



Release Notes

For Albireo VDO v1.4.2

Permabit Technology Corporation
One Alewife Center, Suite 410
Cambridge, MA 02140 USA

Phone: (+1) 617.252.9600
Fax: (+1) 617.252.9977

info@permabit.com
www.permabit.com

Copyright © 2012 – 2014 Permabit Technology Corporation.

All Rights Reserved. Permabit is a registered trademark and the Permabit logo, Permabit Enterprise Archive, and Scalable Data Reduction are trademarks of the Permabit Technology Corporation. All other products or services mentioned may be covered by registered trademarks, trademarks, service marks, or product names as designated by the companies who market those products.

Legal Notices

Portions of this product are Copyright 2009 – 2014 by Permabit Technology Corporation. THESE PORTIONS ARE PROVIDED ON AN “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

You are solely responsible for determining the appropriateness of using the product and assume all risks associated therewith, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations. You may not reverse engineer, decompile, or disassemble this software except as specifically permitted by applicable law.

NEITHER PERMABIT NOR ITS LICENSORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

About Albireo VDO Version 1.4.2

This document describes changes and known issues in Albireo Virtual Data Optimizer (VDO).

Enhancements and Fixes

Enhancements and Changes in VDO Version 1.4.2

The following features were added in VDO v1.4.2:

- **Improved performance for DISCARD/UNMAP/TRIM**

This release improves the performance of processing DISCARD/UNMAP/TRIM commands. This may reduce the time it takes to format a filesystem or run background space reclamation processes.

- **Improved performance on MD (software) RAID5**

When using Linux's "md" subsystem for RAID5, particularly when using hard drives, performance may be improved by up to 50% on some workloads.

- **VDO's write policy (sync or async) can be changed after a volume is created.**

A command `vdo modify --name=myvol writePolicy=sync|async` has been added, which allows the user to change whether or not the volume is in sync mode or async mode. This policy change takes effect only after the volume has been restarted.

- **Fixed dedupe failure when deduplication is rapidly toggled on and off.**

Enhancements and Changes in VDO Version 1.4.1

The following features were added in VDO v1.4.1:

- **Optional 512-byte block support**

VDO can provide software emulation of 512-byte sectors on VDO volumes configured for 4K blocks.

- **Optional mechanism to force offline recovery after entering read-only mode**

Hardware and software system failures separate from VDO can launch VDO into a read-only mode. Using this option will allow administrators to force the VDO volume into an offline recovery mode so the volume can be brought back online and made available if necessary. OEMs making this option available should alert their customers that the operation does have the risk of some data loss and data corruption.

The following changes have been made to existing VDO commands:

- **vdo:**

- An `--enable512e` option has been added. If specified, the VDO volume will emulate a 512-byte block device.
- The `--forceRebuild` option is now used to force an offline rebuild of a VDO volume that has entered read-only mode. In previous releases it was used to force a rebuild when starting VDO volumes as a way to quickly rebuild usage statistics; the new `--rebuildStatistics` option now serves that purpose.
- A new `--rebuildStatistics` option forces an offline rebuild of a VDO volume before starting it to ensure that its free block statistics are accurate.
- The `--vdoBlockSize` parameter that was used to calculate `--vdoBlockMapPageSize` has been removed.

Values specified for `--vdoBlockMapPageSize` must now be multiples of 4096 (which was the default `--vdoBlockSize` value).

- **vdoStats:**

- The `in recovery mode` status has been replaced by `operating mode`, which indicates whether a VDO volume is operating normally, is in recovery mode, or is in read-only mode.
- A new `recovered count` statistic indicates the number of times a VDO volume has been recovered from read-only mode via `vdo start --forceRebuild`.
- A new `512 byte emulation` attribute indicates whether a VDO volume is running in 512 byte emulation mode.
- A new `bios partial in` statistic displays the number of partial block I/O requests received by VDO (applies only to 512 byte emulation mode).

Enhancements and Changes in VDO Version 1.4

The following features and capabilities were added in VDO v1.4:

- **Improved write performance with the asynchronous write policy**

When a VDO volume is configured with `--writePolicy=async`, writes are buffered in DRAM and committed to storage only after deduplication and compression processes have completed. This reduces the number of I/Os issued to the storage media when duplicate or compressible data is being stored, improving performance and (on NAND flash) reducing wear. Applications should issue FLUSH or FUA commands to VDO (or, from user mode, a data integrity operation like `fsync`, `fdatasync` or `sync`) to guarantee the stability of stored data.

