



FuzeDrive Server v1.1

Linux Installation and User Guide

Version: 1.14

Date: 4th September, 2015

Contents

1.0 Important Note	4
2.0 Overview of Enmotus Virtual and Physical Devices.....	4
3.0 Installing and Updating the Tiering Engine Software	5
3.1 Installation	5
3.2 Installing the License.....	6
3.2.1 Online Activation Process	6
3.2.2. Offline Activation Process.....	6
3.2.3 Checking License Status	6
3.2.4 Returning a License	7
3.2.5 Local License Server	7
3.2 Updating the Enmotus Software.....	7
3.3 Uninstalling the Enmotus Software	7
3.4 OEM Specific Versions.....	7
4.0 Manually Starting and Stopping the MicroTiering Engine	7
5.0 Creating Tiered and Virtual Disk Volumes	9
5.1 Creating Single Disk Tiers	10
5.2 Creating Multi-disk Tiers.....	10
5.3 Creating vdrives	10
6.0 Changing the Default Promote Parameters and Policies.....	11
6.1 Changing Promote Rates.....	11
6.2 Changing Promote Policies	12
7.0 Displaying Status and Statistics.....	12
8.0 Removing or Deleting a Tiered Volume	14
9.0 Example Tiered Volume Setup.....	14
10.0 Manually Promoting or Demoting Files using efile.....	17
11.0 Using the Enmotus Monitoring Utilities	18
11.1 eLiveMonitor.....	18
11.2 ePerform	19

12.0 Advanced User Commands	19
12.1 Attach and Detach	19
12.2 Loading and Initializing Tiered Volumes Manually	19
12.3 Resetting a Loaded Tiered Volume Maps and Statistics	20
12.4 Check Integrity of tDisk Metadata	20
12.5 Reset Error Flags	21
12.6 Manually Change the Scan Timer	21
12.7 Freezing Promotes and Stats Engine.....	21
13.0 Applications Guidelines.....	22
13.1 Automated Real Time, Block Level Tiering	22
13.2 Tuning The Tiering Engine.....	22
13.2.1 Setting Promote Rate.....	22
13.2.2 Setting Promote Policy.....	23
13.3 Recommended Settings by Application	23
13.4 Setting Page Size	24
13.5 Fast First, Slow First and Fast Part Settings	25
13.6 Manual Command Line Promote Setup.....	27
13.7 Monitoring Tiered Volumes	28
13.8 Testing Enmotus Tiered Volumes	28
13.9 Adding tDisks to the Logical Volume Manager	30
APPENDIX A: VSP and tDisk Error Codes.....	31
APPENDIX B: Additional Installation Instructions	32
APPENDIX C: Creating FuzeDrive and Preserving Data on one Drive	33

FuzeDrive Server 1.1

Linux Installation and User Guide

1.0 Important Note

The Enmotus FuzeDrive 1.0 software requires a valid license key to operate and use. If you do not have a valid license, please contact your Enmotus representative to obtain a license.

The Enmotus FuzeDrive Server v1.1 Linux Version includes the following components:

- A high performance tiering engine modular driver and daemon
- The command line utility, evsp
- eLiveMonitor visual mapping utility (requires JAVA 1.6 or higher)
- ePerform utility (requires JAVA 1.6 or higher)
- eNotify utility (requires JAVA 1.6 or higher)
- The file pinning command line utility, efile

Linux distributions supported and tested with FuzeDrive Server 1.0 currently include Redhat 6.4/6.5, CentOS 6.4/6.5, Ubuntu 12.04 and Ubuntu 14.04. Other distributions may be made available as well as a generic Linux porting kit in the future. Please contact Enmotus at support@enmotus.com or check at www.enmotus.com for availability of other distributions.

2.0 Overview of Enmotus Virtual and Physical Devices

FuzeDrive Server supports several virtual and physical levels of device hierarchy as illustrated in Fig 1.

Tiered volumes (tdisks or tdrives) are mapped to host luns (luns), which are the system volume presented to the host as a standard block device. Intermediate virtual disks (vdisks or vdrives) are forming future virtualized disks and are reserved for future use. One virtual disk is designated the fast tier and the other the slow tier vdisk. Physical disks (pdisks or drives) are raw block devices or other raw block devices e.g. /dev/sdb or /dev/md0. Raw disks include block level devices from hardware or software RAID devices. These devices may be a mix of 512 or higher (e.g. 4096) sector sized block devices.

Once installed, several commands may be used to view the various layers of block devices:

- > `ecmd --list` lists all compatible system block devices even if not loaded into the tiering engine.
- > `ecmd --list lun` lists all the Enmotus devices registered as `/dev/ebx` devices
- > `ecmd --list tdisk` lists all the configured tiered devices
- > `ecmd --list vdisk` lists all the intermediate virtual block devices (not commonly used)
- > `ecmd --list pdisk` lists all the system block devices currently configured by the tiering engine
- > `ecmd --license` verify and/or display the status of the license or trial period

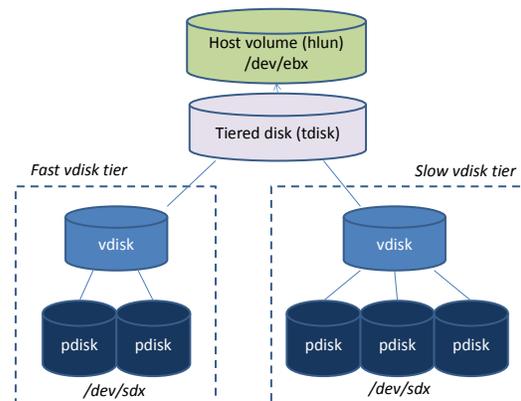


Figure 1: Enmotus MicroTiering Engine Device

3.0 Installing and Updating the Tiering Engine Software

The Enmotus software requires a valid license activation key file to function. A basic license may also be included with a time trial period on a first install. Please contact Enmotus directly if you do not have a valid license file and intend to use the software once the trial period has expired.

3.1 Installation

Download or copy the FuzeDrive Linux installation binary to a temporary installation directory. The filename/command line typically uses the form:

```
> sudo ./FuzeDriveServer_install_Linux64_XXXX
```

Where XXXX indicates the version number of the installation package (no sudo necessary if logged in as root). Everything needed to install the Enmotus software is compacted within this file, included a default trial period mode if installing for the first time.

Open a terminal either with root or sudo privileges, and execute the above installation file. The Linux installation, if supported, will be correctly determined and the core modules and applications installed. For Ubuntu installations, the dkms package is required to ensure the Enmotus software stays current with subsequent kernel updates. If the install fails in Ubuntu, try `apt-get install -f` to install the relevant dependencies. For CentOS and Redhat, standard RPM installation is used.

NOTE: if you received an OEM version as RPMs or in another form, check Appendix C or contact Enmotus for additional instructions.

3.2 Installing the License

If the software is installed, a trial period begins (typically 45 days) after which, the software will enter a state where it will no longer tier hot data to the fast device. In order to continue using the software, an activation key must be obtained from Enmotus or one of its representatives to continue to use the software.

