codes not shown in this table or SPC-5 are reserved for host managed zoned block devices.
(b) Support requirements defined in this table take precedence over support requirements defined in SBC-4.

...

SBC-4 6.6.2 Block Device Characteristics VPD page

The Block Device Characteristics VPD page (see table 259) contains parameters indicating characteristics of the logical unit.

SBC-4 Table 259 -- Block Device Characteristics VPD page

Bit Byte	7	6	5	4	3	2	1	0
...	...							
8	~~Reserved~~ FLEX SUPP	Reserved	ZONED		RBWZ	BOCS	FUAB	VBULS
...	...							

...

The NOMINAL FORM FACTOR field indicates the nominal form factor of the device containing the logical unit and is defined in table 264.

...

A Flex supported (FLEX SUPP) bit set to one indicates that the Flex feature set (see FDI 4.1.1) is supported. A FLEX SUPP bit set to zero indicates that the Flex feature set is not supported.

The ZONED field indicates the type of zoned block capabilities implemented by the device server as defined in table 265.

...

6. SCSI to ATA Translation for Flex

This section specifies the SCSI to ATA translation for Flex including extensions to SAT-4 r6.

This section is in progress and not yet complete.

6.1. FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) commands

[Editor’s Note: This is modeled after SAT-4 9.13 REPORT ZONES command.]

6.1.1. Commands overview

The FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) commands return parameter data that includes descriptors of a set of extents on the device. The FLEX CONVERT (16) and FLEX CONVERT (32) commands request~~s~~ that the device server perform a Flex conversion operation. The FLEX QUERY (16) and FLEX QUERY (32) commands request~~s~~ that the device server perform a Flex query operation. These commands are applicable to ATA Flex devices. If the ATA FLEX CONVERT NON-QUEUED, FLEX CONVERT QUEUED, FLEX QUERY NON-QUEUED and FLEX QUERY QUEUED commands are not supported by the device (i.e., if the FLEX SUPP bit is cleared to zero in the IDENTIFY DEVICE data log), then the command~~s~~ shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE.

The following table shows the translation for fields in the FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) CDBs.

Table 21 -- FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) field translations

Field	Description
OPERATION CODE	Set to D5h for FLEX CONVERT (16) Set to D6h for FLEX QUERY (16) Set to 7Fh for FLEX CONVERT (32) and FLEX QUERY (32)
SERVICE ACTION	Set to F800h for FLEX CONVERT (32) Set to F801h for FLEX QUERY (32)
DIRECTION	The SATL shall set the ATA DIRECTION field in the ATA command to the value specified by the DIRECTION field.
ALL	The SATL shall set the ATA ALL bit in the ATA command to the value specified by the ALL bit.
STARTING ZONE LOCATOR	If the SATL implements direct logical block mapping (see SAT-5 9.1.2), then the SATL shall set the ATA STARTING ZONE LOCATOR field in the ATA command to the value specified in the STARTING ZONE LOCATOR field. If the SATL does not implement direct logical block mapping, then this field is unspecified.
NUMBER OF ZONES	For FLEX CONVERT (32) and FLEX QUERY (32) the SATL shall set the ATA AUXNOZ field in the ATA command to the value specified by the NUMBER OF ZONES field.
ALLOCATION LENGTH	The SATL shall send the ATA FLEX CONVERT or FLEX QUERY command with the ATA RETURN PAGE COUNT field set to INT((ALLOCATION LENGTH + 511)/512).
CONTROL	See SAT-4 6.5.

6.1.2. Command processing

A FLEX CONVERT (16) or FLEX CONVERT (32) command is translated to an ATA FLEX CONVERT NON-QUEUED or FLEX CONVERT QUEUED command. A FLEX QUERY (16) or FLEX QUERY (32) command is translated to an ATA FLEX QUERY NON-QUEUED or FLEX QUERY QUEUED command. The fields in the ATA commands are set as described in the tables above and below.

Table 22 -- ATA FLEX CONVERT NON-QUEUED, FLEX CONVERT QUEUED, FLEX QUERY NON-QUEUED and FLEX QUERY QUEUED field settings

ATA field	Setting
ALL	See the table above.
NOZSRC	For FLEX CONVERT (16) and FLEX QUERY (16) the SATL shall set the ATA NOZSRC field to zero. For FLEX CONVERT (32) and FLEX QUERY (32) the SATL shall set the ATA NOZSRC field to one.
DIRECTION	See the table above.
RETURN PAGE COUNT	See the table above.
STARTING ZONE LOCATOR	See the table above.
AUXNOZ	For FLEX CONVERT (16) and FLEX QUERY (16) ,the setting of the ATA AUXNOZ field is ignored by the ATA device. For FLEX CONVERT (32) and FLEX QUERY (32) see the table above.
COMMAND	See the table above.

