Appendix B- Manual Installation ................................................................. 32
  Manual Installation on Windows Server 2003 ........................................... 32
  Manual Install on Windows Server 2008 .................................................. 41
Appendix C- Command-line Utilities ......................................................... 44
  fio-attach .......................................................... 45
  fio-beacon .......................................................... 45
  fio-bugreport ....................................................... 46
  fio-config .......................................................... 47
  fio-detach .......................................................... 49
  fio-format ........................................................... 50
  fio-pci-check ...................................................... 52
  fio-status .......................................................... 53
  fio-sure-erase ..................................................... 56
  fio-trim-config .................................................... 58
  fio-update-iodrive .................................................. 59
Appendix D- TRIM Support ................................................................. 61
  Platforms ................................................................ 61
  Using the TRIM Service .................................................. 61
  Configurations ........................................................ 63
Appendix E- Monitoring the Health of ioMemory Devices ............................. 64
  NAND Flash and Component Failure ..................................................... 64
  Health Metrics ........................................................ 64
  Health Monitoring Techniques ......................................................... 65
  Software RAID and Health Monitoring .................................................. 65
Appendix F- Using Windows Page Files with the ioMemory Devices ............ 67
  Configuring Device Paging Support ...................................................... 67
  Windows Page File Management ...................................................... 69
  Performance .......................................................... 73
Appendix G- SNMP Test Mode and MIB Support ........................................ 74
  Using Test-Mode Registry Values ......................................................... 74
  SNMP MIB Support ...................................................... 77
Appendix H- SMI-S Interface ............................................................. 79
  Installing the SMI-S WMI Provider on Windows ...................................... 80
  Verifying SMI-S Installation on Windows .............................................. 80
  Description ................................................................ 85
  Implementation .......................................................... 87
Legal Notices

The information contained in this document is subject to change without notice.

Copyright © 2012 Fusion-io, Inc. All rights reserved.

Fusion-io, the Fusion-io logo, ioMemory, Virtual Storage Layer, and ioDrive are registered trademarks, and VSL, ioFX, Flashback, and Flashback Protection are trademarks of Fusion-io, Inc. in the United States and other countries.

The names of other organizations and products referenced herein are the trademarks or service marks (as applicable) of their respective owners. Unless otherwise stated herein, no association with any other organization or product referenced herein is intended or should be inferred.

You may not use the ioMemory VSL software for any competitive benchmarking purpose without prior written consent from Fusion-io. You may not publish any performance data related to the ioMemory VSL software without prior written consent from Fusion-io.

Fusion-io
2855 E. Cottonwood Parkway, Box 100
Salt Lake City, UT 84121
USA

(801) 424-5500

Part Number: D0001658-016_2
Published: June 21, 2012
Introduction

Overview

Congratulations on your purchase of a Fusion-io® solid-state storage device. This guide explains how to install, troubleshoot, and maintain the software for your ioMemory devices.

Throughout this manual, when you see a reference to an ioMemory device, you may substitute your particular device(s), such as an ioDrive2 device, ioFX device, or each of the two ioMemory devices of an ioDrive Duo device.

Products with Multiple Devices
Some products, such as an ioDrive Duo device, are actually comprised of multiple ioMemory devices. If your product consists of multiple ioMemory devices, you will manage each ioMemory device as an independent device.

For example, if you have an ioDrive Duo device, you can independently attach, detach, and/or format each of the two ioMemory devices. Each of the two devices will be presented as an individual device to your system.

About ioMemory Devices

Designed around a revolutionary silicon-based storage architecture, ioMemory devices are the world’s most advanced NAND flash storage devices, with performance comparable to DRAM and storage capacity on par with today’s hard disks – giving you the power to improve storage performance by orders of magnitude. ioMemory devices allow every computer to exceed the I/O performance of an enterprise SAN.

ioMemory devices are data accelerators designed specifically to improve the bandwidth for I/O-bound applications. They are no-compromise solutions for the toughest computational challenges faced by data centers today, putting them in a league of their own.

About the ioMemory Virtual Storage Layer (VSL)

More than just a hardware driver, the ioMemory® Virtual Storage Layer® (VSL™) is the "secret sauce" that gives ioMemory devices their amazing performance. The VSL™ is a hybrid of the RAM virtualization subsystem and the disk I/O subsystem, combining the best of both worlds. It appears like a disk to interface well with block-based applications and software. At the same time, it runs like RAM underneath to maximize performance. This provides the following game-changing benefits:
• Performance: The VSL offers direct and parallel access to multiple CPU cores, enabling near-linear performance scaling, consistent performance across different read/write workloads, and low latency with minimal interrupts and context switching.

• Extensibility: The VSL enables flash-optimized software development, making each ioMemory module a flexible building block for building a flash-optimized data center.

Flash: Not Just Another Disk Drive

Whether on a PCIe card or in a drive bay, other SSD providers besides Fusion-io simply treat flash as another disk drive, putting it behind RAID controllers. This approach has the following limitations:

• Reduced performance and reliability
• Increased latency and complexity
• Limited potential for software development and optimization around the flash storage medium

With the ioMemory VSL, ioMemory devices avoid this limiting approach and provide a wealth of performance and optimization possibilities.

About Flashback Protection Technology

Like many other memory devices, NAND flash eventually fails with use. Those failures can be either permanent or temporary. Fusion’s Flashback™ redundancy is designed to address those ioMemory chips that experience permanent failures, and provides additional protection above and beyond ECC (Error Correction Code) for soft failures.

Flashback technology provides a real-time RAID-like redundancy at the chip-level, without sacrificing user capacity or performance for fault tolerance. Solutions that use physical RAID schemes for redundancy/ protección must in general either sacrifice capacity (RAID 1), or performance (RAID 5).

Fusion’s Flashback Protection™ technology, with self-healing properties, ensures higher performance, minimal failure, and longer endurance than all other flash solutions.
Software Installation

Before continuing with the installation of this software, please read the following:

1. Ensure that your operating system is included in the list of **supported operating systems** contained in the *ioMemory VSL Release Notes* for this release.

2. Before installing the ioMemory VSL, make sure you have properly installed the ioMemory device(s). Refer to the *ioMemory Hardware Installation Guide* for full details and hardware requirements.

```
- Every ioDrive device in a system running ioMemory VSL 3.1.x or later must be upgraded to the latest version of the firmware.

  For example, if you have a system running ioMemory VSL 2.3.1 with ioDrive devices previously installed, and you want to install new ioDrive2 devices (that require the latest version of the firmware), then you will need to upgrade all of the existing devices to the latest firmware version.
```

**Upgrade Previous Devices First**

If you have ioDrive devices configured for ioMemory VSL 2.x or earlier, you must upgrade their firmware before installing new devices in the system. See Appendix J- Upgrading Devices from VSL 2.x to 3.x for the upgrade instructions.

## Installation Overview

1. If you are installing this version of ioMemory VSL on a system with ioDrive devices configured for VSL 2.x, you must carefully follow the instructions in the Appendix J- Upgrading Devices from VSL 2.x to 3.x section.

   If you do not need to upgrade devices to the firmware for VSL 3.x.x, but your system does have previous versions of the ioMemory VSL installed, you will need to uninstall the ioMemory VSL software. See the Existing ioMemory VSL Installation section for instructions. Once you have uninstalled the software, continue with the instructions on that page.

2. **Install** the latest version of the ioMemory VSL.

   For information on capturing an installation log for troubleshooting purposes, see the following Microsoft KB article.
3. **Upgrade the Firmware** to the latest version, if needed (recommended). This applies to ioDrive2 devices that may use a version of the firmware that is earlier than the latest version.

4. Configure the device(s) by [Adding a File System](#), [Creating a RAID Configuration](#), etc.

### New ioMemory VSL Installation

To install the ioMemory VSL software on a new system:

1. Complete all the installation steps given in the *Hardware Installation Guide*.
2. Log into your computer with an account that has Administrative rights.
3. Download the Windows ioMemory VSL installation program from [http://ts.fujitsu.com/support/](http://ts.fujitsu.com/support/) and save it to your desktop or another convenient directory.
4. Run the ioMemory VSL installation program. The installation program presents a custom setup tree-view with options for installation.
5. Select a type of install by selecting components from the drop-down menus. If you change your mind later, you can use the Repair option in Programs and Features, or Add or Remove Programs in the Control Panel.

   - Click on each component to view its description. The descriptions will appear to the right of the install tree.

   1. Click **Next**.

6. Click **Next**.

7. To select a different folder for the installation, browse to the folder and click **OK**. The default folder is `C:\Program Files\Fusion-io ioMemory VSL`.

8. Follow the onscreen prompts to complete the install.

9. Choose **Finish** on the finish screen of the installer.

   - You should be prompted to reboot your system to complete the installation process. If you are not prompted to reboot, you should still reboot your system after completing the installation. If Windows does not recognize the ioMemory device(s) after rebooting, you may need to manually install the ioMemory VSL for the device(s). See [Appendix B- Manual Installation](#) for information on manual installation.

The installation program:

- Creates a folder for the software components (the default path is `C:\Program Files\Fusion-io ioMemory VSL`).
- Installs and loads the ioMemory VSL. (This may require a restart.)
- Installs support for SNMP (if installed, and if Microsoft SNMP is installed and the SNMP service is running).
- Creates a folder for the VSL utilities. The default path is `C:\Program Files\Common Files\VSL Utilites`.
When the install program creates the ioMemory VSL folder on the drive, it also creates these sub-folders:

- `<VSL-version>\Driver` —for manual installations using Device Manager
- Firmware —contains the latest ioMemory device firmware
- SNMP —contains the SNMP components
- SMIS and SDK —if selected for installation

ioSphere is a free browser-based software for managing ioMemory devices. It is also available from http://ts.fujitsu.com/support/, but is in a different location than the ioMemory VSL packages and documentation.

Proceed to the Outdated Firmware Check to continue.

Existing ioMemory VSL Installation

⚠️ Do not install new ioDrive2 devices with previously installed ioDrive devices without first completing the instructions in Appendix J- Upgrading Devices from VSL 2.x to 3.x.

To install the latest ioMemory VSL Windows software on an existing installation,

1. Review the Release Notes file available for this version of the software for additional steps that may be needed to complete the install.
2. Log in as Administrator or have Administrator rights.
3. Uninstall the existing VSL, utilities, etc., using Programs and Features, or Add or Remove Programs (depending on your version of Windows), in the Control Panel.
4. Restart the computer.

The ioMemory VSL installation program will attempt to remove previous versions of the software, however if it fails and a previous version is removed by the user after the newest version is installed, the ioMemory VSL will no longer load after a restart. In that case, you need to a) run the Repair option in the installation program, from Programs and Features (or Add or Remove Programs) in the Control Panel, and b) restart the computer.

5. Download the VSL installation program for Windows from http://ts.fujitsu.com/support/ to your desktop or a convenient directory.
6. Run the ioMemory VSL installation program. The installation program presents a custom setup tree-view with options for installation.
7. Select a type of install by selecting components from the drop-down menus. If you change your mind later, you can use the Repair option in Programs and Features, or Add or Remove Programs in the Control Panel.

Click on each component to view its description. The descriptions will appear to the right of the install tree.

8. Click Next.

9. To select a different folder for the installation, browse to the folder and click OK. The default folder is C:\Program Files\Fusion-io ioMemory VSL.
   - The installer also creates a folder for the VSL utilities. The default path is C:\Program Files\Common Files\VSL Utilis.

10. Follow the onscreen prompts to complete the install.

11. Choose Finish on the finish screen of the installer.

You should be prompted to reboot your system to complete the installation process. If you are not prompted to reboot, you should still reboot your system after completing the installation. If Windows does not recognize the ioMemory device(s) after rebooting, you may need to manually install the ioMemory VSL for the device(s). See Appendix B - Manual Installation for information on manual installation.

ioSphere is a free browser-based software for managing ioMemory devices. It is also available from http://ts.fujitsu.com/support/, but is in a different location than the ioMemory VSL packages and documentation.

Once the system restarts, proceed to the Outdated Firmware Check section to continue.

Silent Install Option

Uninstall Previous
If the you have a version of the ioMemory VSL previously installed, you must uninstall it first (see the information on a Silent Uninstall below). You can manually reboot the computer after installing the new version with the silent install option. This step must be performed prior using any ioMemory VSL utilities or functionality.

If you are installing remotely or with scripts, you can use the silent install option (/quiet) when you run the installation program in the command-line interface.

In the command-line interface, navigate to the folder that contains the .exe installer file, and run this command:

```
<installname>.exe /quiet
```

Where the <installname>.exe is the name of the installer file.
This option installs the ioMemory VSL using its default settings, eliminating the need to "click Next" or select settings during install.

**Installing Components**

The default command-line installation installs the ioMemory VSL (including the command-line utilities) and the firmware file. You can add additional components using the following command:

```
<installname>.exe /quiet ADDLOCAL=SNMP,SDK,SMIS
```

Remove any component (and leading/trailing comma) that you do not wish to install.

**Silent Uninstall**

You may silently uninstall the ioMemory VSL with this command:

```
<installname>.exe /uninstall /quiet
```

**Outdated Firmware Check**

**Check Using the Command-line Interface**

More information on these command-line utilities is available in Appendix C- Command-line Utilities.

1. Run `fio-status` and examine the output.
   - If any device is in minimal mode, then the firmware is outdated.
   - If the firmware listed for any device is a lower number than the latest firmware version as noted in the Release Notes, then the firmware is old, but not outdated.

2. If the firmware is old or outdated, update it using the `fio-update-iodrive` utility.

**Check using the Optional GUI Interface**

You can use the (optional) ioSphere GUI program to check for outdated firmware.

To check for outdated or old firmware:

1. Launch the ioSphere and look for any devices that have a warning symbol.

2. Click on any devices with a warning symbol to ensure that the alert is from outdated firmware.

3. Select all devices requiring firmware update and use the ioSphere to update the firmware. Refer to the ioSphere User Guide for details.
To check for old but not outdated firmware:

1. Find the name of the latest firmware version as noted in the Release Notes.
2. Use the ioSphere to check each ioMemory device's firmware version against the latest.
3. Refer to the ioSphere User Guide for instructions on how to update the firmware.

**Enabling PCIe Power**

If you have installed any dual ioMemory devices, such as the ioDrive2 Duo device, then the device may require additional power to properly function (beyond the minimum 25W provided by PCIe Gen2 slots).

Additional power may be provided through a power cable (see the *ioMemory Hardware Installation Guide*) or through the PCIe slot(s). For instructions on allowing the device to draw additional power from the PCIe slot(s), see [Enabling PCIe Power Override](#) in the Maintenance section.
Device Naming

The ioMemory device receives a name and number as part of the install process for identification. The syntax is fctx where x is the number of the PCIe bus where you installed the ioDrive. Use ioManager to view this bus number, or follow these steps:

1. Choose Start > Control Panel > System > Hardware > Device Manager.
2. Select Fusion ioMemory VSL devices.
3. Click on your ioMemory device in the list. The Properties dialog box appears.

![ioMemory VSL Device Properties](image)

The Location field shows the PCIe bus number for your device (fct9 in this case).

The system manufacturer assigns bus numbers, which can range from 0 on up. These numbers may or may not reflect the physical location of the bus. (For example, the second slot from the edge of the motherboard may be Bus 2, but it could also be Bus 16 or another arbitrary number. Checking Device Manager is one way to confirm the specific bus number for your installation. You can also use ioManager to view this number as well.)
Adding a File System

With ioMemory device(s) and ioMemory VSL installed, you can now use the Windows Disk Management utility to make your device available to applications. Typically, Windows detects the new device, initializes it, and displays it in Disk Management. You can then add partitions, format a volume, or create a RAID configuration on your ioMemory device using the standard Windows procedures (see the Windows Disk Management Utility documentation for more details).