The license may be activated one of two ways: either via an online activation process that handshakes with an Enmotus license server via the Internet, or an offline process if the system is being activated without an Internet connection.

In either case, the form of the activation key is ABCD-EFGH-IJKL-MNOP.

3.2.1 Online Activation Process

This is the simplest method for activating the license and requires an active Internet connection to the machine to the server being licensed:

```
> ecmd --license ABCD-EFGH-IJKL-MNOP
```

Once complete, the license status may be checked as follows:

```
> ecmd --license
```

3.2.2. Offline Activation Process

For offline activation, the following steps are required:

1. At a root or sudo level console on the machine being licensed, use the command:

```
> ecmd --license ABCD-EFGH-IJKL-MNOP -offline
```

2. A local file will be generated and must be copied to an Internet connected computer and the file emailed to Enmotus or their representative
3. A file will be returned which must be copied back to the machine being licensed.
4. The license may then be activated by typing the following command:

```
> ecmd --license <pathname/file_name_returned> --activate
```

3.2.3 Checking License Status

Once successfully installed, the license status may be checked as follows:

```
> ecmd --license
```

3.2.4 Returning a License

To unlicense the server and return the license so that it can be used on a different server, the license may be returned to the license server using the following:

```
> ecmd --license return
```

Use the `--offline` option and follow the offline activation emailing instructions if no Internet connection is available.

3.2.5 Local License Server

Please contact Enmotus for additional details.

3.2 Updating the Enmotus Software

The software is updated using the same installer program used for initial installation and will automatically update the appropriate components of the software according to the type of license deployed on the system.

3.3 Uninstalling the Enmotus Software

Run the same installer program used to install the Enmotus software using the `--uninstall` option. For example:

```
> sudo ./FuzeDriveServer_install_Linux64_1234 --uninstall
```

3.4 OEM Specific Versions

The license may be set for a specific OEM vendor. If that is the case, certain management commands will not be available unless the vendor's device is physically attached and identifiable. In the case of devices hidden behind a RAID controller, please consult Appendix A for further information on how to unlock the management commands.

In addition, some features may not be available in some OEM versions but may be available with a new license key from Enmotus. Please check with your Enmotus representative/

4.0 Manually Starting and Stopping the MicroTiering Engine

The tiering engine will automatically load if installed using the standard Enmotus installer as described earlier. Alternatively, it may also be started, stopped or queried using the system commands (Redhat/CentOS 6.x distributions):

```
> sudo service EnmotusService stop           stop the Enmotus service
```

```
> sudo service EnmotusService start      load and start the Enmotus
                                         service
> sudo service EnmotusService status     show status of the service
```

To aid in system evaluation, it is also possible to load and unload the tiering engine using the low level command utility `evsp`. If required, use the following commands as system admin or via `sudo`:

```
> evsp --autoload                        scans the system for devices with Enmotus
                                         metadata and auto loads them
> evsp --load <device list>             loads with specific devices ignoring any
                                         other devices
```

Example load command:

```
> evsp --load /dev/sdb /dev/sdc
> evsp --load /dev/sd[b-f]
```

If an existing tiered disk configuration exists on the component devices, it will automatically load the configuration from metadata stored on the drives and the tiered volume will be registered and mounted with the host storage block device handler as `/dev/eba` (or `/dev/ebb`, `ebc` if multiple `tdisks` are defined). Note, when loading one or more **existing** tiered volumes defined in metadata, the order of the devices does not matter i.e. the disks will automatically be assembled in the correct order.

Once all the appropriate devices have been manually loaded, to load the tier, use the following:

```
> evsp --autoscan
```

If no devices are specified i.e. `evsp --load`, the tiering engine will simply load into the system without creating any tiered devices and they must be manually added as described later.

To manually unload the tiering engine entirely, as system admin or via `sudo` use:

```
> evsp --unload
```

This will unload all configured tiered volumes unless they are still mounted, then unload the tiering engine modular drivers. Tiered volumes unloaded this way will be available for use later and will retain complete state information, including through a system power cycle and may be reloaded using the `--load <fast-device> <slow-device>` option described earlier.

5.0 Creating Tiered and Virtual Disk Volumes

There are several possible ways to create a tiered volume, depending on the number and configuration of disks required.

The general command `--create` is used to make intermediate virtual volumes (vdisks) that form the tiers as well as the tiered volumes (tdisks). The general format of the command is as follows:

```
> ecmd --create <disk-type> <device1> <device2> ... {options}
```

Where:

`disk-type` = tdisk or vdisk (or tdrive, vdrive¹)

`device1,2...` = the devices to be used to create the virtual device

`options` = type or initial mode of the virtual volume

The following options allow the user to specify the initial mapping of the fast and slow portions of the tier in the virtual disk's logical block address space. Three options are currently provided that allow the initial logical block address order of the fast and slow devices in the virtual address range to be specified:

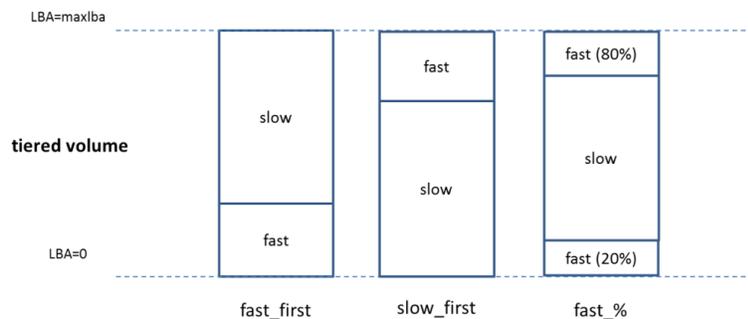


Figure 2: Tiered Volume Initialization Modes

fast_first - the fast device is placed at tiered volume logical block address 0 (default)

slow_first - the slow device is placed at tiered volume logical block address 0

fast_part - 20% of the fast device is placed at logical address 0, with the remaining 80% placed at the end of the tiered volume address

¹ tdrive and tdisk, vdrive and vdisk may be used interchangeably to allow script compatibility with the Windows version of ecmd

5.1 Creating Single Disk Tiers

When combining a single pair of fast and slow devices together into a tdisk, there is no need to create intermediate vdisk devices and a single command line may be used:

```
> ecmd --create <device1> <device2> {options}
```

where device 1 is the fast-device to be used in the fast tier (e.g. an SSD block device) and device2 as the slow-device in the slow tier (e.g. a SATA/SAS MLC SSD hard drive). Note, the fast device must always be provided first to ensure the tier correctly recognizes it as the target fast tiered device.

If no option is specified, a fast_first configuration will be created by default, meaning that the fast device will be located starting at the low order LBA range of the virtual tiered block device.

Example single disk tier tdisk create commands:

```
> ecmd --create /dev/sdb /dev/sdc fast_first
> ecmd --create /dev/md0 /dev/md1 fast_part
```

5.2 Creating Multi-disk Tiers

The create command can use two internal Enmotus virtual disks (vdisks) as its arguments instead of raw block devices. In this case, the vdisks need to be pre-configured prior to being placed into a tier.

