ioMemory VSL 3.2.6

USER GUIDE FOR MICROSOFT WINDOWS

OCTOBER 07, 2013

FUSION-IO

Table of Contents

ioMemory VSL 3.2.6 User Guide for Microsoft Windows

Legal Notices

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

Fusion-io, the Fusion-io logo, ioMemory, Virtual Storage Layer, VSL, ioFX, ioSphere, ioScale, and ioDrive are registered trademarks, and Adaptive Flashback and Sure Erase 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: D0001658-020_1
Published: October 07, 2013

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, ioScale 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 the ioMemory Platform

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

Performance

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

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

Endurance

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

Reliability

The ioMemory platform eliminates concerns about reliability like NAND failures and excessive wear. The all-new intelligent, self-healing feature called Adaptive Flashback provides complete, chip-level fault tolerance. Adaptive

Flashback technology enables an ioMemory product to repair itself after a single chip or a multi-chip failure without interrupting business continuity.

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

> Every ioMemory device in a system must be upgraded to the appropriate firmware.
>
> For example, if you have a system running ioMemory VSL software version 2.2.3 with ioMemory 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 with firmware that supports this version of the ioMemory VSL software. Follow the upgrade path in the *ioMemory VSL Release Notes* to determine the upgrade sequence.

> **Upgrade Previous Devices First**
> If you have ioMemory devices configured for ioMemory VSL software version 2.x or earlier, you must upgrade their firmware before installing new devices in the system. See Upgrading Devices from VSL 2.x to 3.x on page 76 for the upgrade instructions.

If you have ioMemory devices installed and in a RAID configuration, please read Upgrading the Software with a RAID Configuration on page 29 before you upgrade the software and/or firmware.

Installation Overview

1. Download the latest version of the software at http://support.fusionio.com .
2. If you are installing this version of ioMemory VSL software on a system with ioDrive devices configured with firmware for ioMemory VSL software version 2.x, you must carefully follow the instructions in the Upgrading Devices from VSL 2.x to 3.x on page 76. (Follow those instructions instead of the normal installation instructions.)
3. If you have a previous version of the ioMemory VSL software installed, you will need to uninstall the ioMemory VSL software and the utilities.
4. Install the latest version of the ioMemory VSL software and command-line utilities.

> For information on capturing an installation log for troubleshooting purposes, see the following [Microsoft KB article|http://support.microsoft.com/kb/314881].

- **Extracting the MSI File**: If you require the MSI file, you may extract it using the following command:

```
<installname>.exe
```

For example, you may need the MSI file to deploy the software via Group Policy on a Windows Server.

5. Determine if you need to upgrade the firmware to the latest version, see Upgrading the Firmware on page 11.

6. Configure the device(s) by following the configuration instructions, for example Adding a Filesystem on page 19, Creating a RAID Configuration on page 20, etc.

Installing the Software

Do not install new ioDrive2 devices with previously installed ioDrive devices (that are configured for ioMemory VSL software version 2.x) without first completing the instructions in [Appendix J- Upgrading Devices from VSL 2.x to 3.x].

1. Review the *ioMemory VSL Release Notes* available for this version of the software for additional steps that may be needed to complete the install.

2. For new device installations, make sure you have properly installed the device(s) before you install the ioMemory VSL software.

3. Log in as Administrator or have Administrator rights.

4. If needed, uninstall the existing ioMemory VSL software, utilities, etc., using **Programs and Features**, or **Add or Remove Programs** (depending on your version of Windows), in the **Control Panel**.

5. 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 software 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.

6. Download the ioMemory VSL installation program for Windows from http://support.fusionio.com to your desktop or a convenient directory. There are two installation programs available:

   - `<installname>_WinServ2003_2008R1.exe` – For installing on Windows Server 2008.
   - `<installname>_WinServ2008R2_2012_Win7_8.exe` – For installing on Windows Server 2008 R2 and newer.

     Also download the `fio-firmware-<version>.<date>.fff` firmware archive file for this release and save it in the same location.

7. Run the ioMemory VSL installation program. The installation program presents a custom setup tree-view with options for installation.

8. 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.

9. Click **Next**.
10. 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 utilites. The default path is `C:\Program Files\Common Files\VSL Utils`
11. Follow the onscreen prompts to complete the install.
12. 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 software for the device(s). See Manual Installation on page 54 for information on manual installation.

> **Pagefile Support**
> If your ioMemory device is configured for pagefile support, you may need to reboot a second time before Windows can create a permanent pagefile.

> You may also install the ioSphere software (optional GUI management software). ioSphere software and documentation are available as a separate download.

Once the system restarts, proceed to Upgrading the Firmware on page 11.

Silent Install Option

> **Uninstall Previous**
> If the you have a version of the ioMemory VSL software previously installed, you must uninstall it first (see the information on a **Silent Uninstall** below). You can must 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 software using its default settings, eliminating the need to "click Next" or select settings during install.

> Be sure to use the `/quiet` parameter. The command-line quiet install parameter has changed and the installer no longer supports the abbreviated parameter (`/qn`). If you pass the `/qn` parameter to the installer, the installer will ignore the parameter and the installer GUI will launch.

Silent Uninstall

You may silently uninstall the ioMemory VSL software with this command:

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

Upgrading the Firmware

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

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

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

> Do not attempt to downgrade the firmware on any ioMemory device, doing so may void your warranty.

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 software before installing the new devices. Consult the *ioMemory VSL Release Notes* for this ioMemory VSL software 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 VMware host server after you upgrade the device(s). Just restarting the virtual machine won't apply the change.

Command-line Interface

More information on these command-line utilities is available in Command-line Utilities Reference on page 35. All command-line utilities require Administrator rights.

1. Run the `fio-status` utility and examine the output. See fio-status on page 44 for usage information.

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

2. If the firmware is old or outdated, update it using the `fio-update-iodrive` utility. See fio-update-iodrive on page 49 for complete information and warnings.

Optional GUI - ioSphere software

You can use the ioSphere software software to check the status of your ioMemory devices. If the ioSphere software indicates that the device's firmware is outdated, you can also use the ioSphere software to upgrade the device firmware. Consult the ioSphere software documentation for more information on installing and using the software.

Configuration

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

Setting the ioMemory VSL Options

You can configure the ioMemory VSL software using various module parameters. Individual module parameters are described throughout this guide. For a complete list of all parameters and how to implement them see fio-config on page 38.

Enabling PCIe Power Override

If you have installed any products with multiple ioMemory devices, such as the ioDrive Duo device, 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, you may allow the device to draw full power from the PCIe slot by setting an ioMemory VSL software 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, 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 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` (if available) 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 that is 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, 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:

```
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 fio-beacon on page 36 for more information.

Setting the Parameter

Set the module parameter by using the fio-config utility and specifying a new value for the `external_power_override` parameter. 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`

> ⚠ You must reboot or unload/load the driver to enforce any parameter changes.

Virtual Controller Configuration

Depending on your use case and application, you may benefit from configuring supported devices to use Virtual Controller technology.

When configured, each physical ioMemory device is split into two (virtual) logical devices. Splitting the ioMemory device into two virtual devices has the following implications:

- **Latency**: There is no affect on latency.
- **Throughput**: The total peak I/O bandwidth of the device is approximately the same.
- **IOPS**: Depending on the use of the virtual devices (especially the average I/O size), the peak IOPS for each virtual device is about the same for a non-split device. In other words, the combined peak IOPS of the two virtual devices can be nearly double that of a non-split device. For details, see in the *Maintenance* section.
- **Capacity**: Due to virtualization overhead, the combined capacity of the two virtual devices is slightly less than that of a single-controller device. See the *ioMemory VSL Release Notes* for a list of compatible devices and their Virtual Controller capacities.

Converting your ioMemory device to a Virtual Controller configuration will split the ioMemory device into two logical devices.

For 512B I/Os, the combined IOPS performance of the two virtual devices is approximately double that of a single-controller device. For 4kB I/Os, there is more than an 80% improvement in IOPS performance with virtual devices. For 16kB and larger I/Os, there is no improvement of total IOPS performance over a non-Virtual Controller configuration.

Latency in the virtual devices is unaffected, and the combined bandwidth of the two virtual devices is the same as it would be without the split. Due to the overhead of an additional device, the combined capacity of the two virtual devices is slightly less than that of a single-controller device.

Splitting a single physical device into multiple virtualized devices, or merging multiple virtualized devices back to a single physical device, requires a low-level format, which will erase all of the data on the device. Be sure to back up all of your data.

Supported Devices

Only relatively new devices (with few writes performed) may be split or merged. Devices with too much wear are unsuitable for converting to or from a Virtual Controller configuration. Merging virtual devices may also result in additional wear (depending on the wear differences of the two virtual devices).

To be suitable for splitting or merging, devices (including Virtual Controller devices) must have 90% or more of their remaining rated endurance of Petabytes Written (PBW). This rating as well as the current percentage remaining is visible in `fio-status` with the `-a` option. For example:

```
fio-status /dev/fct1 -a
...
Rated PBW: 17.00 PB, 99.95% remaining
```

In the above example, the device is suitable for conversion because it has more than 90% of the rated PBW remaining.

If you attempt to merge or split a device that does not support Virtual Controller technology or a device that has too much wear, the update utility will not allow the conversion and the firmware upgrade will not take place. See the Release Notes for a list of devices that support Virtual Controller technology and their capacities after the conversion.

Multi-device Products

For products with more than one ioMemory device, such as an ioDrive2 Duo device, you must configure all of the ioMemory devices to Virtual Controller technology at the same time. All of the devices must also be merged at the same time. For example, the two ioMemory devices in an ioDrive2 Duo device will be converted into four virtual devices. The utility will not allow a conversion if you attempt to split or merge only one physical device in a multi-device product.

Splitting Controllers

Be sure to use firmware that supports Virtual Controller technology. Consult the Release Notes to determine if the firmware for that release supports Virtual Controller technology.

1. Back up all of your data. Because a low-level format is needed to complete the conversion, all of the user data on your device will be erased.

2. Use the `fio-update-iodrive` command-line utility to configure an ioMemory device to use Virtual Controller technology:

- Use the `--split` option to split the controller.
- Use the `-d` option to specify a device, otherwise all installed devices that can be split will be split.
- Specify the firmware path, and check the *ioMemory VSL Release Notes* to make sure the firmware supports Virtual Controller technology.

  Example:

  ```
  fio-update-iodrive --split -d /dev/fct0 <firmware-path>
  ```

  After rebooting, each physical device will be split into two virtual devices. Each ioMemory device will therefore split into two logical devices, each with a unique device path. For example, `/dev/fct0` may become `/dev/fct0` and `/dev/fct1`. You will manage each device as a unique device.

3. Reboot.
4. Load the ioMemory VSL driver.
5. Run `fio-status` to determine which devices need to be formatted.
6. Low-level format the device(s). For example:

   ```
   fio-format /dev/fct0 /dev/fct1
   ```

   Formatting will erase all user data, be sure to back up your data. You can reverse the split by merging the controllers (without losing data) up until you format the virtual devices.

Merging Controllers

If your ioMemory device (including the two virtual devices) is suitable for merging, then you will be able to use the `fio-update-iodrive` utility to merge the virtual devices back into one physical device.

1. Back up all of your data. Because a low-level format is needed to complete the merge, all of the user data on your device will be erased.
2. Use the `fio-update-iodrive` command-line utility to configure the device for merging:
   a. Use the `--merge` option to merge the virtual devices.
   b. Use the `-d` option to specify a device.

> The `fio-update-iodrive` utility only successfully works against one of the two virtual devices for each physical ioMemory device. Out of the two virtual devices, only the first virtual device (in terms of device numbering) is linked to the physical device (and the firmware). The second virtual device is not linked, and any firmware operation against that second virtual device will fail with this message:
>
> ```
> Error: Device '/dev/fctx' had an error while updating.
> This device does not support firmware update.
> ```
>
> This is expected, and the error will not affect the update/merge of the first (linked) virtual device. The update operation will complete on all devices that can merge and otherwise accept firmware changes.

c. Specify the firmware path, and check the *ioMemory VSL Release Notes* to make sure the firmware supports Virtual Controller technology.

Example:

```
fio-update-iodrive --merge -d /dev/fct0 <firmware-path>
```

3. Reboot.
4. Load the ioMemory VSL driver.
5. Run `fio-status` to determine which devices need to be formatted.
6. Low-level format the device(s). For example:

```
fio-format /dev/fct0
```

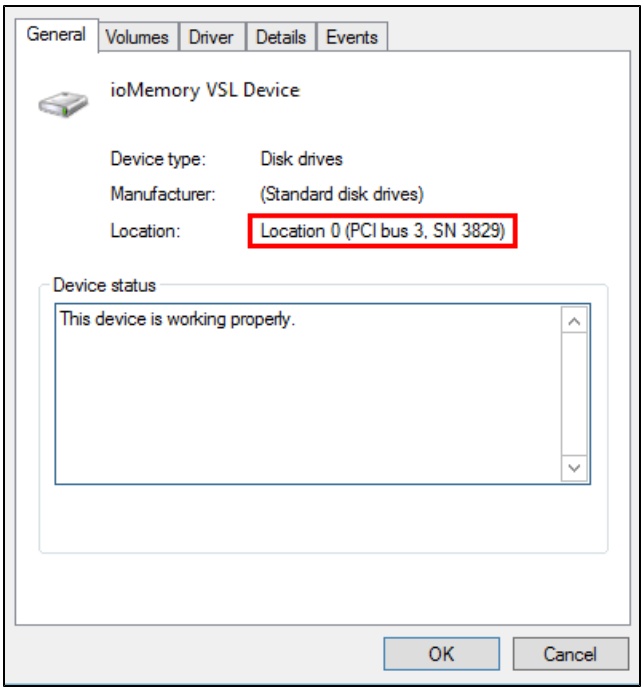
> Formatting will erase all user data, be sure to back up your data. You can reverse the merge by splitting the controllers (without losing data) up until you format the merged device.

Device Naming

The ioMemory device receives a name and number as part of the install process for identification. The syntax is `fctx` where *x* indicates the device number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first ioMemory device installed on the system. Use the ioSphere software or the `fio-status` utility to view this bus number, or follow these steps:

1. Choose **Start** > **Control Panel** > **System** > **Hardware** > **Device Manager**.
2. Select **Disk Drives**.

3. Click on your ioMemory device in the list. The Properties dialog box appears.



The Location field shows the PCIe bus number for your device (`fct3` in this case).

Adding a Filesystem

With ioMemory device(s) and ioMemory VSL software 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).

> **Devices with >2TB Capacities**
> Devices with capacities greater than 2TB require the following partition types:
> - **Single device**: GPT (GUID Partition Table)
> - **Multiple devices** (for a RAID configuration): Dynamic Disk
>
> These devices also require sector sizes greater than 512B (we recommend 4kB sectors). When you format these devices using `fio-format`, the default sector size is 4KB.

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 a Page Files Store

To safely use the ioMemory device with page files (also known as Virtual Memory) requires passing the preallocate_ memory kernel module parameter. To set this parameter, use either the optional ioSphere software (see ioSphere software documentation), or use the fio-config command-line utility (see fio-config on page 38 for the full utility instructions):