If Windows does not initialize the device, you can do so manually. To initialize an ioMemory device,

1. Select **Start > Control Panel.**
2. Click **Administrative Tools.**
3. Click **Computer Management.**
4. Click **Disk Management** in the Storage section of the console tree.
5. Locate and right-click the ioMemory device in the list of storage devices on the right. (If the ioMemory device does not appear in the list, choose **Rescan Disks** from the Action menu. You may also need to restart your computer to display the ioMemory device in the list.)
6. Click **Initialize Disk.**

You can now use the Disk Management Utility to add a file system to your ioMemory device.

Creating a RAID Configuration

You can use your ioMemory device as part of a RAID configuration with one or more additional ioMemory devices. To do so, you must format your ioMemory devices as dynamic volumes. In turn, you can then use these dynamic volumes to create multi-disk RAID configurations (spanned, striped, mirrored, or RAID 5).

For specific steps to perform a RAID configuration, see the Windows Disk Management Utility documentation for details.

If you are using RAID1/Mirroring and one device fails, be sure to run a fio-format on the replacement device (not the remaining good device) before rebuilding the RAID.

Using the Device as Swap

To safely use the ioMemory device as swap space, you need to use the **fio-config** utility to pass a special pre-allocation parameter.
For example:

```
fio-config -p FIO_PREALLOCATE_MEMORY 1072,4997,6710,10345
```

- Where 1072,4997,6710,10345 are device serial numbers obtained from `fio-status` (do not use adapter serial numbers).

A 4K sector size format is required for swap—this reduces the software memory footprint to reasonable levels. See the `fio-format` utility for information on changing the device sector sizes.

- Be sure to provide the serial numbers for the ioMemory device, not the adapter.

- `FIO_PREALLOCATE_MEMORY` is necessary to have the device usable as swap space. This will ensure the device is crash-free during operation. See Appendix I- fio-config Options for more information on setting this parameter.

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

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

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

### Understanding TRIM Support

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

TRIM addresses an issue unique to solid-state storage. When a user deletes a file, the device does not recognize that it can reclaim the space. Instead the device assumes the data is valid.

TRIM is a feature on newer operating systems. 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.

For a complete description of TRIM support on Windows, see the TRIM Support appendix.

- Windows does not support TRIM with a RAID 5 configuration.
TRIM on Windows Server 2008 R2

Windows Server 2008 R2 has built-in TRIM support. With this operating system, ioMemory devices work with Windows TRIM commands by default.

TRIM on Windows Server 2003 and Windows Server 2008 R1

Windows TRIM is not built into Windows Server 2003 or Windows Server 2008 R1. However, the Fusion-io TRIM service is installed with the Windows ioMemory VSL and it provides the necessary TRIM operations.

The Fusion-io TRIM service is enabled by default, unless it detects an operating system that supports TRIM (such as Windows Server 2008 R2). You can disable the Fusion-io TRIM service by using the `fio-trim-config` utility. See `fio-trim-config` for more details.
Maintenance

The ioMemory VSL includes software utilities for maintaining the device. You can also install SNMP as a monitoring option.

Device LED Indicators

The ioMemory device includes three LEDs showing drive activity or error conditions. The LEDs on your device should be similar to one of these configurations:
This table explains the information that these LEDs convey:

<table>
<thead>
<tr>
<th>Green</th>
<th>Yellow</th>
<th>Amber</th>
<th>Indicates</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Power off</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Power on (Driver not loaded and card not attached)</td>
<td>Load driver and attach card</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Power on (Driver loaded but card not attached)</td>
<td>Attach card</td>
</tr>
<tr>
<td>(Flashing)</td>
<td>0</td>
<td>0</td>
<td>Writing (Rate indicates volume of writes)</td>
<td>Can appear in combination with the Read LED</td>
</tr>
<tr>
<td>(Flashing)</td>
<td>0</td>
<td>0</td>
<td>Reading (Rate indicates volume of reads)</td>
<td>Can appear in combination with the Write LED</td>
</tr>
</tbody>
</table>

⚠️ ioFX devices have an additional LED that illuminates the ioFX logo. This LED has no functional significance and it cannot be turned off.

**SNMP Support**

The ioMemory VSL Windows Setup program provides the option to install support for SNMP. If you choose this option, the Setup program installs the components and modifies the Registry for this support. You must also have the Microsoft Windows SNMP Service installed and running on the computer to receive reports.

Once you run the VSL Windows Setup program, it will stop and start the Windows SNMP Service to recognize the VSL’s agent.

If you did not choose to install the SNMP support at Setup, and want to do so later, rerun the Setup program. Choose to install only the SNMP support from the list of items. Once the Setup program completes the install, it will stop and restart the Windows SNMP Service.

For details on using SNMP Test Mode, see Appendix G- SNMP Test Mode and MIB Support.

**GUI Management**

ioSphere is a free browser-based solution for managing ioMemory devices. It is also available from

http://ts.fujitsu.com/support/, but it is located in a different download location along with documentation.
The ioSphere can perform:

- Firmware upgrades
- Low-level formatting
- Attach and detach actions
- Device status and performance information

Command-line Utilities

The Windows Setup package also includes several command-line utilities for managing your

http://ts.fujitsu.com/support/:

- fio-attach
- fio-beacon
- fio-bugreport
- fio-config
- fio-detach
- fio-format
- fio-pci-check
- fio-status
- fio-sure-erase
- fio-trim-config
- fio-update-iodrive

Each of these is described in detail in Appendix C- Command-line Utilities.

Enabling PCIe Power Override

If you have installed any products with multiple ioMemory devices, such as the ioDrive Duo device, then the device may require additional power to properly function (beyond the minimum 25W provided by PCIe Gen2 slots). Even if additional power is not required for your device, all dual ioMemory devices that receive additional power may benefit with improved performance.

ioDrive2 Duo devices, **must** have additional power in order to properly function. For more information on which devices require additional power, see the section on Power Cables for Multi-device Products in the ioMemory Hardware Installation Guide.
This additional power may be provided in two ways:

- **External Power Cable**: This cable ships with all dual ioMemory devices. See the *ioMemory Hardware Installation Guide* for information on installing this cable.

  When a power cable is used, all of the power is drawn from the cable and no power is drawn from the PCIe slot.

- **Enabling Full Slot Power Draw**: Some PCIe slots provide additional power (often up to 75W of power). If your slot is rated to provide at least 55W, then you may allow the device to draw full power from the PCIe slot by setting a VSL module parameter. For more information on enabling this override parameter, see the instructions below in the next section.

  This parameter overrides the setting that prevents device(s) from drawing more than 25W from the PCIe slot. The parameter is enabled per device (using device serial numbers). Once the setting is overridden, each device may draw up to the full 55W needed for peak performance.

  **WARNING**
  If the slot is not capable of providing the needed amount of power, then enabling full power draw from the PCIe slot may result in malfunction or even damage server hardware. You are responsible for any damage to equipment due to improper use of this override parameter and Fusion-io expressly disclaims any liability for any damage arising from such improper use. Contact Fusion-io Customer Support if there are any questions or concerns about the override parameter use.

Before you enable this override parameter, ensure that each PCIe slot you will use is rated to provide enough power for all slots, devices, and server accessories. Consult the server documentation, BIOS interface, setup utility, and/or use `fio-pci-check` to determine the slot power limits.

**Confirm with Server Manufacturer**

Contact the server manufacturer to confirm the power limits and capabilities of each slot, as well as the entire system.
The following are important considerations:

- If you are installing more than one dual ioMemory device, and enabling the override parameter for each device, make sure the motherboard is rated to provide 55W power to each slot used.

  For example, some motherboards safely provide up to 75W to any one slot, but run into power constraints when multiple slots are used to provide that much power. Installing multiple devices in this situation may also result in server hardware damage. Consult with the manufacturer to determine the total PCIe slot power available.

- The override parameter, if enabled correctly, will persist in the system, and will enable full power draw on an enabled device even if the device is removed and then placed in a different slot within the same system. If the device is placed in a slot that is not rated to provide 55W of power, then you may damage your server hardware.

- This override parameter is a setting for the ioMemory VSL software per server, and is not stored in the device. When moved to a new server, the device will default to the 25W power limit until an external power cable is added or this override parameter is enabled for that device in the new server. Consult with the manufacturer to determine the total PCIe slot power available for the new server.

### Enabling the Override Parameter

#### Determine Serial Number(s)

Before you enable this parameter, determine the adapter serial number for each device you will put in a compatible slot. Use the `fio-status` command-line utility to determine the adapter serial number(s).

**Serial Number Label**

You may also inspect the adapter serial number label(s) on the device(s) to determine the serial number(s). However, as a best practice, confirm that each serial number is an adapter serial number by running `fio-status`. The adapter serial number label resides on the back of all ioDrive Duo devices and ioDrive2 Duo devices. On ioDrive Duo devices, it is on the PCB component that is attached to the PCIe connector.

**Using `fio-status`:** Run the `fio-status` command-line utility. Sample output:

```plaintext
fio-status
...
Adapter: Dual Controller Adapter
  Fusion-io ioDrive2 DUO 2.41TB, Product Number:F01-001-2T41-CS-0001, FIO SN:1149D0969
  External Power: NOT connected
  PCIe Power limit threshold: 24.75W
  Connected ioMemory modules:
    fct2: SN:1149D0969-1121
    fct3: SN:1149D0969-1111
```
In this example, 1149D0969 is the adapter serial number.

**Using fio-beacon:** If you have multiple devices installed, you may use the `fio-beacon` utility to verify where each device is physically located. Consult the utility documentation in the appendix for more information.

**Setting the Parameter**

Set the module parameter using the `fio-config` command-line utility. For example:

```
fio-config -p FIO_EXTERNAL_POWER_OVERRIDE <value>
```

Where the `<value>` for this parameter is a comma-separated list of adapter serial numbers. For example: 1149D0969,1159E0972,24589

⚠️ By setting this parameter, you **overwrite previous values**. If you wish to add additional serial numbers to the list, you must list the new serial numbers as well as the previously entered numbers. To clear the list, set the parameter without any values.

The `-p` option makes the parameter persistent after reboot. Reboot the system to enable the parameter.

ℹ️ You can use the `-e` option (enumerate current parameters and values) to see if the serial numbers were properly set, or to see what serial numbers are already set before you add additional numbers to the list.

**Uninstalling the ioMemory VSL**

To uninstall the ioMemory VSL,

1. Go to **Start > Control Panel**.
2. Click **Programs & Files**.
3. Select the **Fusion-io ioMemory VSL** entry.
4. Click **Uninstall**.

Windows uninstalls the ioMemory VSL folder along with all files and folders.

⚠️ The ioMemory VSL Utilities are **not** uninstalled with this procedure. If you are upgrading to a newer version of the ioMemory VSL, you do not need to manually uninstall these utilities. However, if you are uninstalling the software completely, or planning on installing an earlier version of the ioMemory VSL, you should manually remove the following folder and its contents: C:\Program Files\Common Files\VSL Utils
Upgrading the ioMemory VSL - Non-RAID Configuration

⚠️ Be sure to read the Release Notes document that comes with each new release as well as these installation instructions to ensure no loss of data when performing upgrades.

To upgrade the ioMemory VSL in a non-RAID configuration:

1. Follow the steps in Uninstalling the ioMemory VSL earlier.
2. Download the latest driver from http://ts.fujitsu.com/support/
3. Either unzip or run the Windows package to copy the files to a convenient directory.
4. Go to Start > Control Panel.
5. Click Administrative Tools.
6. Click Computer Management.
7. Click Device Manager in the console tree at the left.
8. Expand the Fusion-io Devices item. (Select System Devices with pre-1.2.2 drivers.)
9. Right-click the desired device.
10. Click Update ioMemory VSL Software. If needed, refer to Appendix B- Manual Installation for details on the remaining steps to install the updated ioMemory VSL.

Windows now detects your ioMemory device(s) with the upgraded ioMemory VSL.

Upgrading the ioMemory VSL with a RAID Configuration

⚠️ Be sure to read the Release Notes document that comes with each new release as well as these installation instructions to ensure no loss of data when performing upgrades.

To upgrade the ioMemory VSL with a RAID configuration in place:

1. Shut down any applications that are accessing the ioMemory devices.
2. Open the ioMemory VSL Utilities folder. (The default location for this release is C:\Program Files\Fusion-io ioMemory VSL\Utils.)
3. Double-click the AutoAttachDisable.reg file to add a key to the Windows registry. Your ioMemory device now will not automatically attach the next time you restart the computer.
4. Uninstall the VSL software in Windows Add/Remove Programs.
5. Restart the computer.


7. Unzip and install the ioMemory VSL driver package. While finishing installation, click the "No" button to select a manual restart.

8. Open the VSL Utilities folder. (The default location is `C:\Program Files\Common Files\VSL Utils`.)

9. Double-click the `AutoAttachEnable.reg` file to reset the key in the Windows registry. Your ioMemory device now will automatically attach the next time you restart the computer.

10. Update the firmware of the devices. Follow the steps in Upgrading the Device Firmware, which is the next section.

   ![Restart computer after firmware upgrade](image)

   Restart the computer after the firmware upgrade is complete. The VSL Check Utility will run at next boot.

Windows now detects your devices in the RAID configuration with the upgraded software.

### Upgrading the Device Firmware

⚠️ You should upgrade the firmware only if the System Event Log reports out-of-date firmware, or if instructed to do so by Customer Support or the Release Notes and Errata document.

### Viewing the Firmware Version

The firmware version can be found in the Windows Event Log. It is reported by ioManager and the `fio-status` command-line utility. For more details, see the ioManager User Guide or `fio-status` in Appendix C(Command-line Utilities).

⚠️ Upgrade Path

There is a specific upgrade path that you must take when upgrading ioMemory device. Consult the Release Notes for this ioMemory VSL release before upgrading ioMemory devices.

### Performing the Upgrade

⚠️ You should back up the data on the device prior to any upgrade as a precaution.

To perform the upgrade, use either ioManager (see the `ioManager User Guide`) or the `fio-update-iodrive` command-line utility.

Your ioMemory device may have a minimum firmware label affixed (for example, "MIN FW: XXXXXX"). This label indicates the minimum version of the firmware that is compatible with your device.
Do not attempt to downgrade the firmware on any ioMemory device.

When installing a new ioMemory device along with existing devices, you must upgrade all of the currently installed devices to the latest available versions of the firmware and ioMemory VSL before installing the new devices.

Consult the Release Notes for this ioMemory VSL release for any upgrade considerations.

Upgrading VMware Guest OS

If you are using your ioMemory device with a VMware guest OS (using VMDirectPathIO), you must cycle the power on the host after you upgrade the device(s). Just restarting the virtual machine won't apply the change.

Defragmentation

The ioMemory device does not need to be defragmented. Some versions of Windows, however, run defragmentation as a scheduled task automatically. If necessary, you should turn off automatic defragmentation.

Unmanaged Shutdown Issues

Unmanaged shutdown due to power loss or other circumstances can force the ioMemory device to perform a consistency check during restart. This may take several minutes or more to complete and is shown by a progress percentage during Windows startup.

You can cancel this consistency check by pressing Esc during the first 15 seconds after the "Fusion-io Consistency Check" message appears at the prompt. If you choose to cancel the check, however, the ioMemory device(s) will remain unavailable to users until the check is done. (You can perform this check later on using ioManager's Attach function).

Although data written to the ioMemory device will not be lost due to unmanaged shutdows, important data structures may not have been properly committed to the device. This consistency check repairs these data structures.

Disabling Auto-Attach

The ioMemory VSL defaults to automatically attach (auto-attach) all installed ioMemory devices to the operating system. (If the ioMemory device does not attach, it will not be available to applications or users.) You can disable auto-attach to assist in troubleshooting or diagnostics.
To disable auto-attach:

1. Open the ioMemory VSL Utilities folder. (The default location is C:\Program Files\Fusion-io ioMemory VSL\Utils).
2. Double-click the autoattachdisable.reg file.
3. If you receive a prompt at this point, confirm that you want to modify the registry.