The general command for creating the tdisk from vdisk components is:

```
> ecmd --create tdisk vdrive<p> vdrive<q> {options}
```

Where vdrive p and q are vdrives created using the vdrive create command described below, and options are the same as the single disk per tier i.e. fast_first, slow_first or fast_part.

Example multi disk tier tdisk create commands:

```
> ecmd --create vdrive0 vdrive2 fast_first
> ecmd --create vdrive3 vdrive4 slow_first
```

The index number of the vdrives to be used may be established using the --list vdisks or --list pdisks ('V' column) described earlier.

5.3 Creating vdrives

The tiering engine supports pooling of physical disk drives in either a linear concatenation or stripe mode. Striping is implemented around tiering page boundaries with a stripe element size the same as the page size setting for the tdisk (default 4M).

The command for creating a vdrive from pdisk components is:

```
> ecmd --create vdrive <device1> <device2> ... {mode}
```

Up to a maximum of 16 devices may be combined in a pool. Valid modes include linear (concatenated) and striping.

Example multi disk vdisk create commands:

```
> ecmd --create vdrive /dev/sd[b-d,g] linear
```

creates a 4-disk linear concatenated vdisk pool using devices sdb,c,d and g

```
> ecmd --create vdrive /dev/sdj
```

creates a single disk vdisk using sdj

6.0 Changing the Default Promote Parameters and Policies

The mode of the tiering engine may be changed from its default settings. Promote thresholds, rates and policies can all be setup using the following commands:

```
> ecmd --promote <mode> <tdevice>
```

changes the promote mode for the tiered device, tdisk (usually /dev/eba, ebb, ebc, etc).

```
> ecmd --policy <mode> <tdevice>
```

changes the promote policy used by the tiering engine tdisk (usually /dev/eba, ebb, ebc, etc).

Note: t=<n> may also be used in place of the system device number e.g. t=0 is usually the same as /dev/eba. This applies generally throughout this document.

6.1 Changing Promote Rates

Promote mode options include aggressive (default), normal, slow, on or off and can be set for each individual tiered volume e.g.

```
> ecmd --promote aggressive /dev/eba
```

enables increased promote rate with relatively low activity levels required to promote data for tdisk /dev/eba. See Applications Guidelines section later for more information on the impact of changing the promote parameters.

In addition, it is possible set raw threshold values for the promote engine which correspond to specific IOPs or block activity thresholds. The general command is:

```
> ecmd --promote N /dev/ebx
```

Where 'N' is the specific value to start promoting data. For example, if the policy is set to read only, then N specifies the total number of IOs that must be achieved for a page to promote or if set to a block based policy (see later), then N is the number of sectors written to that page and so on.

For example, to change a tiered disk `/dev/eba` with read IO policy set to promote data once 20 IOs have occurred in any particular tiered page, the following commands are used:

```
> ecmd --promote 20 /dev/eba
```

The promote setting may be applied any time after the tiered volume is created.

6.2 Changing Promote Policies

The default promote policy is to promote on activity based on read IOs only. This is sufficient for most applications. However, it is often necessary to change this to accommodate either write activity or block level activity.

The following *modes* are available for the `--policy` command:

<i>rdio</i>	promote decisions are based on read IO activity only (default)
<i>rwio</i>	promote decisions are based on read and write IO activity
<i>rdblock</i>	promote decisions are based on read block volume activity
<i>rwblock</i>	promote decisions are based on read and write block volume activity

For example, to change the policy to a read plus write IO activity based decision, use:

```
> ecmd --policy rwio /dev/eba
```

The policy command may be issued any time after the tiered volume is created.

7.0 Displaying Status and Statistics

To view the tiering engine status or check to see if the tiering engine is running:

```
> ecmd --status          lists general status information and global
                        configuration info
```

The tiering engine divides the tiered volume (tdisk) into its own pages which are typically a multiple of the virtual volume sector size. For example, for a 512 byte sector sized tdisk, the page size maybe 1024 sectors i.e. 512K. The default is typically 4M. It is these pages that are promoted or demoted.

Given there are often several millions of pages per tiered volume, it is not practical to show statistics on a per page basis. At command line level, the tiering engine allows the user to display information on a summarized page-region basis where each region represents 1/64th of the overall tiered volume. For each region, which may consist of several thousand pages, the use is able to display read, write or block activity as well as determine current mapping information e.g. determine which parts of the tiered volume an SSD based tier is mapped to.

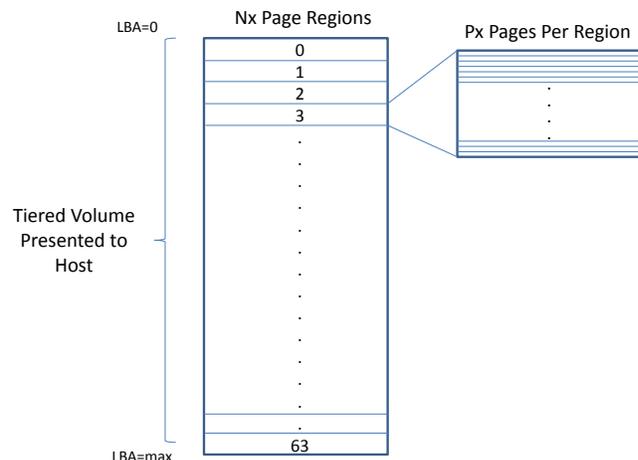


Fig 3: Page Regions for Displaying Stats

The following commands are available to support display of key operating statistics:

- > `ecmd --stats promote_queue /dev/ebx` displays pages waiting to be promoted
- > `ecmd --stats promote_hist /dev/ebx` displays recent promote activity
- > `ecmd --stats count /dev/ebx` displays high level read, write and block counts across the tiered volume page regions
- > `ecmd --stats map /dev/ebx` displays high level fast and slow tier distribution across the tiered volume page regions
- > `ecmd --stats hostio /dev/ebx` displays high level host io statistics
- > `ecmd --stats off /dev/ebx` turns off the statistics counters (useful during formats)
- > `ecmd --stats on /dev/ebx` turns on the statistics counters

8.0 Removing or Deleting a Tiered Volume

The following commands will remove a tiered volume from the system and disconnect the disks from the tiering engine:

```
> ecmd --delete_all      remove all tiered volumes from the system.
> ecmd --delete /dev/ebx delete only the /dev/ebx device from the
                        system
```

Warning! This removes any configuration data on the disks with no ability to undo. All data will be lost on the disks so please backup any important data first. If a valid MBR or GPT partition exists on the tiered drive, the process will abort until the user either manually deletes the partition or uses the force option e.g. `ecmd --delete /dev/eba force`.

9.0 Example Tiered Volume Setup

Note: As the Enmotus engine is a block based tiered engine, it will monitor all basic IOs that it sees to the physical volumes it is managing. For singular events such as formatting the tiered volume, while it doesn't affect long term operation, it is sometimes beneficial to turn off promotes while formatting the volume if aggressive policies and/or *readwrite* are in effect and turn them back on afterward. This prevents unnecessary promotes for non-application activity.