```
fio-config -p FIO_PREALLOCATE_MEMORY 1149D2717-1121,1149D2717-1111,10345
```

- Where `1149D2717-1111,1149D2717-1111,10345` are serial numbers obtained from `fio-status`, see fio-format on page 42.

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

Once you have set this parameter, you can go into the system settings and use the ioMemory device(s) to store the paging files. For more information see Using the Windows Page Files on page 65.

> Be sure to provide the serial numbers for the ioMemory device, not an adapter, when applicable.

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

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

Discard (TRIM) Support

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

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

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

Windows does not support TRIM with a RAID 5 configuration.

TRIM on Windows Server 2008 R2 and Newer

Windows Server 2008 R2 and newer have built-in TRIM support. With these operating systems, 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, and it is therefore not supported.

Performance and Tuning

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

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

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

Disable CPU Frequency Scaling

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

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

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

Limiting ACPI C-States

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

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

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

Setting ACPI C-State Options

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

Setting NUMA Affinity

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

On some servers with NUMA architecture, during system boot, the BIOS will not associate PCIe slots with the correct NUMA node. Incorrect mappings result in inefficient I/O handling that can significantly degrade performance. To prevent this, you must manually assign ioMemory devices optimally among the available NUMA nodes.

See NUMA Configuration on page 72 for more information on setting this affinity.

Setting the Interrupt Handler Affinity

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

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

Windows IRQ Policy

By default, Windows uses a policy of `IrqPolicyAllCloseProcessors` and a priority of IrqPriorityNormal, which should work best for most applications.

If manual tuning is needed, Windows provides the Interrupt Affinity Policy Tool. Information on this tool can be found at: http://msdn.microsoft.com/en-us/windows/hardware/gg463378. The settings that the application changes are listed at: http://msdn.microsoft.com/en-us/library/ff547969(v=vs.85).aspx.

With Windows Sever 2008 or newer on a machine with more than 64 processors, there's an additional GroupPolicy parameter that can be set through the registry in order to set the affinity to a different processor group. This is documented at: http://msdn.microsoft.com/en-us/windows/hardware/gg463349.

Monitoring and Managing Devices

Fusion-io provides many tools for managing your ioMemory devices. These tools will allow you to monitor the devices for errors, warnings, and potential problems. They will also allow you to manage the devices including performing the following functions:

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

Management Tools

Fusion-io has provided several tools for monitoring and managing ioMemory devices. These include stand-alone tools that require no additional software and data-source tools that can be integrated with other applications.

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

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

Stand-alone Tools

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

- **Command-line Utilities**: These utilities are installed with the ioMemory VSL software and are run manually in a terminal. The `fio-status` utility provides status for all devices within a host. The other utilities allow you to perform other management functions. See Command-line Utilities Reference on page 35 for full details.
- **ioSphere software**: The GUI browser-based ioSphere software allows you to monitor and manage every ioMemory device installed in multiple hosts across your network. It collects all of the alerts for all ioMemory devices and displays them in the Alert Tab. You may also set up the ioSphere software to send email or SMS messages for specific types of alerts or all alerts. The ioSphere software packages and documentation are available as separate downloads.

Data-source Tools

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

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

- **SNMP Subagent**: The Fusion-io SNMP AgentX subagent allows you to monitor and manage your ioMemory devices using the Simple Network Management Protocol. You can use a normal SNMP browser, or customize your existing application to interface with the subagent.
- **SMI-S CIM Provider**: The CIM provider allows you to monitor and manage your devices using the Common Information Model. You can use a normal CIM browser, or customize your existing application to interface with the CIM provider.

Example Conditions to Monitor

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

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

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

Device Status

All of the monitoring tools return information on the status of the ioMemory devices, including the following states:

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

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

Required Actions

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

Temperature

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

These temperatures depend on the device. Newer ioMemory devices have higher thermal tolerances. Consult the *ioMemory Hardware Installation Guide* to determine the thermal tolerances of all devices you will monitor. **This table uses the controller thermal tolerances for newer devices** (93°C throttling, 100°C shutdown).

GREEN	<93°C
YELLOW	93-99°C
RED	100°C

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

GREEN	<90°C
YELLOW	90-96°C
RED	97°C

NAND Board Temperature
Newer ioMemory devices also report the temperature of the NAND Boards. This is also a critical temperature to monitor. Consult the *ioMemory Hardware Installation Guide* to see if your device reports this temperature and to see the temperature thresholds.

Required Actions

If the temperature is at or approaching the **YELLOW** condition, you must increase the cooling for your system. This may include increasing the fan speed, bringing down the ambient temperature, reducing write load, or moving the device to a different slot.

Health Reserves Percentage

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

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

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

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

For complete information on Health Reserve conditions and their impact on performance, see Monitoring the Health of Devices on page 63.

Required Actions

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

Write (Health Reserves) Status

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

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

Required Actions

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

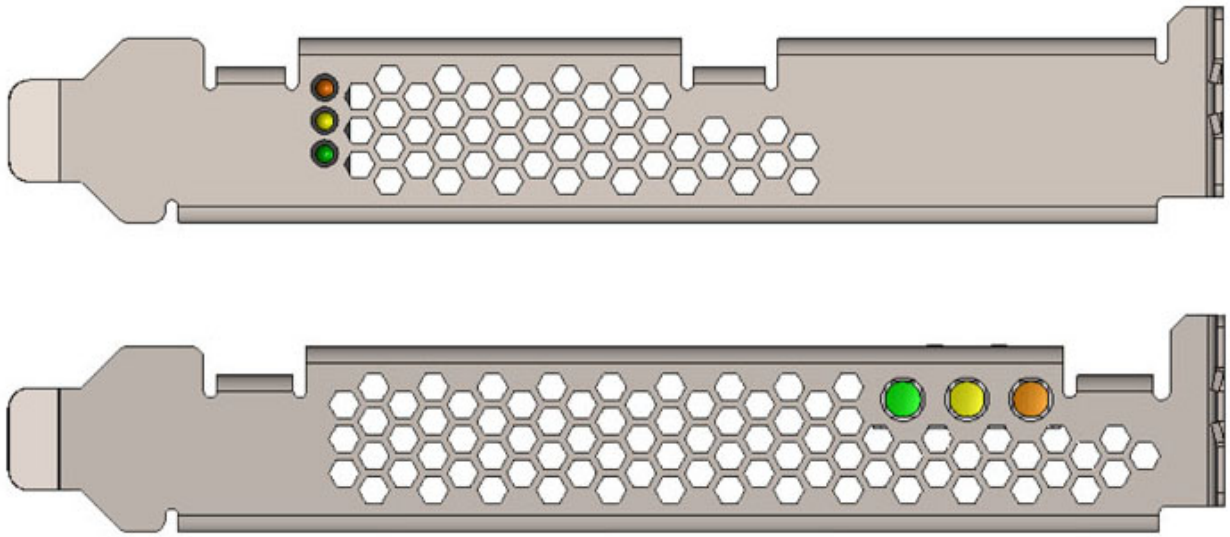
Device LED Indicators

If you have physical access to the devices and depending on your device configuration, you can use the LED indicator (s) on the bracket to monitor their status. Consult the subsections below to determine if your device uses one or more LED indictors and what behaviors are indicated.

> ioMemory devices may have an additional LEDs (that are not on the bracket, as shown below. You can ignore those other LEDs, as they are not meant for monitoring device and software functionality.

Three LEDs

The LEDs on your device should be similar to one of these configurations:

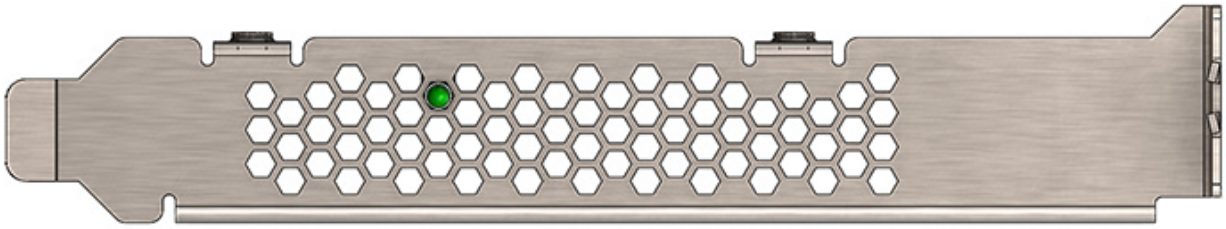


This table explains the information that these LEDs convey:

Green	Yellow	Amber	Indication	Notes
OFF	**OFF**	**OFF**	Power is off.	
OFF	**OFF**	**LIT**	Power is on. Problem with device, or driver not loaded (and device unattached).	Use `fio-status` to view problem, or load driver (and attach device).
LIT	**OFF**	**OFF**	Power is on. Driver loaded (device may or may not be attached).	You may need to attach the device.
LIT	**FLASHING**	**OFF**	Writing (Rate indicates volume of writes).	Can appear in combination with the Read LED indication.
FLASHING	**OFF**	**OFF**	Read (rate indicated volume of reads).	Can appear in combination with the Write LED indication.
LIT	**LIT**	**LIT**	Location Beacon.	Use the `fio-beacon` utility to initiate this behavior.

One LED

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



This table explains the information that the LED conveys:

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

Maintenance

This section explains additional software maintenance functions not covered in the sections Configuration on page 13 and Monitoring and Managing Devices on page 24.

Uninstalling the Software

To uninstall the ioMemory VSL software,

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

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

> ⚠ The ioMemory VSL software utilities are **not** uninstalled with this procedure. If you are upgrading to a newer version of the ioMemory VSL software, 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 software you should manually remove the following folder and its contents: `C:\Program Files\Fusion-io ioMemory VSL`.

Upgrading the Software with a RAID Configuration

> ⚠ Be sure to read the *ioMemory VSL 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 software 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\Common Files\VSL Utils`.)
3. Use the `fio-config` utility to disable auto attach. For example:

   ```
   fio-config -p AUTO_ATTACH 0
   ```

   Your ioMemory device will no longer automatically attach the next time you restart the computer.
4. Uninstall the ioMemory VSL software in **Windows Add/Remove Programs**.
5. Restart the computer.

6. Download the latest ioMemory VSL software package from http://support.fusionio.com .
7. Unzip and install the ioMemory VSL software. While finishing installation, click the "No" button to select a manual restart.
8. Open the ioMemory VSL utilities folder. (The default location is `C:\Program Files\Common Files\VSL Utils`)
9. Use the `fio-config` utility to re-enable auto attach. For example:

```
fio-config -p AUTO_ATTACH 1
```

   Your ioMemory device will now automatically attach the next time you restart the computer.

10. Update the firmware of the devices. Follow the steps in Upgrading the Firmware on page 11.

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

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

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.

Disabling Auto-Attach

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

> You can also use the ioSphere software to enable or disable auto-attach. See the ioSphere software documentation for more information.

1. Open the command-line interface with Administrator permissions.
2. Run the following command:

```
fio-config -p AUTO_ATTACH 0
```

   a. See fio-config on page 38 for more information on setting parameters.

Once you restart your system, your ioMemory device will no longer automatically attach until you re-enable auto attach (see Enabling Auto-Attach on page 31)

When you finish troubleshooting the ioMemory VSL software issue, use the `fio-attach` utility or the ioSphere software 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 command-line interface with Administrator permissions.
2. a. Run the following command:

```
fio-config -p AUTO_ATTACH 1
```

   i. See fio-config on page 38 for more information on setting parameters.

The next time you restart your Windows system, your ioMemory device will automatically attach.

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 ioSphere software's Attach function).

Although data written to the ioMemory device will not be lost due to unmanaged shutdowns, important data structures may not have been properly committed to the device. This consistency check (also called a rescan) repairs these data structures.

Improving Rescan Times

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

Default Fast Rescan

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

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

Faster Rescans Using Module Parameters

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

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

For more information on setting module parameters, see fio-config on page 38.

Here is a quick comparison of the two parameters:

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

RMAP Parameter

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

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

This parameter requires `4.008` bytes of RAM per block of ioMemory device capacity.

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

    This example shows a quick estimation of the number of blocks on a `400GB` device with `512B` size

sectors (2 sectors per KB):

```
400GB * 1000MB/GB * 1000KB/MB * 2 Blocks/kB = 800,000,000 Blocks
```

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

    a. In the example above there were 800 million blocks:

```
800,000,000 Blocks * 4.008B/Block * 1KiB/1024B * 1MiB/1024KiB =
~3058MiB of RAM
```

    b. In this example, you would need about 3100 MiB of RAM available in your system for a 400GB ioMemory device formatted for 512B sectors, and you would need to set the RMAP parameter to 3100.

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

- If the RMAP value is too low for the number of ioMemory device blocks in the system, then the ioMemory VSL software will not use the RMAP process to improve rescan times, it will just use the default fast rescan process. (RMAP is an all-or-nothing setting.)
- If you don't have enough system memory to use the RMAP parameter, consider using the RSORT parameter. The RSORT parameter will use its RAM limit to improve the rescan process, and then the ioMemory VSL softwarerevert to the default fast rescan process to finish the consistency check.

3. Set the module parameter to the value you have determined. See fio-config on page 38 for more information on setting parameters.

RSORT Parameter

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

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

This target is based on 32 bytes per write extent. For example, if your database writes 16kB at a time, there is one write extent per 16kB of ioMemory device capacity.

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

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

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

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

    This example shows a quick estimation of the number of blocks on 1200GB of ioMemory device capacity with 512B size sectors (2 sectors per KB):

    ```
    1200GB * 1000MB/GB * 1000KB/MB * 2 Blocks/kB = 2,400,000,000
    Blocks
    ```

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

    a. In the example above there were `2.4 billion` blocks. We will assume 16KB write extents (`32 blocks` per write on 512B sectors):

    ```
    2,400,000,000 Blocks * 1 Write Extent/32 Blocks = 150,000,000
    Writes
    ```

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

    a. In the example above there were `150 million` write extents:

    ```
    150,000,000 Writes * 32B/Write * 1KiB/1024B * 1MiB/1024KiB =
    ~4578MiB of RAM
    ```

    b. In this example, you would want to set the RSORT limit to about 4600 MiB of RAM available in your system for 1200GB of ioMemory device capacity formatted for 512B sectors.

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

4. Set the module parameter to the value you have determined. See fio-config on page 38 for more information on setting parameters.

Appendix A - Command-line Utilities Reference

The ioMemory VSL software installation packages include various command-line utilities, installed by default in `C:\Program Files\Common Files\VSL Utils`. These provide a number of useful ways to access, test, and manipulate your device.

> 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 ioMemory VSL utilities), and you should not use them directly unless Customer Support advises you to do so.

> **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.

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

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

fio-attach

Description

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

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

> 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 of the device node (`/dev/fctx`), where *x* indicates the device number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first ioMemory device detected on the system.

You can specify multiple ioMemory devices. For example, `/dev/fct1 /dev/fct2` indicates the second and third ioMemory devices installed on the system.

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

fio-beacon

Description

Lights the ioMemory device's LED(s) 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 of the device node (/dev/fctx), where *x* indicates the card number: 0, 1, 2, etc. For example, /dev/fct0 indicates the first ioMemory device detected on the system. The device numbers are visible using fio-status.

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

fio-bugreport

Description

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

Syntax

```
fio-bugreport
```

Notes

This utility captures the current state of the device. When a performance or stability problem occurs with the device, run the fio-bugreport utility and contact Customer Support at 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.

> **Upload Report**
> For best results, do not email the bug report file. Instead please create a case (by emailing Customer Support), and then upload the report to the case using a web browser.

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-20090921_173256.cab.
  Please attach this file to your support case.
```

```
  If you do not have an open support case for this issue, please open a
support
  case with a problem description and then attach this file to your new
case.
```

For example, the filename for a bug report file named `fio-bugreport-20090921_173256.cab` indicates the following:

- Date (20090921)
- Time (173256, or 17:32:56)

fio-config

Description

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

Options	Description
`-e`	Enumerate configuration parameter names and values.
`-g <name>`	Get the configuration parameter.
`-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 software parameters you can set with the `fio-config` utility.

