IBM ioMemory VSL 4.1.0

User Guide for Linux

Table of Contents

IBM ioMemory VSL 4.1.0 User Guide for Linux

Legal Notices

Part Number: D0001565-008_1
Published: July 28, 2014

Introduction

Overview

Congratulations on your purchase of an IBM solid-state storage device. This guide explains how to install, troubleshoot, and maintain the software for your IBM io3 Flash Adapters.

NOTE-

Throughout this manual, when you see a reference to an **IBM io3 Flash Adapter**, you may substitute your particular device(s) from the *Support Devices* list located in the *IBM ioMemory VSL Release Notes*.

About the IBM io3 Flash AdapterPlatform

The IBM io3 Flash Adapter platform combines ioMemory VSL software (VSL stands for Virtual Storage Layer) with IBM io3 Flash Adapter hardware to take enterprise applications and databases to the next level.

Performance

The IBM io3 Flash Adapter platform provides consistent microsecond latency access for mixed workloads, multiple gigabytes per second access and hundreds of thousands of IOPS from a single product. The sophisticated IBM io3 Flash Adapter architecture allows for nearly symmetrical read and write performance with best-in-class low queue depth performance, making the IBM io3 Flash Adapter platform ideal across a wide variety of real world, high-performance enterprise environments.

The IBM io3 Flash Adapter platform integrates with host system CPUs as flash memory to give multiple (and mostly idle) processor cores, direct and parallel access to the flash. The platform's cut-through architecture gives systems more work per unit of processing, and continues to deliver performance increases as CPU power increases.

Endurance

The IBM io3 Flash Adapter platform offers best-in-class endurance in all capacities, which is crucial for caching and write-heavy databases and applications.

Reliability

The IBM io3 Flash Adapter platform eliminates concerns about reliability like NAND failures and excessive wear. The intelligent, self-healing feature called Adaptive Flashback provides complete, chip-level fault tolerance. Adaptive Flashback technology enables an IBM io3 Flash Adapter product to repair itself after a single chip or a multi-chip failure without interrupting business continuity.

System Requirements

Please read the *IBM ioMemory VSL Release Notes* for more information on this release.

Hardware Requirements

- **Hardware Requirements**: These depend on your device (including device capacity, generation, and configuration). Please see the *IBM io3 Flash Adapter Hardware Installation Guide* for requirements on the following:
  - PCIe Slot
  - Cooling
  - Power
- **Supported Devices**: Also see the *IBM io3 Flash Adapter Hardware Installation Guide* for a list of supported IBM io3 Flash Adapters.
- **RAM Requirements**: The *IBM ioMemory VSL Release Notes* contains memory (RAM) requirements for this version of the software.

For specific IBM io3 Flash Adapter System x server configuration information and requirements, refer to the following URL: http://www.ibm.com/support/entry/portal/docdisplay?lndocid=SERV-IO3

Supported Linux Distributions

- Red Hat Enterprise Linux 5 (up to 5.10), 6 (up to 6.5), 7.0
- SUSE Linux Enterprise Server (SLES) 11, 11 SP2, 11 SP3

ioMemory VSL Software Installation

NOTE-

All commands require administrator privileges. Use `sudo` or log in as "root" to run the install.

Installation Overview

1. Download the latest version of the software at http://www.ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-65723 (follow that link and then select **IBM High IOPS and io3 software matrix**).

2. If you have a previous version of the ioMemory VSL software installed, you must uninstall the ioMemory VSL software and the utilities. See See Uninstalling the ioMemory VSL Software on page 34 for instructions. Once you have uninstalled the software, return to this page.

   Attention!

   This version of the ioMemory VSL software only supports third generation devices (such as io3 Enterprise Value Adapters and io3 Enterprise Adapters, and it does not support devices that were compatible with ioMemory VSL software version 3.x.x or earlier.

   Attention!

   Kernel Upgrades

   If you ever plan to upgrade the kernel when the ioMemory VSL software is installed, you **must** follow the procedure for See Upgrading the Kernel on page 35.

3. Install the ioMemory VSL software. The installation instructions will help you determine which of these package types is supported by your kernel:

   - A pre-compiled binary package

   - A source-to-build package

4. Install utilities and management software (included in driver installation instructions).

5. Follow the instructions on See Loading the ioMemory VSL Driver on page 14 and consider See Setting the ioMemory VSL Options on page 19.

6. Determine if you need to upgrade the firmware to the latest version, see See Upgrading the Firmware on page 17.

Installing RPM Packages

To install the Linux ioMemory VSL software and utilities:

1. You will need to install a version of the ioMemory VSL software that is built for your kernel. To determine what kernel version is running on your system, use the following command at a shell prompt:

   ```
   $ uname -r
   ```

2. Compare your kernel version with the binary versions of the software available at http://www.ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-65723 (follow that link and then select **IBM High IOPS and io3 software matrix**).
   - If there is a binary version of the software that corresponds to your kernel version, download that. **For example**:

     ```
     iomemory-vsl4-<kernel-version>-<VSL-version>.x86_64.rpm
     ```

   - If there is no binary version of the software corresponding to your kernel, download the source package. **For example**:

     ```
     iomemory-vsl4-<VSL-version>.src.rpm
     ```

     NOTE-

     Exact package names may vary, depending on software and kernel version chosen.

     Use the source package that is made for your distribution. Source packages from other distributions are not guaranteed to work.

3. Download the support RPM packages you need from http://www.ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-65723 (follow that link and then select **IBM High IOPS and io3 software matrix**). These packages provide utilities, firmware, and other files.

   Examples:

Package	What Is Installed
`fio-util-<VSL-version>.x86_64.rpm`	ioMemory VSL utilities – **Recommended**
`fio-firmware-highiops-<version>.<date>-1.0.noarch.rpm`	Firmware archive package, installs the firmware archive file in a specific location (`/usr/share/fio/firmware`) – **Optional**

`fio-firmware-highiops-<version>.<date>.fff`	Firmware archive file (not an installer package) used to upgrade the firmware, make note of where you store this file – **Recommended**
`libvsl-<version>.x86_64.rpm`	SDK libraries needed for management tools – **Recommended**, see See Monitoring and Managing Devices on page 29 for more information on available management tools.
`lib32vsl-<version>.i386.rpm`	SDK libraries needed for working with 32-bit management applications – **Optional** and only available for certain distributions. **Attention!** You must have a full set of 32-bit system libraries installed (32-bit compatibility layer installed on your 64-bit system) before you can install this package.
`fio-common-<VSL-version>.x86_64.rpm`	Files required for the init script – **Recommended**
`fio-sysvinit-<VSL-version>.x86_64.rpm`	Init script – **Recommended**, see See Loading the ioMemory VSL Driver on page 14 for more information.

4. Change to the directory to where you downloaded the installation packages.
5. If needed, build the ioMemory VSL software from source:
   - **If you downloaded a binary version of the software**: skip the **Building the Software from Source** instructions below and continue to the next step.
   - **If you downloaded the software source package**: follow these **Building the Software from Source** instructions:

Building the Software from Source

You only need to follow these additional instructions if you downloaded the source package.

a. Install the prerequisite files for your kernel version.

> **NOTE-**
>
> Some of the prerequisite packages may already be in the default OS installation. If your system is not configured to get packages over the network, then you may need to mount your install CD/DVD.

- On RHEL 5/6, you need `kernel-devel`, `rpm-build`, GCC4, `make`, and `rsync`.

```
$ yum install rsync tar gcc make kernel-devel-`uname
-r` rpm-build
```

This command will force yum to download the exact versions for your kernel. If the exact versions are no longer available in the repository, then you will have to manually search for them on the Internet.

- On SLES 10/11 you need kernel-syms, make, rpm-build, GCC4, and rsync

```
$ zypper install kernel-syms make rpm gcc rsync
```

b. To build an RPM installation package for the current kernel, navigate to the directory with the downloaded source RPM file and run this command:

```
$ rpmbuild --rebuild iomemory-vsl4-<VSL-version>.src.rpm
```

When using a .rpm source package for a non-running kernel, run this command:

```
$ rpmbuild --rebuild --define 'rpm_kernel_version
<kernel-version>'  iomemory-vsl-<VSL-version>.src.rpm
```

c. The new RPM package is located in a directory that is indicated in the output from the rpmbuild command. To find it, look for the Wrote line. In the following example, the RPM packages are located in the /usr/src/redhat/RPMS/x86_64/ directory.

```
...
Processing files: iomemory-vsl-source-<version>-1.0.x86_
64.rpm
Requires(rpmlib): rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rpmlib(CompressedFileNames) <= 3.0.4-1
Obsoletes: iodrive-driver-source
Checking for unpackaged file(s): /usr/lib/rpm/check-files
/var/tmp/iomemory-vsl-<version>-root
Wrote: /usr/src/redhat/RPMS/x86_64/iomemory-vsl-2.6.18-
128.el5-<version>-1.0.x86_64.rpm
Wrote: /usr/src/redhat/RPMS/x86_64/iomemory-vsl-source-
<version>-1.0.x86_64.rpm
```

In this example, iomemory-vsl4-2.6.18-128.el5-<version>-1.0.x86_64.rpm is the package you will use.

d. Copy your custom-built software installation RPM package into the directory where you downloaded the installation packages and navigate to that directory.

e. Continue to the next step.

6. Enter the following command to install the custom-built software package. Use the package name that you just copied/downloaded into that directory.

```
$ rpm -Uvh iomemory-vsl4-<kernel-version>-<VSL-version>.x86_64.rpm
```

7. Enter the following commands to install the support files:

```
$ rpm -Uvh lib*.rpm
rpm -Uvh fio*.rpm
```

Attention!

If the installation of the `fio-util-<VSL-version>.x86_64.rpm` package fails due to missing dependencies, you will need to install the missing packages before you run the install command again, for example:

```
$ yum install lsof pciutils
```

The ioMemory VSL software and utilities are installed to the following locations:

Package Type	Installation Location
ioMemory VSL software	`/lib/modules/<kernel-version>/extra/fio/iomemory-vsl4.ko`
Utilities	`/usr/bin`

NOTE-

You may also install the IBM Flash Management Console (optional GUI management software). IBM Flash Management Console and documentation are available as a separate download.

Once the packages are installed, continue to See Loading the ioMemory VSL Driver on page 14.

Loading the ioMemory VSL Driver

To load the ioMemory VSL driver:

Run this command:

```
$ modprobe iomemory-vsl4
```

NOTE-

The ioMemory VSL driver automatically loads at system boot. The IBM io3 Flash Adapter is now available to the OS as `/dev/fiox`, where *x* is a letter (i.e., a, b, c, etc.).

To confirm the IBM io3 Flash Adapter is attached, run the `fio-status` utility from the command line. The output lists each drive and its status (attached or not attached).

Attention!

If the IBM io3 Flash Adapter is not automatically attaching, check the `/etc/modprobe.d` files to ensure that the `auto_attach` option is turned on (set to 1).