The following example illustrates how to setup a tier using a single SSD and single hard drive, presented to the system as `/dev/sdb` and `/dev/sdc` respectively. It assumed the user has sudo level privileges or can log in as root user.

Creating the tiered volume:

(a) Check the Raw Physical Devices are Available

```
[user@test64 ~]$ ecmd --list
List system devices:
Device Product ID      Type           GiB  Rev
-----
sda    VMware Virtual S SAS HDD    80   1.0
sdb    VMware Virtual S SAS HDD     8   1.0
sdc    VMware Virtual S SAS HDD    12   1.0
sdd    VMware Virtual S SAS HDD    12   1.0
```

(b) Load the Tiering Engine Manually (only necessary if the service is not started)

```
[user@test64 ~]$ sudo evsp --load /dev/sd[bc]
Loading Enmotus Tiering Module
Loaded OK
Adding /dev/sdb as pDisk[0]
Configuration Detected:  Physical: 0  Virtual: 0  Tiered: 0

Adding /dev/sdc as pDisk[1]
Configuration Detected:  Physical: 0  Virtual: 0  Tiered: 0

Scanning 2 disk(s) for existing configuration ...
pDisk port[0] size=0x1000000, name=/dev/sdb, 8GiB
pDisk port[1] size=0x1800000, name=/dev/sdc, 12GiB
Total disks used = 0, available = 2
```

(c) Creating the Tiered Volume

```
[user@test64 ~]$ ecmd --create /dev/sdb /dev/sdc
Creating new tiered volume
Tiering engine has 2 physical disks loaded
Checking if /dev/sdb is already loaded
Device /dev/sdb already exists

Checking if /dev/sdc is already loaded
Device /dev/sdc already exists

Scanning 2 disk(s) for existing configuration ...
pDisk port[0] size=0x1000000, name=/dev/sdb, 8GiB
pDisk port[1] size=0x1800000, name=/dev/sdc, 12GiB
Total disks used = 0, available = 2

Initializing /dev/sdb
Creating vDisk mode=15 (left=2) f=0
  Assigned vdisk number 0
  vDisk[0] Size : 6GiB, 0xc00000 Sectors
  Page Size : 0x400 (1024)
  Total Pages : 0x3000 (12288)
```

Cont ...

```
Initializing /dev/sdc

Creating vDisk mode=15 (left=1) f=0
  Assigned vdisk number 1
  vDisk[1] Size : 10GiB, 0x1400000 Sectors
  Page Size : 0x400 (1024)
  Total Pages : 0x5000 (20480)

Creating tDisk mode=32 (left=0)
  Using default tiered page size: 1024 (524288 bytes)
  tDisk[0] => 16 GiB sectors=2000000h pages=8000h sectorsize=512
  Total Memory Allocated for tDisk[0]: 6659040 bytes, 6MB
  Initializing vDisk Page Maps for tDisk[0], Mode = 0 ...

Updating metadata on all initialized disks...
Mounting lun[0] => tDisk[0] Enmotus tDisk0 33552384 sectors 512
ssize 15GiB
Registered hlun[0] with host system:
  ID      : ENMOTUS  Enmotus T00 Disk   Rev: A006
  Size   : 15GiB, 33552384 Sectors
```

(d) Unloading the tiered volume:

```
[user@test64 ~]$ sudo evsp --unload
Unloading Tiering Module
Unmounting eba      Enmotus tDisk0   Tiered Disk   15
umount: /dev/eba: not mounted
Unloaded OK
```

(e) Reloading the tiered volume:

```
[user@test64 ~]$ sudo evsp --load /dev/sdb /dev/sdc
Loading Enmotus Tiering Module
Loaded OK
Adding /dev/sdb as pDisk[0]
Configuration Detected:  Physical: 1  Virtual: 1  Tiered: 1

Adding /dev/sdc as pDisk[1]
Configuration Detected:  Physical: 2  Virtual: 2  Tiered: 1

Scanning 2 disk(s) for existing configuration ...
pDisk port[0] size=0x1000000, name=/dev/sdb, 8GiB
pDisk port[1] size=0x1800000, name=/dev/sdc, 12GiB
Loading tiered volume configuration from metadata
lun[0] => tDisk[0] 33552384 sectors 512 ssize 15GiB
Registered hlun[0] with host system:
  ID          : ENMOTUS T00 Enmotus Disk   Rev: A006
  Size       : 15GiB, 33552384 Sectors
Total disks used = 2, available = 0
```

10.0 Manually Promoting or Demoting Files using efile

The tiering engine is often best left automatically promoting. However, it is often preferable to manually pin files to either tier directly. Note, this feature may not be available on all versions of the Enmotus software or OEM versions, but maybe available as a licensable upgrade.

The utility efile is used for promoting or demoting files between the tiers manually and the help menu is obtained by simply typing the following on the command line at a root level or sudo command prompt:

```
> efile --help

efile --version      Prints out version information
efile --promote <file(s)> Promotes one or more files
efile --demote <file(s)> Demotes one or more files
efile --release <file(s)> Releases one or more files
efile --mapping <file(s)> Checks the file mapping of one or more
                        files
efile --tracking <file(s)> Checks the tracking status of one or more
                        files
```

For example, to check if a file **example.txt** on a mounted tiered volume is located on the fast or slow tier, use the following:

```
> efile --mapping example.txt
```

The amount of the file that resides on the fast tier will be shown as a percentage e.g. 80% if partially promoted, 0% if entirely on the slow tier or 100% if entirely on the fast tier. To promote a file that is partially or entirely on the slow tier to the fast tier, use the following:

```
> efile --promote example.txt
```

If the file is partially or fully on the fast tier, to demote to the slow tier use the following:

```
> efile --demote example.txt
```

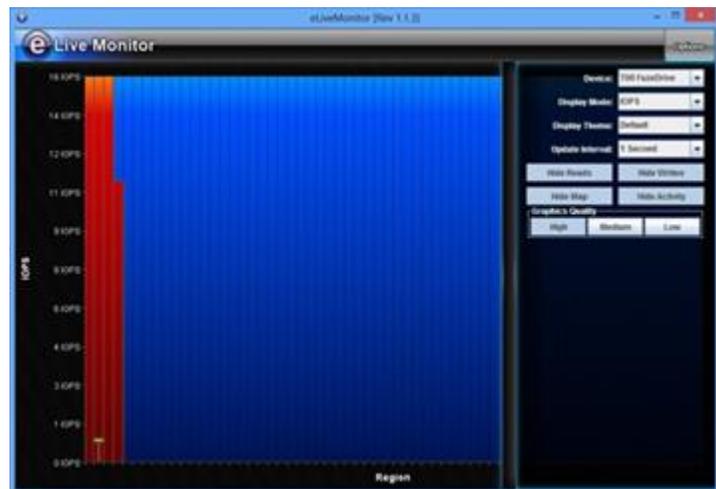
11.0 Using the Enmotus Monitoring Utilities

For Linux installations employing a graphical user interface, several utilities are provided for viewing or measuring performance of tiered and system disks. In CentOS or Redhat, these may be found under the Desktop Applications/Accessories menu.