> ⚠ `fio-config` options must be entered in uppercase to function properly.

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{mdash}they apply to all Fusion-io devices in the computer.

> ⚠ By setting the `FIO_PREALLOCATE_MEMORY` and `FIO_EXTERNAL_POWER_OVERRIDE` 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.

Option	Default (min/max)	Description
`AUTO_ATTACH`	`1 (0, 1)`	Always attach the device on driver load (`1`).
`IODRIVE_TINTR_HW_WAIT`	`0 (0, 255)`	Interval (microseconds) to wait between hardware interrupts.
`FIO_EXTERNAL_POWER_OVERRIDE`	No devices selected	Allows selected devices to draw full power from the PCIe slot. Where the `<value>` for this parameter is a comma-separated list of adapter serial numbers. ⚠ Use with care, see [Enabling PCIe Power Override] for more information.
`FORCE_MINIMAL_MODE`	`0 (0, 1)`	Force minimal mode on the device (`1`), this parameter is set to false (`0`) by default.
`PARALLEL_ATTACH`	`0 (0, 1)`	Enable parallel attach of multiple devices (`1`), this parameter is set to false (`0`) by default.
`FIO_PREALLOCATE_MEMORY`	`0`	For the selected device, pre-allocate all memory necessary to have the device usable as swap space. For example: `fio-config /dev/fct0 -p FIO_PREALLOCATE_MEMORY "1234,54321"` where `"1234"` and `"54321"` are serial numbers obtained from `fio-status`.

Option	Default (min/max)	Description
WIN_LOG_VERBOSE	0 (0, 1)	If enabled (1), the ioMemory VSL software will write additional messages to the event log. This will assist Customer Support in troubleshooting any device issues or failures.
WIN_DISABLE_ALL_AFFINITY	0 (0, 1)	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. The driver must be reloaded for this parameter to take effect.
WIN_DISABLE_DEFAULT_NUMA_AFFINITY	0 (0, 1)	When WIN_DISABLE_DEFAULT_NUMA_AFFINITY 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 "default NUMA affinity". 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 WIN_DISABLE_DEFAULT_NUMA_AFFINITY 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.
FIO_AFFINITY	N/A	FIO_AFFINITY is a list of <affinity specification> 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.

Option	Default (min/max)	Description
		For syntax information and examples showing the use of this parameter, see NUMA Configuration on page 72.

fio-detach

Description

Detaches the ioMemory device and removes the corresponding `fctx`ioMemory device 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 of the device node (`/dev/fctx`), where *x* indicates the card number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first ioMemory device detected on the system.

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

Option	Description
`-f`	Force: Causes an immediate detach (does not save metadata). Although the `-f` (force) option causes the ioMemory device to detach, even in a RAID setup, it is strongly recommended to take the drives/volume offline using the Windows Disk Management plug-in, then perform the detach. Forcing the detach may result in loss of data.
`-q`	Quiet: Disables the display of the progress bar and percentage.

Notes

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

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

> ioMemory devices ship pre-formatted, so `fio-format` is generally not required except to change the logical size or block size of a device, or to erase user data on a device. To ensure the user data is truly erased, use `fio-sure-erase`, see fio-sure-erase on page 46 for more information.

Performs a low-level format of the ioMemory 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. You will be prompted as to whether you want to proceed with the format.

> 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.

> Do not interrupt the formatting! We recommend adding power backup to your system to prevent power failures during formatting. If formatting is interrupted, please contact Customer Support.

Syntax

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

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

Options	Description
`-b <size B|K>`	Set the block (sector) size, in bytes or kibibytes (base 2). The default is 512 bytes. For example: `-b 512B` or `-b 4K` (`B` in `512B` is optional).
`-f`	Force the format size, bypassing normal checks and warnings. This option may be needed in rare situations when `fio-format` does not proceed properly. (The "Are you sure?" prompt still appears unless you use the `-y` option.)

`-q`	Quiet mode: Disable the display of the progress-percentage indicator.
`-s <size M|G|T|%>`	Set the device capacity as a specific size (in TB, GB, or MB) or as a percentage of the advertised capacity: • `T` Number of terabytes (TB) to format • `G` Number of gigabytes (GB) to format • `M` Number of megabytes (MB) to format • `%` Percentage, such as 70% (the percent sign must be included)
`-o <size B|K|M|G|T|%>`	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.) ⚠ Before you use this option, please discuss your use case with Customer Support.
`-R`	Disable fast rescan on unclean shutdown to reclaim some reserve capacity.
`-y`	Auto-answer "yes" to all queries from the application (bypass prompts).

You must re-attach the device in order to use the ioMemory device. See fio-attach on page 36 for details.

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.

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

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

Syntax

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

Option	Description
`-d <value>`	`1` = Disable the link; `0` = bring the link up (Not recommended).
`-e`	Enable PCI-e error reporting.

`-f`	Scan every device in the system.
`-n`	Do not perform any writes to config space. Will prevent errors from being cleared.
`-o`	Optimize the ioMemory device PCIe link settings by increasing the maximum read request size if it is too low.
`-r`	Force the link to retrain.
`-v`	Verbose: Print extra data about the hardware.

fio-status

Description

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

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

Syntax

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

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

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

Option	Description
`-a`	Report all available information for each device.
`-e`	Show all errors and warnings for each device. This option is for diagnosing issues, and it hides other information such as format sizes.
`-c`	Count: Report only the number of ioMemory devices installed.
`-d`	Show basic information set plus the total amount of data read and written (lifetime data volumes). This option is not necessary when the \-a option is used.
`-fj`	Format JSON: creates the output in JSON format.
`-fx`	Format XML: creates the output in XML format.
`-u`	Show unavailable fields. Only valid with `-fj` or `-fx`.
`-U`	Show unavailable fields and details why. Only valid with `-fj` or `-fx`.

	Some `fio-status` fields are unavailable depending on the operating system or device. For example, some legacy fields are unavailable on newer ioMemory devices.
`-F<field>`	Print the value for a single field (see the next option for field names). Requires that a device be specified. Multiple `-F` options may be specified.
`-l`	List the fields that can be individually accessed with `-F`.

Output Change
The standard formatting of `fio-status` ouput has changed compared to the output from ioMemory VSL software version 2.x. 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 devices installed in the system
- ioMemory VSL software version

Adapter information:

- Adapter type
- Product number
- External power status
- PCIe power limit threshold (if available)
- Connected ioMemory devices

Block device information:

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

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

- Physical bytes written
- Physical bytes read

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

Adapter information:

- Manufacturer number
- Part number
- Date of manufacture
- Power loss protection status
- PCIe bus voltage (avg, min, max)
- PCIe bus current (avg, max)
- PCIe bus power (avg, max)
- PCIe power limit threshold (watts)
- PCIe slot available power (watts)
- PCIe negotiated link information (lanes and throughput)

Block device information:

- Manufacturer's code
- Manufacturing date
- Vendor and sub-vendor information
- 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
- RAM usage

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

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

fio-sure-erase

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. **It is not recoverable**.

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

After using `fio-sure-erase`, format the device using `fio-format` before using the device again, see fio-format on page 42.

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

In order to run `fio-sure-erase`, the block device **must be detached**. See fio-detach on page 41 section for more information.

Description

The `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

For information regarding certifications please see http://www.fusionio.com/sureerase/. See below for more information on Clear and Purge support.

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 of the device node (`/dev/fctx`), where *x* indicates the card number: 0, 1, 2, etc. For example, `/dev/fct0` indicates the first ioMemory device detected on the system. Use `fio-status` to view this device node, see fio-status on page 44.

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.

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

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

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

Clear Support

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

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

The following describes the steps taken in the Clear operation:

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

Purge Support

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

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

The following describes the steps taken in the Purge operation:

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

fio-update-iodrive

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 ioMemory VSL driver after each firmware upgrade step. Otherwise the on-device 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 at http://support.fusionio.com if you have issues with your upgrade.

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

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 *ioMemory VSL Release Notes* for this ioMemory VSL software 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.

To update one or more specific devices:

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

Syntax

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

where `<firmware-path>` is the full path to the firmware archive file `fusion_<version>.<date>.fff` available at http://support.fusionio.com . If you downloaded the `.fff` firmware archive file, then the firmware is most likely with the other downloaded packages.

Option	Description
`-d`	Updates the specified devices (by `fctx`, where {i}x{i} is the number of the device shown in `fio-status`). If this option is not specified, all devices are updated. Use the `-d` option with care, as updating the wrong ioMemory device could damage your device.
`-f`	Force upgrade (used when directed by Customer Support). Use the `-f` option with care, as it could damage your card.
`-l`	List the firmware available in the archive.
`-p`	Pretend: Shows what updates would be done. However, the actual firmware is not modified.
`-c`	Clears locks placed on a device.
`-q`	Runs the update process without displaying the progress bar or percentage.
`-y`	Confirm all warning messages.
`--split`	Split the ioMemory device into virtual devices.
`--merge`	Merge the virtual devices of an ioMemory device.