If the ATA command completes without error, then:

1) the parameter data are translated as described below;
2) the translated parameter data are transferred to the initiator; and
3) the SCSI FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) or FLEX QUERY (32) command is completed with GOOD status.

If the ATA command completes with an error, then the SCSI FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) or FLEX QUERY (32) command is terminated with CHECK CONDITION status and sense data as described in SAT-4 clause 11.

6.1.3. Parameter data

The translation of the FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) parameter data are defined in the following table.

Table 23 -- FLEX CONVERT (16), FLEX CONVERT (32), FLEX QUERY (16) and FLEX QUERY (32) parameter data translation

Field or Bit	Description
EXTENT LIST LENGTH	This field shall be set to the contents of the ATA EXTENT LIST LENGTH field in the ATA parameter data.
SZLBV	This bit shall be set to the contents of the ATA VSZLBV field in the ATA parameter data.
NOZIZ	This bit shall be set to the contents of the ATA NOZIZ field in the ATA parameter data.
SEHMZT	This bit shall be set to the contents of the ATA SEHMZT field in the ATA parameter data.
FZSV	This bit shall be set to the contents of the ATA FZSV field in the ATA parameter data.
FZTBV	This bit shall be set to the contents of the ATA FZTBV field in the ATA parameter data.
FZTBZL	This bit shall be set to the contents of the ATA FZTBZL field in the ATA parameter data.
SZSV	This bit shall be set to the contents of the ATA SZSV field in the ATA parameter data.
SZTBV	This bit shall be set to the contents of the ATA SZTBV field in the ATA parameter data.
SZTBZL	This bit shall be set to the contents of the ATA SZTBZL field in the ATA parameter data.
Converted	This bit shall be set to the contents of the ATA Converted field in the ATA parameter data.
Direction	This field shall be set to the contents of the ATA Direction field in the ATA parameter data.
All	This bit shall be set to the contents of the ATA ALL bit in the ATA parameter data.
Number Of Zones	This field shall be set to the contents of the ATA Number Of Zones field in the ATA parameter data.
ZONE TYPE [a]	This field shall be set to the contents of the ATA ZONE TYPE field in the ATA parameter data.
ZONE CONDITION [a]	This field shall be set to the contents of the ATA ZONE CONDITION field in the ATA parameter data.
NUMBER OF ZONES [a]	This field shall be set to the contents of the ATA ZONE NUMBER OF ZONES field in the ATA parameter data.
STARTING ZONE LOCATOR [a]	This field shall be set to the contents of the ATA STARTING ZONE LOCATOR field in the ATA parameter data.
[a] These fields are replicated in each extent descriptor.	

The number of bytes returned shall be the smaller of:

a) 32 plus the contents of the EXTENT LIST LENGTH field in the translated parameter data; and
b) the contents of the ALLOCATION LENGTH field in the CDB.

6.2. Flex Control Mode Page

This section would be added to SAT-4 10.4.

The Flex Control mode page defines parameters that affect the behavior of Flex operations in the ATA device. The following table shows the translation of fields in the Flex Control mode page.

Table 24 -- Flex Control mode page fields

Field	Description
PS	Unspecified
SPF	Shall be set to one
PAGE CODE	Shall be set to 3Dh
SUBPAGE CODE	Shall be set to 08h
PAGE LENGTH	Shall be set to 01FDh
FSSZT	If processing a MODE SENSE command, then the SATL shall determine the setting of the FSSZT device parameter from ATA IDENTIFY DEVICE data log, Flex Capabilities data log page. If processing a MODE SELECT command, then the SATL shall issue an ATA SET FEATURES command, Flex Control subcommand with the FSSZT field set to: a) SEQUENTIAL WRITE PREFERRED if the value of this field is set to SEQUENTIAL WRITE PREFERRED, or b) SEQUENTIAL WRITE REQUIRED if the value of this field is set to SEQUENTIAL WRITE REQUIRED.
FSNOZ	If processing a MODE SENSE command, then the SATL shall determine the setting of FSNOZ from ATA IDENTIFY DEVICE data log,Flex Capabilities data log page. If processing a MODE SELECT command, then the SATL shall issue an ATA SET FEATURES command, Flex Control subcommand with FSNOZ set to the value of this field.

6.3. Flex Capabilities VPD Page

This section would be added to SAT-4 10.5.

The `Flex Capabilities` VPD page indicates Flex capabilities in the ATA device. The following table shows the translation of fields in the `Flex Capabilities` VPD page.