11.1 eLiveMonitor

This utility provides two primary functions:

- Observation of where the fast tier devices are currently mapped in relation to the tiered virtual disk logical block address range
- Displaying of real-time read, write IO and Mbytes data overlaid on the tiered volume
- Displaying of real-time fast, slow read and write IO and Mbytes statistics data over time.



To load and run the utility, from a terminal window type

```
> elivemonitor
```

For UI users, an icon is typically found in the standard Applications or standard desktop search capabilities and searching for elivemonitor.

11.2 ePerform

ePerform visually graphs IO performance for any volume in the system. It performs a sequential sweep across the selected volume which can be any standard block device in the system, including the tiered volumes created using the Enmotus command line.

12.0 Advanced User Commands

12.1 Attach and Detach

It is sometimes useful for debug purposes to forcibly attach or mount a broken tDisk to a host lun for diagnostics purposes or simply detach/unmount it from the file system without destroying the tier.

The following commands are used:

```
> ecmd --attach t=0 {force}  attach tdisk0 as a /dev/ebx device if
                             currently offline. Use the force option if
                             the disk is in an error'd state.

> ecmd --detach t=4         detach the tdisk4
```

12.2 Loading and Initializing Tiered Volumes Manually

WARNING: Some of the following command(s) will remove or delete configuration information. Backup all important data before using.

The load, create and delete commands described earlier are the usually sufficient to setup or tear down a single tiered volume. However, it is sometimes necessary to manually setup one or more tiered volumes in discrete steps to support specific setup options.

To setup a tiered volume manually, several additional commands are used that first add disks one at a time, initialize the disks with a new metadata structure, then create a tiered volume from the component disks.

The following commands are available:

```
> ecmd --add /dev/<dev>      add a disk <dev> to the tiering
                             engine driver

> ecmd --clear_meta {/dev/<dev>}  clear metadata off all the added
                             pdisks or a specific pdisk

> evsp --init /dev/<dev> {force}  initialize disk with an undefined
                             metadata structure and force it to
                             override any legacy disk protection
                             if 'force' specified
```

```
> ecmd --create tdisk /dev/<fast-device> /dev/<slow-device> create
    tiered volume

> ecmd --rescan_meta          forces the driver to reload it's
                             configuration from metadata all
                             newly added pdisks
```

For example, to load a prior configuration on sdb and sdc without bringing the tier on line and clear it of all metadata, use the following:

```
> sudo evsp --load
> ecmd --add /dev/sdb
> ecmd --add /dev/sdc
> ecmd --clear_meta
> sudo evsp --unload
```

To load a tier manually one disk one at a time to verify a disk is loading ok, use:

```
> sudo evsp --load
> ecmd --add /dev/sdb
> ecmd --add /dev/sdc
> ecmd --rescan_meta
```

12.3 Resetting a Loaded Tiered Volume Maps and Statistics

To reset a tiered volume's statistics gathering only without changing the data configuration on the tDisk, use the following:

```
> evsp --reset stats /dev/ebx    resets a tdisk's statistics only. This
                                is a non-destructive command.
```

The above command may be used on a running tiered volume as the virtual mapping is not changed by the reset command.

WARNING: The following command(s) will remove or delete configuration information. Backup all important data before using and also dismount the tiered volume from the file system.

12.4 Check Integrity of tDisk Metadata

The tiering engine automatically checks the state of the metadata on each drive in the system. As this can take a long time for virtual pooled disks, it cycles around each of the member pDisks in the vDisks or pools and compares the stored metadata tables with the active tables in RAM.

To force an immediate manual run of all or one of the disks in the tiered volume, use the following:

```
> ecmd --check /dev/ebx          checks tdisk 'n' metadata for each of
                                the member pdisks
```

12.5 Reset Error Flags

The status display will show if there are any errors immediately after the global status if a global error, or under the individual tdisk listing if related to a single tdisk.

To reset these error flags, use the following:

```
> ecmd --reset error            resets the global VSP level error flags
> ecmd --reset error /dev/ebx  resets the tdisk /dev/ebx error flags
```

See Appendix B for the error code meanings.

12.6 Manually Change the Scan Timer

It is possible to manually set the scan timer which tells the tiering engine how often to check for promote candidates and kick off a promote cycle. The `ecmd --promote` options described earlier sets up default values, however the default scan timer value may be overwritten using the following:

```
> ecmd --scantime <value> {s,m,h,d} {t=N} sets the global scan timer
                                e.g. --scantime 10s = 10 seconds;
                                maximum is 7 days. If t=N (N=tdisk
                                number) present, sets just for a
                                particular tdisk.
```

12.7 Freezing Promotes and Stats Engine

It is often beneficial to freeze the state of the promote engine or particular tiered volume while performing maintenance functions or formatting the volume for the first time.

To stop a tiered volume from moving data automatically or preventing file pinning operations, use the following:

```
> ecmd --promote off /dev/ebx    # use t=N in place of /dev/ebx
```

To also freeze the statistics gathering for a particular tier use:

```
> ecmd --stats off /dev/ebx
```

13.0 Applications Guidelines

13.1 Automated Real Time, Block Level Tiering

The Enmotus core tiering software is designed to operate automatically at sub-file level in real time i.e. it has no direct awareness of files in a system, only storage IO accesses, and moves data frequently until the system is load balanced. Statistics are gathered continuously, so once the software has learned where to best place data, it then continues to adjust data placement as storage IO patterns change or new files are added to or deleted from the system.

Real-time block based tiering has several advantages over traditional batch or scheduled tiering:

- Only the portions of the files that need to be on the fast storage are moved
- Data is moved within seconds or minutes of being accessed as opposed to daily or weekly
- Once the system has tuned to the storage IO patterns, it continues to adjust where active data is stored

Real time tiering, especially given the increased availability of more cost effective SSDs, makes a big difference to daily server operation given today’s highly dynamic data environments and increasing use of unstructured data in everyday business decisions.

13.2 Tuning The Tiering Engine

There are several tunable parameters which the user may use to change the behavior of the tiering software to suit a particular application.

The two primary high level tunable parameters are ***promote rate*** and ***promote policy*** which load a set of pre-defined values into the tiering engine and may be set for each individual tiered volume at create time using the options menu or modified later (see installation section earlier). For users wishing to customize these settings directly, see the Advanced Settings section later.

13.2.1 Setting Promote Rate

The promote rate setting decides how often data is moved and at what activity thresholds. Depending on the activity level of the server, this setting may be tuned depending on the anticipated loading levels of the server. Three levels are provided as shown in Table 1.