If you arrived at this section from Upgrading the Firmware on page 11, you should return to that section.

Appendix B - Troubleshooting Event Log Messages

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

> Each ioMemory device is numbered from 0 upwards. Use the `fio-status` utility or ioSphere software to view this number for your device.

> While these messages are very useful for troubleshooting purposes, the ioMemory VSL log messages are not designed for continual monitoring purposes (as each is based on a variety of factors that could produce different log messages depending on environment and use case). For best results, use the tools described in Monitoring and Managing Devices on page 24 to regularly monitor your devices.

Verbose Event Log Parameter

If you begin experiencing issues with your ioMemory devices, you should enable the `WIN_LOG_VERBOSE` ioMemory VSL parameter. This will expand the extent of the ioMemory VSL error log messages in the event log and provide additional crucial information for troubleshooting any issues.

Sample Command:

```
fio-config.exe -p WIN_LOG_VERBOSE 1
```

For more information on enabling parameters, see fio-config on page 38.

Viewing Logs

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**.
6. Select **System**.

Error Messages

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

Message	Suggested Solution
Error: ioDrive(x) firmware is too old. The firmware must be updated.	Follow the instructions in Upgrading the Firmware on page 11 to update the firmware.
Error: ioDrive initialization failed with error code `0xerrorcode*`	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` utility to rebuild the drive.	This error may appear after an unmanaged shutdown. You can use either the `[fio-attach]` command-line utility or ioSphere software 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 the ioSphere software. 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 on page 31.

* Where `0xerrorcode` is one of the following:

Error Code	Description
0xFFFFFC00	Uncorrectable ECC Error
0xFFFFFBFF	Uncorrectable ECC Error
0xFFFFFBFE	Invalid Media Format
0xFFFFFBFD	Unknown Error

Or one of the 43 standard Windows errno definitions found at http://msdn.microsoft.com/en-us/library/t3ayayh1%28v=vs.110%29.aspx

⚠ The error code is converted to a negative number, and then reported in hexadecimal format. For example, `errno=1` is converted to `-1` and is represented as `0xFFFFFFFF`, and `errno=1024` is converted to `-1024` and is represented as `0xFFFFFC00`.

Informational Messages

The following is a common Event Log informational message:

Message	Additional Information
Affinity not set for ioMemory VSL device fct119 because either WIN_DISABLE_ALL_AFFINITY is set to true or "SetWorkerAffinity119" does not exist in the registry and WIN_DISABLE_DEFAULT_NUMA_AFFINITY is set to true.	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 on page 38 for more information about affinity settings.

Appendix C - Manual Installation

The Windows Setup program will install ioMemory VSL 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 software is installed for a particular device. Repeat the steps for each device, if needed.

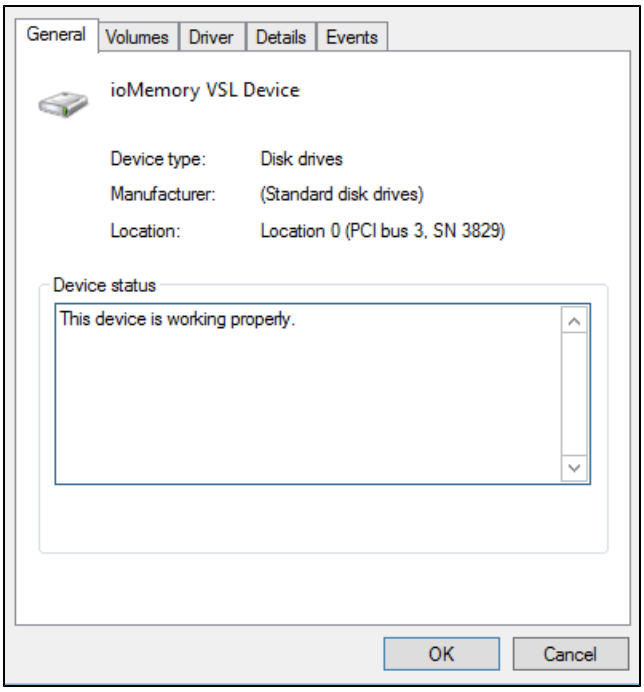
Manual Installation on Windows Server 2003

Before you manually install the ioMemory VSL software, make sure you have downloaded and run the ioMemory VSL Windows Setup program from http://support.fusionio.com . This will install the ioMemory VSL software on the system, and you will now be able to install the ioMemory VSL software for each ioMemory device.

The Windows Driver Wizard may automatically detect the new ioMemory device and starts to locate its ioMemory VSL software 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`.



a. If the **Device Status** reads `This device is working properly`, then the ioMemory VSL software 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.

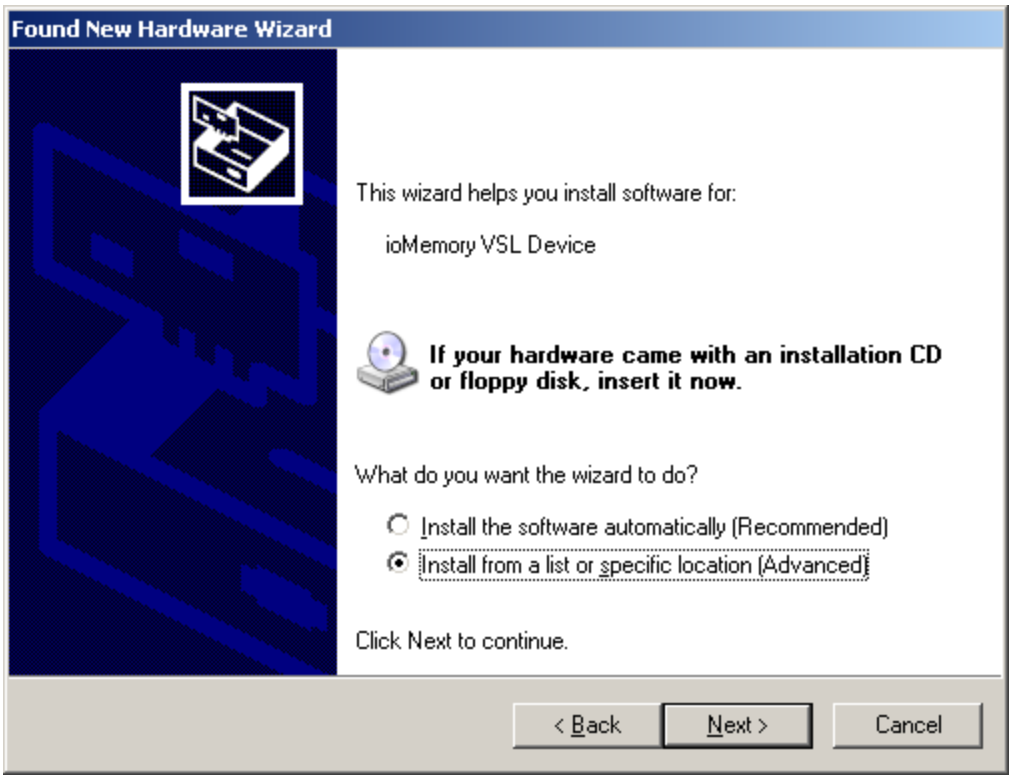
5. Right-click on the device and choose **Update Driver**.

6. Follow the instructions below.

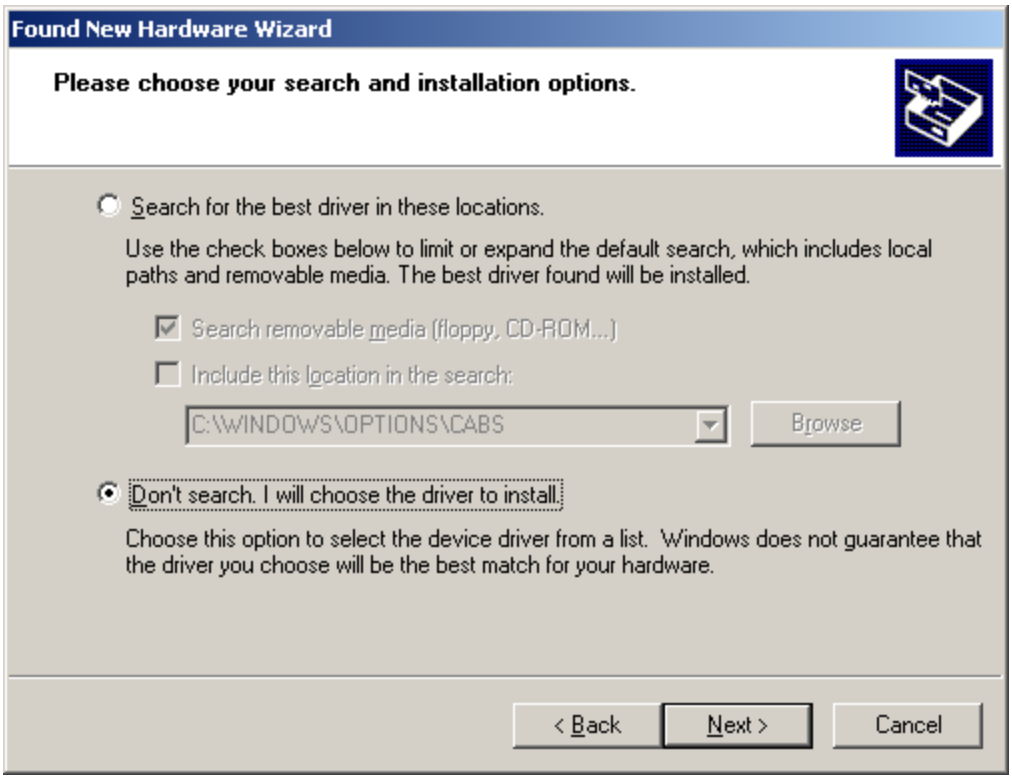
Installation Wizard

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

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

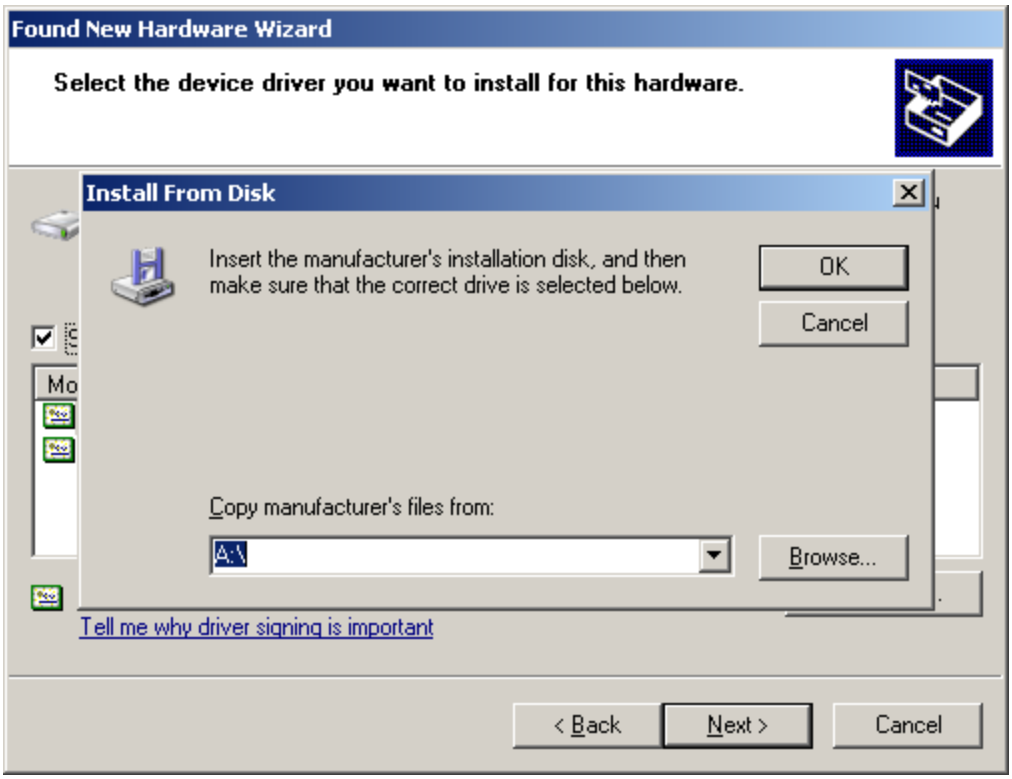


3. Choose **Don't Search. I will choose the driver to install**. Click **Next**

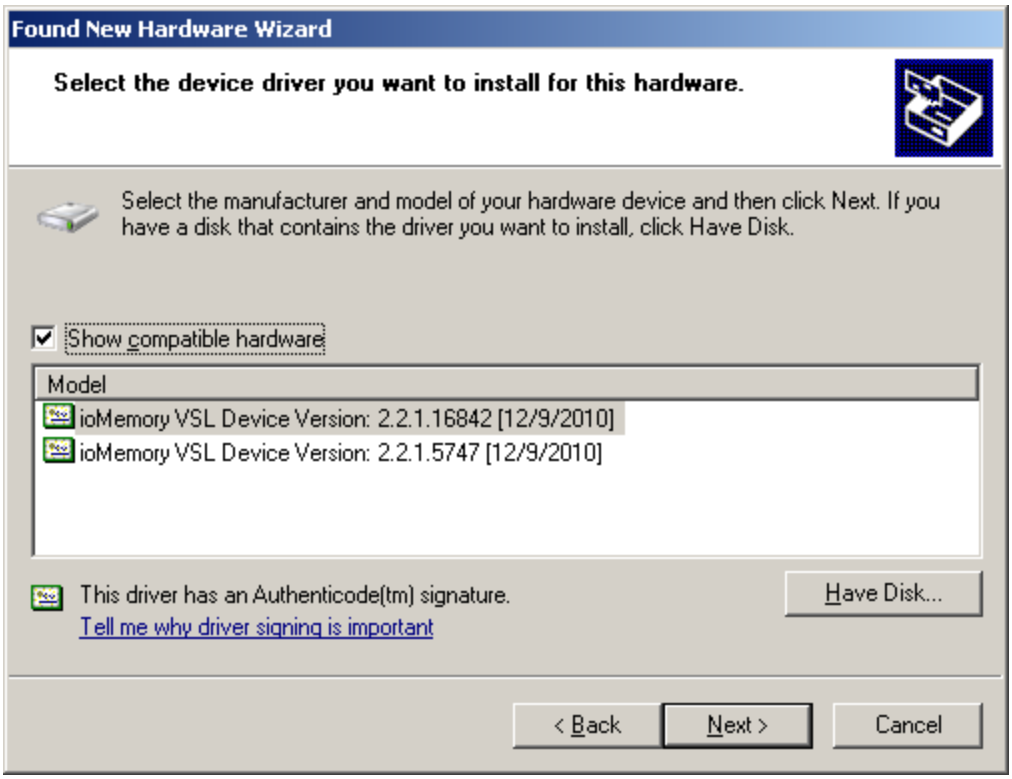


4. Click **Have Disk** to bring up a browsing dialog.

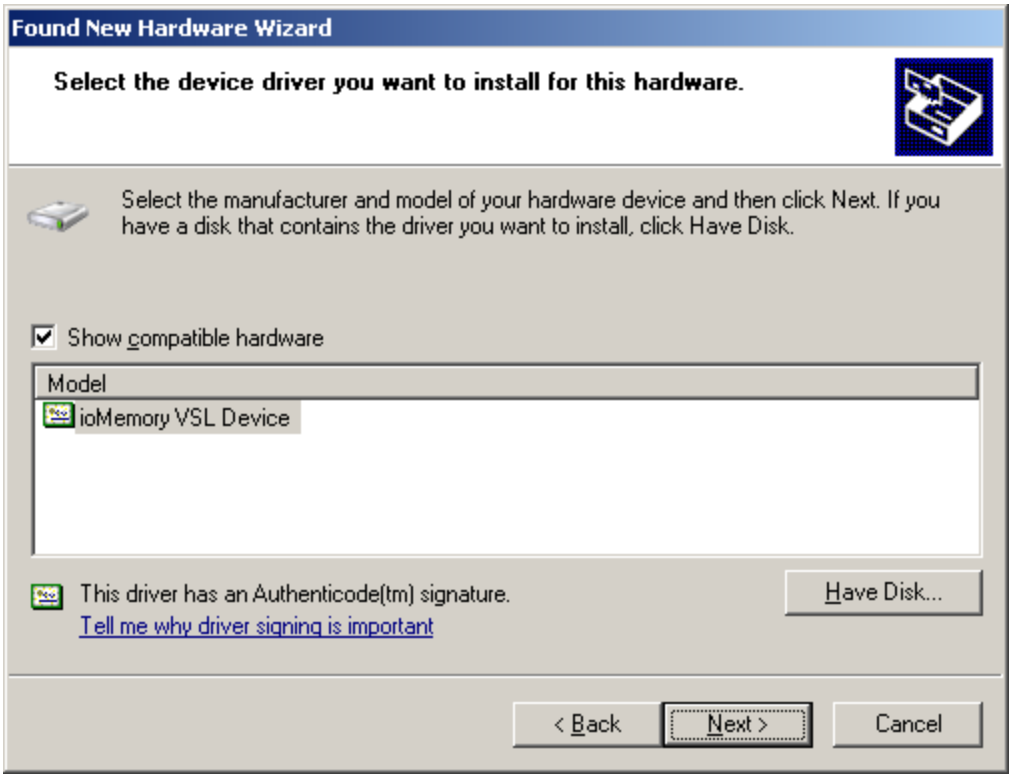
5. Browse to the folder with the ioMemory VSL software (the default is `C:\Program Files\Fusion-io ioMemory VSL\<VSL-Version>\Driver\`). \\


Found New Hardware Wizard
Select the device driver you want to install for this hardware.
Install From Disk
Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.
OK
Cancel
Copy manufacturer's files from:
A:\
Browse...
Tell me why driver signing is important
< Back
Next >
Cancel

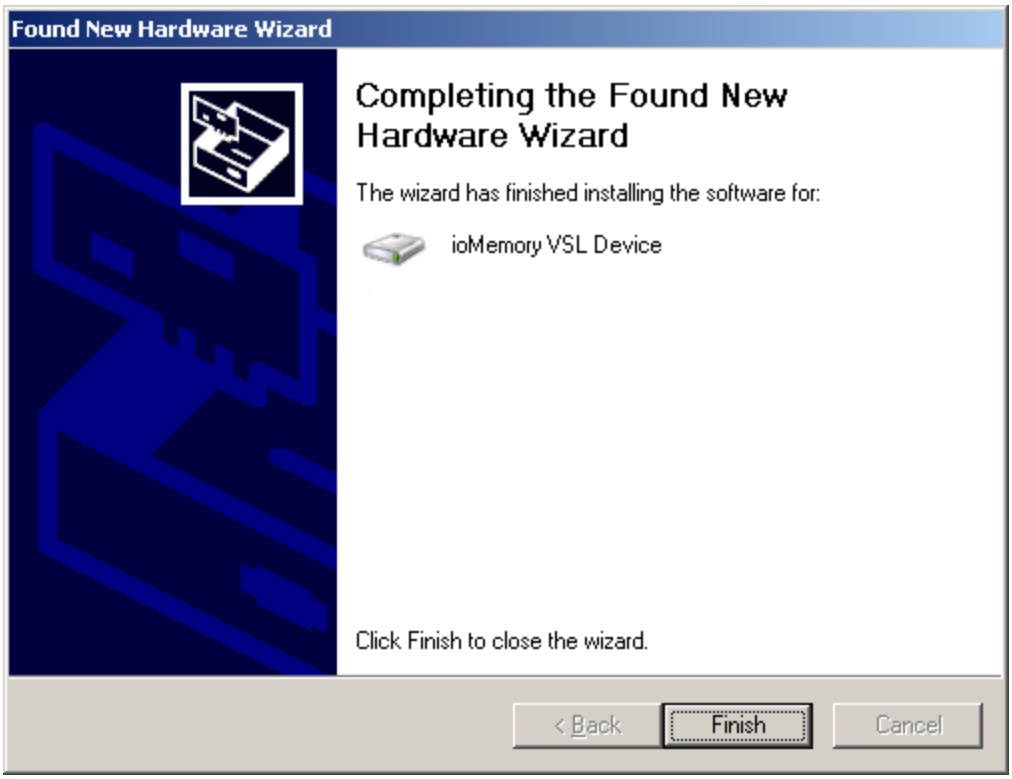
6. Click **OK**.


Found New Hardware Wizard
Select the device driver you want to install for this hardware.
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.
Show compatible hardware
Model
ioMemory VSL Device Version: 2.2.1.16842 [12/9/2010]
ioMemory VSL Device Version: 2.2.1.5747 [12/9/2010]
This driver has an Authenticode(tm) signature.
Tell me why driver signing is important
Have Disk...
< Back
Next >
Cancel

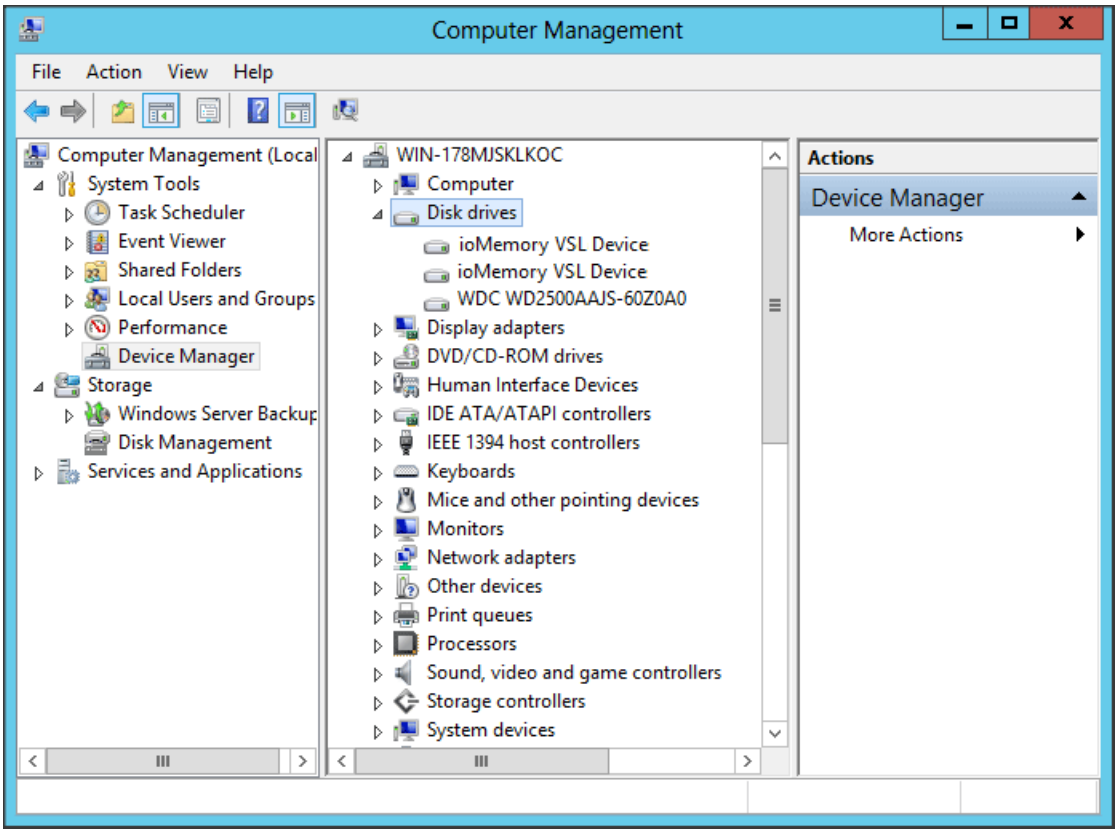
7. The driver you selected should now be in the compatible hardware window. Make sure it is selected and choose **Next**.



When the ioMemory VSL install completes, Windows displays this message:



Now you can view the ioMemory devices in device manager.



If you need to update your firmware, see Upgrading the Firmware on page 11.

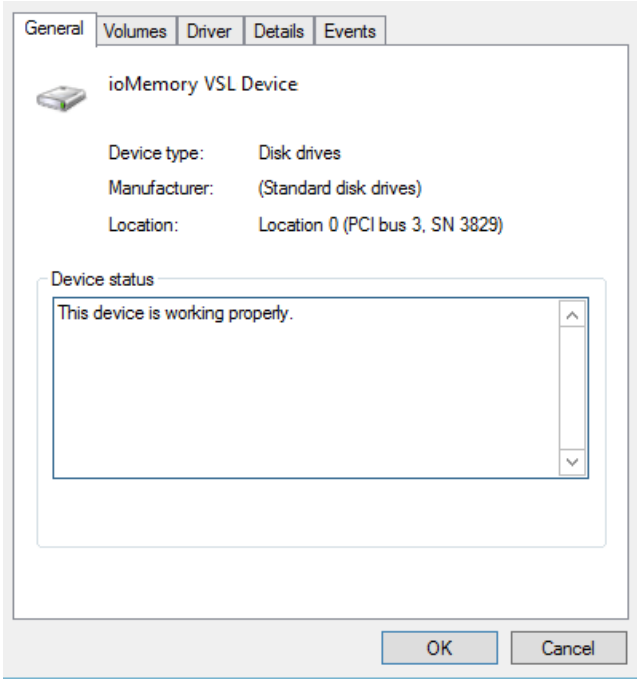
Manual Install on Windows Server 2008 and 2012

Before you manually install the ioMemory VSL software, make sure you have downloaded and run the ioMemory VSL Windows Setup program from http://support.fusionio.com . This will install the ioMemory VSL software on the system, and you will now be able to install the ioMemory VSL software for each ioMemory device.

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

1. Launch the Device Manager.
   - In Windows Server 2008, choose **Start** > **Control Panel** > **Device Manager**.
   - In Windows Server 2012, from the **Server Manager** select **Tools** (in the upper right) > **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`.



   a. If the **Device Status** reads `This device is working properly`, then the ioMemory VSL software 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.

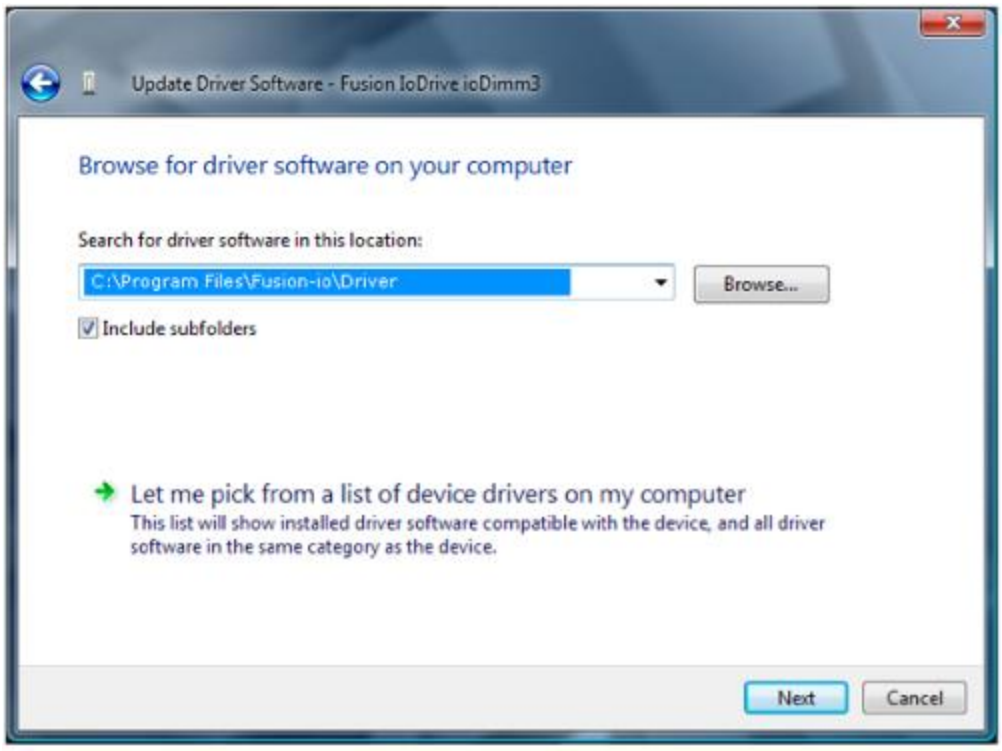
4. Close the Properties dialog box.

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

4. Click **OK**.

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

6. Restart the computer.

7. Proceed to Upgrading the Firmware on page 11 to continue.

Appendix D - Monitoring the Health of Devices

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

Health Metrics

The ioMemory VSL software manages block retirement using pre-determined retirement thresholds. The ioSphere software 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 Health Monitoring Techniques on page 63 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.

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 form which you purchased the device.

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

Health Monitoring Techniques

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

```
Found 3 ioMemory devices in this system
Fusion-io driver version: 3.x.x build xxxx

Adapter: Single Adapter
        Fusion-io ioDrive 1.30TB, Product Number:F00-001-1T30-CS-0001,
SN:1133D0248, FIO SN:1134D9565
...
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
```

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

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

ioSphere software: 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.

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 E - Using the Windows Page Files

Introduction

This appendix describes how to effectively use paging (also called swap or Virtual Memory) 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 software can be configured to support paging files on one or more ioMemory devices. This requires that each ioMemory device 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 device 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.

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 to 75% 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

software 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 *ioMemory VSL Release Notes* for this version of the ioMemory VSL software 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 \[1024x1024 = 1 MiB\]). The total RAM on the system is 8000 MB and the OS is Server 2008 R2.

First, covert the 576 MiB into MB: `576 MiB * (1 MB/1.048576 MiB) = ~549 MB`

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

```
(8000 MB x 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 software initialization. To enable paging support, you must enable the `FIO_PREALLOCATE_MEMORY` configuration item. This can be done using the `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 software 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.

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

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

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

To query the current value, run this command:

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

An alternate method to manage (enable or disable) paging support is to use the ioSphere software.

You must reload the ioMemory VSL software 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

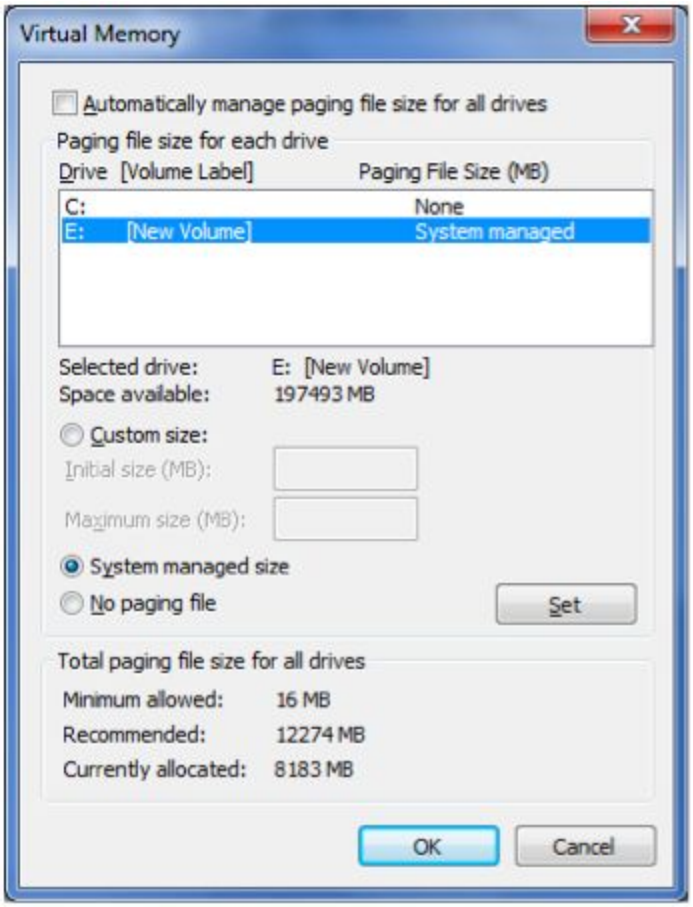
By default, the ioMemory VSL software disables support for page files. The previous section described how to enable support for page files on one or more ioMemory devices. The following describes how to work with the built-in Windows control panels to configure and set up paging files on ioMemory devices.

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.



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. When prompted to restart, click **Yes**. This is necessary for the new page file settings to take effect.
5. Click **Set** to save the setting. Do not omit this step, or your changes will be lost.
6. Click **OK**.

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.

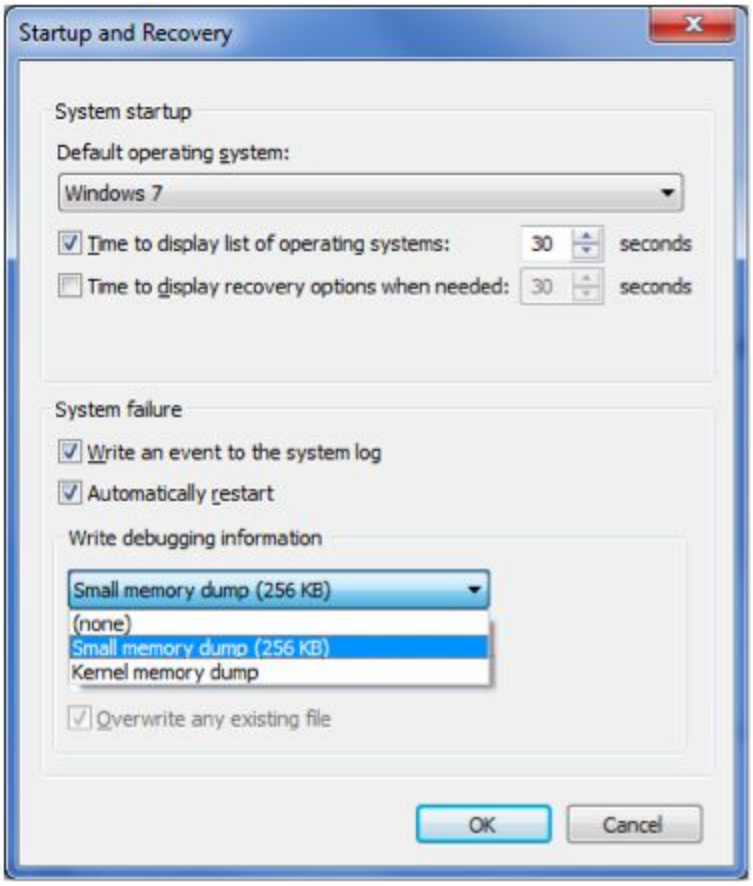
3. Create a page file large enough for a full-kernel memory dump. This typically requires a page file at least the size of installed RAM, with some recommending the size equal to RAM x 1.5.

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.



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:

```
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.

Page File Performance

Using an 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 F - NUMA Configuration

About NUMA Architecture

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

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

Using the `FIO_AFFINITY` Parameter

Use this parameter to map devices with specific 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 and determine the `<device-id>` (see below). Then use the example below of setting the `FIO_AFFINITY` parameter as a template and modify it for your particular system.

Determining the Bus Number

In order to create the correct settings for your specific system, use `fio-status` to list all of the devices' bus numbers. For example:

```
fio-status
Found 2 ioMemory devices in this system
...
       PCI:04:00.0
...
       PCI:0F:00.0
```

> In Windows, the bus number is displayed in hex, but you need to enter the number in decimal. In this example, `04` is `4` and `0F` is `15`.

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

FIO_AFFINITY Parameter

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

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

Syntax:

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

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

```
<[domain number:]bus number>,[g|n]<group or node number>[,<hex mask>]
```

If domain number is not specified, it will be set to `0` (most common).

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 1:7,n0,0xf;20,n1;80,7;2:132,4,0xff0
```

Has the effect of creating :

PCI Address (domain:bus)	Node/Group	Processor Affinity
1:7	node 0	processors 0 to 3 in the node (mask 0xf)
0:20	node 1	all processors in the node (no hex mask)
0:80	group 7	all processors in the group (no hex mask)
2:132	group 4	processors 4 to 11 in the group (mask 0xff0)

Advanced Configuration

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

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

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!
```

```
# fio-status
Found 2 ioMemory devices in this system
...
       PCI:04:00.0
...
       PCI:0F:00.0
```


In Windows, the bus number is displayed in hex, but you need to enter the number in decimal. In this example, `04` is `04` and `0F` is `15`.

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

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

FIO_AFFINITY Parameter

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

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

Syntax:

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

Where each <affinity specification> has the following syntax:

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

Simple Example:

```
fio-config -p FIO_AFFINITY 0000:04:00.0=1,0000:1d:00.0=0,0000:05:00.0=2,
0000:1e:00.0=3
```

Has the effect of creating :

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

Advanced Configuration

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

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

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 G - Upgrading Devices from VSL 2.x to 3.x

This version of the ioMemory VSL software supports new features, including the latest generation of ioMemory architecture and improved Flashback protection. These features require the latest version of the ioMemory device 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.x ioMemory VSL software 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. Consult the ioMemory VSL software for the upgrade path. Visit http://support.fusionio.com 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://support.fusionio.com) for the version you will upgrade the device to. Then follow the upgrade instructions in that version's user guide for your operating system (including the firmware update instructions).

Upgrade Procedure

Be sure to follow the upgrade path in the *ioMemory VSL Release Notes*. Make sure that all previously installed ioDrive devices are updated with the appropriate 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 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` 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 on page 41.

2. Uninstall the 2.x ioMemory VSL software.

    a. Go to **Start** > **Control Panel**.

    b. Click **Programs & Features**.

    c. Select the ioMemory VSL entry.

    d. Click **Uninstall**.

    e. Restart the computer.

3. Install the new ioMemory VSL software.

    a. Download the ioMemory VSL installation program for Windows at http://support.fusionio.com

    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 default install folder (`C:\Program Files\Fusion-io ioMemory VSL`).

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 fio-update-iodrive on page 49.

Sample syntax:

```
fio-update-iodrive <firmware-path>
```

Where `<firmare-path>` is the full path to the firmware archive file (`fusion_<version>.<date>.fff`) available at http://support.fusionio.com . This command will update all of the devices to the selected firmware. If you wish to update specific devices, consult fio-update-iodrive on page 49 for more options.

5. Reboot the system

6. **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 .

7. 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.

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

8. Attach all ioDrive devices, for example:

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

9. 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 software. You may now install any ioDrive2 devices.

Fusion Powered Support

We offer Fusion-io Customer Services and Support by phone, e-mail and on the Web. For the most up-to-date contact information visit at http://support.fusionio.com .

E-Mail

Our support e-mail address is: support@fusionio.com

E-mail is the fastest way to get simple questions answered. Please give a detailed description of your problem with your complete contact information (name, phone number, email address, location address).

Warranty Support

Warranty Support is available via support@fusionio.com and at http://support.fusionio.com .

Telephone Support

ioFX Support

North America: (855) 322-5767

Enterprise Support

North America: (877) 816-5740

Country Numbers

For product support outside of North America, please use the number for the country/region closest to you. If that is not possible, please contact North America at (801) 424 5474.

Australia 1800 353 941 (02) 8278 1489	**Belgium** 02 700 74 86	**China** 40-08866109	**Denmark** 4331 4999	**Finland** 097 251 9979	**France** 01 57 32 48 90
Germany (069) 17 07 76 790	**Hong Kong** 3071 3587	**Italy** 02 23331509	**Japan** (03) 6743-9765	**Luxembourg** (224) 87 19 84	**Mexico** 01 882 816 5740
Netherlands 070 7703993	**Norway** 23 02 49 99	**Singapore** 6818 5692	**South Korea** 02 3483 6689	**Sweden** 08 593 663 99	**United Kingdom** (020) 3564 9935

Web

Go online to find tips, FAQs, and troubleshooting help, or download the latest user guides, software, and support packages at: http://support.fusionio.com .