- **Optimized reads of compressed data**

VDO will cache reads of blocks containing compressed fragments, reducing the number of I/Os to the storage device.

- **Optional online recovery after failover or system failure**

VDO volumes may be configured to resume operating quickly (typically in less than 30 seconds) following a crash, and perform most of the recovery while online and available. OEMs using this feature must reserve a portion of the volume for handling write I/Os during recovery. Users will experience degraded performance during the online recovery process, though the amount of degradation can be controlled by setting throughput limits on the recovery process.

- **Additional OS support**

VDO is now tested on RHEL 6.4, CentOS 6.4, and Debian 7.

The following changes have been made to existing VDO commands:

- **vdo:**
 - The `--vdoRecoveryReserveSize`, `--vdoRecoveryScanRate`, and `--vdoRecoverySweepRate` parameters were added to `vdo create` to support online recovery.
 - The `--vdoReadCacheSize` parameter was added to `vdo create` to support read caching.
- **vdoStats:**
 - A new `--version` option displays the current version.
 - The `--verbose` parameter displays additional statistics.

The following issues from VDO v1.3 have been fixed:

- **The `vdo --noRun` option may falsely report errors**

Use of the `--noRun` option with some commands, such as `vdo create`, may cause errors to be incorrectly reported. Despite any errors reported, `--noRun` does function as designed to prevent commands from altering the existing VDO environment.

- **VDO does not enforce the minimum `--blockMapCacheSize`**

Do not change `--blockMapCacheSize` to be less than the required minimum of 128 MB (the default setting).

Enhancements and Changes in VDO Version 1.3

The following features and capabilities were added in VDO v1.3:

- **VDO plug-ins:**

Two Albireo plug-ins were introduced to extend VDO's data optimization capabilities:

- **Albireo REPLICA** is a remote file copy service that works in concert with Albireo VDO to provide data optimization for content both over the wire and at rest. See the *Albireo REPLICA Integration Guide* for more information.
- **Albireo COMPRESS** works in conjunction with VDO to compress data, providing additional storage efficiency. See the *Albireo VDO Integration Guide* for more information.

- **Forced rebuild**

The `--forceRebuild` option causes VDO to rebuild its metadata. This option is useful in testing.

- **Enhanced statistics reporting**

The `vdoStats --human-readable` and `--si` options display block count data in Base 2 (1 KB = 2^{10} bytes = 1024 bytes) and Base 10 formats (1 KB = 10^3 bytes = 1000 bytes), respectively.

The following changes have been made to existing VDO commands:

- **vdo:** The `--enableDeduplication` and `--disableDeduplication` options now operate on running VDO volumes. The `--startDeduplication` and `--stopDeduplication` options have been removed, as they are no longer necessary.
- **vdoStats:** The `Dedupe %` statistic has been replaced with `Space Savings %`. For more detailed information see the `vdoStats` section of the *Albireo VDO Integration Guide*.

Enhancements and Changes in VDO Version 1.2

The following changes were included in VDO v1.2.0.79:

- Adds several enhancements to improve the performance of recovery time. The VDO Integration Guide has been updated to describe how long recovery should take under different circumstances.
- Fixes a potential crash if the kernel attempts to tear down VDO without first suspending it.
- Fixes spurious log messages of the form "page cache save took X heartbeats".

The following change was included in VDO v1.2.0.59:

- Improved default naming scheme for logical volumes.

The following changes were included in VDO 1.2.0.41:

- Eliminated reallocation of memory during `growPhysical`.
- Fixed allocation of extremely large page caches.

The following changes were included in Changes in VDO 1.2.0.29 (1.2 GA release):

- **Online volume resizing**

The existing `growPhysical` subcommand now operates on running volumes; previously, volumes had to be stopped before they could be resized.

- **Page cache customization**

Two new options allow the VDO page cache to be customized:

- The new `--blockMapPageSize` option sets the page size of the block map.

The default of 32 KB is recommended for use with HDD; for SSD, 4 KB is recommended.

- The new `--blockMapCacheSize` option sets the memory allocation for cached block map pages.

Prior versions of VDO had a fixed RAM requirement of 150 MB. VDO v1.2 requires 22 MB plus the size of the page cache (which defaults to the minimum 128 MB). Increasing the cache size may increase performance in some use cases.