This creates a new DWORD parameter registry key called AutoAttach in:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fiodrive\Parameters
```

Once you restart your system, your ioMemory device will no longer automatically attach until you re-enable auto attach (see Enabling Auto-Attach).

When you finish troubleshooting the ioMemory VSL issue, use ioManager to attach the ioMemory device(s) and make them available to Windows.

**Enabling Auto-Attach**

To re-enable auto-attach after disabling it using the method described in Disabling Auto-Attach:

1. Open the ioMemory VSL Utilities folder. (The default location is C:\Program Files\Fusion-io ioMemory VSL\Utils).
2. Double-click the autoattachenable.reg file.
3. If you receive a prompt at this point, confirm that you want to modify the registry.

This resets the AutoAttach parameter in the Registry. The next time you restart your Windows system, your ioMemory device will automatically attach.
Performance and Tuning

ioMemory devices provide high bandwidth, and high Input/Output per Second (IOPS), and are specifically designed to achieve low latency.

As ioMemory devices improve in 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 ioMemory devices.

While Fusion-io devices generally perform well out of the box, this section describes some of the common areas where tuning may help achieve optimal performance.

Disabling DVFS

Dynamic Voltage and Frequency Scaling, or 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 maximum IOPS. When tuning for maximum performance, you may benefit from reducing or disabling DVSF completely, even though this may increase power consumption.

DVFS, if available, should be 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 are known to 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.

These power savings techniques are known to have a negative impact on I/O latency and maximum IOPS. 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.
If your processor has ACPI C-states available, you can typically limit/disable them in the BIOS interface (sometimes referred to as a Setup Utility). ACPI C-states may be part of the Advanced Configuration and Power Interface (ACPI) menu; consult your computer documentation for details.

**Setting NUMA Affinity**

Servers with a NUMA (Non-Uniform Memory Access) architecture require special installation instructions in order to maximize ioMemory device performance. These servers include the HP DL580, HP DL980, or the IBM 3850 server.

On servers with NUMA architecture, during system boot, the BIOS on some systems will not distribute PCIe slots evenly among the NUMA nodes. Each NUMA node contains multiple CPUs. This imbalanced distribution means that, during high workloads, half or more of the CPUs may remain idle while the the rest are 100% utilized. To prevent this imbalance, you must manually assign ioMemory devices equally among the available NUMA nodes.

For information on setting NUMA affinity, see [Appendix I- NUMA Configuration](#).

**Setting the Interrupt Handler Affinity**

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

Many operating systems will attempt to dynamically place interrupts for you and generally make good decisions. Hand tuning interrupt placement 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 interrupts for a given device to specific CPUs.
Appendix A- Troubleshooting Event Log Messages

The Windows System Event Log displays the following fiodrive messages concerning the ioMemory device: Informational, Warnings, and Errors.

Each ioMemory device is numbered from 0 upwards. These numbers reflect the PCIe bus number where you installed the device. Use the fio-status utility or ioManager to view this number for your device.

To open the Windows Event Viewer,

1. Click Start.
2. Click Computer and right-click Manage.
3. Expand Diagnostics.
4. Expand Event Viewer.
5. Expand Windows Logs.

Error Messages

The following are common Event Log error messages, along with suggested solutions:

<table>
<thead>
<tr>
<th>Message</th>
<th>Suggested Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error: ioDrive(s) firmware is too old. The firmware must be updated.</td>
<td>Use the firmware upgrade instructions in the Maintenance section to update the firmware.</td>
</tr>
</tbody>
</table>
| Error: ioDrive initialization failed with error code 0xerrorcode (where errorcode is a number that may vary) | 1. Reinstall the Windows ioMemory VSL.  
2. Remove and reseat the ioMemory device.  
3. Remove and insert the ioMemory device in a different PCIe slot |
| Error: ioDrive was not attached. Use the fio-attach command-line utility or ioManager to re-attach the device. This attach process may take up to ten minutes as the utility performs a consistency check on the device(s). |
Warning: ioDrive was not loaded because auto-attach is disabled.

The ioMemory device must attach to the Windows operating system to be available to users and applications. (This attach normally occurs at boot time.) As part of this attach process, the ioMemory VSL checks to see if there is an AutoAttach parameter in the Windows registry. If you create this Registry parameter to disable auto-attach, the attach operation does not complete.

To attach an unattached device,

1. Run ioManager.
2. Select your unattached ioMemory device from the Device Tree.
3. Click Attach.
4. Confirm the Attach operation.

Your device now attaches to the Windows operating system. To re-enable Auto-Attach at boot time, refer to Enabling Auto-Attach in the Maintenance section.

### Informational Messages

The following is a common Event Log informational message:

<table>
<thead>
<tr>
<th>Message</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affinity not set for ioMemory VSL device fct119 because either WIN_DISABLE_ALL_AFFINITY is set to true or &quot;SetWorkerAffinity119&quot; does not exist in the registry and WIN_DISABLE_DEFAULT_NUMA_AFFINITY is set to true.</td>
<td>When WIN_DISABLE_ALL_AFFINITY is set to 0, the driver will enable interrupt and worker thread affinity in the driver. When WIN_DISABLE_ALL_AFFINITY is set to 1, the driver will disable all affinity settings. This is an override of any other affinity settings. Refer to fio-config for more information about affinity settings.</td>
</tr>
</tbody>
</table>
Appendix B - Manual Installation

The Windows Setup program will install ioMemory VSL and software on your Windows operating system. However, there are some instances where you may need to manually install the software for a particular ioMemory device, including:

- After a software installation (including upgrade), ioMemory devices don't show up in fio-status.
- You install new ioMemory devices on a system that has previously installed ioMemory devices and ioMemory VSL software.

Follow the steps below for Windows Server 2003 or Windows Server 2008. This will ensure that the ioMemory VSL is installed for a particular device. Repeat the steps for each device, if needed.

Manual Installation on Windows Server 2003

Before you manually install the ioMemory VSL driver, make sure you have downloaded and run the ioMemory VSL Windows Setup program from [http://ts.fujitsu.com/support/](http://ts.fujitsu.com/support/). This will install the ioMemory VSL on the system, and you will now be able to install the ioMemory VSL for each ioMemory device.

The Windows Driver Wizard may automatically detect the new ioMemory device and starts to locate its ioMemory VSL after you restart the system. If this happens, you may skip to the Installation Wizard procedure below.

1. Choose **Start > Control Panel > Administrative Tools > Computer Management > Device Manager**.
2. Select **Fusion ioMemory VSL devices**.
3. Click on your ioMemory device(s) in the list. The Properties dialog box appears.

⚠️ The device may be titled Mass Storage Controller.

![ioMemory VSL Device Properties](image)

- a. If the **Device Status** reads *This device is working properly*, then the ioMemory VSL has been installed.

- b. If the device is not working correctly, you will need to manually install the software for that device. Continue with the manual installation.

4. Close the Properties dialog box.

5. Right-click on the device and choose **Update Driver**.

6. Follow the instructions below.
1. Windows will ask if it is okay to connect to Windows update to find the driver. Select **No, not this time** and choose **Next**.

![Found New Hardware Wizard](image)

**Welcome to the Found New Hardware Wizard**

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

[Online privacy information](#)

Can Windows connect to Windows Update to search for software?

- [ ] Yes, this time only
- [ ] Yes, now and every time I connect a device
- [x] **No, not this time**

Click Next to continue.
2. Click **Install from a list or specific location**. Click **Next**.

![Found New Hardware Wizard](image)

**This wizard helps you install software for:**

- ioMemory VSL Device

**If your hardware came with an installation CD or floppy disk, insert it now.**

**What do you want the wizard to do?**

- ![Install the software automatically (Recommended)]
- ![Install from a list or specific location (Advanced)]

Click **Next** to continue.
3. Choose **Don't Search. I will choose the driver to install.** Click **Next.**

   ![Found New Hardware Wizard]

   **Please choose your search and installation options.**

   - **Search for the best driver in these locations.**
     - Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
     - **Search removable media (floppy, CD-ROM...)**
     - **Include this location in the search:**
       - C:\WINDOWS\OPTIONS\CABS
     - **Browse**
   - **Don't search. I will choose the driver to install.**
     - Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

   ![Next button]

4. Click **Have Disk** to bring up a browsing dialog.
5. Browse to the folder with the ioMemory VSL (the default is `C:\Program Files\Fusion-io ioMemory VSL\<VSL-Version>\Driver`).
6. Click **OK**.
7. The driver you selected should now be in the compatible hardware window. Make sure it is selected and choose Next.

When the ioMemory VSL install completes, Windows displays this message:
Now you can view the ioMemory VSL in device manager.
If you need to update your firmware, review the Outdated Firmware Check and Upgrading the Device Firmware sections of this guide.

**Manual Install on Windows Server 2008**

Before you manually install the ioMemory VSL driver, make sure you have downloaded and run the ioMemory VSL Windows Setup program from [http://ts.fujitsu.com/support/](http://ts.fujitsu.com/support/). This will install the ioMemory VSL on the system, and you will now be able to install the ioMemory VSL for each ioMemory device.

The Windows Driver Wizard may automatically detect the new ioMemory device and starts to locate its ioMemory VSL after you restart the system. If this happens, you may skip to the Installation Wizard procedure below.

1. Choose **Start > Control Panel > Device Manager**.

2. Select **Fusion ioMemory VSL devices**.
3. Click on your ioMemory device(s) in the list. The Properties dialog box appears.

⚠️ The device may be titled Mass Storage Controller.

![ioMemory VSL Device Properties]

- a. If the **Device Status** reads *This device is working properly*, then the ioMemory VSL has been installed.

- b. If the device is not working correctly, you will need to manually install the software for that device. Continue with the manual installation.

4. Close the Properties dialog box.

5. Right-click on the device and choose **Update Driver**.

6. Follow the instructions below.
Installation Wizard

1. Windows will ask you to locate the software driver.

2. Click **Browse** next to the path field. Windows displays a file dialog.

3. Select the folder with the ioMemory VSL (the default is `C:\Program Files\Fusion-io ioMemory VSL\<VSL-Version>\Driver\`).

4. Click **OK**.

5. Click **Next**.
   - Windows finds the correct driver and installs the device software. When the driver installation completes,

6. Restart the computer.

7. Proceed to the [Outdated Firmware Check](#) section to continue.
Appendix C- Command-line Utilities

The Windows Setup package installs various utilities into the C:\Program Files\Common Files\VSL Utils folder, described in the table below.

There are some additional utilities installed in the C:\Program Files\Common Files\VSL Utils directory that are not listed below. Those additional utilities are dependencies (used by the main VSL utilities), and you should not use them directly unless Customer Support advises you to do so.

<table>
<thead>
<tr>
<th>Utility</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>fio-attach</td>
<td>Makes an ioMemory device available to the OS</td>
</tr>
<tr>
<td>fio-beacon</td>
<td>Lights the ioMemory device's external LEDs</td>
</tr>
<tr>
<td>fio-bugreport</td>
<td>Prepares a detailed report for use in troubleshooting problems</td>
</tr>
<tr>
<td>fio-config</td>
<td>Enables configuration parameters for device operation</td>
</tr>
<tr>
<td>fio-detach</td>
<td>Temporarily removes an ioMemory device from OS access</td>
</tr>
<tr>
<td>fio-format</td>
<td>Used to perform a low-level format of an ioMemory device</td>
</tr>
<tr>
<td>fio-pci-check</td>
<td>Checks for errors on the PCI bus tree, specifically for ioMemory device</td>
</tr>
<tr>
<td>fio-status</td>
<td>Displays information about the device</td>
</tr>
<tr>
<td>fio-trim-config</td>
<td>Enables or disables the TRIM feature</td>
</tr>
<tr>
<td>fio-update-iodrive</td>
<td>Updates the ioMemory device's firmware</td>
</tr>
<tr>
<td>fio-sure-erase</td>
<td>Clears or purges data from the device</td>
</tr>
</tbody>
</table>

**Administrator Rights:** The command-line utilities require administrator rights in order to run under Windows (right-click the Command Prompt menu item and select Run as administrator.)

To run these utilities from a command line, you must either change to the directory which contains them (by default, C:\Program Files\Common Files\VSL Utils) or add that directory to your system path. As a convenience, if you used the Windows installer then the utilities directory has been added to the system path for you. Otherwise, see the documentation for your version of Windows for information about adding a directory to the system path.
**fio-attach**

**Description**

Attaches the ioMemory device and makes it available to Windows. You can then partition the ioMemory device, or set it up as part of a RAID array, using the Windows Disk Management utility. This command displays a progress bar and percentage as it completes the attach process.

> In most cases, the ioMemory VSL 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's auto_attach parameter to 0.

> If the ioMemory device 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 given by the ioMemory VSL to your device. This name is `/dev/fctx`, where `x` indicates the PCIe bus number where you installed the ioMemory device. (For example, the name `/dev/fct4` refers to the ioMemory device installed in PCIe Bus 4 in your Windows system. Use ioManager or `fio-status` to view this bus number.)

You can specify multiple ioMemory devices. For example, `/dev/fct1 /dev/fct2` indicates the ioMemory devices installed in PCIe Buses 1 and 2 in your Windows system.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c</td>
<td>Attach only if clean.</td>
</tr>
<tr>
<td>-q</td>
<td>Quiet: disables the display of the progress bar and percentage.</td>
</tr>
</tbody>
</table>

**fio-beacon**

**Description**

Lights the ioMemory device's three LEDs to locate the device. You should first detach the ioMemory device and then run fio-beacon.
Syntax

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

where `<device>` is the name given by the ioMemory VSL to your device. This name is `/dev/fctx`, where `x` indicates the device number.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0</td>
<td>Off: (Zero) Turns off the three LEDs</td>
</tr>
<tr>
<td>-1</td>
<td>On: Lights the three LEDs</td>
</tr>
<tr>
<td>-p</td>
<td>Prints the PCI bus ID of the device at <code>&lt;device&gt;</code> to standard output. Usage and error information may be written to standard output rather than to standard error.</td>
</tr>
</tbody>
</table>

**fio-bugreport**

**Description**

Prepares a detailed report of the device for use in troubleshooting problems.

**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 send the output to support@fusionio.com for assistance in troubleshooting.

`fio-bugreport` runs several information-gathering utilities and combines the resulting data into a text file. The results are saved in the `utils` directory (default installation path is `C:\Program Files\Common Files\VSL\Utils`) in a `.cab` file that indicates the date and time the utility was run.

You are then prompted to send an e-mail describing the problem to support@fusionio.com with the bug report file attached.
Sample Output

```
C:\Users\username>"\Program Files\Fusion-io\Utils\fio-bugreport.exe"
Generating bug report. Please wait, this may take a while...
---------------------------------------------
Gathering all Windows Event Logs...DONE
Gathering Fusion-io Windows Event Logs...DONE
Gathering System Information...DONE
Running fio utilities...DONE
Compressing to CAB file...DONE
Bug report has successfully been created:
fio-bugreport-20100222_192621.cab.
Please e-mail this file to support@fusionio.com
```

For example, the filename for a bug report file named fio-bugreport-20090921.192621.cab indicates the following:

- Date (20090921)
- Time (192621, or 19:26:21)

**fio-config**

**Description**

Sets and gets ioMemory VSL configuration parameters for device operation. For a list of parameters, see [Parameters](#) Reference below.

In order for the parameter value(s) to be enforced, you must either reboot the system or first disable and then re-enable all ioMemory devices in the [Device Manager](#). This will reload the ioMemory VSL with the values(s) enabled. Be sure to use the `-p` option if you plan to reboot.

**Syntax**