On SLES systems, you must also allow unsupported modules for this command to work. Modify the `/etc/modprobe.d/iomemory-vsl4.conf` file and uncomment the appropriate line:

```
# To allow the ioMemory VSL driver to load on SLES11, uncomment below
allow_unsupported_modules 1
```

Controlling Driver Loading

You can control driver loading either through the init script or through udev.

In newer Linux distributions, users can rely on the udev device manager to automatically find and load drivers for their installed hardware at boot time, though udev can be disabled and the init script used in nearly all cases. We recommend using the init script to load the ioMemory VSL driver if you are managing a RAID array using LVM, mdadm, or Veritas Storage Foundation.

For older Linux distributions without udev functionality, users must rely on a boot-time init script to load needed drivers.

Using the init Script

On systems where udev loading of the driver doesn't work, or is disabled, the init script may be enabled to load the driver at boot. On some distros it may be enabled by default.

NOTE-

The init Script is part of the `fio-sysvinit` package, which must be installed before you can enable it.

You can disable this loading of the driver with the following command:

```
$ chkconfig --del iomemory-vsl4
```

To re-enable the driver loading in the init script, use the following command:

```
$ chkconfig --add iomemory-vsl4
```

The ioMemory VSL software install process places an init script in `/etc/init.d/iomemory-vsl4`. In turn, this script uses the setting options found in the options file in `/etc/sysconfig/iomemory-vsl4`. The options file must have ENABLED set (non-zero) for the init script to be used:

```
ENABLED=1
```

The options file contains documentation for the various settings—two of which, `MOUNTS` and `KILL_PROCS_ON_UMOUNT`, are discussed further in the section See Handling Driver Unloads on page 17.

Mounting Filesystems when Using the init Script

Because the ioMemory VSL driver does not load by the standard means (in the `initrd`, or built into the kernel), using the standard method for mounting filesystems (`/etc/fstab`) for filesystems hosted on the ioMemory VSL software does not work. To set up auto-mounting of a filesystem hosted on an IBM io3 Flash Adapter:

1. Add the filesystem mounting command to `/etc/fstab` as normal.
2. You must add the '`noauto`' option and the '`0 0`' flag to `/etc/fstab` as in the two following sample entries (modify the samples to fit your use case, including your filesystem).

```
/dev/fioa /mnt/fioa ext3 defaults,noauto 0 0
/dev/fiob1 /mnt/ioDrive ext3 defaults,noauto 0 0
```

(where the `a` in `fioa` can be `a`, `b`, `c`, etc., depending on how many IBM io3 Flash Adapters you have installed in the system).

Attention!

Failure to add '`noauto 0 0`' to `fstab` may cause a boot failure.

To have the init script mount these drives after the driver is loaded and unmounted and before the driver is unloaded, add a list of mount points to the options file using the procedure documented there.

For the filesystem mounts shown in the earlier example, the line in the options file would look like this:

```
MOUNTS="/mnt/fioa /mnt/iodrive"
```

Using udev

On systems that rely on udev to load drivers, users need to modify an ioMemory VSL software options file if they want to prevent udev from auto-loading the ioMemory VSL at boot time. To do this, locate and edit the `/etc/modprobe.d/iomemory-vsl4.conf` file that already has the following line:

```
# blacklist iomemory-vsl4
```

To disable loading, remove the "#" from the line and save the file.

With the blacklist command in place, restart Linux. The ioMemory VSL driver will not be loaded by udev.

To restore the udev-loading of the driver, replace the "#" to comment out the line.

On either udev or init script systems

Users can disable the loading of the driver at boot time, and thus prevent the auto-attach process for diagnostic or troubleshooting purposes on either udev or init script systems. Follow the steps in the section See Disabling Auto-Attach on page 35 to disable or re-enable the auto-attach functionality.

IBM io3 Flash Adapters and Multipath Storage

If you are using IBM io3 Flash Adapters along with multipath storage, you must blacklist the IBM io3 Flash Adapters to prevent `device-mapper` from trying to create a `dm-device` for each IBM io3 Flash Adapter. **This must be done prior to activating `dm-multipath` and/or loading the driver**. If IBM io3 Flash

Adapters are not blacklisted, they will appear busy and you will not be able to attach, detach, or update the firmware on the devices.

To blacklist IBM io3 Flash Adapters, edit the `/etc/multipath.conf` file and include the following:

```
blacklist {
devnode          "^fio[a-z]"
}
```

Handling Driver Unloads

Special consideration must be taken during ioMemory VSL driver unload time. By default, the init script searches for any processes holding open a mounted filesystem and kills them, thus allowing the filesystem to be unmounted. This behavior is controlled by the option `KILL_PROCS_ON_UMOUNT` in the options file. If these processes are not killed, the filesystem cannot be unmounted. This may keep the ioMemory VSL software from unloading cleanly, causing a significant delay on the subsequent boot.

Upgrading the Firmware

With the ioMemory VSL software loaded, you need to check to ensure that the IBM io3 Flash Adapter's firmware is up-to-date and then update the firmware if needed. You can do this with either the command-line utilities or the optional IBM Flash Management Console (GUI).

NOTE-

Make sure you have downloaded the firmware archive file that goes with this version of the ioMemory VSL software.

Attention!

Do not attempt to downgrade the firmware on any IBM io3 Flash Adapter, doing so may void your warranty.

Attention!

When installing a new IBM io3 Flash Adapter along with existing devices, you must upgrade all of the currently installed devices to the latest available versions of the firmware and ioMemory VSL software before installing the new devices. Consult the *IBM ioMemory VSL Release Notes* for this ioMemory VSL software release for any upgrade considerations.

Attention!

Upgrading Guest OS

If you are using your IBM io3 Flash Adapter within a guest OS (for example, using VMDirectPathIO), you must power cycle the host server after you upgrade the device(s). Just restarting the virtual machine will not apply the the firmware update.

Command-line Interface

More information on these command-line utilities is available in See Command-line Utilities Reference on page 41. All command-line utilities require root permission . Follow these steps to upgrade the firmware:

1. Run the `fio-status` utility and examine the output. See See fio-status on page 48 for usage information.

   - If any device is in minimal mode and the reason is outdated firmware then you must update the firmware.

   - If the a device is not in minimal mode, but the firmware listed for that device is a lower number than the latest firmware version available with this version of the ioMemory VSL software, then the firmware is old, but not outdated.

2. If the firmware is old or outdated, update it using the `fio-update-iodrive` utility. See See fio-update-iodrive on page 52 for complete information and warnings.

Optional GUI - IBM Flash Management Console

You can use the IBM Flash Management Console to check the status of your IBM io3 Flash Adapters. If the IBM Flash Management Console indicates that the device's firmware is outdated, you can also use the IBM Flash Management Console to upgrade the device firmware. Consult the IBM Flash Management Console documentation for more information on installing and using the software.

Configuration

Once you have your IBM io3 Flash Adapter and ioMemory VSL software installed and loaded, and the firmware on the device is current, you may need to configure the device and/or software. This section outlines some of the common configurations that you may need to consider.

Setting the ioMemory VSL Options

This section explains how to set ioMemory VSL software options. For more information about setting specific options, see See Using Module Parameters on page 58.

One-Time Configuration

ioMemory VSL software options can be set at install time, on the command line of either `insmod` or `modprobe`. For example, to set the `auto_attach` option to 0, run the command:

```
$ modprobe iomemory-vsl4 auto_attach=0
```

This option takes effect only for this load of the ioMemory VSL software; subsequent calls to `modprobe` or `insmod` will not have this option set.

Persistent Configuration

To maintain a persistent setting for an option, add the desired option to `/etc/modprobe.d/iomemory-vsl4.conf` or a similar file. To prevent the IBM io3 Flash Adapters from auto-attaching, add the following line to the `iomemory-vsl4.conf` file:

```
options iomemory-vsl4 auto_attach=0
```

This option then takes effect for every subsequent ioMemory VSL software load, as well as on autoload of the ioMemory VSL software during boot time.

Using the Device as Swap

To safely use the IBM io3 Flash Adapter as swap space requires passing the preallocate_memory kernel module parameter. To set this parameter, use either the optional IBM Flash Management Console (see IBM Flash Management Console documentation), or add the following line to the `/etc/modprobe.d/iomemory-vsl4.conf` file (see See Using Module Parameters on page 58 for more information on using parameters):

```
options iomemory-vsl4 preallocate_memory=1149D2717-1121,1149D2717-1111,
10345
```

- Where `1149D2717-1111`,`1149D2717-1111`,`10345` are serial numbers obtained from `fio-status`, see See fio-format on page 45.

A 4K sector size format is required for swap—this reduces the ioMemory VSL software memory footprint. Use `fio-format` to format the IBM io3 Flash Adapter with 4K sector sizes.

Attention!

You must have enough RAM available to enable the IBM io3 Flash Adapter with pre-allocation enabled for use as swap. Attaching an IBM io3 Flash Adapter, with pre-allocation enabled, without sufficient RAM may result in the loss of user processes and system instability.

Consult the *IBM ioMemory VSL Release Notes* for RAM requirements with this version of the ioMemory VSL software.

NOTE-

The `preallocate_memory` parameter is recognized by the ioMemory VSL software at load time, but the requested memory is not actually allocated until the specified device is attached.

Using the Logical Volume Manager

The Logical Volume Manager (LVM) volume group management application handles mass storage devices like IBM io3 Flash Adapters if you add the IBM io3 Flash Adapter as a supported type:

1. Locate and edit the `/etc/lvm/lvm.conf` configuration file.

2. Add an entry similar to the following to that file:

```
types = [ "fio", 16 ]
```

The parameter "16" represents the maximum number of partitions supported by the device.

NOTE-

If using LVM or MD, do not use udev to load the ioMemory VSL driver. The init script will ensure that the LVM volumes and MD devices are detached before attempting to detach the IBM io3 Flash Adapter.

Mounting Filesystems when Using the init Script

If you are using the init script to load the ioMemory VSL software, once you have configured your devices you will need to follow the instructions in the *Mounting Filesystems when Using the init Script* subsection of the section See Controlling Driver Loading on page 15.

Configuring RAID using the Logical Volume Manager

The simplest way to using your IBM io3 Flash Adapters with LVM is to use the entire block device with it (for example: `/dev/fioa` rather than `/dev/fioa1`, `/dev/fioa2`, etc.). This way the block device does not need to be partitioned ahead of time, though partitioning is also supported. The examples that follow assumes the entire block device is used.