Table 25 -- `Flex Capabilities` VPD page fields

Field	Description
`URFZ`	This field shall be set to the contents of the ATA `URFZ` field in the ATA `Flex Capabilities` log.
`FSWR`	This field shall be set to the contents of the ATA `FSWR` field in the ATA `Flex Capabilities` log.
`FSWP`	This field shall be set to the contents of the ATA `FSWP` field in the ATA `Flex Capabilities` log.
`FLEX VERSION`	This field shall be set to the contents of the ATA `FLEX VERSION` field in the ATA `Flex Capabilities` log.
`FMRmax`	This field shall be set to the contents of the ATA `FMRmax` field in the ATA `Flex Capabilities` log.
`SMRmin`	This field shall be set to the contents of the ATA `SMRmin` field in the ATA `Flex Capabilities` log.
`SMRmax`	This field shall be set to the contents of the ATA `SMRmax` field in the ATA `Flex Capabilities` log.

6.4. Extensions to SAT-4

In this section indented items indicate changes to existing contents of SAT-4 r06. The section and table numbers are from the respective standard and are disambiguated by being prefixed with the respective document name, and references to other clauses in this document are disambiguated by being prefixed with FDI.

Existing text is in grey, new or changed text is in blue, deleted text in ~~red strikethrough~~ and ellipses show omitted unchanged content.

6.4.1. Extensions to parameters for SAT implementations

6.4.1.1. Extensions to vital product data parameters

6.4.1.1.1. Extensions to Block Device Characteristics VPD Page

> ### SAT-4 10.5.8 Block Device Characteristics VPD page
>
> SAT-4 table 157 shows the translation of fields in the Block Device Characteristics VPD page.
>
> SAT-4 Table 157 -- Block Device Characteristics VPD page field translations

Field	Description
...	...
NOMINAL FORM FACTOR	Shall be set to the value contained in the ATA NOMINAL FORM FACTOR field in the ATA IDENTIFY DEVICE data log
FLEX SUPP	The SATL shall set this bit to the value contained in the ATA FLEX SUPP bit in the Supported Capabilities data log page (page 03h of ATA IDENTIFY DEVICE data log (log address 30h))
ZONED	The SATL shall set this field to the value contained in the ATA ZONED field in the ATA IDENTIFY DEVICE data log
...	...

7. Examples

The following examples do not include security considerations such as TCG Band configurations and crypto erase of data written to LBAs before those LBAs go offline.

7.1. Example Flex Drive

Consider an example Flex drive that is 12 TB as 100% CMR and 15% more capacity as SMR. Logical blocks (LBAs) are 512 bytes. Both spaces are organized as 256-MiB zones which are the atomic unit of conversion; that is, each zone is either fully offline or online and thus at each seam there is typically some media that is not provisioned because it cannot supply a full zone.

In these descriptions of the layout, note that zone number is an abstraction that does not appear in the interface.

7.1.1. Example zone layout

This example drive has a 100% CMR capacity that complies with SFF-8447 for a 12-TB drive. The CMR LBA space is organized as 44,704 FMR zones. FMR zones boundaries are detailed in the following table.

Table 26 -- FMR zones for the example drive

Zone number	ZoneStartLBA
0	0 [0 0000 0000h]
1	524,288 [0 0008 0000h]
2	1,048,576 [0 0010 0000h]
...	...
44,699	23,435,149,212 [5 74D8 0000h]
44,700	23,435,673,600 [5 74E0 0000h]
44,701	23,436,197,888 [5 74E8 0000h]
44,702	23,436,722,176 [5 74F0 0000h]
44,703	23,437,246,464 [5 74F8 0000h]

The SMR LBA space is organized as 51,409 zones, each of which is 256 MiB or almost 13.8 TB. SMR zones boundaries are detailed in the following table.

Table 27 -- SMR zones for the example drive

Zone number	ZoneStartLBA
65,536	34,359,738,368 [8 0000 0000h]
65,537	34,360,262,656 [8 0008 0000h]
65,538	34,360,786,944 [8 0010 0000h]
...	...
116,942	61,311,287,296 [E 4670 0000h]
116,943	61,311,811,584 [E 4678 0000h]
116,944	61,312,335,872 [E 4680 0000h]

The following figures show the physical correspondence of CMR and SMR zones.