- **Accurate deduplication reporting**

VDO now tracks the number of logical blocks mapped to accurately report deduplication statistics.

- **Adds support for additional Linux distributions**

Albireo VDO now supports Red Hat Enterprise Linux (RHEL) 6.3 and CentOS 6.3.

Enhancements and Changes in VDO Version 1.1

The following features and capabilities were added in VDO v1.1:

- **Synchronous writes**

Albireo VDO has a synchronous mode in which writes are not acknowledged until data resides on stable storage. This enables VDO to recover from an unclean shutdown with no loss of data.

The `--writePolicy` controls this feature. Setting this option to `async` enables asynchronous mode, where writes are acknowledged after data has been *cached* for writing to stable storage.

- **Increased scalability**

VDO supports physical volume sizes of up to 256 TB.

- **Reduced RAM requirements**

See the *Albireo VDO Integration Guide* for more information.

Compatibility

Albireo VDO v1.4.2 is not backward-compatible with volumes created by previous versions of VDO.

Known Issues

The following issues are known to exist in Albireo VDO Version 1.4.2.

- **A VDO installation failure may cause subsequent installation attempts to fail**

VDO installation may fail if compiled against a mismatched kernel and headers, leaving the install in an unclean state and causing subsequent VDO installs to fail. To resolve this issue, uninstall VDO (`vdoInstaller uninstall`) and then retry the installation.

- **VDO volumes may not start at boot**

VDO volumes may occasionally fail to start automatically on some systems due to vdo manager command locking. Should this occur, remove the file `/tmp/vdo.lock` if it exists and run the following script as the root user.

```
/etc/init.d/kvdo start
```

Issues in Third Party Software

These are the known issues in third party software that may affect Albireo VDO.

- **Many versions of XFS do not correctly handle power failure when operating on top of storage with volatile writeback caches, including VDO in "async" mode.**

It is recommended to use "sync" mode when running VDO on XFS. See the following URL for a discussion on this issue: http://xfs.org/index.php/XFS_FAQ#Q:_What_is_the_problem_with_the_write_cache_on_journaled_filesystems.3F

- **Installing VDO on SLES requires 500 MB of additional RAM.**

Extensive testing has shown that the minimum memory requirements for operating VDO on SLES 11 are higher than for other Linux distributions. To achieve expected performance, VDO on SLES requires an additional 500 MB of available memory.

- **Debian 6.0 ISO distribution will not load VDO.**

Workaround: VDO is built using the 3.2.0 kernel from Debian 7.0. The Debian 6.0 backported kernel 3.2.0-0.bpo.2-amd64 has been tested and does work with VDO. This can be obtained from the following website: <http://backports-master.debian.org/Mirrors/>

- **Installing VDO on a SLES 11 system may fail with a message stating the system does not allow unsupported modules.**

Workaround: Modify the setting in `/etc/modprobe.d/unsupported-modules` to allow unsupported modules. This is the intended behavior of SLES 11, which by default allows only approved modules to be

installed. See the Knowledge Base article at <http://www.novell.com/support/kb/doc.php?id=7002793> for more information.

- **VDO may require that the `kvdo` module be listed as dependent upon a hardware device driver before being used on a device.**

Workaround: Add a dependency on the hardware device driver for `kvdo` using `inserv` to ensure that VDO does not try to load before the necessary underlying storage devices. See <http://www.novell.com/support/kb/doc.php?id=7002295> for documentation.

- **VDO performance on SSD is significantly reduced when using I/O schedulers other than `noop`.**

On Debian systems, this can be controlled by installing the `sysfsutils` package and adding the following line to `/etc/sysfs`:

```
block/device_name/queue/scheduler = noop
```

On SLES, this setting can be adjusted at boot time by adding the following line to `/etc/rc.local`:

```
echo noop > /sys/block/device_name/queue/scheduler
```

- **LVM on SLES 11 SP2 may not restart on reboot automatically.**

If LVM does not start automatically at boot, VDO will be unable to start and the volumes will not be available.

This setting can be changed to start LVM at boot time by running the following command as root:

```
chkconfig boot.lvm on
```

- **The Google Chrome PDF viewer on Linux may not properly display the product documentation.**

If a PDF appears poorly formatted, try an alternate browser/reader.