Whether you plan to stripe (RAID 0) or mirror (RAID 1) the devices, the first two steps are the same:

1. First, create physical volumes, for example:

```
$ pvcreate /dev/fioa  /dev/fiob
```

2. Next add these physical volumes to a volume group, for example:

```
$ vgcreate iomemory_vg /dev/fioa /dev/fiob
```

Creating a Striped Volume (RAID 0) Using LVM

With the volume group created, you can create logical volumes within this volume group. In this instance, only one will be created and we name it `iomemory_lv`:

1. Create the striped volume using the `-i2` option for two stripes, for example:

```
$ lvcreate -l 100%VG -n iomemory_lv -i2 iomemory_vg
```

2. Create a file system on the newly created volume, for example:

```
$ mkfs.ext3 /dev/iomemory_vg/iomemory_lv
```

3. In order to make this persistently mount at boot time, edit the `/etc/sysconfig/iomemory-vsl` file by adding the volume group path under the example line:

```
Example: LVM_VGS="/dev/vg0 /dev/vg1"
LVM_VGS="/dev/iomemory_vg"
```

Be sure to just add the volume group path and not the logical volume as well. For more information on using the init script and the `/etc/sysconfig/iomemory-vsl4` file, see See Loading the ioMemory VSL Driver on page 14.

Creating a Mirrored Volume (RAID 1) Using LVM

With the volume group created, you can create logical volumes within this volume group. In this instance, only one will be created and we name it `iomemory_lv`:

1. Create the mirrored volume using the `-m1` option for an original linear volume plus one copy, for example:

```
$ lvcreate -l 50%VG -n iomemory_lv -m1 --corelog iomemory_vg
```

You can monitor the progress of the initial mirror synchronization using the lvs command. For example:

```
lvs -a -o +devices
```

Attention!

With the \--corelog option, LVM uses an in-memory region or log to track the state of the mirror legs. Since it is in-memory and lost at reboot, it must be regenerated at boot by scanning the mirror legs. For alternative configurations, consult the LVM documentation.

2. Create a file system on the newly created volume, for example:

```
$ mkfs.ext3 /dev/iomemory_vg/iomemory_lv
```

3. In order to make this persistently mount at boot time, edit the /etc/sysconfig/iomemory-vsl4 file by adding the volume group path under the example line:

```
# Example: LVM_VGS="/dev/vg0 /dev/vg1"
LVM_VGS="/dev/iomemory_vg"
```

Be sure to just add the volume group path and not the logical volume as well. For more information on using the init script and the /etc/sysconfig/iomemory-vsl4 file, see See Loading the ioMemory VSL Driver on page 14.

Configuring RAID Using mdadm

You can configure two or more IBM io3 Flash Adapters into a RAID array using software-based RAID solutions.

NOTE-

If you are using RAID1/Mirrored and one device fails, be sure to run fio-format on the replacement device (not the existing, good device) before rebuilding the RAID. Following are some examples of some common RAID configurations using the mdadm utility.

Attention!

The Linux kernel RAID 5 implementation performs poorly at high data rates. This is an issue in the Linux kernel. Alternatives include using RAID 10, or possibly a third-party RAID stack.

Mounting Arrays

Once you are done making your array (by following one of the configuration samples below), you must edit the /etc/sysconfig/iomemory-vsl4 file and add the array path under the example line. This will make the array mount at boot time.

In all of the examples below, we create arrays named md0. Here is is how you would edit the /etc/sysconfig/iomemory-vsl4 file to include the array using md0 as an example (you add the array just below the example line in that file):

```
# Example: MD_ARRAYS="/dev/md0 /dev/md1"
MD_ARRAYS="/dev/md0"
```

If you are using the init script to load the ioMemory VSL software, once you have configured your devices you will need to follow the instructions in the *Mounting Filesystems when Using the init Script* subsection of the section See Controlling Driver Loading on page 15.

RAID 0

To create a striped set, where `fioa` and `fiob` are the two IBM io3 Flash Adapters you want to stripe, run this command:

```
$ mdadm --create /dev/md0 --chunk=256 --level=0 --raid-devices=2 /dev/fioa
/dev/fiob
```

Making the Array Persistent (Existing after Restart)

On some versions of Linux, the configuration file is in `/etc/mdadm/mdadm.conf`, not `/etc/mdadm.conf`.

Inspect `/etc/mdadm.conf`. If there are one or more lines declaring the devices to inspect, make sure one of those lines specifies "partitions" as an option. If it does not, add a new DEVICE line to the file specifying "partitions" like this:

```
DEVICE partitions
```

Also add a device specifier for the fio IBM io3 Flash Adapters:

```
DEVICE /dev/fio*
```

To see if any updates are needed to `/etc/mdadm.conf`, issue the following command:

```
$ mdadm --examine --scan
```

Compare the output of this command to what currently exists in `mdadm.conf` and add any needed sections to `/etc/mdadm.conf`.

NOTE-

For example, if the array consists of two devices, there will be three lines in the output of the command that are not present in the `mdadm.conf` file: one line for the array, and two device lines (one line for each device). Be sure to add those lines to the `mdadm.conf` so it matches the output of the command.

For further details please see the `mdadm` and `mdadm.conf` man pages for your distribution.

With these changes, on most systems the RAID 0 array will be created automatically upon restart. However, if you have problems accessing `/dev/md0` after restart, run the following command:

```
$ mdadm --assemble --scan
```

You may also want to disable udev loading of the ioMemory VSL driver, if needed, and use the init script provided for driver loading. Please see the *Using the init Script* subsection of the section See Controlling Driver Loading on page 15 for further details on how to use the init script.

NOTE-

In SLES 11, you may need to run the following commands to make sure these services are run on boot:

```
chkconfig boot.md on chkconfig mdadmd on
```

RAID 1

To create a mirrored set using the two IBM io3 Flash Adapters `fioa` and `fiob`, run this command:

```
$ mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/fioa /dev/fiob
```

RAID 10

To create a striped, mirrored array using four IBM io3 Flash Adapters (fioa, fiob, fioc, and fiod), run this command:

```
$ mdadm --create /dev/md0 -v --chunk=256 --level=raid10 --raid-devices=4
/dev/fioa /dev/fiob /dev/fioc /dev/fiod
```

Discard (TRIM) Support

With this version of the ioMemory VSL software, Discard (also known as TRIM) is enabled by default.

Discard addresses an issue unique to solid-state storage. When a user deletes a file, the device is not aware that it can reclaim the space. Instead the device must assume the data is valid.

Discard is a feature on most modern filesystem releases. It informs the device of logical sectors that no longer contain valid user data. This allows the wear-leveling software to reclaim that space (as reserve) to handle future write operations.

Discard (TRIM) on Linux

Discard is enabled by default in the ioMemory VSL software release. However, for discard to be implemented, the Linux distribution must support this feature, and discard must be turned on.

In other words, if your Linux distribution supports discard, and discard is enabled on the system, then discard will be implemented on your IBM io3 Flash Adapter.

Under Linux, discards are not limited to being created by the filesystem, discard requests can also be generated directly from userspace applications using the kernel's discard ioctl.

Attention!

There is a known issue that ext4 in Kernel.org 2.6.33 or earlier may silently corrupt data when discard is enabled. This has been fixed in many kernels provided by distribution vendors. Please check with your kernel provider to be sure your kernel properly supports discard. For more information, see the Errata in the *IBM ioMemory VSL Release Notes* for this version of the software

NOTE-

On some Linux distributions, MD and LVM do not pass discards to underlying devices and any IBM io3 Flash Adapters that are part of an MD or LVM array will not receive discards sent by the filesystem. Please see your distribution's documents for exact details.

Starting in RHEL 6.5, MD does support discard for RAID-0, RAID-1, and RAID-10.

Performance and Tuning

IBM io3 Flash Adapters provide high bandwidth, high Input/Output per Second (IOPS), and are specifically designed to achieve low latency.

As IBM io3 Flash Adapters improve IOPS and low latency, the device performance may be limited by operating system settings and BIOS configuration. These settings may need to be tuned to take advantage of the revolutionary performance of IBM io3 Flash Adapters.

While IBM io3 Flash Adapters generally perform well out of the box, this section describes some of the common areas where tuning may help achieve optimal performance.

Disable CPU Frequency Scaling

Dynamic Voltage and Frequency Scaling (DVFS) are power management techniques that adjust the CPU voltage and/or frequency to reduce power consumption by the CPU. These techniques help conserve power and reduce the heat generated by the CPU, but they adversely affect performance while the CPU transitions between low-power and high-performance states.

These power-savings techniques are known to have a negative impact on I/O latency and IOPS. When tuning for performance, you may benefit from reducing or disabling DVFS completely, even though this may increase power consumption.

DVFS, if available, is often configurable as part of your operating systems power management features as well as within your system's BIOS interface. Within the operating system and BIOS, DVFS features are often found under the Advanced Configuration and Power Interface (ACPI) sections; consult your computer documentation for details.

Limiting ACPI C-States

Newer processors have the ability to go into lower power modes when they are not fully utilized. These idle states are known as ACPI C-states. The `C0` state is the normal, full power, operating state. Higher C-states (`C1`, `C2`, `C3`, etc.) are lower power states.

While ACPI C-states save on power, they can have a negative impact on I/O latency and maximum IOPS. With each higher C-state, typically more processor functions are limited to save power, and it takes time to restore the processor to the `C0` state.

When tuning for maximum performance you may benefit from limiting the C-states or turning them off completely, even though this may increase power consumption.

Setting ACPI C-State Options

If your processor has ACPI C-states available, you can typically limit or disable them in the BIOS interface (sometimes referred to as a Setup Utility). APCI C-states may be part of of the Advanced Configuration and Power Interface (ACPI) menu. Consult your computer documentation for details.

C-States Under Linux

Newer Linux kernels have drivers that may attempt to enable APCI C-states even if they are disabled in the BIOS. You can limit the C-state in Linux (with or without the BIOS setting) by adding the following to the kernel boot options:

```
intel_idle.max_cstate=0 processor.max_cstate=0
```

In this example, the maximum C-state allowed will be `C0` (disabled).

Setting NUMA Affinity

Servers with a NUMA (Non-Uniform Memory Access) architecture may require special installation instructions in order to maximize IBM io3 Flash Adapter performance. This includes most multi-socket servers.

On some servers with NUMA architecture, during system boot, the BIOS will not associate PCIe slots with the correct NUMA node. Incorrect mappings result in inefficient I/O handling that can significantly degrade performance.

The ioMemory VSL software automatically reassigns devices with the appropriate NUMA node. However, you may still use deprecated NUMA node parameters to manually assign devices to the available NUMA nodes.

See See NUMA Configuration on page 60 for more information on these deprecated parameters for setting this affinity.

Setting the Interrupt Handler Affinity

Device latency can be affected by placement of interrupts on NUMA systems. We recommend placing interrupts for a given device on the same NUMA node that the application is issuing I/O from. If the CPUs on this node are overwhelmed with user application tasks, in some cases it may benefit performance to move the the interrupts to a remote node to help load-balance the system.

Many operating systems will attempt to dynamically place interrupts across the nodes, and generally make good decisions.