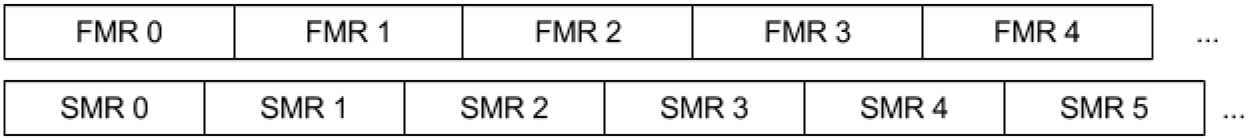


Figure 7: Zone correspondence for the example drive at lower CMR LBAs

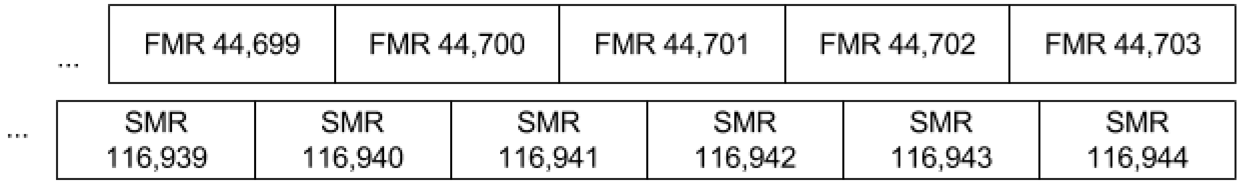


Figure 8: Zone correspondence for the example drive at higher SMR LBAs

7.1.2. Unprovisioned media at a seam

The preceding figures visually expose how same-sized FMR zones lead to an sub-optimal capacity due to some media remaining unprovisioned at a seam. For instance, the following figure shows a configuration with SMR zones 51,405 through 51,408, inclusive, as the only online SMR zones.

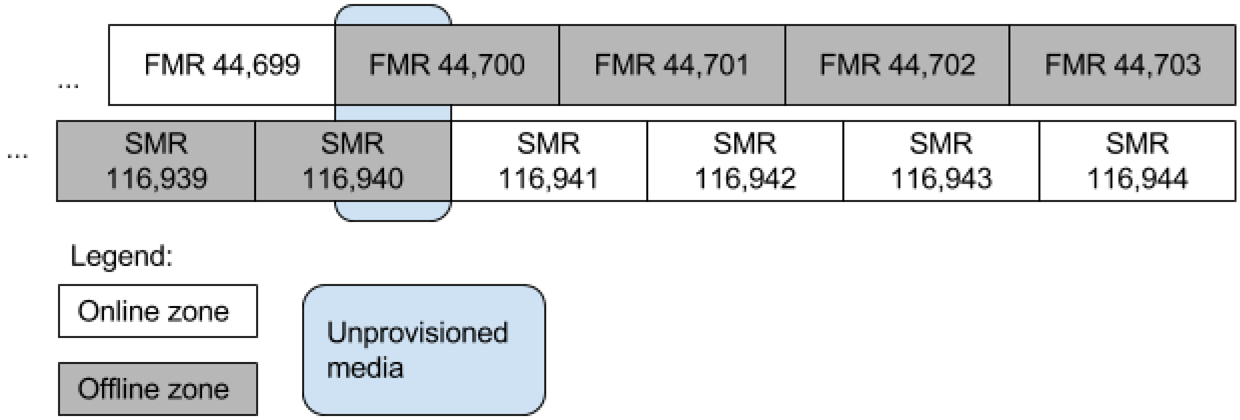


Figure 9: Example of unprovisioned media at a seam

The amount of unprovisioned media due to zone granularity is the size of an FMR zone, and the average amount is one-half the size of an FMR zone. With 256-MiB FMR zones and 10 seams per TB the average magnitude of the impact of same-sized zones and zone granularity conversions is 128 MiB per 100 GB or less than 0.14%.

7.1.3. Example drive configurations

A selected set of configurations that just exceed a multiple of 100 GB of SMR capacity are shown in the following table.

Table 28 -- Example configurations at 100 GB SMR increments

SMR online capacity (GB)	CMR online capacity (GB)	Total capacity (GB)	Online SMR zones	Online FMR zones	SMR portion
0.00	12,000.14	12,000.14	0	44,704	0.00%
100.13	11,912.90	12,013.02	373	44,379	0.83%
200.25	11,825.92	12,026.18	746	44,055	1.67%
300.11	11,738,95	12,039.06	1,118	43,731	2.49%
400.24	11,651.98	12,052.22	1,491	43,407	3.32%
500.10	11,565.00	12,065.10	1,863	43,083	4.14%
600.22	11,478.03	12,078.25	2,236	42,759	4.97%
700.08	11,391.33	12,091.41	2,608	42,436	5.79%
800.21	11,304.09	12,104.29	2,981	42,111	6.61%
900.06	11,217.38	12,117.44	3,353	41,788	7.43%
1000.19	11,130.14	12,130.33	3,726	41,463	8.25%

A selected set of configurations that just exceed a multiple of 10% SMR are shown in the following table.