Table 1: Promote Rate Aggressive, Normal and Slow Parameters

User Setting	Scan Cycle Time	Promote Threshold per Page IOs	Promote Threshold per Page Mbytes	Max Page Promotes Per Cycle
Aggressive*	2s	1 IOs	1M	32

Normal	8s	4 IOs	4M	16
Slow	60s	50 IOs	50M	2

* Default Setting

13.2.2 Setting Promote Policy

Statistics about hosts IO made to the tiered volume are recorded by the automated tiering engine and decisions are made if and when to move data from from the slow to the fast storage tier based on a the policy setting for that tiered volume. These statistics are maintained on a per-page basis (see page size settings in installation or later sections).

Table 2: IO and MB Policy Settings

Policy Setting	Description
Read Only IO*	Promotes data based on the frequency of read IO requests per page
Read-Write IO	Promotes data based on the frequency of read AND writes IO requests per page
Read Blocks	Promote data based on amount of data read per page
Read-Write Blocks	Promote data base don amount of data read AND written per page

* Default Setting

13.3 Recommended Settings by Application

Table 3 illustrates the recommended settings that may be used for different types of applications. Typical application profiles include data base, email and web server configurations.

Table 3: Recommended Settings by Application

Application Profile	Traffic Type	Promote Rate	Promote Policy
*Small Business Server – General	Random Unstructured	Normal	Read IO
Database Server - Read Intensive	Random Unstructured	Normal	Read IO

Database Server - Frequent Updates	Random Unstructured	Normal	Read-Write IO
Database Server - Medium Large File	Sequential/Random Mix	Normal	Read-Write MB
Exchange/Sharepoint Server	Random Unstructured	Normal	Read IO
VDI Server	Medium-Large Streaming	Aggressive	Read MB
General Host Virtual Server	Medium-Large Streaming	Normal	Read MB
Large File/Video Server	Large Streaming	Aggressive	Read-Write MB
Big Data Storage-Server	Random Unstructured	Normal	Read IO
Unified Storage Server (Windows)	Random Unstructured	Normal	Read IO

* Default Setting

13.4 Setting Page Size

At create time, it is possible to change the default page size for the tiered volume. The size of the page determines a number of operating parameters within the tiering engine such as:

- Level and number of statistics that are maintained across the volume
- The level of activity for the amount of data read or written
- The amount of data moved per promote cycle
- The amount of system RAM required to maintain the runtime statistics tables

The default setting is the maximum 4Mbytes per page and should be sufficient for most applications outlined above. However, it may be desirable to change the default to a smaller amount.

The general guidelines in Table 4 are provided, however please note these are dependent on the user environment and hardware and may require some experimentation for the particular environment the server and software is operating in.

Table 4: Page Size Setting Recommendations

Page Size (M,K bytes)	General Usage Guideline
4M*	<ul style="list-style-type: none"> Volumes >= 50TB Large File Based Applications e.g. Video Servers or Virtual Desktop Servers
1-2M	<ul style="list-style-type: none"> Volumes 10-50TB Medium to Large File Based Applications
512K	<ul style="list-style-type: none"> Volumes < 10TB Small-Medium File Based Applications
64K-256K	<ul style="list-style-type: none"> Volumes < 5TB Small File Based Applications

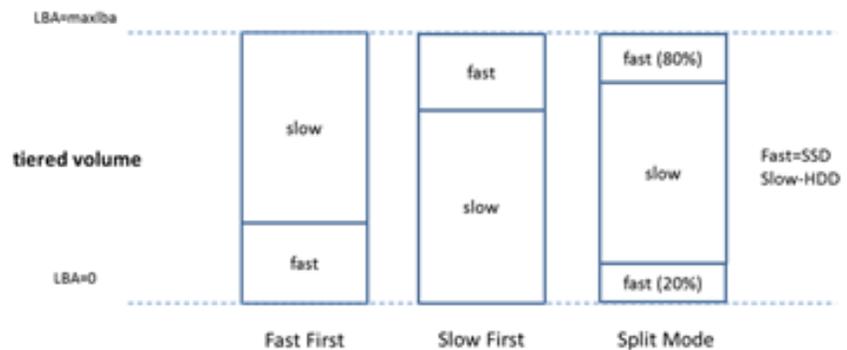
* Default Setting

13.5 Fast First, Slow First and Fast Part Settings

As briefly described earlier, during the create process, the user has the ability to define how the initial tiered disk map may be constructed in terms of fast and slow device ordering in terms of the tiered disk logical block

addressing. A tiered volume maps host IO requests on the fly into physical device IO requests by consulting a RAM table mapping table. It is this mapping table that is affected by these particular options.

Figure 1: Initial Tier Mapping Options



From an application standpoint, these settings determine the initial starting performance levels of the volume and are dependent on the nature and data access patterns. For example, if we use fast first (default) setting, as the operating system copies data to the volume, it typically copies/moves files into the lower addressable area of any

drive, hence for a fresh tiered volume, initial data will be copied to the SSD first for example. If we used slow first, initial data would be copied/moved to the HDD first for example. The former means the data is pre-warmed or tiered immediately at data copy, create or move time. The latter means the data is stored on the cold tier initially and will only be promoted if accessed later.

Table 5: Initial Tiered Map Setting

Initial Map Setting	Description	Typical Usage
Fast First*	Fast tier (e.g. SSD) is mapped to the first logical block of the tiered volume, occupying the lower addressable range of the tiered disk.	Fast warming, applications where it is desired to have SSD performance immediately until the capacity of the SSD has been fully utilized. Works well when significantly less data than the SSD size is initially being placed onto the tiered volume.
Slow First	Slow tier (e.g. HDD) is mapped to the first logical block of the tiered volume. SSD is placed at the end of the disk in this case.	Depending on the policy setting, all data is first copied to the slow tier (HDD) and only migrated to the SSD if subsequently read. Works well with larger data sets than the SSD size and the user wishes to selectively promote only the active portions of the data after it is copied to the tiered drive.
Fast Part	20% of the fast tier is mapped to the first logical block, followed by the complete slow tier, with the remaining 80% mapped to the end of the tiered volume.	A mix of the above, and especially useful for larger SSDs where only part of the data needs to be pre-warmed after initial create and the larger portion going to the hard drive. Also has the effect of setting an initial 80% of the SSD as a resource pool of fast storage that gets assigned to the active data as it is used.

* Default Setting

Note, these setting only has an impact during the initial use period of the tiered volume, as over time, the automated load balancing engine will redefine the tied map based on the relative activity and promote settings.

13.6 Manual Command Line Promote Setup

For expert or experimental users, the following command line options exists to allow the aforementioned parameters to be individually tuned. It is recommended that any experimentation with these parameters be done on an offline or non-production test server wherever possible to ensure minimal disruption to end users before using on a production server.

To run the following commands, open a Linux terminal as root or sudo and use the commands listed in Table 5. Also see Enmotus Command Line section earlier in this document.

Table 5: Advanced Command Line Configuration Options

Description	Command Line Format	Example
Promote Scan Time	evsp --scantime <time> {s,m,h} where s=secs, m=mins, h=hrs	evsp --scantime 5m
Promote Rate	evsp --promote {aggressive, normal, slow} /dev/ebX where T is the number of the tiered volume in disk manager e.g. /dev/eba	evsp --policy aggressive /dev/eba
Promote Threshold	evsp --promote N /dev/ebX where N is the threshold value in IOs or Sectors (depending on the promote policy setting)	evsp --promote 100 /dev/eba (100 IOs per page if rdio or 100*512=51,200 bytes if byte policy)
Promote Policy	ecmd --policy {rdio, rwio, rdblock, rwblock} /dev/ebX	ecmd --policy rdblock /dev/ebb

13.7 Monitoring Tiered Volumes

There are several options available for monitoring storage activity of tiered volumes as well as assess how the fast tier (e.g. SSD) is mapped in relation to current and past activity levels. One of the most important views to see how well the fast tier has adapted to activity levels on the volume, and also quickly assess the locality information of data with respect to the fast-slow tiers, is the eLive Monitor utility region view. Using this view, the fast regions of the volume are shown in a highlighted color that represents the percentage of fast pages mapped to that region.

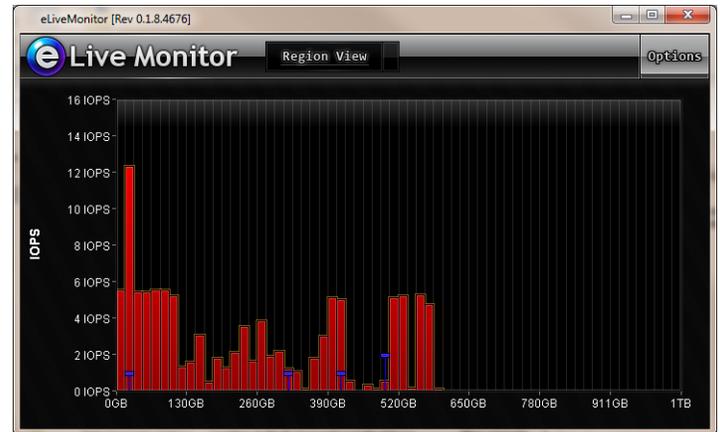
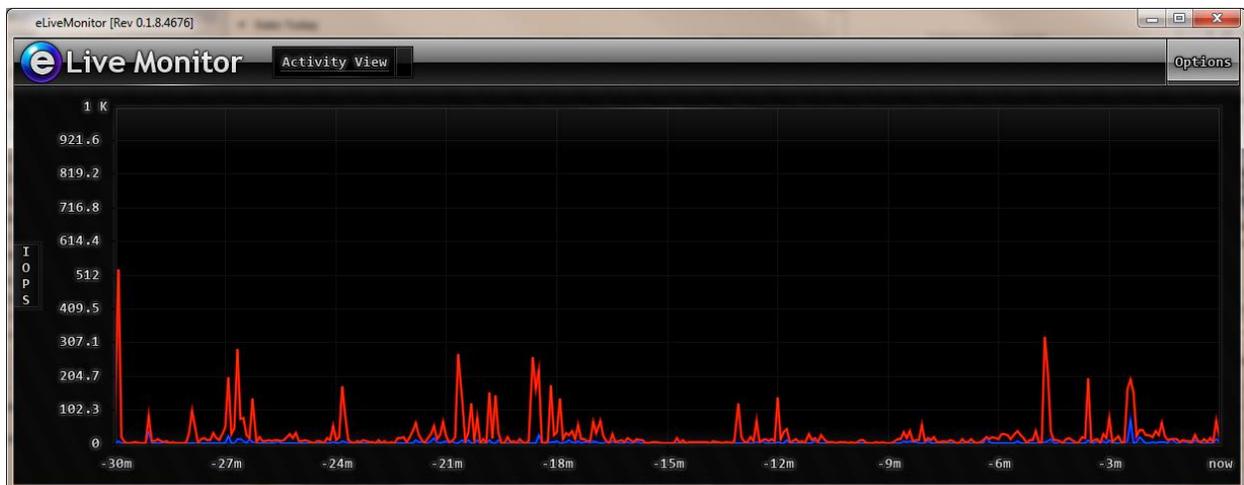


Figure 2: Viewing Fast Tier Mapping Information



13.8 Testing Enmotus Tiered Volumes

Real-time tiering volumes behave differently to conventional caching or big iron tiering solutions. Tiers treat their different classes of storage as primary storage which is directly mapped to user storage allowing full performance of the media to be utilized for both reads and writes.

However, initial testing using many popular benchmarking programs can provide misleading results, principally because they tend to assume a single class of physical is being tested not a dynamic, mixed storage media device such as that enabled by tiering. Hence, broad sweeping benchmarks that exercise an entire all at once will create a result that looks not much better than the performance of the slow media device. This is because a tier blends the IOs from the two tiers and given the slow tier is usually significantly dominates the mix of fast and slow, the result is marginally better than slow.

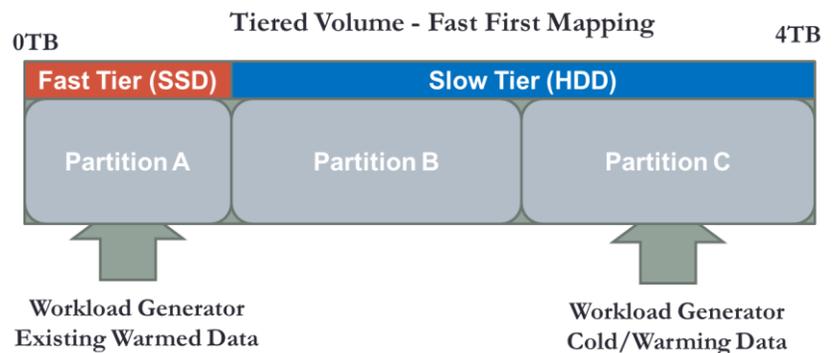


Figure 3: Test Partitions for Tier Analysis

In real world applications, only a small portion of any volume is active at any one time in 80%+ of all applications. Hence, when testing a tiered volume it is far more meaningful to test data in its two primary modes of operation:

- Pre-warmed or existing tiered data performance
- Cold or data that is being warmed

The easiest way to do this is set up a test tiered volume using the fast first (default) configuration, then create three file partitions on the volume as follows:

- One partition at the beginning of the volume, roughly 5G smaller than the size of the raw fast device (e.g. SSD) - Partition A
- A second partition to fill the “middle” portion of the tiered volume - Partition B
- A final partition that is again smaller than the fast tier size - Partition C

Once the partitions are setup, then use an off the shelf benchmark program or file exerciser on partition A to measure what the performance of the fast tier is. This should be close to the raw performance of the device used to form the fast tier.

To test the slowest performance of the tier, use the same exerciser or workload generator and target partition C which using the default setup, should be mapped to the slow portion of the tier. It is suggested that the benchmark or exerciser should be run for a short while, then left for an equal length of time to allow the tier to move data without host io activity. The initial performance run should show slow media rates e.g. HDD IO or MByte rates. Each subsequent run after that should show increasingly better performance until it reaches the fast tier rates.