Linux IRQ Balancing

In Linux this dynamic placement is called IRQ Balancing. You can check to see if the IRQ balancer is effective by checking `/proc/interrupts`. If the interrupts are unbalanced (too many device interrupts on one node) or on an overwhelmed node, you may need to stop the IRQ balancer and manually distribute the interrupts in order to balance the load and improve performance.

NOTE-

Restarting the IRQ Balancer after the ioMemory VSL software loads (and the IBM io3 Flash Adapters are attached) may resolve interrupt affinity issues. For example, run one of the following commands (depending on your distribution):

```
/etc/init.d/irqbalance restart
```

```
/etc/init.d/irq_balancer restart
```

If that does not resolve the affinity issues, then we recommend manual pinning the device interrupts to specific nodes.

Hand-tuning interrupt placement in Linux is an advanced option that requires profiling of application performance on any given hardware. Please see your operating system documentation for information on how to pin specific device interrupts to specific nodes.

Monitoring and Managing Devices

IBM provides many tools for managing your IBM io3 Flash Adapters. These tools will allow you to monitor the devices for errors, warnings, and potential problems. They will also allow you to manage the devices including performing the following functions:

- Firmware upgrades
- Low-level formatting
- Attach and detach actions
- Device status and performance information
- Configuring Swap and Paging
- Generating bug reports

Management Tools

IBM has provided several tools for monitoring and managing IBM io3 Flash Adapters. These include stand-alone tools that require no additional software and data-source tools that can be integrated with other applications.

Consider the descriptions of each tool to decide which tool (or combination of tools) best fits your needs.

> **Attention!**
>
> The ioMemory VSL software does print some error messages to the system logs, and while these messages are very useful for troubleshooting purposes, the ioMemory VSL software log messages are not designed for continual monitoring purposes (as each is based on a variety of factors that could produce different log messages depending on environment and use case). For best results, use the tools described in this section to regularly monitor your devices.

Stand-alone Tools

These stand-alone tools do not require any additional software.

- **Command-line Utilities**: These utilities are installed with the ioMemory VSL software and are run manually in a terminal. The `fio-status` utility provides status for all devices within a host. The other utilities allow you to perform other management functions. See See Command-line Utilities Reference on page 41 for full details.
- **IBM Flash Management Console**: The GUI browser-based IBM Flash Management Console allows you to monitor and manage every IBM io3 Flash Adapter installed in multiple hosts across your network. It collects all of the alerts for all IBM io3 Flash Adapters and displays them in the Alert Tab. You may also set up the IBM Flash Management Console to send email or SMS messages for specific types of alerts or all alerts. The IBM Flash Management Console packages and documentation are available as separate downloads.

Data-source Tools

These data-source tools provide comprehensive data, just like the stand-alone tools, but they do require integration with additional software. At a minimum, some tools can interface with a browser. ***However, the benefit of these tools is that they can be integrated into existing management software that is customized for your organization.***

These tool packages and documentation are also available as separate downloads (separate from the ioMemory VSL software packages).

- **SNMP Subagent**: The IBM SNMP AgentX subagent allows you to monitor and manage your IBM io3 Flash Adapters using the Simple Network Management Protocol. You can use a normal SNMP browser, or customize your existing application to interface with the subagent.
- **SMI-S CIM Provider**: The CIM provider allows you to monitor and manage your devices using the Common Information Model. You can use a normal CIM browser, or customize your existing application to interface with the CIM provider.
- **ioMemory VSL Management SDK**: This C programing API allows you to write customize applications for monitoring and managing IBM io3 Flash Adapters.

Example Conditions to Monitor

This section gives examples of conditions you can monitor. It is intended as an introduction and not as a comprehensive reference. These conditions will have slightly different names, states, and values, depending on the tool you choose. For example, an SNMP MIB may have a different name than a SMI-S object or an API function.

In order to properly monitor these conditions, you should become familiar with the tool you choose to implement and read the documentation for that tool. You may also discover additional conditions that you wish to frequently monitor.

For quick reference, the possible states/values of these conditions are described as Normal (**GREEN**), Caution/Alert (YELLOW), or Error/Warning (**RED**). You may implement your own ranges of acceptable states/values, especially if you use a data-source tool.

Device Status

All of the monitoring tools return information on the status of the IBM io3 Flash Adapters, including the following states:

GREEN	Attached
YELLOW	Detached, Busy (including: Detaching, Attaching, Scanning, Formatting, and Updating)
RED	Minimal Mode, Powerloss Protect Disabled

If the device is in Minimal Mode, the monitoring tool can display the reason for the Minimal Mode status.

Required Actions

If the device is in Minimal Mode, the action will depend on the reason. For example, if the reason is outdated firmware, then you will need to update the firmware.

Temperature

IBM io3 Flash Adapters require adequate cooling. In order to prevent thermal damage, the ioMemory VSL software will start throttling write performance once the on-board controller reaches a specified temperature. If the controller temperature continues to rise, the software will shut down the device once the controller temperature reaches the maximum operating temperature.

These temperatures depend on the device. Newer IBM io3 Flash Adapters have higher thermal tolerances. Consult the *IBM io3 Flash Adapter Hardware Installation Guide* to determine the thermal tolerances of all devices you will monitor. **This table uses the controller thermal tolerances for newer devices** (79°C throttling, 85°C shutdown).

GREEN	<79°C
YELLOW	79-84°C
RED	85°C

You may wish to shift the conditions by a few degrees so the **YELLOW** condition exists before throttling occurs. For example:

GREEN	<75°C
YELLOW	75-83°C
RED	83°C

Attention!

NAND Temperature

IBM io3 Flash Adapters also report the temperature of the NAND. This is also a critical temperature to monitor. Consult the *IBM io3 Flash Adapter Hardware Installation Guide* to see if your device reports this temperature and to see the temperature thresholds.

Required Actions

If the temperature is at or approaching the **YELLOW** condition, thermal mitigation steps may be necessary. Evaluate the server environment and system requirements necessary to operate the IBM io3 Flash Adapters. Server operating conditions are documented in the user guides for the server and the requirement to operate IBM io3 Flash Adapters is at the following website, which may include updates to uEFI and IMM code levels:

http://www.ibm.com/support/entry/portal/docdisplay?lndocid=SERV-IO3

Health Reserves Percentage

IBM io3 Flash Adapters are highly fault-tolerant storage subsystem with many levels of protection against component failure and the loss nature of solid-state storage. As in all storage subsystems, component failures may occur.

By pro-actively monitoring device age and health, you can ensure reliable performance over the intended product life. The following table describes the Health Reserve conditions.

GREEN	>10%
YELLOW	0-10%
RED	0%

At the 10% healthy threshold, a one-time warning is issued. At 0%, the device is considered unhealthy. It enters *write-reduced* mode. After the 0% threshold, the device will soon enter *read-only* mode.

For complete information on Health Reserve conditions and their impact on performance, see See Monitoring the Health of Devices on page 56.

Required Actions

The device needs close monitoring as it approaches 0% reserves and goes into write-reduced mode, which will result in reduced write performance. Prepare to replace the device soon.

Write (Health Reserves) Status

In correlation with the Health Reserves Percentage, the management tools will return write states similar to these:

GREEN	Device is healthy
YELLOW	Device is getting close to entering reduced write mode.
RED	Device has entered reduced-write or read-only mode to preserve the flash from further wearout.

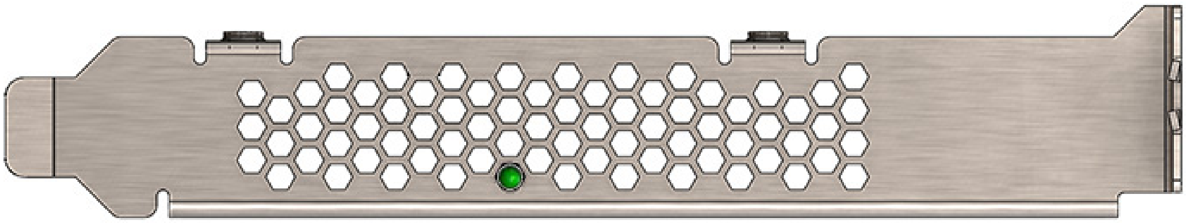
Required Actions

The device needs close monitoring as it approaches 0% reserves and goes into write-reduced mode, which will result in reduced write performance. Prepare to replace the device soon.

Device LED Indicator

If you have physical access to the devices and depending on your device configuration, you can use the LED indicator on the bracket to monitor their status.

If your device has one LED, it should be similar to this configuration:



This table explains the information that the LED conveys:

LED	Indications	Notes
LIT	Power is on and driver is working.	
FLASHING (Fast)	Read and/or write activity.	The faster flashing only indicates activity, it does not reflect the amount of data that is read or written. The flashing may not indicate reads from empty sectors (all zeros).
FLASHING (Slow)	Location beacon.	Use the `fio-beacon` utility to initiate this behavior.
OFF	This indicates one of the following: Power is off, driver is not loaded, or driver is not working.	Check `fio-status` to see if device is attached and there are no errors.

Maintenance

This section explains additional software maintenance functions not covered in the sections See Configuration on page 19 and See Monitoring and Managing Devices on page 29.

NOTE-

All commands require administrator privileges. Log in as "root" or use sudo to run the commands.

Uninstalling the ioMemory VSL Software

Attention!

This version of the ioMemory VSL software only supports third generation devices (such as io3 Enterprise Value Adapters), and it does not support devices that were compatible with ioMemory VSL software version 3.x.x or earlier.

Attention!

You must uninstall all previous versions of the ioMemory VSL software before you install the lastest version.

NOTE-

If you came to this section from the *Installation Overview* section, return to See Installation Overview on page 10 after you uninstall previous versions of the ioMemory VSL software and utilities.

Uninstalling the ioMemory VSL Utilities and Other Support Packages

Uninstalling 2.x Support Packages

To uninstall the support RPM packages, run this command (adding or removing package names as needed):

```
$ rpm -e fio-util fio-snmp-agentx fio-common fio-firmware iomanager-gui
iomanager-jre libfio libfio-doc libfusionjni fio-sysvinit fio-smis fio-
snmp-mib libfio-dev
```

Uninstalling 3.x Support Packages

To uninstall the support RPM packages, run this command (adding or removing package names as needed):

```
$ rpm -e fio-util fio-snmp-agentx fio-common fio-firmware libvsl libvsl-
doc fio-sysvinit fio-smis fio-snmp-mib libvsl-dev
```

Uninstalling the ioMemory VSL RPM Package

With versions 2.x and later (including 3.x releases) of the ioMemory VSL software, you must specify the kernel version of the package you are uninstalling. Run this command to find the installed driver packages:

```
$  rpm -qa | grep -i iomemory
```

Sample output:

```
iomemory-vsl-2.6.18-194.el5-2.2.2.82-1.0
```

Uninstall the ioMemory VSL software by running a command similar to this example (specify the kernel version of the driver you wish to uninstall):

```
$ rpm -e iomemory-vsl-2.6.18-194.el5-2.2.0.82-1.0
```

Unloading the Software Driver

To unload the driver, run this command:

```
$ modprobe -r iomemory-vsl4
```

Upgrading the Kernel

If you ever plan to upgrade the kernel when the ioMemory VSL software is installed, you **must**:

1. Unload the ioMemory VSL driver.
2. Uninstall the ioMemory VSL software.
3. Upgrade the kernel.
4. Install an ioMemory VSL software package that is compiled for the new kernel.