Table 29 -- Example configurations at 10% increments

SMR portion	Online SMR zones	Online FMR zones	SMR online capacity (GB)	CMR online capacity (GB)	Total capacity (GB)
0%	0	44,704	0	12,000.14	12,000.14
10.001%	4,530	40,704	1,216.01	10,942.77	12,158.78
20.001%	9,180	36,721	2,464.24	9,857.22	12,312.46
30.001%	13,958	32,567	3,746.82	8,742.14	12,488.96
40.001%	18,866	28,298	5,064.30	7,596.19	12,660.49
50.001%	23,912	23,911	6,418.83	6,418.56	12,837.39
60.001%	29,100	19,399	7,811.47	5,207.38	13,018.85
70.001%	34,437	14,758	9,244.11	3,961.57	13,205.68
80.001%	39,930	9,982	10,718.63	2,679.52	13,398.15
90.002%	45,585	5,064	12,236.63	1,359.36	13,595.99
100%	51,409	0	13,800.0	0	13,800.0

7.2. Example Conversions

For these examples, the setting of the `FSSZT` device parameter is `Sequential Write Required`.

7.2.1. Converting the first FMR zone to offline

This example is a scenario in which taking the first FMR zone online brings the first SMR zone online. The following diagram shows the physical correspondence of the zones for this example.

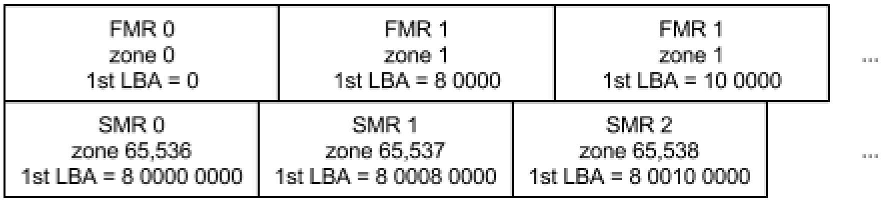


Figure 10: Example correspondence of zones
Zone numbers are in decimal, LBAs are in hex.

From a starting state of 100% CMR and all SMR zones offline, to take the first FMR zone offline and bring SMR space online, use the following operation sequence:

```
FlexQuery(Direction=FromCmrToSmr, All=false, StartingZoneLocator=0,
    NumberOfZones=1)
ResetWritePointer(ResetAll=false, ZoneId=0, ZoneCount=1)
FlexConvert(Direction=FromCmrToSmr, All=false, StartingZoneLocator=0,
    NumberOfZones=1)
```

The `Flex Query` and `Flex Convert` operations both return:

```
All status bits = 0
Direction = From Cmr To Smr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 0
      Number Of Zones = 1
Extent Descriptor (last)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = 8 0000 0000h
      Number Of Zones = 1
```

As the return data for the Flex Query operation, this indicates that zone 0 (starting at LBA 0) can convert to zone 65,536 (starting at LBA 8 0000 0000h). As the return data for the Flex Convert operation, this indicates that zone 0 was converted to zone 65,536.

The result of the Flex Convert operation is no change to the drive state.

The result of the Reset Write Pointer operation is zone 0 changes to the Empty state, a prerequisite for conversion.

The result of the Flex Convert operation is zone 0 is unprovisioned, goes offline, and changes to the Offline state, and zone 65,536 is provisioned as SMR, comes online, and changes to the Empty state.

7.2.2. Converting the second SMR zone to online

This example is a scenario in which bringing an SMR zones online leaves unprovisioned media that could provision a second SMR zone, but this second SMR zone does not come only. The diagram in the previous section shows the physical correspondence of the zones for this example.

From a starting state of 100% CMR and all SMR zones offline, to bring the second SMR zone online, use the following operation sequence:

```
FlexQuery(Direction=FromCmrToSmr, All=false,
    StartingZoneLocator=8 0008 0000, NumberOfZones=1)
ResetWritePointer(ResetAll=false, ZoneId=0, ZoneCount=2)
FlexConvert(Direction=FromCmrToSmr, All=false,
    StartingZoneLocator=8 0008 0000, NumberOfZones=1)
```

The Flex Query and Flex Convert operations both return:

```
All status bits = 0
Direction = From Cmr To Smr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 0
      Number Of Zones = 2
Extent Descriptor (last)
```

```
        Zone Type = Sequential Write Required
        Zone Condition = Empty
        Starting Zone Locator = 8 0008 0000h
        Number Of Zones = 1
```

As the return data for the Flex Query operation, this indicates that zones 0 (starting at LBA 0) and 1 can convert to zone 65,537 (starting at LBA 8 0008 0000h). As the return data for the Flex Convert operation, this indicates that zones 0 and 1 went offline and zone 65,537 came online.

The result of the Flex Convert operation is no change to the drive state.

The result of the Reset Write Pointer operation is zones 0 and 1 change to the Empty state, a prerequisite for conversion.

The result of the Flex Convert operation is zones 0 and 1 are unprovisioned, go offline, and change to the Offline state; and zone 65,637 is provisioned as SMR, comes online, and changes to the Empty state.

7.2.3. Converting an SMR zone to online without an FMR zone going offline

This example is a scenario in which an SMR zone comes online without any FMR zone going offline. The diagram for the previous section shows the physical correspondence of the zones for this example.

From a starting state of the ending state of the previous section with the first two FMR zones offline, all other FMR zones online, the second SMR zone online, and all other SMR zones offline, use the following operation sequence:

```
FlexQuery(Direction=FromCmrToSmr, All=false,
    StartingZoneLocator=8 0000 0000, NumberOfZones=1)
FlexConvert(Direction=FromCmrToSmr, All=false,
    StartingZoneLocator=8 0000 0000, NumberOfZones=1)
```

The Flex Query and Flex Convert operations both return:

```
All status bits = 0
Direction = From Cmr To Smr
Extent Descriptor (first)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = 8 0000 0000h
      Number Of Zones = 1
```

As the return data for the Flex Query operation, this indicates that zone 65,536 (starting at LBA 8 0000 0000h) can come online without any modifications to any FMR zones. As the return data for the Flex Convert operation, this indicates that zone 65,536 came online. This indicates that no Reset Write Pointer operation is needed to transition an FMR zone to the Empty state.

The result of the Flex Convert operation is no change to the drive state.

The result of the `Flex Convert` operation is zone 65,536 is provisioned as SMR, comes online, and changes to the `Empty` state.

7.2.4. Converting the last SMR zone to online

From a starting state of 100% CMR and all SMR zones offline, to bring the last SMR zone online, use the following operation sequence:

```
FlexQuery(Direction=FromCmrToSmr, All=false,
   StartingZoneLocator=E 4680 0000h, NumberOfZones=1)
ResetWritePointer(ResetAll=false, ZoneId=5 74F8 0000h, ZoneCount=1)
FlexConvert(Direction=FromCmrToSmr, All=false,
   StartingZoneLocator=E 4680 0000h, NumberOfZones=1)
```

The `Flex Query` and `Flex Convert` operations both return:

```
All status bits = 0
Direction = From Cmr To Smr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 5 74F8 0000h
      Number Of Zones = 1
Extent Descriptor (last)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = E 4680 0000h
      Number Of Zones = 1
```

As the return data for the `Flex Query` operation, this indicates that zone 44,703 (starting at LBA 5 74F8 0000h) can convert to zone 116,944 (starting at LBA E 4680 0000h). As the return data for the `Flex Convert` operation, this indicates that zone 44,703 was converted to zone 116,944.

The result of the `Flex Convert` operation is no change to the drive state.

The result of the `Reset Write Pointer` operation is zone 44,703 changes to the `Empty` state.

The result of the `Flex Convert` operation is zone 44,703 is unprovisioned, goes offline, and changes to the `Offline` state, and zone 116,944 is provisioned as SMR, comes online, and changes to the `Empty` state.

Note that this conversion changes the last FMR zone to offline, but it does not change the reported capacity or the LBAs in the Report Zones header.

7.2.5. Converting the highest possible 100 GiB of SMR space to online while leaving the last FMR zone online

This example shows how to bring a specific amount SMR space online at high LBAs while leaving the highest number FMR zone online. There is a first query to determine which SMR zones have to stay offline to allow this FMR zone to stay online, and then a second query to determine which FMR zones have to come offline for the SMR capacity to come online.

From a starting state of 100% CMR and all SMR zones offline, to bring the 100 GiB of SMR space online at high LBAs while leaving the last FMR zone online, use the following operation sequence:

```
FlexQuery(Direction=FromCmrToSmr, All=false,
   StartingZoneLocator=5 74F8  0000h, NumberOfZones=1)
FlexQuery(Direction=FromCmrToSmr, All=false,
   StartingZoneLocator=E 39F8  0000h, NumberOfZones=400)
ResetWritePointer(ResetAll=false, ZoneId=5 6A08 0000h, ZoneCount=350)
FlexConvert(Direction=FromCmrToSmr, All=false,
   StartingZoneLocator=E 39F8 0000h, NumberOfZones=400)
```

The first `Flex Query` operation returns:

```
All status bits = 0
Direction = From Cmr To Smr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 5 74F8 0000h
      Number Of Zones = 1
Extent Descriptor (last)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = E 4678 0000h
      Number Of Zones = 2
```

The second `Flex Query` operation and the `Flex Convert` operation both return:

```
All status bits = 0
Direction = From Cmr To Smr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 5 6A08 0000h
      Number Of Zones = 350
Extent Descriptor (last)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = E 39F8 0000h
      Number Of Zones = 400
```

The return data for the first `Flex Query` operation indicates that, in this example, the last FMR zone, zone 44,703 starting at LBA 5 74F8 0000, can convert to two SMR zones starting at LBA E 4678 0000h which is zone 116,943.

This information is used to construct the second Flex Query operation to specify the 400 SMR zones that are the next lower in LBA space; thus, the query is constructed for 400 SMR zones starting at LBA E 39F8 0000h (where E 39F8 0000h = E 4678 0000h - (400 * 2^19)).

The return data for the second Flex Query operation indicates that, in this example, 350 FMR zones starting at LBA 5 6A08 0000h can convert to 400 SMR zones starting at LBA E 39F8 0000h.

This information is used to construct the Reset Write Pointer operation of 350 zones starting at LBA 5 6A08 0000h.

The result of the two Flex Convert operations is no change to the drive state.

The result of the Reset Write Pointers operation is 350 zones starting at LBA 5 6A08 0000h change to the Empty state.

The result of the Flex Convert operation is those 350 zones are unprovisioned, go offline, and change to the Offline state, and 400 zones are provisioned as SMR, come online, and change to the Empty state. The following figure and tables depict this configuration.

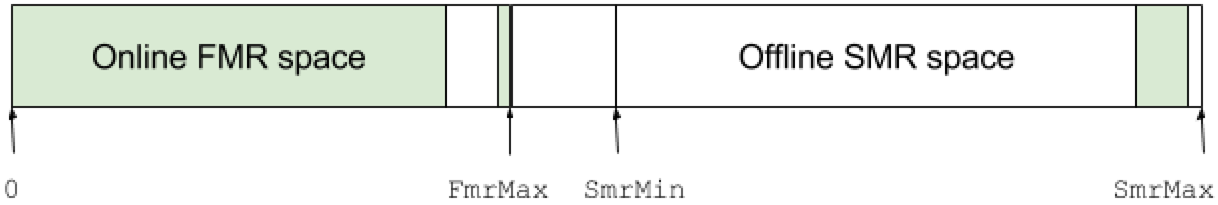


Figure 11: LBA layout for example 100 GiB of SMR space online followed single FMR zone online at the highest LBAs

Table 30 -- Seam locations for example 100 GiB of SMR space online followed single FMR zone online at the highest LBAs

	CMR space	SMR space
Minimum LBA	0	SMRmin = 8 0000 0000h
First seam	5 6A08 0000h	E 39F8 0000h
Second seam	5 74F8 0000h	E 4678 0000h
Maximum LBA	FMRmax = 5 74FF FFFFh	SMRmax = E 4687 FFFFh

Table 31 -- Online extents for example 100 GiB of SMR space online followed single FMR zone online at the highest LBAs

Meda stroke position	Online space size and type
Below the first seam	44,343 zones of CMR space, approximately 11,905.92 GB

From the first to second seam	400 zones of SMR space, exactly 100 GiB
Above the second seam	1 zone of CMR space, exactly 256 MiB

7.2.6. Converting the last two SMR zones to CMR

From the state where the last 373 SMR zones are online as with the ending state of the previous example, to take the last two SMR zones offline and bring CMR space online, use the following operation sequence.

```
FlexQuery(Direction=FromSmrToCmr, All=false,
   StartingZoneLocator=E 4678 0000h, NumberOfZones=2)
ResetWritePointer(ResetAll=false, ZoneId=E 4678 0000h, ZoneCount=2)
FlexConvert(Direction=FromSmrToCmr, All=false,
   StartingZoneLocator=E  4678 0000h, NumberOfZones=2)
```

The Flex Query and Flex Convert operations both return:

```
All status bits = 0
Direction = From Smr To Cmr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 5 74F8 0000h
      Number Of Zones = 1
Extent Descriptor (last)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = E 4678 0000
      Number Of Zones = 2
```

As the return data for the Flex Query operation, this indicates that zones 51,407 and 51,408 (starting at LBA E 4678 0000h) can convert to zone 44,703 (starting at LBA 5 74F8 0000h). As the return data for the Flex Convert operation, this indicates that zones 51,407 and 51,408 were converted to zone 44,703.

The result of the Flex Convert operation is no change to the drive state.

The result of the Reset Write Pointer operation is that zones 51,407 and 51,408 are in the Empty state.

The result of the Flex Convert operation is that zones 51,407 and 51,408 are unprovisioned, go offline, and changes to the Offline state, and zone 44,703 is provisioned as CMR, comes online, and changes to the Empty state.

7.2.7. Converting the next-to-last two SMR Zones 2 and 3 to CMR

From the state where all but the last two of the last 373 SMR zones are online as with the ending state of the previous example, to take the next-to-last two of SMR zones offline and bring CMR space online, use the following operation sequence.

```
FlexQuery(Direction=FromSmrToCmr, All=false,
    StartingZoneLocator=E 4668 0000h, NumberOfZones=2)
ResetWritePointer(ResetAll=false, ZoneId=E 4668 0000h, ZoneCount=2)
FlexConvert(Direction=FromSmrToCmr, All=false,
    StartingZoneLocator=E 4668 0000h, NumberOfZones=2)
```

The Flex Query and Flex Convert operations both return:

```
All status bits = 0
Direction = From Smr To Cmr
Extent Descriptor (first)
      Zone Type = FMR
      Zone Condition = Offline
      Starting Zone Locator = 5 74E8 0000h
      Number Of Zones = 2
Extent Descriptor (last)
      Zone Type = Sequential Write Required
      Zone Condition = Empty
      Starting Zone Locator = E 4668 0000h
      Number Of Zones = 2
```

As the return data for the Flex Query operation, this indicates that zones 51,405 and 51,406 (starting at LBA E 4668 0000h) can convert to zones 44,701 and 44,702 (starting at LBA 5 74E8 0000h). As the return data for the Flex Convert operation, this indicates that zones 51,405 and 51,406 were converted to zones 44,701 and 44,702.

The result of the Flex Convert operation is no change to the drive state.

The result of the Reset Write Pointer operation is that SMR zones 51,405 and 51,406 are in the Empty state.

The result of the Flex Convert operation is that SMR zones 51,405 and 51,406 are unprovisioned, go offline, and change to the Offline state, and zones 44,701 and 44,703 are provisioned for CMR, come online, and change to the Empty state.

7.3. Example Use of ATA PASS-THROUGH (16)

The following table shows ATA PASS-THROUGH (16) transporting an ATA FLEX CONVERT NON-QUEUED command. EXTEND is set to one so that a 48-bit STARTING ZONE LOCATOR value can be specified. In this example ALL is set to zero. Since AUXNOZ cannot be included in ATA PASS-THROUGH (16), NOZSRC is set to zero, specifying that Number Of Zones comes from the FSNOZ device paramter and not the AUXNOZ command parameter.

Table 32 -- ATA PASS-THROUGH for a FLEX QUERY NON-QUEUED command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (85h)							
1	Obsolete			PROTOCOL				EXTEND (1)
2	OFF_LINE		CK_COND	T_TYPE	T_DIR	BYTE_ BLOCK	T_LENGTH	
3	Reserved		ALL (0)	Reserved				
4	Reserved					NOZSRC (0)	DIRECTION	
5 6	RETURN PAGE COUNT							
7 … 12	STARTING ZONE LOCATOR							
13	DEVICE							
14	COMMAND (C1h)							
15	CONTROL							

8. Appendix A: Capabilities Reporting

This following table lists the ATA and SCSI syntax for Flex capabilities.

Table 33 -- Capabilities Reporting in ATA and SCSI

Field	ATA	SCSI
Device Type	Device Signature = ATA device	PERIPHERAL DEVICE TYPE = 0 (direct access block device)
ZONED	= 0 (not reported)	= 0 (not reported)
miscellaneous	ZAC Minor Version = FFFFh	n/a
100% CMR Capacity = FMRmax+1	IDENTIFY DEVICE offsets 100..103, 230..233; and ACCESSIBLE CAPACITY field	Mode parameter block descriptors
FMRmax	Flex Capabilities log	Flex Capabilities VPD page; and READ CAPACITY (16)
SMRmin	Flex Capabilities log	Flex Capabilities VPD page
SMRmax	Flex Capabilities log; and REPORT ZONES EXT header	Flex Capabilities VPD page; and REPORT ZONES header
URFZ	Flex Capabilities log	Flex Capabilities VPD page
FLEX SUPP	Supported Capabilities log	Block Device Characteristics VPD page
FLEX VERSION	Flex Capabilities log	Flex Capabilities VPD page
FSWP	Flex Capabilities log	Flex Capabilities VPD page
FSWR	Flex Capabilities log	Flex Capabilities VPD page
FSSZT	Reporting: Flex Capabilities log	Flex Control mode page
	Setting: SET FEATURES command, Flex Control subcommand	
FSNOZ	Reporting: Flex Capabilities log	Flex Control mode page
	Setting: SET FEATURES command, Flex Control subcommand	