```
fio-config [options] [<parameter>] [<value>]
```

where `<parameter>` is the ioMemory VSL parameter you wish to set, and `<value>` is the value you wish to set for the parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-e</td>
<td>Enumerate configuration parameter names and values.</td>
</tr>
<tr>
<td>-g &lt;name&gt;</td>
<td>Get the configuration parameter.</td>
</tr>
</tbody>
</table>
### fio-config Options

Options must be entered in uppercase to function properly.

- **-p** `<name>`: Set and make the configuration parameter persistent. **Use this option if you want the parameter setting to remain after a reboot.**
- **-s** `<name>`: Set the configuration parameter in memory only.
- **-V**: Print verbose information.
- **-v**: Print version information.

### Parameters Reference

The following table describes the ioMemory VSL parameters you can set with the `fio-config` utility.

* MSI (Message Signaled Interrupts) is enabled by default for this platform, and it cannot be disabled using `fio-config`.

* Other than `FIO_PREALLOCATE_MEMORY` and `FIO_EXTERNAL_POWER_OVERRIDE`, all `fio-config` options are global—they apply to all Fusion-io devices in the computer.

* By setting the `FIO_PREALLOCATE_MEMORY` and `FIO_EXTERNAL_POWER_OVERRIDE` parameters, you overwrite previous values. If you wish to add additional serial numbers to the list, you must list the new serial numbers as well as the previously entered numbers. To clear the list, set the parameter without any values.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default (min/max)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AUTO_ATTACH</code></td>
<td>1 (0, 1)</td>
<td>Always attach the device on driver load (1).</td>
</tr>
<tr>
<td><code>IODRIVE_TINTR_HW_WAIT</code></td>
<td>0 (0, 255)</td>
<td>Interval (microseconds) to wait between hardware interrupts.</td>
</tr>
<tr>
<td><code>FIO_EXTERNAL_POWER_OVERRIDE</code></td>
<td>No devices selected</td>
<td>Allows selected devices to draw full power from the PCIe slot. Where the <code>&lt;value&gt;</code> for this parameter is a comma-separated list of adapter serial numbers.</td>
</tr>
<tr>
<td><code>FORCE_MINIMAL_MODE</code></td>
<td>0 (0, 1)</td>
<td>Force minimal mode on the device (1), this parameter is set to false (0) by default.</td>
</tr>
<tr>
<td><code>PARALLEL.Attach</code></td>
<td>0 (0, 1)</td>
<td>Enable parallel attach of multiple devices (1), this parameter is set to false (0) by default.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FIO_PREALLOCATE_MEMORY</td>
<td>0</td>
<td>For the selected device, pre-allocate all memory necessary to have the drive usable as swap space. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>fio-config /dev/fct0 -p FIO_PREALLOCATE_MEMORY = &quot;1234,54321&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>where &quot;1234&quot; and &quot;54321&quot; are serial numbers obtained from <code>fio-status</code>.</td>
</tr>
<tr>
<td>WIN_DISABLE_ALL_AFFINITY</td>
<td>0</td>
<td>When <code>WIN_DISABLE_ALL_AFFINITY</code> is set to 0, the driver will enable interrupt and worker thread affinity in the driver. When <code>WIN_DISABLE_ALL_AFFINITY</code> is set to 1, the driver will disable all affinity settings. This is an override of any other affinity settings. The driver must be reloaded for this parameter to take effect.</td>
</tr>
<tr>
<td>WIN_DISABLE_DEFAULT_NUMA_AFFINITY</td>
<td>0</td>
<td>When <code>WIN_DISABLE_DEFAULT_NUMA_AFFINITY</code> is set to 0, during initialization, the driver will query Windows for the affinity settings assigned to the adapter by the OS. This is what is known as the &quot;default NUMA affinity&quot;. Once the affinity is queried correctly, the driver sets the affinity of the adapter's interrupt and associated worker threads to the default OS setting. This generally has the effect of setting the affinity of the interrupt and worker threads to all processors on a single NUMA node in the system. When <code>WIN_DISABLE_DEFAULT_NUMA_AFFINITY</code> is set to 1, the driver will ignore the affinity settings assigned to the adapter by the OS. The driver must be reloaded for this parameter to take effect.</td>
</tr>
<tr>
<td>FIO_AFFINITY</td>
<td>N/A</td>
<td>FIO_AFFINITY is a list of <code>&lt;affinity specification&gt;</code> triplets to specify the affinity settings of all adapters in the system. Each item in the triplet is separated by a comma, and each triplet set is separated by a semicolon. For syntax information and examples showing the use of this parameter, see Appendix I- NUMA Configuration.</td>
</tr>
</tbody>
</table>
**fio-detach**

**Description**

Detaches the ioMemory device and removes the corresponding fctx 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.

⚠️ 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 given by the ioMemory VSL to your device. This name is `/dev/fctx`, where `n` indicates the device number. (The number reflects the PCIe bus for the ioDrive.) For example, the name `/dev/fctx4` refers to the ioMemory device installed in PCIe Bus 4 in your system. (Use fio-status to view this number.)

You can specify multiple ioMemory devices. For example, `/dev/fctx1 /dev/fctx2` indicates the ioDrives installed in PCIe Buses 1 and 2 in your Windows system.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f</code></td>
<td>Force: Causes an immediate detach (does not save metadata).</td>
</tr>
<tr>
<td><code>-i</code></td>
<td>Immediate: Causes a forced immediate detach (does not save metadata). This will fail if the device is in use by the OS.</td>
</tr>
<tr>
<td><code>-q</code></td>
<td>Quiet: Disables the display of the progress bar and percentage.</td>
</tr>
</tbody>
</table>

**Notes**

Attempting to detach an ioMemory device may fail with an error indicating that the device is busy. This typically may occur if the device is part of a software RAID (0,1,5) volume. Windows refuses the request to detach the drive associated with the ioMemory device because it is part of a RAID volume and may cause the volume to fail. This does not occur with simple volumes (such as a single ioMemory device). To detach in this case, take the volume offline using the Disk Management MMC plug-in application.
**fio-format**

**Description**

The ioMemory device ships pre-formatted, so `fio-format` is generally not required except to change the logical size or block size of the device, or to erase user data on the device.

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

- **Use this utility with care, as it deletes all user information on the device.**
- **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.**
- **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.**

**Syntax**

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

where `<device>` is the name given by the ioMemory VSL to your device. This name is `/dev/fctx`, where `x` indicates the PCIe bus number where you installed the ioMemory device. (For example, the name `/dev/fct4` refers to the ioMemory device installed in PCIe Bus 4 in your Windows system. Use ioManager or `fio-status` to view this bus number.)

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`-b &lt;size B</td>
<td>K&gt;`</td>
</tr>
<tr>
<td><code>-f</code></td>
<td>Force the format size, bypassing normal checks and warnings. This option may be needed in rare situations when <code>fio-format</code> does not proceed properly. (The &quot;Are you sure?&quot; prompt still appears unless you use the <code>-y</code> option.)</td>
</tr>
<tr>
<td><code>-q</code></td>
<td>Quiet mode: Disable the display of the progress-percentage indicator.</td>
</tr>
</tbody>
</table>
### Set the device capacity as a specific size (in TB, GB, or MB) or as a percentage of the advertised capacity:

- **s <size M|G|T|%>**
  - 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)

### Over-format the device size (to greater than the advertised capacity), where the maximum size equals the maximum physical capacity. If a percentage is used, it corresponds to the maximum physical capacity of the device. (Size is required for the -o option; see the -s option above for size indicator descriptions.)

**-o <size B|K|M|G|T|>%**

⚠️ Before you use this option, please discuss your use case with Customer Support.

- **-R** Disable fast rescan on unclean shutdowns at the cost of some capacity.
- **-y** Auto-answer "yes" to all queries from the application (bypass prompts).

### fio-pci-check

**Description**

Checks for errors on the PCI bus tree, specifically for ioMemory devices. This utility displays the current status of each ioMemory device. It also prints the standard PCI Express error information and resets the state.

ℹ️ It is perfectly normal to see a few errors (perhaps as many as five) when fio-pci-check is initially run. Subsequent runs should reveal only one or two errors during several hours of operation.

⚠️ The ioMemory VSL must be loaded to run this utility. Some PCI errors cannot be reset in Windows.

**Syntax**

```
fio-pci-check [options]
```
"Yes" is forced when the user is asked to continue.

**fio-status**

**Description**

Provides detailed information about the installed ioMemory devices. This utility operates on either fctx or fiox devices. The utility depends on running as root and having the ioMemory VSL loaded. If no ioMemory VSL 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 given by the ioMemory VSL to your device. This name is /dev/fctx, where *x* indicates the device number. (The number reflects the PCIe bus for the ioMemory device.) For example, the name /dev/fctx4 refers to the ioMemory device installed in PCIe Bus 4 in your system. (*fio-status* displays this number.)

If *<dev>* is not specified, *fio-status* displays information for all devices in the system. If the ioMemory VSL is not loaded, this parameter is ignored.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>Report all available information for each device.</td>
</tr>
<tr>
<td>-e</td>
<td>Show all errors and warnings for each device. This option is for diagnosing issues, and it hides other information such as format sizes.</td>
</tr>
<tr>
<td>-c</td>
<td>Count: Report only the number of ioMemory devices installed.</td>
</tr>
<tr>
<td>-d</td>
<td>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.</td>
</tr>
<tr>
<td>-fj</td>
<td>Format JSON: creates the output in JSON format.</td>
</tr>
<tr>
<td>-fx</td>
<td>Format XML: creates the output in XML format.</td>
</tr>
<tr>
<td>-u</td>
<td>Show unavailable fields. Only valid with -fj or -fx.</td>
</tr>
<tr>
<td>-U</td>
<td>Show unavailable fields and details why. Only valid with -fj or -fx.</td>
</tr>
<tr>
<td>-F&lt;field&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>-l</td>
<td>List the fields that can be individually accessed with -F.</td>
</tr>
</tbody>
</table>
Output Change

Starting with version 3.0.0 and later, the standard formatting of `fio-status` output has changed. This will affect any custom management tools that used the output of this utility.

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

- Number and type of cards installed in the system
- Software version

Adapter information:

- Adapter type
- Product number
- External Power Supply
- PCI power limit threshold (if available)
- Connected ioMemory devices

Block device information:

- Attach status
- Product name
- Product number
- Serial number
- PCIe slot number
- Firmware version
- Size of the device, out of total capacity
- Internal temperature (average and maximum, since ioMemory VSL 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
- Date of manufacture
- Power loss protection status
- PCIe bus voltage (avg, min, max)
- PCIe bus current (avg, max)
- PCIe power limit threshold (watts)
- PCIe slot available power (watts)
- PCIe negotiated link information (lanes and throughput)

Block device information:

- Part number
- Manufacturer's code
- Manufacturing date
- Vendor and sub-vendor information
- Size of the device, out of total capacity
- Format status and sector information (if device is attached)
- FPGA 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

Error Mode Information: If the ioMemory VSL 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.

\textbf{fio-sure-erase}

As a best practice, do not use this utility if there are any ioMemory devices 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. \textbf{It is not recoverable.}

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

After using \texttt{fio-sure-erase}, format the device using \texttt{fio-format} before using the device again.

If the device is in Read-only mode, perform a format using \texttt{fio-format} before running \texttt{fio-sure-erase}. If the device is in Minimal mode, then \texttt{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 support@fusionio.com for further assistance.

In order to run \texttt{fio-sure-erase}, the block device \textbf{must be detached}. See the \texttt{fio-detach} section for more information.

\textbf{Description}

The \texttt{fio-sure-erase} is a command-line utility that securely removes data from ioMemory devices. 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.

\textbf{Registry Requirement}

On Windows, a registry key must be created to configure the driver for ECC-bypass mode:
1. Locate the following key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fiodrive\Parameters
```

2. Create a DWORD key underneath it called "BypassECC" and set the value to "1".

3. Restart the computer before running the utility.

**Syntax**

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

Where `<device>` is the name given by the ioMemory VSL to your device. This name is `/dev/fctx`. For example, the name `/dev/fct4` refers to the ioMemory device installed in PCIe Bus 4 in your Windows system. Use `fio-status` to view this bus number.

**Products with Multiple Devices**

`fio-sure-erase` works on individual ioMemory devices. For example, if you are planning to purge an ioDrive Duo device, you will need to perform this operation on each of the product's two ioMemory devices.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-p</code></td>
<td>Purge instead of Clear: performs a write followed by an erase. For more information on Purge, see below.</td>
</tr>
<tr>
<td></td>
<td><strong>⚠️</strong> Purging the device may take hours to accomplish, depending on the size of the device that needs to be purged.</td>
</tr>
<tr>
<td><code>-y</code></td>
<td>No confirmation: do not require a yes/no response to execute the utility.</td>
</tr>
<tr>
<td><code>-q</code></td>
<td>Quiet: do not display the status bar.</td>
</tr>
</tbody>
</table>

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

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-trim-config**

**Description**

Enables or disables the TRIM feature (Windows service), which reclaims available space from the file system. It is generally recommended that TRIM be enabled for best performance. TRIM is enabled by default. Running this utility with no options displays the current TRIM status.

```
This utility affects all ioMemory devices in the system. You cannot enable or disable TRIM for only selected devices. Settings take place immediately, there is no need to reboot the system or the device.
```

**Syntax**

```
fio-trim-config [options]
```
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>Disable TRIM on this computer.</td>
</tr>
<tr>
<td>-e</td>
<td>(Default) Enable TRIM on this computer.</td>
</tr>
</tbody>
</table>

---

**fio-update-iodrive**

⚠️ You should back up the data on the ioMemory device prior to any upgrade as a precaution.

**Description**

Updates the ioMemory device's firmware. This utility scans the PCIe bus for all ioMemory devices and updates them. A progress bar and percentage are shown for each device as the update completes.

⚠️ 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.

⚠️ Note that when running multiple firmware upgrades in sequence, it is critical to load the driver after each firmware upgrade step. Otherwise the on-drive format will not be changed, and there will be data loss.

⚠️ Do not use this utility to downgrade the ioMemory device to an earlier version of the firmware. Doing so may result in data loss and void your warranty. Contact customer support at [http://ts.fujitsu.com/support/](http://ts.fujitsu.com/support/) if you need to downgrade your firmware.

⚠️ The default action (without using the -d option) is to upgrade all ioMemory devices with the firmware contained in the `<iodrive_version.fff>` 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.

⚠️ You must detach all ioMemory devices before updating the firmware.

⚠️ **Upgrade Path**

There is a specific upgrade path that you must take when upgrading ioMemory device. Consult the Release Notes for this ioMemory VSL release before upgrading ioMemory devices.

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

```
fio-update-iodrive [options] <iodrive_version.fff>
```

where `<iodrive_version.fff>` is the path and firmware archive file provided by Fusion-io. The default path is `C:\Program Files\Fusion-io ioMemory VSL\Firmware`. This parameter is required.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Use the <code>-d</code> option with care, as updating the wrong ioMemory device could damage your device.</td>
</tr>
<tr>
<td>-f</td>
<td>Force upgrade (used primarily to downgrade to an earlier firmware version). If the ioMemory VSL is not loaded, this option also requires the <code>-d</code> option.</td>
</tr>
<tr>
<td></td>
<td>Use the <code>-f</code> option with care, as it could damage your card.</td>
</tr>
<tr>
<td>-l</td>
<td>List the firmware available in the archive.</td>
</tr>
<tr>
<td>-c</td>
<td>Clears locks placed on a device.</td>
</tr>
<tr>
<td>-p</td>
<td>Pretend: Shows what updates would be done. However, the actual firmware is not modified.</td>
</tr>
<tr>
<td>-q</td>
<td>Runs the update process without displaying the progress bar or percentage.</td>
</tr>
<tr>
<td>-y</td>
<td>Confirm all warning messages.</td>
</tr>
</tbody>
</table>

All three external LED indicators light up during the update process.
Appendix D - TRIM Support

Introduction

TRIM is used to address a unique property of solid-state devices. The problem stems from the fact that when a user deletes a file, the device that contains the file does not recognize that it can reclaim that space. Instead, the device assumes the data is valid until the system informs the device it can overwrite that data. This is fine for a normal hard device, because you can continually write to the same sector of the device without significant degradation. A solid-state device, on the other hand, writes to different flash memory areas when the same logical sector is written to.

Because of the way flash memory handles writes, having a full SSD can cause significant reduction in write throughput. Currently available storage stacks and file systems were not designed with this caveat in mind. The SSD software must assume that all sectors contain valid data, and therefore always treat the device as full. In reality, a file system does not normally contain a full device's worth of data.