Failure to follow this procedure may result in driver load issues.

Disabling the ioMemory VSL Software

The ioMemory VSL software automatically loads by default when the operating system starts. You can disable the ioMemory VSL software for diagnostic or troubleshooting purposes.

To disable auto-load, uninstall the ioMemory VSL software to keep it from loading, or move it out of the `/lib/modules/<kernel_version>` directory.

Disabling Auto-Attach

When the ioMemory VSL software is installed, it is configured to automatically attach any devices when the ioMemory VSL software is loaded. Sometimes you may want to disable the auto-attach feature (to assist in troubleshooting or diagnostics). To do so:

NOTE-

You can also use the IBM Flash Management Console to enable or disable auto-attach. See the IBM Flash Management Console documentation for more information.

1. Edit the following file:

```
/etc/modprobe.d/iomemory-vsl4.conf
```

2. Uncomment the following line in that file:

```
options iomemory-vsl4 auto_attach=0
```

3. Save the file. To re-enable auto-attach, simply edit the file and remove that line, comment it out, or change it to the following:

```
options iomemory-vsl4 auto_attach=1
```

Unmanaged Shutdown Issues

Unmanaged shutdowns due to power loss or other circumstances can force the IBM io3 Flash Adapter to perform a consistency check during the restart. This may take several minutes or more to complete.

Although data written to the IBM io3 Flash Adapter is not lost due to unmanaged shutdowns, important data structures may not have been properly committed to the device. This consistency check (also called a rescan) repairs these data structures.

Improving Rescan Times

The rescan of the device (also called a consistency check) the ioMemory VSL software performs after an unmanaged shutdown may take an extended period of time depending on the total capacity of the device(s) that the ioMemory VSL software needs to scan.

Default Fast Rescan

By default, all IBM io3 Flash Adapters formatted with the fio-format utility or ioSphere are formatted to have improved rescan times. You can disable this default fast rescan by reformatting the device and using the `-R` option. Disabling this feature will reclaim some reserve capacity that is normally set aside to help improve rescan times.

If you leave the default fast rescan feature in place you can also take further steps to improve rescan times by implementing one of the following module parameters.

Faster Rescans Using Module Parameters

These two module parameters require the default fast rescan formatting structure, and they also use system memory (RAM) to help improve rescan times. The extra memory enables the rescan process to complete faster, which reduces downtime after a hard shutdown. This memory allocation is only temporary and is freed up after the rescan process is complete.

If you decide to use one of these parameters, you will need to set the upper limit of RAM used by that parameter. To do this, you will need to determine how much RAM each parameter may use in your scenario, how much system RAM is available, and (therefore) which parameter is more suited for your use case.

For more information on setting module parameters, see See Using Module Parameters on page 58.

Here is a quick comparison of the two parameters:

- **RMAP Parameter**
  - **Fastest**: This improvement results in the fastest rescan times.
  - **Less Scalable**: (All or nothing.) This parameter requires enough RAM to function. If the RAM limit is set too low, then the ioMemory VSL software will not use RMAP at all, and it will revert back to the default fast rescan process.
  - **Target Scenario**: This parameter will improve any use case if there is enough RAM available for the parameter. It is more suited for smaller capacity IBM io3 Flash Adapters and/or systems with fewer IBM io3 Flash Adapters installed. We also recommend it for devices that have been used for many small random writes.
- **RSORT Parameter**
  - **Faster**: This improves rescan times over the default fast rescan process.
  - **Scalable**: With this parameter, the ioMemory VSL softwareworks with the system RAM to improve rescan times until it reaches the RAM limit set in the parameter. At that point, the software reverts back to the default fast rescan process.
  - **Target Scenario**: This parameter will improve rescan times in any use scenario. It is especially useful in systems with multiple IBM io3 Flash Adapters and/or larger-capacity IBM io3 Flash Adapters. We also recommend it when IBM io3 Flash Adapters are used to store databases.

RMAP Parameter

The `rmap_memory_limit_MiB` parameter sets the upper memory (RAM) limit (in mebibytes) used by the ioMemory VSL software to perform the RMAP rescan process. You should only use this option if you have enough memory for all of your IBM io3 Flash Adapters in the system. If you do not have enough memory to use this option, use the RSORT parameter instead.

Because this parameter requires a set amount of memory, it often works best with fewer IBM io3 Flash Adapters and/or smaller-capacity IBM io3 Flash Adapters in a system, but the determining factor is how much memory is in the system and whether there is enough to set the appropriate memory limit.

This parameter requires `4.008` bytes of RAM per block of IBM io3 Flash Adapter capacity.

1. First determine the number of blocks that are formatted for each device.
   a. This information is visible when you format the device using the `fio-format` utility.
   b. Or you can estimate the number of blocks using the device capacity and the formatted sector

size.

This example shows a quick estimation of the number of blocks on a `1000GB` device with `512B` size sectors (`2` blocks per KB):

```
1000GB * 1000MB/GB * 1000KB/MB * 1000B/KB * 1 Block/512B = 1,
953,125,000 Blocks
```

2. Multiply the number of blocks by `4.008` bytes of RAM per block (and translate that into MiB) to determine the memory limit that is required for this parameter to function.

    a. In the example above there were about `1.95 billion` blocks:

    ```
    1,953,125,000 Blocks * 4.008B/Block * 1KiB/1024B * 1MiB/1024KiB
    = ~74656MiB of RAM
    ```

    b. In this example, you would need about 7465 MiB of RAM available in your system for a 1000GB IBM io3 Flash Adapter formatted for 512B sectors, and you would need to set the RMAP parameter to `7500`.

    > **NOTE-**
    >
    > **Default Value**
    > The RMAP parameter is, by default, set to `3100`. It is set to this low default value so the rescan process does not use all of the RAM in systems that have less available memory.

    - If the RMAP value is too low for the number of IBM io3 Flash Adapter blocks in the system, then the ioMemory VSL software will not use the RMAP process to improve rescan times, it will just use the default fast rescan process. (RMAP is an all-or-nothing setting.)

    - If you don't have enough system memory to use the RMAP parameter, consider using the RSORT parameter. The RSORT parameter will use its RAM limit to improve the rescan process, and then the ioMemory VSL softwarerevert to the default fast rescan process to finish the consistency check.

3. Set the module parameter to the value you have determined. See See Using Module Parameters on page 58 for more information on setting parameters.

RSORT Parameter

The `rsort_memory_limit_MiB` parameter sets the memory (RAM) limit used by the ioMemory VSL software to perform the RSORT rescan process. The RSORT rescan process is faster than the default rescan process and we recommend using it to rescan devices that are used datastores for databases.

If this parameter is given any memory limit, the ioMemory VSL software will use the RSORT process until either the rescan is done or it consumes the memory limit. If the process runs out of memory, it will revert to the default fast rescan process. However, in order to optimize the use of this process, you can calculate the target RAM usage and set the limit based on that target. There is no penalty for setting a high limit, the RSORT process will only use the RAM it needs (up to the limit that is set).

This target is based on `32` bytes per write extent. For example, if your database writes 16KB at a time, there is one write extent per 16KB of IBM io3 Flash Adapter capacity.

NOTE-

Blocks per Write Extent
One measure of the the benefits of the RSORT process is to see how many blocks are written per write extent. The RSORT process improves rescan times over the default fast rescan process on when a device has `8` or more blocks written per extent. For example, if your IBM io3 Flash Adapter is formatted to `512B` sector sizes (`2` blocks per KB), and your database writes in `8KB` chunks, then your database writes `16` blocks per write extent and RSORT would improve the rescan times.

---

1. First determine the number of blocks that are formatted for each device.

   a. This information is visible when you format the device using the `fio-format` utility.

   b. Or you can estimate the number of blocks using the total device capacities and their formatted sector sizes.

      This example shows a quick estimation of the number of blocks on `1000GB` of IBM io3 Flash Adapter capacity with `512B` size sectors (`2` sectors per KB):

```
1000GB * 1000MB/GB * 1000KB/MB * 1000B/KB * 1 Block/512B = 1,
953,125,000 Blocks
```

2. Divide the number of blocks by the write extents per block to determine the total possible number of write extents on the device(s).

   a. In the example above there were `1.95 billion` blocks. We will assume `16KB` write extents (`32 blocks` per write on 512B sectors):

```
1,953,125,000 Blocks * 1 Write Extent/32 Blocks = 61,035,156
Writes
```

3. Multiply the number of writes by 32 bytes of RAM per write (and translate that into MiB) to determine the memory target for this parameter.

   a. In the example above there were `61 million` write extents:

```
61,035,156 Writes * 32B/Write * 1KiB/1024B * 1MiB/1024KiB =
~1863MiB of RAM
```

   b. In this example, you would want to set the RSORT limit to about 2300 MiB of RAM available in your system for 1000GB of IBM io3 Flash Adapter capacity formatted for 512B sectors.

NOTE-

Default Value

The RMAP parameter is, by default, set to `0`m and it has a maximim of 100000 (100GiB).

4. Set the module parameter to the value you have determined. See See Using Module Parameters on page 58 for more information on setting parameters.

Appendix A - Command-line Utilities Reference

The ioMemory VSL software installation packages include various command-line utilities, installed by default in `/usr/bin`. These provide a number of useful ways to access, test, and manipulate your device.

Attention!

There are some additional utilities installed in the `/usr/bin` directory that are not listed below. Those additional utilities are dependencies (used by the main ioMemory VSL utilities), and you should not use them directly unless Customer Support advises you to do so.

Utility	Purpose
`fio-attach`	Makes an IBM io3 Flash Adapter available to the OS.
`fio-beacon`	Lights the IBM io3 Flash Adapter's external LEDs.
`fio-bugreport`	Prepares a detailed report for use in troubleshooting problems.
`fio-detach`	Temporarily removes an IBM io3 Flash Adapter from OS access.
`fio-format`	Used to perform a low-level format of an IBM io3 Flash Adapter.
`fio-pci-check`	Checks for errors on the PCI bus tree, specifically for IBM io3 Flash Adapters.
`fio-status`	Displays information about the device(s).
`fio-sure-erase`	Clears or purges data from the device.
`fio-update-iodrive`	Updates the IBM io3 Flash Adapter's firmware.

NOTE-

There are `-h` (Help) and `-v` (Version) options for all of the utilities. Also, `-h` and `-v` cause the utility to exit after displaying the information.

fio-attach

Description

Attaches the IBM io3 Flash Adapter and makes it available to the operating system. This creates a block device in `/dev` named `fiox` (where `x` is `a`, `b`, `c`, etc.). You can then partition or format the IBM io3 Flash Adapter, or set it up as part of a RAID array. The command displays a progress bar and percentage as it operates.

NOTE-

In most cases, the ioMemory VSL software automatically attaches the device on load and does a scan. You only need to run fio-attach if you ran fio-detach or if you set the ioMemory VSL software's auto_attach parameter to 0.

NOTE-

If the IBM io3 Flash Adapter is in minimal mode, then auto-attach is disabled until the cause of the device being in minimal mode is fixed.

Syntax

```
fio-attach <device> [options]
```

where `<device>` is the name of the device node (`/dev/fctx`), where *x* indicates the device number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first IBM io3 Flash Adapter detected on the system.

You can specify multiple IBM io3 Flash Adapters. For example, `/dev/fct1 /dev/fct2` indicates the second and third IBM io3 Flash Adapters installed on the system.

Option	Description
`-r`	Force a metadata rescan. This may take an extended period of time, and is not normally required. **Attention!** Only use this option when directed by Customer Support.
`-c`	Attach only if clean.
`-q`	Quiet: disables the display of the progress bar and percentage.
`-Q`	Quiet: disables the display of the progress bar only.

fio-beacon

Description

Lights the IBM io3 Flash Adapter's LED to locate the device. You should first detach the IBM io3 Flash Adapter and then run `fio-beacon`.

Syntax

```
fio-beacon <device> [options]
```

where `<device>` is the name of the device node (`/dev/fctx`), where *x* indicates the card number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first IBM io3 Flash Adapter detected on the system. The device numbers are visible using `fio-status`.

Option	Description
`-0`	Off: (Zero) Turns off the beacon.
`-1`	On: Lights the LED beacon (slow flash).
`-p`	Prints the PCI bus ID of the device at `<device>` to standard output. Usage and error information may be written to standard output rather than to standard error.

fio-bugreport

Description

Prepares a detailed report of the device for use in troubleshooting problems. The results are saved in the `/tmp` directory in the file that indicates the date and time the utility was run.

Example:

```
/tmp/fio-bugreport-20100121.173256-sdv9ko.tar.bz2
```

Syntax

```
fio-bugreport
```

Notes

This utility captures the current state of the device. When a performance or stability problem occurs with the device, run the `fio-bugreport` utility and contact Customer Support at http://www.ibm.com/systems/support for assistance in troubleshooting.

Sample Output

```
-bash-3.2# fio-bugreport
BUGREPORT_VERSION 2.0
Running fio-read-lebmap /dev/fct0
Running fio-read-lebmap -m /dev/fct0
Running fio-get-erase-count /dev/fct0
Running fio-get-erase-count -b /dev/fct0
Running fio-kinfo -i /dev/fct0
Running fio-kinfo -i --driver-data /dev/fct0
Running dc-status -v
Running dc-status -b -fj -U
Collecting /proc/buddyinfo
Collecting /proc/cgroups
Collecting /proc/cmdline
...
Collecting /etc/init.d/iomemory-vsl4
Collecting /etc/modprobe.d/iomemory-vsl4.conf
Collecting /etc/sysconfig/iomemory-vsl4
Building tar file...
```

```
Please attach the bugreport tar file
   /tmp/fio-bugreport-20090921.173256-sdv9ko.tar.bz2
 to your support case, including steps to reproduce the problem.
 If you do not have an open support case for this issue, please open a
support
 case with a problem description and then attach this file to your new
case.
```

For example, the filename for a bug report file named `fiobugreport-20090921.173256-sdv9ko.tar.bz2` indicates the following:

- Date: 20090921 (YYYY:MM:DD)
- Time: 173256, or 17:32:56
- Misc. information: `sdv9ko.tar.bz2`

fio-detach

Description

Detaches the IBM io3 Flash Adapter and removes the corresponding `fctx`IBM io3 Flash Adapter block device from the OS. The `fio-detach` utility waits until the device completes all read/write activity before executing the detach operation. By default, the command also displays a progress bar and percentage as it completes the detach.

Attention!

Before using this utility, ensure that the device you want to detach is **NOT** currently mounted and in use.

Syntax

```
fio-detach <device> [options]
```

where `<device>` is the name of the device node (`/dev/fctx`), where *x* indicates the card number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first IBM io3 Flash Adapter detected on the system.

You can specify multiple IBM io3 Flash Adapters. For example, `/dev/fct1 /dev/fct2` indicates the second and third IBM io3 Flash Adapters installed on the system. You can also use a wildcard to indicate all IBM io3 Flash Adapters on the system. For example, `/dev/fct*`

Option	Description
`-q`	Quiet: Disables the display of the progress bar and percentage.
`-Q`	Quiet: disables the display of the progress bar only.

Notes

Attempting to detach an IBM io3 Flash Adapter may fail with an error indicating that the device is busy. This typically may occur if the IBM io3 Flash Adapter is part of a software RAID (0,1,5) volume, is mounted, or some process has the device open.

The tools fuser, mount, and lsof can be helpful in determining what is holding the device open.

fio-firmware

Description

This utility is useful for diagnosing system issues. The utility reports the version and revision of the firmware on all IBM io3 Flash Adapters without requiring you to load the ioMemory VSL software driver. By indicating the firmware version(s) on installed devices, this utility can help you determine which version of the ioMemory VSL software to load.

Output is in CSV format, with each IBM io3 Flash Adapter on a separate line:

```
<PCIe address>, <FW Version>, <FW Revision>
```

Syntax

```
fio-firmware [options]
```

Option	Description
`-f`	Print only x.y.z firmware version.
`-n`	Disable printing header information on columns.
`-p`	Print pending version of firmware to be activated on next reboot
`-r`	Print only firmware revision.

fio-format

Description

NOTE-

IBM io3 Flash Adapters ship pre-formatted, so `fio-format` is generally not required except to change the logical size or block size of a device, or to erase user data on a device. To ensure the user data is truly erased, use `fio-sure-erase`, see See fio-sure-erase on page 50 for more information.

Performs a low-level format of the IBM io3 Flash Adapter. By default, `fio-format` displays a progress-percentage indicator as it runs.

Attention!

Use this utility with care, as it deletes all user information on the device. You will be prompted as to whether you want to proceed with the format.

NOTE-

Using a larger block (sector) size, such as 4096 bytes, can significantly reduce worst-case ioMemory VSL host memory consumption. However, some applications are not compatible with non-512-byte sector sizes.

NOTE-

If you do not include the `-s` or `-o` options, the device size defaults to the advertised capacity. If used, the `-s` and `-o` options must include the size or percentage indicators.

Attention!

We recommend adding power backup to your system to prevent power failures during formatting.

Syntax

```
fio-format [options] <device>
```

where `<device>` is the name of the device node (`/dev/fctx`), where *x* indicates the device number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first IBM io3 Flash Adapter detected on the system. Use `fio-status` to view this number.

Options	Description
`-b <size B|K>`	Set the block (sector) size, in bytes or kibibytes (base 2). Only 512b or 4KiB sector sizes are supported. For example: `-b 512B` or `-b 4K` (`B` in `512B` is optional). io3 Enterprise Value Adapters and io3 Enterprise Adapters ship with 4KiB sectors sizes. If you do not specify a sector size, the utility will format the sectors to the default size of 4KiB sector sizes.
`-f`	Force the format size, bypassing normal checks and warnings. This option may be needed in rare situations when `fio-format` does not proceed properly. (The "Are you sure?" prompt still appears unless you use the `-y` option.)
`-q`	Quiet: Disable the display of the progress and percentage indicators.
`-Q`	Quiet: disables the display of the progress bar only.

-s <size M|G|T|%>	Set the device capacity as a specific size (in TB, GB, or MB) or as a percentage of the advertised capacity: • T Number of terabytes (TB) to format • G Number of gigabytes (GB) to format • M Number of megabytes (MB) to format • % Percentage, such as 70% (the percent sign must be included)
-R	Disable fast rescan on unclean shutdown to reclaim some reserve capacity.
-y	Auto-answer "yes" to all queries from the application (bypass prompts).

You must re-attach the device in order to use the IBM io3 Flash Adapter. See See fio-attach on page 41 for details.

fio-pci-check

Description

Checks for errors on the PCI bus tree, specifically for IBM io3 Flash Adapters. This utility displays the current status of each IBM io3 Flash Adapter. It also prints the standard PCI Express error information and resets the state.

> **NOTE-**
> It is perfectly normal to see a few correctable errors when `fio-pci-check` is initially run. Subsequent runs should reveal only one or two errors during several hours of operation.

Syntax

```
fio-pci-check [options]
```

Option	Description
-d <value>	1 = Disable the link; 0 = bring the link up (Not recommended).
-e	Enable PCI-e error reporting.
-f	Scan every device in the system.
-n	Do not perform any writes to config space. Will prevent errors from being cleared.
-o	Optimize the IBM io3 Flash Adapter PCIe link settings by increasing the maximum read request size if it is too low.
-r	Force the link to retrain.
-v	Verbose: Print extra data about the hardware.

fio-status

Description

Provides detailed information about the installed devices. This utility operates on either `fctx` or `fiox` devices. The utility depends on running as root and having the ioMemory VSL driver loaded. If no driver is loaded, a smaller set of status information is returned.

`fio-status` provides alerts for certain error modes, such as a minimal-mode, read-only mode, and write-reduced mode, describing what is causing the condition.

Syntax

```
fio-status [<device>] [<options>]
```

where `<device>` is the name of the device node (`/dev/fctx`), where *x* indicates the card number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first IBM io3 Flash Adapter detected on the system.

If `<device>` is not specified, `fio-status` displays information for all cards in the system. If the ioMemory VSL driver is not loaded, this parameter is ignored.

Option	Description
`-a`	Report all available information for each device.
`-e`	Show all errors and warnings for each device. This option is for diagnosing issues, and it hides other information such as format sizes.
`-c`	Count: Report only the number of IBM io3 Flash Adapters installed.
`-d`	Show basic information set plus the total amount of data read and written (lifetime data volumes). This option is not necessary when the \-a option is used.
`-fj`	Format JSON: creates the output in JSON format.
`-fx`	Format XML: creates the output in XML format.
`-u`	Show unavailable fields. Only valid with `-fj` or `-fx`.
`-U`	Show unavailable fields and details why. Only valid with `-fj` or `-fx`. **NOTE-** Some `fio-status` fields are unavailable depending on the operating system or device. For example, some legacy fields are unavailable on newer IBM io3 Flash Adapters.
`-F<field>`	Print the value for a single field (see the next option for field names). Requires that a device be specified. Multiple `-F` options may be specified.
`-l`	List the fields that can be individually accessed with `-F`.
`-L`	List all the available ioMemory devices on the system.

Basic Information: If no options are used, `fio-status` reports the following basic information:

- Number and type of devices installed in the system
- ioMemory VSL software version

Adapter information:

- Adapter type
- Product number
- Product UUID
- PCIe power limit threshold (if available)
- Connected IBM io3 Flash Adapters

Block device information:

- Attach status
- Product name
- Product number
- Serial number
- PCIe address and slot
- Firmware version
- Size of the device, out of total capacity
- Internal temperature (average and maximum, since ioMemory VSL software load) in degrees Centigrade
- Health status: healthy, nearing wearout, write-reduced or read-only
- Reserve capacity (percentage)
- Warning capacity threshold (percentage)

Data Volume Information: If the `-d` option is used, the following data volume information is reported *in addition* to the basic information:

- Physical bytes written
- Physical bytes read

All Information: If the `-a` option is used, all information is printed, which includes the following information *in addition* to basic and data volume information:

Adapter information:

- Manufacturer number
- Part number
- Date of manufacture
- Power loss protection status
- PCIe bus voltage (avg, min, max)
- PCIe bus current (avg, max)

- PCIe bus power (avg, max)
- PCIe power limit threshold (watts)
- PCIe slot available power (watts)
- PCIe negotiated link information (lanes and throughput)
- Product UUID

Block device information:

- Manufacturer's code
- Manufacturing date
- Vendor and sub-vendor information
- Format status and sector information (if device is attached)
- Controller ID and Low-level format GUID
- PCIe slot available power
- PCIe negotiated link information
- Card temperature, in degrees Centigrade
- Internal voltage (avg and max)
- Auxiliary voltage (avg and max)
- Percentage of good blocks, data and metadata
- Lifetime data volume statistics
- RAM usage

Error Mode Information: If the ioMemory VSL software is in minimal mode, read-only mode, or write-reduced mode when `fio-status` is run, the following differences occur in the output:

- Attach status is "Status unknown: Driver is in MINIMAL MODE:"
- The reason for the minimal mode state is displayed (such as "Firmware is out of date. Update firmware.")
- "Geometry and capacity information not available." is displayed.
- No media health information is displayed.

fio-sure-erase

Attention!

As a best practice, do not use this utility if there are any IBM io3 Flash Adapters installed in the system that you do not want to clear or purge. First remove any devices that you do not want to accidentally erase. Once the data is removed with this utility it is gone forever. **It is not recoverable**.

Attention!

Before you use this utility, be sure to back up any data that you wish to preserve.

NOTE-

After using fio-sure-erase, format the device using fio-format before using the device again, see See fio-format on page 45.

Attention!

If the device is in Read-only mode, perform a format using fio-format before running fio-sure-erase. If the device is in Minimal mode, then fio-sure-erase cannot erase the device. Updating the firmware may take the device out of Minimal Mode. If the device remains in Minimal mode, contact Customer Support at http://www.ibm.com/systems/support for further assistance.

In order to run fio-sure-erase, the block device **must be detached**. See See fio-detach on page 44 section for more information.

Description

The fio-sure-erase is a command-line utility that securely removes data from IBM io3 Flash Adapters. It complies with the "Clear" and "Purge" level of destruction from the following standards:

1. DOD 5220.22-M - Comply with instructions for Flash EPROM
2. NIST SP800-88- Comply with instructions for Flash EPROM

See below for more information on Clear and Purge support.

Syntax

```
fio-sure-erase [options] <device>
```

Where <device> is the name of the device node (/dev/fctx), where *x* indicates the card number: 0, 1, 2, etc. For example, /dev/fct0 indicates the first IBM io3 Flash Adapter detected on the system. Use fio-status to view this device node, see See fio-status on page 48.

Option	Description
-p	Purge instead of Clear: performs a write followed by an erase. For more information on Purge, see below. **Attention!** Purging the device may take hours to accomplish, depending on the size of the device that needs to be purged.
-y	No confirmation: do not require a yes/no response to execute the utility.
-t	Do not preserve current format parameters, including device and sector size (reset to default).
-q	Quiet: do not display the status bar.

NOTE-

If you run `fio-sure-erase` with no options, a Clear is performed. For more information, see below.

When the utility completes, each block of memory consists of uniform 1 bits or 0 bits.

Clear Support

A "Clear" is the default state of running `fio-sure-erase` (with no options), and refers to the act of performing a full low-level erase (every cell pushed to "1") of the entire NAND media, including retired erase blocks.

Metadata that is required for operation will not be destroyed (media event log, erase counts, physical bytes read/written, performance and thermal history), but any user-specific metadata will be destroyed.

The following describes the steps taken in the Clear operation:

1. Creates a unity map of every addressable block (this allows `fio-sure-erase` to address every block, including previously unmapped bad blocks).
2. For each block, performs an erase cycle (every cell is pushed to "1").
3. Restores the bad block map.
4. Formats the device (the purpose of this is to make the device usable again, the utility erases all of the headers during the clear).

Purge Support

A "Purge" is implemented by using the `-p` option with `fio-sure-erase`. Purge refers to the act of first overwriting the entire NAND media (including retired erase blocks) with a single character (every cell written to logical "0"), and then performing a full chip erase (every cell pushed to "1") across all media (including retired erase blocks).

Metadata that is required for operation will **not** be destroyed (media event log, erase counts, physical bytes read/written, performance and thermal history), but any user-specific metadata will be destroyed.

The following describes the steps taken in the Purge operation:

1. Creates a unity map of every addressable block (this allows `fio-sure-erase` to address every block, including previously unmapped bad blocks).
2. For each block, performs a write cycle (every cell written to "0").
3. For each block, performs an erase cycle (every cell pushed to "1").
4. Restores the bad block map.
5. Formats the drive (the purpose of this is to make the drive usable again, the utility erases all of the headers during the clear).

fio-update-iodrive

Attention!

You should back up the data on the IBM io3 Flash Adapter prior to any upgrade as a precaution.

Description

Updates the IBM io3 Flash Adapter's firmware. This utility scans the PCIe bus for all IBM io3 Flash Adapters and updates them. A progress bar and percentage are shown for each device as the update completes.

Attention!

It is extremely important that the power not be turned off during a firmware upgrade, as this could cause device failure. If a UPS is not already in place, consider adding one to the system prior to performing a firmware upgrade.

Attention!

Note that when running multiple firmware upgrades in sequence, it is critical to load the ioMemory VSL driver after each firmware upgrade step. Otherwise the on-device format will not be changed, and there will be data loss. For more information on staged upgrades, see the *IBM ioMemory VSL Release Notes*.

Attention!

Do not use this utility to downgrade the IBM io3 Flash Adapter to an earlier version of the firmware. Doing so may result in data loss and void your warranty. Contact IBM System x Customer Support at http://www.ibm.com/systems/support if you have issues with your upgrade.

NOTE-

The default action (without using the `-d` or `-s` option) is to upgrade all IBM io3 Flash Adapters with the firmware contained in the `fio-firmware-highiops-<version>.<date>.fff` firmware archive file. Confirm that all devices need the upgrade prior to running the update. If in doubt, use the `-p` (Pretend) option to view the possible results of the update.

Attention!

Upgrade Path

There is a specific upgrade path that you must take when upgrading IBM io3 Flash Adapter. Consult the *IBM ioMemory VSL Release Notes* for this ioMemory VSL software release before upgrading IBM io3 Flash Adapters.

NOTE-

If you receive an error message when updating the firmware that instructs you to update the midprom information, contact Customer Support.

To update one or more specific devices:

- If the ioMemory VSL driver is loaded, use the `-d` option with the device number.

Online Firmware Updates

This utility will upgrade compatible IBM io3 Flash Adapters while attached. The utility will check to make sure the device(s) support live firmware upgrades (while attached).

NOTE-

The firmware update will not take effect until you restart the system. If you have updated the firmware, but not restarted, the fio-status utility will report the following:

```
...
Firmware vX.X.X, rev 115781 Public
       Unactivated Firmware vX.X.Y, rev XXXXXX -- Reboot required to
activate     <-------
```

If a device is not compatible, the `fio-update-iodrive` utility will return an error that the device is not supported. In this case, you will need to first detach the device and then run this utility again. In either case, you must restart the system before the updated firmware will take effect.

Syntax

```
fio-update-iodrive [options] <firmware-path>
```

where `<firmware-path>` is the full path to the firmware archive file `fio-firmware-highiops-<version>.<date>.fff` available at http://www.ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-65723 (follow that link and then select **IBM High IOPS and io3 software matrix**). If you downloaded the `.fff` firmware archive file, then the firmware is most likely with the other downloaded packages. If you installed the firmware from the firmware package, the default path is `/usr/share/fio/firmware/`. This parameter is required.

Option	Description
`-d`	Updates the specified devices (by `fctx`, where *x* is the number of the device shown in `fio-status`). If this option is not specified, all devices are updated. **Attention!** Use the `-d` or `-s` option with care, as updating the wrong IBM io3 Flash Adapter could damage your device.
`-f`	Force upgrade (used only when directed by Customer Support). If the ioMemory VSL driver is not loaded, this option also requires the `-s` option. **Attention!** Use the `-f` option with care, as it could damage your card.
`-l`	List the firmware available in the archive.

Option	Description
`-p`	Pretend: Shows what updates would be done. However, the actual firmware is not modified.
`-c`	Clears locks placed on a device.
`-q`	Quiet: Runs the update process without displaying the progress bar or percentage.
`-Q`	Quiet: Only disable the progress bar.
`-y`	Confirm all warning messages.
`-s`	Updates the devices in the specified slots using '*' as a wildcard for devices. The slots are identified in the following PCIe format (as shown in lspci): `[[[[<domain>]:]<bus>]:][<slot>][.[<func>]]`

If you arrived at this section from See Upgrading the Firmware on page 17, you should return to that section.

Appendix B - Monitoring the Health of Devices

This section describes how the health of IBM io3 Flash Adapters can be measured and monitored in order to safeguard data and prolong device lifetime.

Health Metrics

The ioMemory VSL software manages block retirement using pre-determined retirement thresholds. The IBM Flash Management Console and the `fio-status` utilities show a health indicator that starts at 100 and counts down to 0. As certain thresholds are crossed, various actions are taken.

At the 10% healthy threshold, a one-time warning is issued. See See Health Monitoring Techniques on page 56 for methods for capturing this alarm event.

At 0%, the device is considered unhealthy. It enters *write-reduced* mode, which somewhat prolongs its lifespan so data can be safely migrated off. In this state the IBM io3 Flash Adapter behaves normally, except for the reduced write performance.

After the 0% threshold, the device will soon enter *read-only* mode -- any attempt to write to the IBM io3 Flash Adapter causes an error. Some filesystems may require special mount options in order to mount a read-only block device in addition to specifying that the mount should be read-only.

For example, under Linux, `ext3` requires that "`-o ro,noload`" is used. The "`noload`" option tells the filesystem to not try and replay the journal.

Read-only mode should be considered a final opportunity to migrate data off the device, as device failure is more likely with continued use.

The IBM io3 Flash Adapter may enter failure mode. In this case, the device is offline and inaccessible. This can be caused by an internal catastrophic failure, improper firmware upgrade procedures, or device wear out.

NOTE-

For service or warranty-related questions, contact IBM System x Customer Support.

NOTE-

For products with multiple IBM io3 Flash Adapters, these modes are maintained independently for each device.

Health Monitoring Techniques

`fio-status -a`: Output from the `fio-status` utility (using the `-a` option) shows the health percentage and device state. These items are referenced as "`Reserve space status`" in the sample output below.

```
Found 1 ioMemory device in this system
Driver version: 4.x.x build xxxx
...
Reserve space status: Healthy; Reserves: 100.00%, warn at 10.00%
Lifetime data volumes:
   Physical bytes written: 6,423,563,326,064
   Physical bytes read   : 5,509,006,756,312
```

The following Health Status messages are produced by the `fio-status` utility:

- Healthy
- Read-only
- Reduced-write
- Unknown

IBM Flash Management Console: You may also find information about device health within the IBM Flash Management Console. See the IBM Flash Management Console documentation for more information.

Software RAID and Health Monitoring

Software RAID stacks are typically designed to detect and mitigate the failure modes of traditional storage media. The IBM io3 Flash Adapter attempts to fail as gracefully as possible, and these new failure mechanisms are compatible with existing software RAID stacks. An IBM io3 Flash Adapter in a RAID group will fail to receive data at a sufficient rate if: a) the device is in a write-reduced state, and b) it is participating in a write-heavy workload. In this case, the device will be evicted from the RAID group. A device in read-only mode will be evicted when write I/Os are returned from the device as failed. Catastrophic failures are detected and handled just as though they are on traditional storage devices.

Appendix C - Using Module Parameters

The following table describes the module parameters you can set by editing the `/etc/modprobe.d/iomemory-vsl4.conf` file and changing their values.

Each module parameter in the configuration file must be preceded by `options iomemory-vsl4`. The `/etc/modprobe.d/iomemory-vsl4.conf` file has some example parameters that are commented out. You may use these examples as templates and/or uncomment them in order to use them.

NOTE-

These changes must be completed before the ioMemory VSL software is loaded in order to take effect.

Module Parameter	Default (min/max)	Description
`auto_attach`	`1` (`0`, `1`)	`1` (default) = Always attach the device(s) on driver load. `0` = Don't attach the device(s) on driver load.
`fio_dev_wait_timeout_secs`	`30`	Number of seconds to wait for `/dev/fio*` files to show up during driver load. For systems not using `udev`, this should be set to `0` to disable the timeout and avoid an unneeded pause during driver load.
`force_minimal_mode`	`0`	`1` = Force minimal mode on the device. `0` = Do not force minimal mode on the device.
`numa_node_forced_local`	`0`	`1` = Enable, `0` = Disable. See See NUMA Configuration on page 60 for more information.

Module Parameter	Default (min/max)	Description
numa_node_override	Nothing Selected	A list of <affinity specification> couplets that specify the affinity settings of all devices in the system. Each item in the couplet is separated by a colon, and each couplet set is separated by a comma. Where each <affinity specification> couplet has the following syntax: `<device-id>=<node-number>` See See NUMA Configuration on page 60 for more information on using this parameter.
preallocate_memory	No devices selected	For the selected devices, pre-allocate all memory necessary to have the drive usable as swap space. Where the <value> for this parameter is a comma-separated list of device serial numbers.
rmap_memory_limit_MiB	3100 (0, 100000)	Amount of system memory (RAM) in MiB allocated for improving rescan times after an unclean shutdown. For more information, see See Improving Rescan Times on page 36.
rsort_memory_limit_MiB	0 (0, 100000)	Amount of system memory (RAM) in MiB allocated for improving rescan times after an unclean shutdown. For more information, see See Improving Rescan Times on page 36.
use_workqueue	0 (0 or 3)	Linux only: 3 = use standard OS I/O elevators; 0 = bypass.

NOTE-

Other than and preallocate_memory, module parameters are global — they apply to all IBM io3 Flash Adapters in the computer.

Appendix D - NUMA Configuration

Attention!

Deprecated Parameters
The NUMA node parameters are deprecated. The ioMemory VSL software now automatically creates interrupt affinity between IBM io3 Flash Adapters and NUMA nodes. However, you may still use this appendix and these parameters to manually configure NUMA node affinity.

About NUMA Architecture

Servers with a NUMA (Non-Uniform Memory Access) architecture may require special installation instructions in order to maximize IBM io3 Flash Adapter performance. This includes most multi-socket servers.

On some servers with NUMA architecture, during system boot the BIOS will not associate PCIe slots with the correct NUMA node. Incorrect mappings result in inefficient I/O handling that can significantly degrade performance. Here are two methods for correcting the NUMA affinity:

- **`numa_node_forced_local` Parameter**: Forces I/O completions to happen on the same CPU that is running VSL processes for a particular device. This parameter is simple to implement (enabled or disabled), and is persistent.
- **`numa_node_override` Parameter**: Used to manually assign devices to specific NUMA nodes. This parameter is more complex. For optimal implementation, you should understand the NUMA architecture of the system in order to assign devices to the nodes that are closely linked to the device's PCIe slot. This parameter is persistent until any new I/O devices are installed in the system (thus changing the PCIe bus numbers in the system.

Using the numa_node_forced_local Parameter

The `numa_node_forced_local` parameter is either enabled or disabled, and therefore does not offer as much user control as other options. It is persistent and it is enabled by modifying the `/etc/modprobe.d/iomemory-vsl4.conf` file and editing or adding the following line:

```
numa_node_forced_local=1
```

This parameter forces an I/O completion for a particular device to happen on a CPU within the local `numa_node` that the other ioMemory VSL processes are running on (rather than trying to complete an I/O on the CPU that the host issued it on). Because the I/O completions are grouped with other ioMemory VSL processes, they are less likely to compete with processes from other device drivers.

This parameter may or may not improve performance depending on your configuration and workloads. You should test this parameter with your use case to determine if it improves performance.

Using the numa_node_override Parameter

Use this parameter to map devices with specific NUMA nodes.

Attention!

The example below shows the final implementation of custom affinity settings. This implementation required an analysis of the specific system, including the system architecture, type and number of IBM io3 Flash Adapters installed, and the particular PCIe slots that were used. Your particular circumstances will require a custom analysis of your set-up. This analysis requires understanding of your system's NUMA architecture compared to your particular installation.

Your actual settings may be different than the example below, depending on your server configuration. In order to create the correct settings for your specific system, use `fio-status` to list all of the devices and determine the `<device-id>` (see below). Then use the example below of setting the `numa_node_override` parameter as a template and modify it for your particular system.

Determining the Device ID

You should present each `<device-id>` in the following format:

```
<domain>:<bus>:<device>.<function>
```

Typically the domain is `0000`, consult your server documentation to determine your device ID. The remainder of the device ID string is visible in `fio-status` output. For example:

```
# fio-status
Found 2 ioMemory devices in this system
...
       PCI:04:00.0
...
       PCI:15:00.0
```

In the example above the device IDs would be `0000:04:00.0` and `0000:15:00.0` on a system that had a domain of `0000`.

Attention!

Note that the PCI device ID, including the bus number, may change if you change any of the PCI devices in the system. For example, if you add a network card or another IBM io3 Flash Adapter. If the device ID changes, you will have to update the configuration.

numa_node_override Parameter

Configuring your IBM io3 Flash Adapters for servers with NUMA architecture requires the use of the `numa_node_override` parameter by modifying the `iomemory-vsl4.conf` file.

The `numa_node_override` parameter is a list of `<affinity specification>` couplets that specify the affinity settings of all devices in the system. Each item in the couplet is separated by an equal sign (=), and each couplet set is separated by a comma.

Syntax:

```
numa_node_override=<affinity specification>[,<affinity specification>...]
```

Where each `<affinity specification>` has the following syntax:

```
<device-id>=<node-number>
```

Simple Example:

```
numa_node_override=0000:04:00.0=1,0000:1d:00.0=0,0000:05:00.0=2,
0000:1e:00.0=3
```

Has the effect of creating :

<device-id>	Node/Group	Processor Affinity
0000:04:00.0	node 1	all processors in node 1
0000:1d:00.0	node 0	all processors in node 0
0000:05:00.0	node 2	all processors in node 2
0000:1e:00.0	node 3	all processors in node 3

Advanced Configuration

If your server has multiple NUMA nodes and multiple IBM io3 Flash Adapters installed, you will need to make sure that the IBM io3 Flash Adapters are spread out among the various nodes.

While it may be optimal to pair devices to nodes that are electronically closer each device's PCIe slot (which would require an advanced understanding of your server's NUMA architechure and an analysis of the device installation), just simply spreading out all of the devices' node affinity among the available nodes should result in improved performance.

In the example above the device IDs would be `0000:04:00.0` and `0000:15:00.0` on a system that had a domain of `0000`.

Attention!

Note that the PCI device ID may change if you change any of the PCI devices in the system. For example, if you add a network card or another IBM io3 Flash Adapter. If the device ID changes, you will have to update the configuration.

numa_node_override Parameter

Configuring your IBM io3 Flash Adapters for servers with NUMA architecture requires the use of the `numa_node_override` parameter by modifying the `iomemory-vsl.conf` file.

The `numa_node_override` parameter is a list of `<affinity specification>` couplets that specify the affinity settings of all devices in the system. Each item in the couplet is separated by an equal sign (=), and each couplet set is separated by a comma.

Syntax:

```
numa_node_override=<affinity specification>[,<affinity specification>...]
```

Where each `<affinity specification>` has the following syntax:

```
<device-id>=<node-number>
```

Simple Example:

```
numa_node_override=0000:04:00.0=1,0000:1d:00.0=0,0000:05:00.0=2,
0000:1e:00.0=3
```

Has the effect of creating :

<device-id>	Node/Group	Processor Affinity
0000:04:00.0	node 1	all processors in node 1
0000:1d:00.0	node 0	all processors in node 0
0000:05:00.0	node 2	all processors in node 2
0000:1e:00.0	node 3	all processors in node 3

Advanced Configuration

If your server has multiple NUMA nodes and multiple IBM io3 Flash Adapters installed, you will need to make sure that the IBM io3 Flash Adapters are spread out among the various nodes.

While it may be optimal to pair devices to nodes that are electronically closer each device's PCIe slot (which would require an advanced understanding of your server's NUMA architechure and an analysis of the device installation), just simply spreading out all of the devices' node affinity among the available nodes should result in improved performance.

IBM Support

IBM io3 Flash Adapter software and documentation are available on the web at the following address:

http://www.ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-65723 (follow that link and then select **IBM High IOPS and io3 software matrix**).