Note, in some cases, the maximum fast tier IO access rates may not be fully achieved unless the test is run with a read-write promote policy instead of the default read-only policy. It may also require the test to be run with promote rate set to aggressive. This is because some benchmarks may write to a location but not subsequently access a tiered page sufficiently to trigger a promote event to occur. Using the aggressive rate ensures that only a small amount of activity is necessary to trigger a promote and will attempt to move the data as quickly as possible (see earlier sections on promote policy and rate settings).

13.9 Adding tDisks to the Logical Volume Manager

It is possible to add a tDisk as a physical device to the Linux Logical Volume Manager.

To enable the LVM environment to recognize the Enmotus devices, the following line must be added to the appropriate section of the `/etc/lvm/lvm.conf` configuration file:

```
types = [ "en-block", 16 ]
```

Check that the line was added correctly by typing the system command (as root):

```
> pvcreate dumpconfig | grep types
```

Once successfully added, follow the standard process for forming logical volumes starting with for example (as root):

```
> pvcreate /dev/eba
```

Check that the logical volume has been added by using the `lsblk` command if supported.

APPENDIX A: VSP and tDisk Error Codes

VSP Error Bits:

Bit	Hex Val	Description
0	1h	Fatal Error - cannot recover
1	2h	VMAP error detected on a tdisk
2	4h	Memory error during tdisk create or load
3	8h	tDisk failed
4	10h	tDisk vmap errors were recovered
5	20h	RSVD
6	40h	RSVD
7	80h	DDF I/O Error
8	100h	DDF CRC error was detected
9	200h	Maximum capacity error (license parameter exceeded)
10	400h	License keep alive timer has expired
11	800h	License 24 hour grace period timer has expired (demotes begin)
12	1000h	eboot FEh code error (client/bootable only)
13	2000h	pdisk I/O error
14	4000h	promote error detected
15	8000h	License parameter error (non-capacity)
15+		RSVD

tDisk Error Bits:

Bit	Hex Val	Description
0	1h	DDF I/O error
1	2h	Promote error
2	4h	Out of range error
3	8h	VMAP compare error
4	10h	VMAP checksum error
5	20h	VMAP error but was recovered
6	40h	VMAP flags error
7	80h	Pageregion error
8	100h	Memory allocation error
9	200h	Capacity max error (license parameter exceeded)

APPENDIX B: Additional Installation Instructions

If you received your software distribution as standard distribution packages, then use the following installation steps:

For Redhat and CentOS 6.x distributions, type the following to install the kernel level tiering drivers using yum:

```
> sudo yum install kmod-en-tier-XXXX-Y.enmotus.x86_64.rpm
```

and the following to install the user level utilities.

```
> sudo yum install en-tier-manage-XXXX-Y.enmotus.x86_64.rpm
```

where XXXX-Y is the Enmotus version number. For example:

```
> sudo yum install en-tier-manage-4694-1.enmotus.x86_64.rpm kmod-en-  
tier-4694-1.enmotus.x86_64.rpm
```

Alternatively, use rpm to install as follows:

```
> sudo rpm -ihv en-tier-manage-XXXX-1.enmotus.x86_64.rpm kmod-en-tier-  
XXXX-1.enmotus.x86_64.rpm
```

Once installed successfully, start the Enmotus tiering engine service using:

```
> sudo service EnmotusService start
```

Note: OEM versions may differ slightly in name e.g. en-tier-manage-{revision}-{oemname}.x86_64.rpm.

APPENDIX C: Creating FuzeDrive and Preserving Data on one Drive

FuzeDrive software supports creation of a FuzeDrive while preserving data on one drive. This is helpful in the situation where a HDD is running slower than desired and needs more performance. Another application is when an SSD is running out of space and needs more capacity, but only a portion of the SSD is being actively used.

The data to be preserved can be either the fast or slow portion of a FuzeDrive. This feature is performed by using the “ecmd –create” command. Note, this operation uses a different command line syntax “ecmd” instead of “evsp.”

The drive that contains the partition to be preserved will be recognized by the presence of the partition. The other drive in the FuzeDrive must not contain a partition. The FuzeDrive creation process will detect the partition that is to be retained and the user will be prompted to confirm the partition is to be preserved.

See example command and console response below:

This example adds an SSD to the existing HDD with a pre-existing partition

```
[root@centos-6u5-test19 Btest]# sudo ecmd --create /dev/sde /dev/sdb
Ecmd will now do drive discovery, this can take a while.
*** Drive /dev/sdb has a partition and Drive /dev/sde has none.
You can create a tier and preserve the data on /dev/sdb?
Create tier preserving data? (y/n) > y
Create tier preserving drive /dev/sdb data
      O
      O
      O
Tier successfully created and data preserved.
```

```
/dev/sde is the fast device
/dev/sdb is the slow device
```

Before FuzeDrive Creation:

```
[root@centos-6u5-test19 Btest]# ecmd -list
Ecmd will now do drive discovery, this can take a while.
System drives:
```

Device	Vendor	ProductId	Type	Rev	GiB
sdb	TOSHIBA	MK2002TSKB	SATA HDD	MT2A	1863
sde	KINGSTON	SH103S3120G	SATA SSD	524ABBF0	111
sdd	TOSHIBA	THNSNC256GBSJ	SATA SSD	CJGA0202	238
sdc	HGSTHDN	724040ALE640	SATA HDD	MJAOA5E0	3726
*sda	Hitachi	HDS721010KLA330	SATA HDD	GKAOA70F	931

Enmotus pDisks:

ID	ST	T	V	NAME	PRODUCTID	REV	SECTORS AVAIL	SECTORS USED	SIZE GiB
=====									

* Indicates the Boot Drive

> Indicates the Drive is used in a FuzeDrive

After FuzeDrive Creation:

```
[root@centos-6u5-test19 Btest]# ecmd -list
Ecmd will now do drive discovery, this can take a while.
System drives:
```

Device	Vendor	ProductId	Type	Rev	GiB
>sdb	TOSHIBA	MK2002TSKB	SATA HDD	MT2A	1863
>sde	KINGSTON	SH103S3120G	SATA SSD	524ABBF0	111
sdd	TOSHIBA	THNSNC256GBSJ	SATA SSD	CJGA0202	238
sdc	HGSTHDN	724040ALE640	SATA HDD	MJAOA5E0	3726
*sda	Hitachi	HDS721010KLA330	SATA HDD	GKAOA70F	931
eba	ENMOTUS	T01FuzeDrive	VDD TDD	P001	1970

Enmotus pDisks:

ID	ST	T	V	NAME	PRODUCTID	REV	SECTORS AVAIL	SECTORS USED	SIZE GiB
=====									

0	2	1	0	sde	KINGSTONSH103S31	524A	0	db94000	111
1	2	1	1	sdb	TOSHIBA MK2002TS	MT2A	0	e8a08000	1863

* Indicates the Boot Drive

> Indicates the Drive is used in a FuzeDrive