TRIM is an enhancement to existing file systems that informs the SSD software of logical sectors that do not contain valid user data. TRIM retrieves this information from a file system when a file has been deleted, and it informs the device that it can reclaim the space held by that file. This allows the wear-leveling software to reclaim that space as reserve to handle future write operations.

In order for TRIM to do this, it continually runs in the background and monitors the file system(s) being used on Fusion-io devices. The process is meant to be "lazy" so as not to noticeably impact the performance of the computer.

Platforms

The TRIM feature is available on ioMemory devices for Windows operating systems starting with Windows Server 2003.

TRIM capabilities have been built into Windows 7 and Windows Server 2008 R2 operating systems. ioMemory devices support Windows 7 TRIM, which means that Fusion-io TRIM is unnecessary under Windows Server 2008 R2, and it will not run when this operating system is present. No system changes are needed. The Fusion-io TRIM service will automatically shut off if it detects the Windows Server 2008 R2 operating system.

Using the TRIM Service

TRIM runs quietly in the background as a Windows Service a few minutes after the computer is booted, so as not to slow down the boot process. However, you can fine-tune TRIM by using the configurable settings described below.
Starting and Stopping TRIM

To start or stop the TRIM service through the Windows Services Manager,

1. Search for "Fusion-io Trim Service".
2. Click "Start service" or "Stop service".

Note that stopping the service only stops it for that session. Restarting the computer causes TRIM to run again.

Enabling TRIM

To enable the Fusion-io TRIM service,

1. Open a registry editor.
2. Navigate to

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fiodrive\Parameters

3. Add the TrimEnabled value as a DWORD (see the description below). Stop the TRIM service (by either of the two previously mentioned methods) and restart it; or, restart the computer.

   The fio-trim-config utility will set the TrimEnabled flag for you and does not require a reboot of your system or a manual restart of the service for the changes to take effect. This utility is the preferred method of enabling and disabling TRIM as it prevents the user from having to edit the registry themselves which has the potential to produce undesired consequences.

TrimEnabled

This registry value controls whether TRIM will run, regardless of the user starting and stopping TRIM through either of the two previously mentioned methods. If this value is present and set to 1, then TRIM will run as normal (unless you have stopped the service using the previously mentioned methods). If this value is set to 0 before attempting to start the Fusion-io TRIM service, the service will not run until the value is changed to 1 (or removed from the registry) and the service manually restarted. If this value is not present, TRIM assumes it should run unless you stop it. Note that this value persists across sessions, so if you restart the computer and set this value to 0, TRIM will not run when the computer starts again.

Controlling TRIM Aggressiveness

You can control how aggressive the TRIM service is. The default settings for TRIM minimize impact on computer performance while allowing the ioMemory device(s) to perform at their best. However, if a drive is being used with a write-intensive application, then you may want TRIM to be more aggressive. To do this, you'll need to add a few registry entries.
1. Open a registry editor.

2. Navigate to

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fiodrive\Parameters
```

3. Add the TrimTimeDelayInSeconds value as a DWORD (see the description below).

4. Add the BitmapRetrievalDelayInMilliSeconds value as a DWORD (see the description below).

5. Stop the TRIM service (by either of the two previously mentioned methods) and restart it; or, restart the computer.

**TrimTimeDelayInSeconds**

This registry value controls how quickly TRIM will start running again after it has finished processing all the volumes on a system. For example, if this is set to 600 (10 minutes in seconds), then TRIM will process all the volumes on a system, sleep for 10 minutes, and then process the volumes again and continue this cycle. This value can be any number between zero and 4,294,967,295. If this value is not set or present in the registry, then TRIM assumes a default of 900 seconds (or 15 minutes).

**BitmapRetrievalDelayInMilliSeconds**

This value most affects how much impact TRIM will have on the CPU. It currently defaults to 25 milliseconds but can be any number between zero (most aggressive) and 4,294,967,295 (pretty much useless).

**Configurations**

TRIM can be used with the following configurations and features:

- RAID volumes (mirrored, spanned, or striped). Note: RAID 5 (striped with parity) is not currently supported.
- Simple volumes (no RAID)
- Any combination of the above RAID levels across multiple devices, so long as there is at least one ioMemory device in the RAID set.
- Multiple partitions on the same ioMemory device
- NTFS and FAT32 file systems
- Volumes with mount points (no <drive letter> is defined)
- Compressed volumes
- Different cluster sizes, packet sizes, and sector sizes
- Extended and shrunk volumes
Appendix E- Monitoring the Health of ioMemory Devices

This section describes how the health of ioMemory devices can be measured and monitored in order to safeguard data and prolong device lifetime.

NAND Flash and Component Failure

An ioMemory device is a highly fault-tolerant storage subsystem that provides 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 properly selecting NAND flash media for the hosted application and proactively monitoring device age and health, you can ensure reliable performance over the intended product life.

Health Metrics

The ioMemory VSL manages block retirement using pre-determined retirement thresholds. The ioSphere 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 the 'monitoring' section below 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 ioMemory device 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 ioMemory device 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 ioMemory device 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 wearout.

For service or warranty-related questions, contact the company from which you purchased the device.

For products with multiple ioMemory devices, these modes are maintained independently for each device.

Health Monitoring Techniques

fio-status: Output from the fio-status utility shows the health percentage and device state. These items are referenced as "Media status" in the sample output below.

```
Found 1 ioDrive in this system
Fusion-io driver version: 2.2.3 build 240
Adapter: ioDrive
Fusion-io ioDrive 160GB, Product Number:FS1-002-161-ES

Media status: Healthy; Reserves: 100.00%, warn at 10.00%; Data: 99.12%
Lifetime data volumes:
    Physical bytes written: 6,423,563,326,064
    Physical bytes read   : 5,509,006,756,312
```

ioSphere: In the Device Report tab, look for the Reserve Space percentage in the right column. The higher the percentage, the healthier the drive is likely to be.

SNMP/SMI-S: On Windows or Linux, see the corresponding appendix for details on how to configure SNMP or SMI-S health indicators.

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

- Healthy
- Low metadata
- Read-only
- Reduced-write
- Unknown
Software RAID and Health Monitoring

Software RAID stacks are typically designed to detect and mitigate the failure modes of traditional storage media. The ioMemory device attempts to fail as gracefully as possible, and these new failure mechanisms are compatible with existing software RAID stacks. An ioMemory device 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 F- Using Windows Page Files with the ioMemory Devices

Introduction

This appendix describes how to effectively use paging (swap) files on ioMemory devices with Windows.

Using a page file with a traditional disk drive places practical limits on the usable size of the page file and virtual memory, due to the poor performance of disk drives in relation to RAM. Placing the OS paging file on one or more ioMemory devices allows much larger page files and usable virtual memory. This is due to the much faster response times and bandwidth on ioMemory devices versus hard disks.

Configuring Device Paging Support

The ioMemory VSL can be configured to support paging files on one or more ioMemory devices. This requires that each ioDrive used with a paging file pre-allocates the worst-case amount of memory it may need in any possible I/O scenario. This is done on a per-adapter (ioDIMM) instance.

Because of the extra host RAM memory use, paging should be enabled only on ioMemory devices that will actually hold a paging file. It is possible to place a single paging file on more than one ioMemory device. In this case Windows will stripe paging I/O across all available paging files, possibly providing additional performance to the Virtual Memory (VM) subsystem.

ioMemory VSL RAM Consumption

The amount of RAM pre-allocated per ioMemory device depends on the device's total size and the sector (block) size selected when formatting the drive (with fio-format).

Consult the Release Notes for this version of the software for RAM usage per GB of ioMemory device.

Using a larger sector size significantly reduces the amount of host memory consumption needed for paging support. It is recommended that a 4K sector size be used because a) that is generally the natural size of a host memory page, and b) it minimizes overall host memory consumption. In Windows, NTFS will generally use a cluster size of 4K, so formatting to 512 is not useful except for applications that compatible only with 512-byte sector sizes (such as Windows XP and Windows 2003).
The indicated amount is needed per ioMemory device that supports paging. You must carefully plan which ioMemory device(s) will be used to hold a paging file.

**Non-paged Memory Pool**

Pre-allocated memory for the ioMemory device comes from the Windows kernel non-paged memory pool. This pool dynamically grows as system components consume additional kernel memory. The maximum size of this pool is restricted as follows:

- Server 2003, 2008 R1/R2 - 75% of RAM up to a maximum of 128GB.
- Vista/Windows 7 - 40% of RAM up to a maximum of 128GB.

The amount of in-use, non-paged pool memory should be noted when planning page file usage. This is because the ioMemory device pre-allocates RAM, and that reduces the available physical non-paged memory. The ioMemory VSL will fail to load if the total pre-allocated memory plus the in-use, non-paged memory exceeds the maximum non-paged memory pool.

To determine the total non-paged memory pool use for two ioMemory devices, let's use the following example:

- One ioMemory device that requires 850 MB of RAM, and the other requires 1700 MB or RAM.

  Consult the Release Notes for this version of the ioMemory VSL for RAM requirements.

- Both are formatted with a 4K sector size
- Both will support paging files

The current allocated non-paged pool is obtained from Task Manager and, in this example, has a value of 576 MiB. (Values shown in Task Manager are in MiB \[1024 \times 1024 = 1 \text{ MiB}\]). The total RAM on the system is 8000 MB and the OS is Server 2008 R2.

First, covert the 576 MiB into MB: \[576 \text{ MiB} \times \left(\frac{1 \text{ MB}}{1.048576 \text{ MiB}}\right) = \sim 549 \text{ MB}\]

To calculate the total available non-paged pool, use the following formula:

\[(8000 \text{ MB} \times 0.75) - 549 - 850 - 1700\]

which still leaves 2901 MB available for the non-paged pool.

**Enabling/Disabling Paging Support**

Memory pre-allocation occurs during ioMemory VSL initialization. To enable paging support, you must enable the \texttt{FIO\_PREALLOCATE\_MEMORY} configuration item. This can be done using the \texttt{fio-config} command-line utility. This parameter is assigned a string with a list of decimal serial numbers of the ioMemory devices that will support a paging file. The ioMemory VSL performs memory pre-allocation for those instances.
Below is an example of using the fio-config utility to enable paging and pre-allocation on two ioMemory devices with serial numbers 1234 and 17834. Serial number information can be obtained using the fio-status utility.

```bash
fio-config -p FIO_PREALLOCATE_MEMORY "1234,17834"
```

To disable paging support on all devices, use a value of 0 for FIO_PREALLOCATE_MEMORY:

```bash
fio-config -p FIO_PREALLOCATE_MEMORY "0"
```

To query the current value, run this command:

```bash
fio-config -g FIO_PREALLOCATE_MEMORY
```

An alternate method to manage (enable or disable) paging support is to use ioManager.

You must reload the ioMemory VSL for the new pre-allocation setting to take effect. Typically this can be done by restarting the machine or using disable/enable within Device Manager for each ioMemory device instance.

Also, using the Windows System Properties to change paging file configuration requires a system restart before the properties are applied. Therefore, you can change both FIO_PREALLOCATE_MEMORY and the system page file configuration and then apply both with a single restart.

## Windows Page File Management

By default, the ioMemory VSL disables support for page files. The previous section described how to enable support for page files on one or more ioMemory devices. The following describes how to work with the built-in Windows control panels to configure and set up paging files on ioMemory devices.

### Setting Up Paging File(s)

To set up page files in Windows,

1. Go to **Control Panel** and double-click System.
2. Click **Advanced system settings** from the Task pane.
3. On the Advanced tab, click **Settings**. The Performance Options dialog opens.
4. On the Advanced tab, click **Change**. The Virtual Memory dialog opens.

![Virtual Memory Dialog](image)

Using this dialog, you can configure a page file for each available drive in the system. Selecting the "Automatically manage paging file size for all drives" checkbox causes Windows to create a single page file on the system drive, which is the drive the OS is started from. This checkbox should be cleared when using an ioMemory device with a paging file.

Windows supports up to 16 distinct paging files. To enable a page file on an ioMemory device,

1. Choose the ioMemory device from the device list.
2. Select the **Custom size** radio button.
3. Provide values in the **Initial size** and **Maximum size** fields.
4. Click **Set** to save the setting. Do not omit this step, or your changes will be lost.
5. Click **OK**.
6. When prompted to restart, click **Yes**. This is necessary for the new page file settings to take effect.

To remove a paging file on the drive, follow the steps earlier but select **No paging file**. For performance reasons,
typically you will remove all paging files on any system hard disk.

The Virtual Memory dialog allows page files to be configured on available ioMemory devices, even if the ioMemory device has not been configured to support a page file. Even though the dialog allows enabling of the page file, following the required restart you'll notice that no page file was created on the device. Follow the directions earlier in this document to properly enable page file support on one or more ioMemory devices.

### System Drive Paging File Configuration

By default Windows creates and manages a page file on the system boot drive (typically a hard disk), which is typically where Windows is installed. Keeping a regular page file on the system hard disk is generally not optimal, because the hard disk's I/O performance is many orders of magnitude slower than an ioMemory device. To remedy this, you can eliminate or minimize the size of the system boot drive page file, as explained later. Enabling page files on ioMemory devices (but not the system drive) improves Virtual Memory (VM) subsystem performance, as the VM manager stripes I/O across all available page files. Additionally, the ioMemory devices act as a very large memory store, which can greatly improve memory usage for large applications.

The Windows kernel uses the system disk page file to store crash dumps. Crash dumps may be small (mini-dumps) or large (full-kernel memory dumps). Typically, running without dump file support or with a small dump file is adequate. There are several possible system drive page file configurations:

1. Eliminate all page files on any hard disks, including the system boot drive. Although this maximizes paging I/O on ioMemory devices, no post-mortem crash dump file will be available if a system crash occurs. However, it may be possible to re-enable a page file on the system drive and then reproduce the crash scenario.

2. Create a minimal-size page file on the system boot drive. The recommended minimum size is 16MB, although Windows may warn that a minimum 400MB page file is needed.

3. Create a page file large enough for a full-kernel memory dump. This typically requires a page file at least the size of installed RAM, with some recommending the size equal to RAM x 1.5.

Fusion-io is actively working on adding support for an ioMemory device being a crash dump target.

To view or change the crash dump configuration,

1. Go to the **System Properties** dialog.
2. Click the Advanced tab.
3. In the Startup and Recovery section, click **Settings**. The Startup and Recovery dialog opens.

   ![Startup and Recovery dialog](image)

   In the System Failure section you can change settings to handle the system log, restart, and debugging information.

   **Guaranteeing Minimum Committable Memory**

   If you enable "System managed size" or set a "Custom size" in the Virtual Memory dialog, you should do so with care. If the initial size is less than the desired amount of committable virtual memory, this can cause an application to have memory allocation failures if the amount of committed memory exceeds the currently allocated page file size or the initial size value. When committed memory exceeds the current page file size, a request to allocate additional memory will fail. The Windows Virtual Memory manager will slowly increase the size of the paging file up to the available size of its drive or to the "Maximum size" custom setting, whichever is smaller.

   If you want to use a large amount of committed virtual memory (more than 1.5 times the amount of RAM) and avoid application memory allocation errors, the initial and maximum committed memory should be explicitly set for the expected application committed memory usage. These values should generally be the same.

   **How Large Should You Make the Paging File?**
The following articles explain in great detail how to size the page file appropriately.

- Main Article Link: Pushing the Limits of Windows
- Specific section that documents virtual memory: Pushing the Limits of Windows: Virtual Memory

Verifying Page File Operation

To verify that a page file is actively placed on an ioMemory device, you can browse for hidden files at the drive's root. For example, run the following command at a prompt:

```bash
dir c: /ah
```

In the output listing there should be a file called `pagefile.sys`. If no page file is present, then recheck the page file configuration in the Virtual Memory dialog and verify that page file support has been enabled on the queried ioMemory device.

Performance

Using the ioMemory device as the paging store can improve overall Virtual Memory system performance. Actual benefits will vary widely with an application's virtual memory usage and with hardware platform/performance.
Appendix G- SNMP Test Mode and MIB Support

Introduction

This document explains how you can set up a test mode with your VSL Windows SNMP agent. This enables you to set test values in a Windows registry and force SNMP traps without having to create the actual conditions on the device.

For example, you can use the SNMP test mode to change the non-writeable indicator and generate a trap, or simulate a change to the physical or logical size of the device, etc.

To use SNMP Test Mode, you must have installed the SNMP option with your ioMemory VSL.

Using Test-Mode Registry Values

The picture below shows the registry entries included for SNMP test values.
Each of these entries is described below. Entries marked by an asterisk (*) generate SNMP traps when set to the indicated values, and the fusionIoDimmMIBCondition and fusionIoDimmInfoStatus MIB variables may be affected because of the changes.

All entries, except those marked by **, reflect your registry changes immediately. Entries marked by ** require a restart of the Windows SNMP agent for the changes to take effect.

<table>
<thead>
<tr>
<th>SNMP Test Registry Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ioDimmExtnAvailLogCapacityL</td>
<td>Lower word of the available logical capacity in bytes</td>
</tr>
<tr>
<td>ioDimmExtnAvailLogCapacityU</td>
<td>Upper word of the available logical capacity in bytes</td>
</tr>
<tr>
<td>ioDimmExtnBytesReadL</td>
<td>Lower word of the total number of bytes read since the device was formatted</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>ioDimmExtnBytesReadU</code></td>
<td>Upper word of the total number of bytes read since the device was formatted</td>
</tr>
<tr>
<td><code>ioDimmExtnLogBytesWrittenL</code></td>
<td>Lower word of the number of user data bytes written</td>
</tr>
<tr>
<td><code>ioDimmExtnLogBytesWrittenU</code></td>
<td>Upper word of the number of user data bytes written</td>
</tr>
<tr>
<td><code>ioDimmExtnPhysBytesWrittenL</code></td>
<td>Lower word of the total physical bytes written</td>
</tr>
<tr>
<td><code>ioDimmExtnPhysBytesWrittenU</code></td>
<td>Upper word of the total physical bytes written</td>
</tr>
<tr>
<td><code>ioDimmExtnTotalLogCapacityL</code></td>
<td>Lower word of the total logical capacity in bytes as formatted</td>
</tr>
<tr>
<td><code>ioDimmExtnTotalLogCapacityU</code></td>
<td>Upper word of the total logical capacity in bytes as formatted</td>
</tr>
<tr>
<td><code>ioDimmExtnTotalPhysCapacityL</code></td>
<td>Lower word of the total logical capacity in bytes as formatted</td>
</tr>
<tr>
<td><code>ioDimmExtnTotalPhysCapacityU</code></td>
<td>Upper word of the total logical capacity in bytes as formatted</td>
</tr>
<tr>
<td><code>ioDimmExtnUsablePhysCapacityL</code></td>
<td>Lower word of the useable physical capacity in bytes. This is space that is holding valid data, or is erased and ready for writing, or is waiting to be reclaimed via garbage collection.</td>
</tr>
<tr>
<td><code>ioDimmExtnUsablePhysCapacityU</code></td>
<td>Upper word of the useable physical capacity in bytes. This is space that is holding valid data, or is erased and ready for writing, or is waiting to be reclaimed via garbage collection.</td>
</tr>
<tr>
<td>*<code>ioDimmInfoInternalTemp</code></td>
<td>Current internal temperature of the device in degrees Celsius. If this value is set above 78 degrees Celsius for ioDimm cards, a trap is generated. If set above 90 degrees for HP Mezzanine cards, a trap is generated.</td>
</tr>
<tr>
<td>*<code>ioDimmInfoFlashbackIndicator</code></td>
<td>1 = flashback redundancy is degraded; 2 = false</td>
</tr>
<tr>
<td>*<code>ioDimmInfoNonWritableIndicator</code></td>
<td>1 = device is no longer writable because it has surpassed the read-only threshold; 2 = false</td>
</tr>
<tr>
<td><code>ioDimmInfoPercentLifeRemaining</code></td>
<td>Upper word of the total logical capacity in bytes as formatted</td>
</tr>
<tr>
<td>*<code>ioDimmInfoState</code> (trap generated if state = 4)</td>
<td>Current state of the attached client device: unknown(0), detached(1), attached(2), minimal(3), error(4), detaching(5), attaching(6), scanning(7), formatting(8), updating(9), attach(10), detach(11), format(12), update(13)</td>
</tr>
<tr>
<td>*<code>ioDimmInfoWearoutIndicator</code></td>
<td>Boolean: True = device has surpassed the wearout threshold</td>
</tr>
<tr>
<td><code>ioDimmTestMode</code></td>
<td>Set test mode on or off</td>
</tr>
<tr>
<td><strong>ioDimmTestModeIndex</strong></td>
<td>Number indicating the selected Fusion-io mib</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><strong>mib</strong></td>
<td>Name of the MIB in use</td>
</tr>
<tr>
<td><strong>pathname</strong></td>
<td>Path to the driver, set at installation</td>
</tr>
<tr>
<td><strong>traps</strong></td>
<td>Set trap generation on or off</td>
</tr>
<tr>
<td><strong>updateDelay</strong></td>
<td>Number of milliseconds to wait until getting the next value from the ioMemory VSL to generate a trap</td>
</tr>
</tbody>
</table>

### SNMP MIB Support

The following SNMP MIB fields are supported in Windows:

<table>
<thead>
<tr>
<th>fusionIoDimmMibRevMajor</th>
<th>fusionIoDimmInfoAdapterType</th>
</tr>
</thead>
<tbody>
<tr>
<td>fusionIoDimmMibRevMinor</td>
<td>fusionIoDimmInfoAdapterPort</td>
</tr>
<tr>
<td>fusionIoDimmMIBCondition</td>
<td>fusionIoDimmInfoAdapterSerialNumber</td>
</tr>
<tr>
<td>fusionIoDimmInfoIndex</td>
<td>fusionIoDimmInfoAdapterExtPowerPresent</td>
</tr>
<tr>
<td>fusionIoDimmInfoStatus</td>
<td>fusionIoDimmInfoPowerlossProtectDisabled</td>
</tr>
<tr>
<td>fusionIoDimmInfoName</td>
<td>fusionIoDimmInfoInternalTempHigh</td>
</tr>
<tr>
<td>fusionIoDimmInfoSerialNumber</td>
<td>fusionIoDimmInfoAmbientTemp</td>
</tr>
<tr>
<td>fusionIoDimmInfoPartNumber</td>
<td>fusionIoDimmInfoPCIBandwidthCompatibility</td>
</tr>
<tr>
<td>fusionIoDimmInfoSubVendorPartNumber</td>
<td>fusionIoDimmInfoPCIPowerCompatibility</td>
</tr>
<tr>
<td>fusionIoDimmInfoSparePartNumber</td>
<td>fusionIoDimmInfoActualGoverningLevel</td>
</tr>
<tr>
<td>fusionIoDimmInfoAssemblyNumber</td>
<td>fusionIoDimmInfoLifespanGoverningLevel</td>
</tr>
<tr>
<td>fusionIoDimmInfoFirmwareVersion</td>
<td>fusionIoDimmInfoPowerGoverningLevel</td>
</tr>
<tr>
<td>fusionIoDimmInfoDriverVersion</td>
<td>fusionIoDimmInfoThermalGoverningLevel</td>
</tr>
<tr>
<td>fusionIoDimmInfoUID</td>
<td>fusionIoDimmInfoLifespanGoverningEnabled</td>
</tr>
<tr>
<td>fusionIoDimmInfoState</td>
<td>fusionIoDimmInfoLifespanGoverningTgtDate</td>
</tr>
<tr>
<td>fusionIoDimmInfoClientDeviceName</td>
<td>fusionIoDimmExtnIndex</td>
</tr>
<tr>
<td>fusionIoDimmInfoBeacon</td>
<td>fusionIoDimmExtnTotalPhysCapacityU</td>
</tr>
<tr>
<td>fusionIoDimmInfoPCIAddress</td>
<td>fusionIoDimmExtnTotalPhysCapacityL</td>
</tr>
<tr>
<td>fusionIoDimmInfoPCIDeviceID</td>
<td>fusionIoDimmExtnTotalLogCapacityU</td>
</tr>
<tr>
<td>fusionIoDimmInfoPCISubdeviceID</td>
<td>fusionIoDimmExtnTotalLogCapacityL</td>
</tr>
<tr>
<td>fusionIoDimmInfoPCIVendorID</td>
<td>fusionIoDimmExtnBytesReadU</td>
</tr>
<tr>
<td>fusionIoDimmInfoPCISubvendorID</td>
<td>fusionIoDimmExtnBytesReadL</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>fusionIoDimmInfoPCISlot</td>
<td>fusionIoDimmExtnBytesWrittenU</td>
</tr>
<tr>
<td>fusionIoDimmInfoWearoutIndicator</td>
<td>fusionIoDimmExtnBytesWrittenL</td>
</tr>
<tr>
<td>fusionIoDimmInfoFlashbackIndicator</td>
<td>fusionIoDimmExtnFormattedBlockSize</td>
</tr>
<tr>
<td>fusionIoDimmInfoWritableIndicator</td>
<td>fusionIoDimmExtnCurrentRAMUsageU</td>
</tr>
<tr>
<td>fusionIoDimmInfoInternalTemp</td>
<td>fusionIoDimmExtnCurrentRAMUsageL</td>
</tr>
<tr>
<td>fusionIoDimmInfoHealthPercentage</td>
<td>fusionIoDimmExtnPeakRAMUsageU</td>
</tr>
<tr>
<td>fusionIoDimmInfoMinimalModeReason</td>
<td>fusionIoDimmExtnPeakRAMUsageL</td>
</tr>
<tr>
<td>fusionIoDimmInfoReducedWriteReason</td>
<td>fusionIoDimmWearoutTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliVolts</td>
<td>fusionIoDimmNonWritableTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliVoltsPeak</td>
<td>fusionIoDimmFlashbackTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliVoltsMin</td>
<td>fusionIoDimmTempHighTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliWatts</td>
<td>fusionIoDimmTempOkTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliWattsPeak</td>
<td>fusionIoDimmErrorTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliAmps</td>
<td>fusionIoDimmPowerlossProtectTrap</td>
</tr>
<tr>
<td>fusionIoDimmInfoMilliAmpsPeak</td>
<td></td>
</tr>
</tbody>
</table>
Appendix H- SMI-S Interface

Introduction to the SMI-S Interface

The SMI-S interface is based on Web-Based Enterprise Management (WBEM) and provides a Common Information Model (CIM) model that represents the ioMemory device and associated software, in accordance with existing Distributed Management Task Force (DMTF) and Storage Networking Industry Association (SNIA) Storage Management Initiative Specification (SMI-S) standards. This model permits backward-compatible extension, accommodating new hardware and software features developed by Fusion-io.

It is assumed that the reader is versed in WBEM, SMI-S, and DMTF standards. This document and associated model may change at any time as feedback is received.

References

- CIM Schema v2.22
  http://www.dmtf.org/standards/cim/cim_schema_v2220

- DMTF DSP1011, Physical Asset Profile
  http://www.dmtf.org/standards/published_documents/DSP1011_1.0.2.pdf

- DMTF DSP1023, Software Inventory Profile
  http://www.dmtf.org/standards/published_documents/DSP1023_1.0.1.pdf

- DMTF DSP1033, Profile Registration Profile
  http://www.dmtf.org/standards/published_documents/DSP1033_1.0.0.pdf

- DMTF DSP1075 PCI Device Profile
  http://www.dmtf.org/standards/published_documents/DSP1075_1.0.0.pdf

- DMTF DSP1002, Diagnostics Profile
  http://www.dmtf.org/standards/published_documents/DSP1002_2.0.0.pdf

- SMI-S v1.4 Architecture

- SMI-S v1.4 Common Profiles
SMI-S v1.4 Host Profiles  

SMI-S v1.4 Common Diagnostic Model  
http://www.dmtf.org/standards/mgmt/cdm/

Installing the SMI-S WMI Provider on Windows

To install the Fusion-io SMI-S WMI provider on Windows,

1. Go to Control Panel > Add & Remove Programs.

2. Right-click Management and Monitoring Tools and select Details. Make sure the WMI Windows Installer Provider is selected.

The SMI-S WMI provider for ioMemory devices will be installed and the WMI service will be restarted automatically.

Expected Warning Message

When you install the WMI provider, a warning will appear in the Windows event log with the following description:

A provider, fio-smis-wmi, has been registered in the Windows Management Instrumentation namespace root\fio to use the LocalSystem account. This account is privileged and the provider may cause a security violation if it does not correctly impersonate user requests.

This warning is expected. The WMI provider only interfaces with the ioMemory VSL and does not modify system data.

Verifying SMI-S Installation on Windows

To verify the Fusion-io SMI-S WMI provider on Windows,
1. Run the wbemtest.exe program. The WMI Tester window appears.

![WMI Tester Window]

2. Click Connect to display the Connect dialog. The CIM provider namespace is `root\fio`
3. Type the namespace value shown in the screenshot above and click Connect.

The WMI Tester window appears, with the namespace value filled in.

![WMI Tester Window](image)

4. Click Enum Instances (second button on the first row) to bring up the Class Info dialog.

![Class Info Dialog](image)
5. Type FIO_IoMemoryPort as shown above and then click OK.

If the provider is installed correctly, the result will look like the following example, with an entry for each ioMemory device in the system:
6. Double-click an entry to bring up detailed information, such as in this example:

![Object editor for FIO IoMemoryPort CreationClassName="FIO_IoMemoryPort", Description=...]

**Manual Registration**

If the automatic installation fails to register the provider, follow these steps to manually register it:

1. Stop the WMI (`winmgmt`) service via the services tool or the following command line:

   ```
   net stop winmgmt
   ```

2. Browse to the `Fusion-io\SMIS\cim-schema` directory using the command-line interface and run the following:

   ```
   mofcomp fio-reg-wmi.mof
   ```

3. Browse to `Fusion-io\SMIS\WMI` directory
4. Un-register and re-register the fio-smis-wmi.dll using the following commands:

```plaintext
regsvr32 /u fio-smis-wmi.dll
regsvr32 fio-smis-wmi.dll
```

5. Start the winmgmt service via the services tool or the following command line:

```plaintext
net start winmgmt
```

**Description**

SMI-S is a collection of specifications that traditionally focus on Storage Area Network (SAN) systems based on the SCSI command set, such as Fibre Channel, iSCSI, and SAS. However, the general pattern used to model these storage systems can be applied to solid-state, direct-attached storage systems such as those provided by Fusion-io.

IoMemory devices are modeled using the SMI-S patterns established in the Storage HBA, Direct Attached (DA) Ports, and Host Discovered Resources Profiles. The physical aspects of the IoMemory device and all firmware and IoMemory VSL software are modeled using published DMTF specifications, including the Physical Asset, Software Inventory, PCI Device Profiles, and Common Diagnostic Model Profile.

The following chart describes the Fusion-Io SMI-S CIM model, with IoMemory devices and their associated firmware and software. For simplicity, the prefix FIO_ has been removed from the class names.
A: IOMemoryPort Class

The central instance of the model is of the IOMemoryPort class (A in the figure), a logical representation of the ioMemory device. It supports the extrinsic methods necessary to provision the drive. An instance of PCIDevice (B) and IOMemoryPort exist for each installed ioMemory device, and they are associated with instances of ConcreteIdentity (1). An instance of SSDStatistics (C), which contains important performance and capacity data for the device, is associated by an ElementStatisticalData association (2) to each IOMemoryPort. IOMemoryPort is scoped by an instance of the ComputerSystem class. The SystemDevice (3) aggregation aggregates IOMemoryPort within the containing ComputerSystem.

E: IOMemoryPortController Class

An instance of IOMemoryPortController (E) represents the ioMemory VSL used to control the installed ioMemory devices. IOMemoryPortController specializes CIM_PortController, and it aggregates IOMemoryPort with the ControlledBy (4) aggregation. The software version and vendor information are represented by the SoftwareIdentity (F) instance that is associated to IOMemoryPortController (E) via ElementSoftwareIdentity (5). The SoftwareIdentity that represents the installed ioMemory VSL software is associated to the scoping ComputerSystem using the InstalledSoftwareIdentity association (6).

An instance of the ProtocolEndpoint class (G) represents both ends of the logical data path between the IOMemoryPort and the solid-state storage. This aspect of the model is derived from the pattern in the DA Ports Profile, where the port is both an initiator and target. ProtocolEndpoint is associated to the IOMemoryPort by DeviceSAPImplementation (7) and to the ComputerSystem by HostedAccessPoint (8).
H: LogicalSSD Class (Block Device)

The block device exposed to applications (file systems, database, and logical volume manager) is modeled using an instance of LogicalSSD (H), a subclass of CIM_DiskDrive. It is associated with a StorageExtent (J) using the MediaPresent association (9), but the StorageExtent will always be present. It is also associated to the ProtocolEndpoint (G) representing the IOMemoryPort using SAPAvailableForElement (10) and to the scoping ComputerSystem using SystemDevice (3).

ioMemory devices, being PCIe devices, are also represented by an instance of the PCIDevice class (B). IOMemoryPort is an alternate representation of the PCIDevice and its associated control device. It is associated to it by the ConcreteIdentity association.

K: SoftwareIdentity

The ioMemory VSL software is also represented with SoftwareIdentity, which is associated to the PCIDevice by the ElementSoftwareIdentity association (11). The SoftwareIdentity (firmware) is associated to the scoping ComputerSystem by the InstalledSoftwareIdentity association (12). An instance of SoftwareInstallationService (L) is associated with each PCIDevice, which can be used to update device firmware.

M: Physical Aspects

The physical aspects of ioMemory devices are represented by an instance of the PhysicalPackage class (M), which is associated to the PCIDevice by Realizes (13) and to the scoping ComputerSystem by SystemPackaging (14). The temperature sensors on ioMemory devices are represented by an instance of TemperatureSensor (N) and is associated to the PhysicalPackage by AssociatedSensor.

Implementation

This section describes the arrangement of instances and associations for the Fusion-io device CIM model. Not all class properties are described in detail. Consult the CIM schema for detailed description of all properties.

A WBEM CIM provider based on this model will be developed in the future. Fusion-io intends to support popular CIMOMs, including OpenPegasus, OpenWBEM, SFCB, and Windows WMI.

The device health is indicated by the value of the HealthLevel property. Values include: Healthy, Warning, Reduced Write, and Read Only. These values are mapped to standardHealthState values – OK, Degraded/Warning, and Critical Failure – as appropriate.

Extrinsic methods for device provisioning include attach, detach, format, and update. The attach method creates a block device for the ioMemory device. Detach disables the block device. A format option enables users to specify the device size in either megabytes or a percentage. The update method allows users to upgrade the firmware on the device.

Device longevity is indicated by the value of the HealthPercentage property. FlashbackAvailability indicates whether or not this feature of the ioMemory device is online.
IOMemoryPorts are aggregated by IOMemoryPortController via the ControlledBy aggregation. Instances of IOMemoryPort are associated to their corresponding PCIDevice with the ConcreteIdentity association. The IOMemoryPort is a logical device of the scoping ComputerSystem and is indicated as such by the SystemDevice aggregation.

Products with two or more ioMemory devices, such as the ioDrive Duo device do appear like two separate ioMemory devices. For products with multiple devices, the IOMemoryPort class is extended to include information about the carrier card type, serial number, and external power connection for the product as a whole.

**IOMemoryPort**

One instance of IOMemoryPort exists for each ioMemory device installed in the ComputerSystem.

The LocationIndicator property reflects the state of the device indicator beacon (e.g., all LEDs on solid). Reading the value gives the current state of the indicator. Writing the value with "On" or "Off" turns the indicator on or off and can be used to determine the device's physical location.

**SSDStatistics**

One instance of SSDStatistics exists for each IOMemoryPort instance. Properties of this object provide performance and capacity information. Some of this information is only available when the drive is attached (i.e., the state of the associated IOMemoryPort is "Attached").

**IOMemoryPortController**

Only one instance of IOMemoryPortController exists, representing the ioMemory VSL software used to control IOMemoryPorts. The IOMemoryPortController specializes the CIM_PortController.

IOMemoryPortController is aggregated to the scoping ComputerSystem using the SystemDevice aggregation. IOMemoryPortController is associated with a SoftwareInventory instance representing the ioMemory VSL software properties via the ElementSoftwareIdentity association.

**ProtocolEndpoint**

One instance of ProtocolEndpoint exists for each instance of IOMemoryPort. It is associated to the IOMemoryPort using DeviceSAPImplementation and to LogicalSSD using SAPAvailableForElement. Because an IOMemoryPort represents both the initiator and target ports, only one ProtocolEndpoint per IOMemoryPort is needed to model the connection between IOMemoryPort and LogicalSSD.

**LogicalSSD**

One instance of LogicalSSD, a subclass of CIM_DiskDrive, exists for each block device (/dev/fioX) exposed by an ioMemory device. Correlatable IDs are used, based on operating system device names. This enables client applications to associate block devices discovered through this model with resources discovered from other SMI-S models instrumented on the host system.

ComputerSystem aggregates LogicalSSDs via SystemDevice. The LogicalSSD instances are associated to their ProtocolEndpoints via SAPAvailableForElement. If the IOMemoryPort associated to the endpoint is not
attached, then the Availability property is set to "Off Line," and the DeviceID property value is "Unknown."

**StorageExtent**

One instance of StorageExtent is associated with each LogicalSSD and represents the logical storage of the associated device.

**SoftwareIdentity**

One instance of SoftwareIdentity exists to represent the ioMemory VSL software. The firmware is also modeled using SoftwareIdentity but requires an instance for each ioDrive installed. The IsEntity property has a value of True, indicating that the SoftwareIdentity instance corresponds to a discrete copy of the ioMemory VSL software or firmware. The MajorVersion, MinorVersion, RevisionNumber, and BuildNumber properties convey the driver/firmware version information. The Manufacturer property can be used to identify Fusion-io.

Another option for the firmware is to omit the InstalledSoftwareIdentity association with ComputerSystem, because the firmware is not really installed on ComputerSystem. This option would depend on how users want to model the firmware.

**SoftwareInstallationService**

An instance of SoftwareInstallationService exists for each PCIDevice and can be used to update the associated device's firmware.

**PCIDevice**

An instance of PCIDevice is instantiated for each ioMemory device (PCIe card) in the computer. Properties are set as follows:

- **BusNumber** – bus number where the PCIe device exists
- **DeviceNumber** – device number assigned to the PCI device for this bus.
- **FunctionNumber** – set to the function number for the PCI device.
- **SubsystemID, SubsystemVendorID, PCIDeviceID, VendorID, and RevisionID** are optional but can be populated if values can be extracted from the configuration registers of the PCI device.

PCIDevice is associated with IOMemoryPort, its alternate logical representation, using ConcreteIdentity. The PCIDevice is also associated with PhysicalPackage, representing the physical aspects of the ioDrive, via Realizes.

**PhysicalPackage**

One instance of PhysicalPackage exists for each discrete, physical ioDrive installed in the computer system. The Manufacturer, Model, SKU, SerialNumber, Version, and PartNumber properties can be used to describe these aspects of the physical card. PhysicalPackage is associated with PCIDevice via Realizes and the scoping ComputerSystem via SystemPackaging.
TemperatureSensor

One instance of TemperatureSensor exists for each PhysicalPackage. Temperature information for the drive is stored in the properties of this object.

Diagnostic Test

One instance of DiagnosticTest will exist. The RunDiagnostic() method will trigger a snapshot of device status for the specified ManagedElement which must be an instance of IoMemoryPort. The diagnostic run is synchronous and runs instantaneously. The resulting ConcreteJob object will associate to the originating DiagnosticTest instance and the respective IoMemoryPort instance that was specified (see Figure 2). At this time, RunDiagnostic() can only be used with the default DiagnosticSettingData provided.

Each run will add a single entry of DiagnosticSettingDataRecord and associated DiagnosticCompletionRecord in the DiagnosticLog. The RecordData property of the DiagnosticCompletionRecord will record critical device status at the time of the run. The format of the RecordData string can be found in the RecordFormat property.

The format is a series of status strings, each of which can hold one of the following values delimited by an asterisk (*) character: "Unknown", "OK", "Warning", or "Error". Currently, seven status values are recorded: WearoutStatus, WritabilityStatus, FlashbackStatus, TemperatureStatus, MinimalModeStatus, PciStatus and InternalErrorStatus. All of these should report "OK" under normal operating conditions.

WearoutStatus will be set to "Warning" when less than 10% reserve space is left on the device. It will be set to "Error" when there is no more reserved space.

- WritabilityStatus will be set to "Error" whenever the device is write throttling or in read-only mode. This can happen due to a variety of conditions including device wearout and insufficient power.
- FlashbackStatus will report "Warning" if a catastrophic error causes Flashback protection to be degraded.
- TemperatureStatus will report "Warning" when the device temperature is nearing the maximum safe temperature and "Error" when the maximum safe temperature is reached or surpassed.
- MinimalModeStatus will report either "Warning" or "Error" whenever the device is in minimal mode.
- PciStatus will report "Warning" or "Error" if there are compatibility problems with the host PCIe bus.
- InternalErrorStatus will report "Error" if there are any internal problems with the ioMemory VSL.

The CompletionState property will summarize the results and may be set to Unknown, OK, Warning or Failed. If any status is in error the state will report as Failed. Otherwise, if there is any warning status the state will report Warning. The Message property will be set to indicate the appropriate action if there are any warnings or errors.

DiagnosticSetting Data

There will be an instance of DiagnosticSettingData associated with the DiagnosticTest instance (see Figure 2). It records the default settings for each call to RunDiagnostic.
DiagnosticServiceCapabilities

There is an instance of DiagnosticServiceCapabilities associated with the DiagnosticTest instance which records the capabilities of the DiagnosticTest service.

DiagnosticLog

An instance of DiagnosticLog is associated with the DiagnosticTest instance and will store the results of each run.

DiagnosticSettingRecord

A copy of the default DiagnosticSettingData will be stored in a DiagnosticSettingDataRecord each time a diagnostic is run and will be associated with an instance of DiagnosticCompletionRecord.

DiagnosticCompletionRecord

An instance of DiagnosticCompletionRecord will store the results of each RunDiagnostic execution. The details are explained in DiagnosticTest.

RegisteredDiskDriveLiteProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Disk Drive Lite Profile. The following properties are set as follows:

- InstanceID – set to "SNIA:DiskDriveLiteProfile-1.4.0"
- RegisteredOrganization – set to "11" (SNIA)
- RegisteredName – set to "DirectAccess Ports Profile"
- RegisteredVersion – set to "1.4.0"

RegisteredDAPortsProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the DA Ports Profile. The properties are set as follows:

- InstanceID – set to "SNIA:DAPortsProfile-1.4.0"
- RegisteredOrganization – set to "11" (SNIA)
- RegisteredName – set to "DirectAccess Ports Profile"
- RegisteredVersion – set to "1.4.0"
RegisteredStorageHBAProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Storage HBA Profile. The properties are set as follows:

- InstanceID – set to "SNIA:StorageHBAProfile-1.4.0"
- RegisteredOrganization – set to "11" (SNIA)
- RegisteredName – set to "Storage HBA Profile"
- RegisteredVersion – set to "1.4.0"

RegisteredHostDiscoveredResourcesProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Host Discovered Resources Profile. The properties are set as follows:

- InstanceID – set to "SNIA:HostDiscoveredResourcesProfile-1.2.0"
- RegisteredOrganization – set to "11" (SNIA)
- RegisteredName – set to "Host Discovered Resources Profile"
- RegisteredVersion – set to "1.2.0"

RegisteredPCIDeviceProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the PCI Device Profile. The properties are set as follows:

- InstanceID – set to "DMTF:DSP1075-PCIDevice-1.0.0a"
- RegisteredOrganization – set to "2" (DMTF)
- RegisteredName – set to "PCIDevice Profile"
- RegisteredVersion – set to "1.0.0a"

RegisteredSoftwareInventoryProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Software Inventory Profile. The properties are set as follows:

- InstanceID – set to "DMTF:DSP1023-SoftwareInventory-1.0.1"
- RegisteredOrganization – set to "2" (DMTF)
- RegisteredName – set to "Software Inventory Profile"
- RegisteredVersion – set to "1.0.1"
RegisteredSoftwareUpdateProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Software Update Profile. The properties are set as follows:

- **InstanceID** – set to "DMTF:DSP1023-SoftwareUpdate-1.0.0"
- **RegisteredOrganization** – set to "2" (DMTF)
- **RegisteredName** – set to "Software Update Profile"
- **RegisteredVersion** – set to "1.0.0"

RegisteredPhysicalAssetProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Physical Asset Profile. The properties are set as follows:

- **InstanceID** – set to "DMTF:PhysicalAssetProfile-1.0.2"
- **RegisteredOrganization** – set to "2" (DMTF)
- **RegisteredName** – set to "Physical Asset Profile"
- **RegisteredVersion** – set to "1.0.2"

RegisteredSensorsProfile

Only one instance of this class is needed. It resides in the /root/interop namespace and indicates the implementation of the Sensors Profile. The properties are set as follows:

- **InstanceID** – set to "SNIA:SensorsProfile-1.0.0"
- **RegisteredOrganization** – set to "11" (SNIA)
- **RegisteredName** – set to "Sensors Profile"
- **RegisteredVersion** – set to "1.0.0"

RegisteredCommonDiagnosticProfile

Only one instance of this class is needed. It will reside in the /root/interop namespace and indicate the implementation of the Common Diagnostic Model Profile. The **InstanceID** property will be set to a value of "DMTF:DiagnosticsProfile-2.0.0a". The **RegisteredOrganization** property will be set to a value of "2" (DMTF). The **RegisteredName** property will be set to a value of "Diagnostics Profile". The **RegisteredVersion** property will be set to a value of "2.0.0a".
Indications

An indication will be generated periodically when a serious condition exists for a particular ioMemory device. The WBEM provider currently supports six types of indications. They alert users of the SMI-S provider to conditions such as imminent wearout, degradation of writability, degradation of the flashback feature, higher temperature, and internal error states.

The indications will be instances of the FIO_AlertIndication class which simply specializes the CIM_AlertIndication class.

The values for the properties of the FIO_AlertIndication instances are under development and may change as testing proceeds and feedback is received.
FIO_AlertIndication

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td>See below for each type</td>
</tr>
<tr>
<td>IndicationTime</td>
<td>Timestamp when sent</td>
</tr>
<tr>
<td>AlertingManagedElement</td>
<td>IoMemoryPort.DeviceID=&lt;device ID&gt;</td>
</tr>
<tr>
<td>AlertingElementFormat</td>
<td>CIMObjectPath (2)</td>
</tr>
<tr>
<td>AlertType</td>
<td>Device Alert (5)</td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>See below for each type</td>
</tr>
<tr>
<td>ProbableCause</td>
<td>See below for each type</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>&quot;FIO_AlertIndication&quot;</td>
</tr>
<tr>
<td>SystemName</td>
<td>&lt;hostname&gt;</td>
</tr>
<tr>
<td>ProviderName</td>
<td>&quot;fiosmis&quot;</td>
</tr>
<tr>
<td>CorrelatedIndications</td>
<td>Not used</td>
</tr>
<tr>
<td>Description</td>
<td>Class description</td>
</tr>
<tr>
<td>OtherAlertType</td>
<td>Not used</td>
</tr>
<tr>
<td>OtherSeverity</td>
<td>Not used</td>
</tr>
<tr>
<td>ProbableCauseDescription</td>
<td>Not used</td>
</tr>
<tr>
<td>EventID</td>
<td>Same as IndicationIdentifier</td>
</tr>
<tr>
<td>OwningEntity</td>
<td>&lt;vendor&gt;</td>
</tr>
<tr>
<td>MessageID</td>
<td>TBD</td>
</tr>
<tr>
<td>Message</td>
<td>TBD</td>
</tr>
<tr>
<td>MessageArguments</td>
<td>TBD</td>
</tr>
</tbody>
</table>

Reduced Writability Indication

The ioMemory VSL can dramatically reduce write throughput to manage device conditions such as excessive wear, high temperature, and insufficient power. The reduced writability indication is generated while the drive is in this mode. If the triggering condition is excessive wear, the IoMemoryPort health percentage will report 0% health.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td>&lt;mfr&gt;&quot;&quot;:&lt;hostname&gt;&quot;write&quot;</td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>Degraded/Warning (3)</td>
</tr>
</tbody>
</table>
Read-only Indication

When the drive has reached the end-of-life, it can no longer be written to and can only be read from. The read-only indication will be sent when this occurs. The IoMemoryPort health percentage will continue to report 0% health when this happens.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td><code>&lt;mfr&gt;&quot;&quot;:&quot;&lt;hostname&gt;&quot;:read_only&quot;</code></td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>Degraded/Warning (3)</td>
</tr>
<tr>
<td>ProbableCause</td>
<td>Threshold Crossed (52)</td>
</tr>
</tbody>
</table>

Wearout Indication

As the drive wears out, this indication is generated as a warning when the drive health percentage drops below 10%, before write throughput is reduced.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td><code>&lt;mfr&gt;&quot;&quot;:&quot;&lt;hostname&gt;&quot;:wearout&quot;</code></td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>Degraded/Warning (3)</td>
</tr>
<tr>
<td>ProbableCause</td>
<td>Threshold Crossed (52)</td>
</tr>
</tbody>
</table>

Flashback Indication

If a catastrophic part failure degrades the effectiveness of the flashback feature, this indication will be sent.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td><code>&lt;mfr&gt;&quot;&quot;:&quot;&lt;hostname&gt;&quot;:flashback&quot;</code></td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>Degraded/Warning (3)</td>
</tr>
<tr>
<td>ProbableCause</td>
<td>Loss of Redundancy (88)</td>
</tr>
</tbody>
</table>

High Temperature Indication

This indication will be sent when the temperature of the card becomes excessive.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td><code>&lt;mfr&gt;&quot;&quot;:&quot;&lt;hostname&gt;&quot;:temperature&quot;</code></td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>Critical (6)</td>
</tr>
</tbody>
</table>
ProbableCause | Temperature Unacceptable (51)

**Error Indication**

If the ioMemory VSL is in an error state the error indication will be sent.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndicationIdentifier</td>
<td>&lt;mfr&gt;&quot;:&lt;hostname&gt;&quot;::error&quot;</td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>Major (6)</td>
</tr>
<tr>
<td>ProbableCause</td>
<td>Other (1)</td>
</tr>
</tbody>
</table>
Appendix I- NUMA Configuration

About NUMA Architecture

Servers with a NUMA (Non-Uniform Memory Access) architecture require special installation instructions in order to maximize ioMemory device performance. These servers include the HP DL580, HP DL980, or the IBM 3850 server.

On servers with NUMA architecture, during system boot, the BIOS on some systems will not distribute PCIe slots evenly among the NUMA nodes. Each NUMA node contains multiple CPUs. This imbalanced distribution means that, during high workloads, half or more of the CPUs will remain idle while the the rest are 100% utilized. To prevent this imbalance, you must manually assign ioMemory devices equally among the available NUMA nodes.

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 ioMemory devices 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 (fct numbers). Next, use fio-beacon to identify each of the devices in their respective PCIe slots. Then use the example below of the fio-config syntax as a template and modify it for your particular system.

Configuring your ioMemory devices for servers with NUMA architecture requires the use of the FIO_AFFINITY parameter with the fio-config utility.

FIO_AFFINITY Parameter

The FIO_AFFINITY parameter is a list of <affinity specification> triplets that specify the affinity settings of all adapters in the system. Each item in the triplet is separated by a comma, and each triplet set is separated by a semicolon.

Syntax:

```shell
fio-config -p FIO_AFFINITY <affinity specification>[:<affinity specification>...]
```

Where each <affinity specification> has the following syntax:
If there is no g or n character before the group/node number, then the number is assumed to be a group number.

The hex mask is optional. If it is not present, the mask is assumed to be 0xffffffffffffffff. Also, the 0x prefix is optional.

If the hex mask is a node mask, then the mask is relative to the node, not the group to which the node belongs.

**Simple Example:**

```
fio-config -p FIO_AFFINITY 4,n1,0xf;5,n0;7,g1;9,g2,0xff0
```

Has the effect of creating:

<table>
<thead>
<tr>
<th>Device</th>
<th>Node/Group</th>
<th>Processor Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>fct4</td>
<td>node 1</td>
<td>processors 0 to 3 in the node (mask 0xf)</td>
</tr>
<tr>
<td>fct5</td>
<td>node 0</td>
<td>all processors in the node (no hex mask)</td>
</tr>
<tr>
<td>fct7</td>
<td>group 1</td>
<td>all processors in the group (no hex mask)</td>
</tr>
<tr>
<td>fct9</td>
<td>group 2</td>
<td>processors 4 to 11 in the group (mask 0xff0)</td>
</tr>
</tbody>
</table>

**Advanced Configuration Example**

This sample server has 4 NUMA nodes with 8 hyper-threaded cores per node (16 logical processors per node, a total of 64 logical processors in the system). This system also uses the expansion configuration and has 11 PCIe expansion slots. During system boot, the system's BIOS will assign PCIe slots 1-6 to NUMA node 2 and PCIe slots 7-11 to NUMA node 0. NUMA nodes 1 and 3 will have no assigned PCIe slots. This creates a load balancing problem in the system when ioMemory devices are under heavy traffic. Specifically, during these periods of high use, half of the CPUs in the system will sit idle while the other half of the CPUs are 100% utilized, thus limiting the throughput of the ioMemory devices.

To avoid this problem, you must manually configure the affinity of the ioMemory devices using the `FIO_AFFINITY` configuration parameter to distribute the work load across all NUMA nodes. This parameter will override the default behavior of the ioMemory VSL driver. For more information about the `FIO_AFFINITY` configuration parameter, refer to the syntax explanation above.

What follows is an example of how to manually configure 10 ioDrive Duo devices (each with two ioMemory devices). Slot 1 is a Generation 1 PCI-e slot, so it is not compatible with an ioDrive Duo device. Thus we can fill slots 2-11 with ioDrive Duo devices.
Because each ioDrive Duo device has two ioMemory devices, there are two device numbers for each ioDrive Duo device (one for each ioMemory device). There will therefore be two device numbers for each slot.

When the system boots, the default BIOS NUMA node assignments are:

<table>
<thead>
<tr>
<th>BIOS Assigned NUMA Node</th>
<th>PCI-e Slots</th>
<th>FCT device numbers</th>
<th>Processor Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7-11</td>
<td>8,9,13,14,18,19,23,24,28,29</td>
<td>all processors in the node</td>
</tr>
<tr>
<td>1</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>2</td>
<td>2-6</td>
<td>135,136,140,141,145,146,150,151,155,156</td>
<td>all processors in the node</td>
</tr>
<tr>
<td>3</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
</tbody>
</table>

Here, the BIOS creates a load imbalance by assigning the cards to only two NUMA nodes in the system. In order to balance the work load, we want to make the following manual settings:

<table>
<thead>
<tr>
<th>Assigned NUMA Node</th>
<th>PCI-e Slots</th>
<th>FCT device numbers</th>
<th>Processor Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7-9</td>
<td>8,9,13,14,18,19</td>
<td>all processors in the node (no hex mask)</td>
</tr>
<tr>
<td>1</td>
<td>10-11</td>
<td>23,24,28,29</td>
<td>all processors in the node (no hex mask)</td>
</tr>
<tr>
<td>2</td>
<td>2-3</td>
<td>135,136,140,141</td>
<td>all processors in the node (no hex mask)</td>
</tr>
<tr>
<td>3</td>
<td>4-6</td>
<td>145,146,150,151,155,156</td>
<td>all processors in the node (no hex mask)</td>
</tr>
</tbody>
</table>

In order to configure the ioMemory VSL driver with these override settings, run `fio-config` with the following string:

```bash
fio-config -p FIO_AFFINITY
8,n0;9,n0;13,n0;14,n0;18,n0;19,n0;23,n1;24,n1;28,n1;29,n1;
135,n2;136,n2;140,n2;141,n2;145,n3;146,n3;150,n3;151,n3;155,n3;156,n3
```

The above example contains a line break for formatting purposes. There would be no line breaks in a real implementation of `FIO_AFFINITY`.

Note that no `<hex mask>` was stipulated for any of the devices in this example (making each `<affinity specification>` a couplet rather than a triplet). This means that each device is shared among all of the processors in the specified nodes. If you wish to fine tune the configuration, you can stipulate the hex mask in each `<affinity specification>` and specify the processors for each device.
Checking the Log for Errors

If you enter a configuration that is not valid, then the settings will be disabled and an error will be available in the system logs.

For example:

```
$ fio-config -p FIO_AFFINITY 5,g0,0xf;6,0xf
```

In this example, the affinity for device fct6 is set incorrectly, because there is no group/node number before the mask. The following errors appear in the system log:

```
2011-09-09T12:22:15.176086800Z - ERROR - FusionEventDriver - FIO_AFFINITY: Invalid group or node number
2011-09-09T12:22:15.176086800Z - ERROR - FusionEventDriver - Invalid FIO_AFFINITY parameter syntax at character 13: "5,g0,0xf;6,0x". Manual affinity settings are disabled!
```
Appendix J- Upgrading Devices from VSL 2.x to 3.x

This version of the ioMemory VSL supports new features, including the latest generation of ioMemory architecture and improved Flashback protection. These features require the latest version of the Fusion-io firmware. Every ioMemory device in a system running 3.1.x or later must be upgraded to the latest version of the firmware.

For example, if you have a system running 2.3.1 ioMemory VSL with ioDrive devices previously installed, and you want to install new ioDrive2 devices (that require the latest version of the firmware), then you will need to upgrade all of the existing devices to the latest firmware version.

You cannot revert a device's firmware to an earlier version once you have upgraded the device (without voiding your warranty). If you experience problems with your upgrade, please contact Customer Support at support@fusionio.com.

Upgrading devices (previously configured for VSL 2.x.x) to work with VSL 3.x.x will require a low-level media format of the device. No user data will be maintained during the process. Be sure to backup all data as instructed.

Upgrade Path
Depending on the current firmware version of your devices, you may need to upgrade your device's firmware multiple times in order to preserve internal structures. The following is the minimum upgrade path that must be followed. Upgrade the ioMemory VSL software on the system (and upgrade the firmware version for each version of the software) in this order:

1.2.4 -> 1.2.7 -> 2.1.0 -> 2.3.1 -> 3.1.x

For example, if your device is using the firmware for ioMemory VSL version 2.2.0, upgrade to 2.3.1 (both ioMemory VSL and compatible firmware) and then continue on the path. Visit http://ts.fujitsu.com/support/ for all of the required software and firmware versions.

For more information on upgrading from one version to the next, see the ioMemory VSL Release Notes (available at http://ts.fujitsu.com/support/) for the version you will upgrade the device to. Then follow the upgrade instructions in version's user guide for your operating system (including the firmware update instructions).
Upgrade Procedure

Be sure to follow the upgrade path listed above. Make sure that all previously installed ioDrive devices are updated with the appropriate 2.3.1-compatible firmware.

⚠️ If you plan to use ioDrive devices and ioDrive2 devices in the same host, perform this upgrade on all existing ioDrive devices before installing the new ioDrive2 devices.

1. Prepare each existing ioDrive device for upgrade.
   a. Backup user data on each ioDrive device.
      - The upgrade process will require a low-level media format of the device. No user data will be maintained during the process; be sure to make a complete backup.

      Use a backup method of your choice. For best results, use software and backup devices that have proven effective in the past. Do not backup the data onto another ioMemory device on the same system. The back up must be to a local disk or to an externally attached volume.

   b. Run the `fio-bugreport` command-line utility and save the output. This will capture the device information for each device in the system. This device information will be useful in troubleshooting any upgrade issues. Sample command:

      ```
fio-bugreport
      ```

   c. Detach ioDrive devices, for example:

      ```
fio-detach /dev/fct*
      ```

      For more information, see `fio-detach`

2. Uninstall the 2.x ioMemory VSL software
   a. Go to Start > Control Panel.

   b. Click Programs & Files.

   c. Select the ioMemory VSL (Fusion-io) entry.

   d. Click Uninstall.

   e. Restart the computer.
3. Install the new VSL.
   b. Run the ioMemory VSL installation program. The installation program presents a custom setup tree-view with options for installation.
      Mouse over a component in the tree view to see its description.
   c. Click Next.
   d. To select a different folder for the installation, browse to the folder and click OK. The default folder is C:\Program Files\Fusion-io ioMemory VSL.
      The uninstaller file is placed in the root of the Fusion-io ioMemory VSL (default install folder).
   e. Follow the onscreen prompts to complete the install.
   f. Choose Reboot Now on the finish screen of the installer.
      For full installation instructions, see [Existing ioMemory VSL Installation](#).

4. Update the firmware on each device to the latest version using fio-update-iodrive.

   **Prevent Power Loss**
   Take measures to prevent power loss during the update, such as a UPS. Power loss during an update may result in device failure. For all warnings, alerts, and options pertaining to this utility, see the [fio-update-iodrive](#) utility reference in the appendix.

   Sample syntax:
   ```
   fio-update-iodrive <iodrive_version.fff>
   ```
   Where `<iodrive_version.fff>` is the path to the firmware archive (the default file path is `C:\Program Files\Fusion-io ioMemory VSL\Firmware\iodrive_<version>.fff`). This command will update all of the devices to the selected firmware. If you wish to update specific devices, consult the [utility reference](#) for more options.
5. **Reboot the system**

   ![Tip](image)

   If run, `fio-status` will warn that the upgraded devices are missing a lebmap. This is expected, and will be fixed in the next step.

   **Destructive Step**

   Running `fio-format` in the next step will erase the entire device, including user data. Once this format is started, the device cannot be downgraded to the 2.x driver without voiding your warranty. If you experience problems with your upgrade, please contact Customer Support at `support@fusionio.com`.

6. Format each device using `fio-format`, for example:

   ```
   fio-format <device>
   ```

   You will be prompted to confirm you wish to erase all data on the device.

   ![Warning](image)

   The format may take an extended period of time, depending on the wear on the device.

7. Attach all `ioDrive` devices, for example:

   ```
   fio-attach /dev/fct*
   ```

8. Check the status of all devices using `fio-status`, for example:

   ```
   fio-status -a
   ```

Your `ioDrive` devices have now been successfully upgraded for this version of the `ioMemory VSL`. You may now install any `ioDrive2` devices.
Appendix K- Documentation Permissions

The AVR bootloader and the tree.h file, which ship in binary form with the driver, contain content that have the following documentation copyright requirements:

**AVR Bootloader**


All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**tree.h**

Copyright 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Fusion Powered Support

We offer ioDrive Customer Support by phone, e-mail and on the Web.

E-Mail

Please use the contact form at:
http://support.ts.fujitsu.com/contact/

For the Japanese market, please use the following URL:
http://primeserver.fujitsu.com/primergy/support/

Telephone Support

For customer support phone numbers, please refer to the Fujitsu Technology Solutions service desk at http://ts.fujitsu.com/support/servicedesk.html

For the Japanese market, please use the following URL:
http://primeserver.fujitsu.com/primergy/support/supportdesk.html

Web

It is recommended to use the latest firmware / driver / BIOS versions on servers and components.

Software packages are available for download at:
http://support.ts.fujitsu.com > Server > Drivers & Downloads

For the Japanese market, please use the following URL:
http://primeserver.fujitsu.com/primergy/downloads

Corresponding user documentation for servers and optional components can be found online at
http://manuals.ts.fujitsu.com > Industry Standard Servers > Expansion Cards > PCIe SSD Cards or on the ServerView Suite Documentation DVD.

For the Japanese market, please use the following URL: