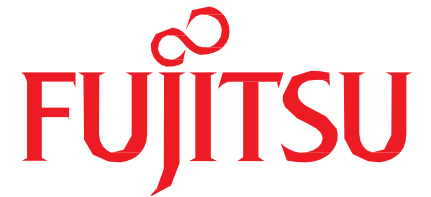


User Guide - 日本語



PCIe SSD PACC EP PX600
ioMemory VSL for Linux

2015 年 6 月

富士通株式会社

著作権および商標

Copyright 2015 FUJITSU LIMITED

商標の確認

Fusion、ioMemoryのロゴデザインやブランド名および製品名は、SanDiskの商標または登録商標です。Microsoft および Windows は、Microsoft Corporation の登録商標です。Linux は、Linus Torvalds の登録商標です。その他すべてのブランド名および製品名は各社の商標です。

目次

著作権および商標.....	2
目次	3
概要.....	5
Fusion ioMemory プラットフォームについて.....	5
パフォーマンス.....	5
耐久性.....	5
信頼性.....	5
Fusion ioMemory VSL ソフトウェアのインストール.....	6
インストールの概要.....	6
RPM パッケージのインストール	6
Fusion ioMemory VSL Driver のロード.....	7
ドライバロードの制御.....	8
init スクリプトの使用	8
init スクリプトを使用している場合のファイルシステムのマウント	8
udev の使用.....	9
udev と init スクリプトの両方	9
Fusion ioMemory デバイスとマルチパスストレージ.....	10
ドライバのアンロードの制御.....	10
ファームウェアのアップグレード.....	10
コマンドラインインターフェース.....	10
設定.....	12
Fusion ioMemory VSL オプションの設定.....	12
一度だけの設定.....	12
固定的な設定.....	12
PCIe Power Override の有効化	12
自動的な電力取得.....	12
最大電力の有効化.....	13
Override パラメータの有効化.....	13
シリアル番号の特定.....	13
パラメータを設定する	14
デバイスを Swap として使用する	14
メモリ割り当て予約量を設定.....	15
Logical Volume Manager を使用する	15
init スクリプトを使用している場合のファイルシステムのマウント	16
Logical Volume Manager を使用して RAID を設定する	16
LVM で stripe ボリューム (RAID 0) を作成する	16
LVM で mirror ボリューム (RAID 1) を作成する	16
Mdmadm を使用して RAID を設定する	17
アレイのマウント.....	17
RAID 0.....	17
アレイを固定する(再起動時にも検出).....	18
RAID 1.....	18
RAID 10.....	19
Discard (TRIM) のサポート	19
Linux での Discard (TRIM)	19
パフォーマンスとチューニング.....	20
CPU Frequency Scaling の無効化.....	20
ACPI C-State を制限する.....	20
ACPI C-State オプションを設定する	20
Linux での C-State	20
NUMA Affinity の設定.....	21
Interrupt Handler Affinity の設定	21
Linux での IRQ Balancing.....	21

デバイスの監視・管理	22
管理ツールの使用	22
監視すべき状態の表示例	22
デバイスのステータス	22
必要なアクション	23
温度	23
必要なアクション	23
Health Reserves のパーセンテージ	23
必要なアクション	23
書き込み (Health Reserves) のステータス	23
必要なアクション	24
デバイスの LED Indicator	24
メンテナンス	25
Fusion ioMemory VSL ソフトウェアのアンインストール	25
ソフトウェアドライバのアンロード	25
Kernel のアップグレード	25
Fusion ioMemory VSL ソフトウェアの無効化	25
Auto-Attach の無効化	26
予期しないシャットダウンの問題	26
Rescan の時間を改善する	26
デフォルトの Fast Rescan	26
モジュールパラメータを使った Faster Rescan	26
RMAP パラメータ	27
RSORT パラメータ	28
付録 A - コマンドラインユーティリティのリファレンス	31
fio-attach	31
fio-beacon	32
fio-bugreport	32
fio-detach	33
fio-firmware	34
fio-format	35
fio-pci-check	36
fio-status	36
fio-sure-erase	39
Clear のサポート	40
Purge のサポート	40
fio-update-iodrive	41
オンラインでのファームウェアアップデート	41
付録 B - デバイスの Health 情報を監視	44
Health 情報の法則	44
Health 監視の手法	44
ソフトウェア RAID と Health 監視	45
付録 C - モジュールパラメータの使用	46
付録 D - NUMA の設定	48
NUMA アーキテクチャーについて	48
numa_node_override パラメータの使用	48
Device ID を特定する	48
numa_node_override パラメータ	49
構文:	49
簡単な例:	49
拡張設定	49
お客様サポート	50

概要

弊社のソリッドステートストレージデバイスをご購入いただき、ありがとうございます。このガイドでは、Fusion ioMemory デバイスのための Fusion ioMemory VSL ソフトウェアをインストール、トラブルシュート、また管理する手順を説明しています。

Fusion ioMemory プラットフォームについて

Fusion ioMemory プラットフォームは、Fusion ioMemory VSL (Virtual Storage Layer)ソフトウェアと Fusion ioMemory ハードウェアで構成されており、エンタープライズアプリケーションやデータベースを次世代のレベルへ上げることができます。

パフォーマンス

Fusion ioMemory プラットフォームは、複雑なワークロードにも安定的にマイクロ秒単位のレイテンシ、複数ギガバイト秒のアクセス、また 10 万以上の IOPS を提供します。洗練された Fusion ioMemory アーキテクチャーにより、読み込みと書き込みでほぼ同一かつ高い水準のパフォーマンスを可能にし、Fusion ioMemory プラットフォームによりエンタープライズ環境のシステムを改善することができます。

Fusion ioMemory プラットフォームは、ホストの CPU とフラッシュメモリ間を、複数のコアにまたがって直結します。このプラットフォームにより、システムの CPU あたりの稼働率を上昇させることができます。

耐久性

ioMemory プラットフォームは、すべての容量のデバイスで高い水準の耐久性を持っており、キャッシュや高負荷のデータベース書き込みアクセスの要件を満たします。

信頼性

Fusion ioMemory プラットフォームは、NAND の故障や寿命などの懸念を縮小します。高度な自己修復機能である Adaptive Flashback® により、チップレベルでの冗長性を持っています。Adaptive Flashback テクノロジーにより、Fusion ioMemory 製品はチップ単体または複数の不良から復旧し、ビジネスを中断させません。

Fusion ioMemory VSL ソフトウェアのインストール

インストールを継続する前に、次のことを確認してください：

1. ご使用のオペレーティングシステムに対応していることを確認してください。
2. Fusion ioMemory デバイスが正しく取り付けられていることを確認してください。

i すべてのコマンドには root 権限が必要です。インストールするには、sudo または"root"でログインしてください。

インストールの概要

1. 最新版のソフトウェアをダウンロードします。<http://support.ts.fujitsu.com> (Global 市場の場合) / <http://jp.fujitsu.com/platform/server/primergy> (日本市場の場合)
2. すでに以前のバージョンの Fusion ioMemory VSL ソフトウェアがインストールされている場合、あらかじめアンインストールします。

- このバージョンの Fusion ioMemory VSL ソフトウェアは、Fusion ioMemory PX600 デバイスのみをサポートしており、以前のデバイスとは互換性はありません。

カーネルのアップグレード

カーネルのアップデートが予定されている場合、[別ページにある手順](#)を参照してください。

3. Fusion ioMemory VSL ソフトウェアをインストールします。
4. ユーティリティや管理ソフトウェアをインストールします。
4. Fusion ioMemory VSL ドライバをロードし、また必要に応じてオプションを設定します。
5. ファームウェアの更新が必要かを確認し、また必要に応じて更新を行います。

RPM パッケージのインストール

Linux 用の Fusion ioMemory VSL ソフトウェアをインストールするには：

1. カーネルに合った Fusion ioMemory VSL ソフトウェアのバージョンを使用する必要があります。カーネルのバージョンを参照するには、シェルプロンプトで次のコマンドを実行します：

```
$ uname -r
```

2. カーネルのバージョンと、ダウンロードしたソフトウェアのバージョンが一致していることを確認します。
 - 例：

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

w パッケージ名はソフトウェアやカーネルのバージョンにより異なります。

3. 追加のユーティリティ、ファームウェア等をダウンロードします。

例：

パッケージ	内容
fio-common-<version>.rpm	Fusion ioMemory VSL ユーティリティ – 推奨
fio-sysvinit-<version>.rpm	

パッケージ	内容
fio-util-<version>.rpm	
fio-firmware-fusion_<version>-<date>-1.0.noarch.rpm	ファームウェアアーカイブを特定の場所(/usr/share/fio/firmware)へインストールするためのパッケージ- オプション
fio-firmware-fusion_<version>-<date>.fff	ファームウェアアーカイブ - 推奨

- ダウンロードしたパッケージのあるディレクトリへ移動します。
- パッケージをインストールするため、次のコマンドを実行します。ファイル名はダウンロードしたパッケージにより異なります :

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

- 追加のパッケージをインストールするため、次のコマンドを実行します :

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

i fio-util-<VSL-version>.x86_64.rpm パッケージのインストールが依存関係の問題で失敗した場合、不足しているパッケージをインストールします。例 :

```
$ yum install lsof pciutils
```

Fusion ioMemory VSL ソフトウェアとユーティリティは、次の場所にインストールされます :

パッケージ種類	インストール先
Fusion ioMemory VSL ソフトウェア	/lib/modules/<kernel-version>/extra/fio/iomemory-vsl4.ko
ユーティリティ	/usr/bin

i 必要に応じて、追加で ServerView RAID Manager をインストールします。詳細な手順は同ソフトウェアのドキュメントを参照してください。


Fusion ioMemory VSL Driver のロード

Fusion ioMemory VSL Driver をロードするには、次のコマンドを実行します :

```
$ modprobe iomemory-vsl4
```

i Fusion ioMemory VSL Driver はシステムのブート時に自動的にロードされます。また、デバイスは /dev/fiox (x は文字、a、b、c など) として有効になります。

Fusion ioMemory デバイスが attach されていることを確認するには、fio-status コマンドを実行します。出力にドライブとステータスが表示されます。

 Fusion ioMemory デバイスが自動的に attach されなかった場合、`/etc/modprobe.d` ファイルを確認し `auto_attach` オプションが有効になっていること(1 に設定されていること)を確認してください。

SLES システムでは、未サポートのモジュールを実行できるよう許可する必要があります。
`/etc/modprobe.d/iomemory-vsl4.conf` を編集し該当の行をコメントアウトします：

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

ドライバロードの制御


ドライバのロードは、`init` スクリプトまたは `udev` で制御できます。

新しい Linux ディストリビューションでは、ユーザーは `udev` デバイスマネージャーに従いブート時にドライバとインストールされたハードウェアを見つけることができるため、`udev` を無効化し `init` スクリプトは必要なくなります。`init` スクリプトでの Fusion ioMemory VSL Driver のロードは、LVM、`mdadm` や Veritas Storage Foundation を使用した RAID アレイ管理を行う場合に推奨します。

古い Linux ディストリビューションで、`udev` 機能がない場合、ユーザーは `init` スクリプトを使用して必要なドライバをロードすることになります。

init スクリプトの使用

`Udev` でのロードが動作しないか無効化されたシステムでは、`init` スクリプトが使用されます。いくつかのディストリビューションでは、これがデフォルトで有効になっています。

 `init` スクリプトは `fiio-sysvinit` パッケージの一部になっており、有効化する前にあらかじめインストールする必要があります。

ドライバのロードは、次のコマンドで無効化することができます：

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

再度有効化するには、次のコマンドです：

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

Fusion ioMemory VSL ソフトウェアインストールのプロセスは、`init` スクリプトを `/etc/init.d/iomemory-vsl4` に配置します。このとき、スクリプトは設定オプションを `/etc/sysconfig/iomemory-vsl4` から読み取ります。オプションのファイルは必ず有効(ゼロではない)に設定されている必要があります：

```
ENABLED=1
```

オプションのファイルは、`MOUNTS` や `KILL_PROCS_ON_UMOUNT` など複数のスイッチが用意されています。詳細は別のページで説明しています。


init スクリプトを使用している場合のファイルシステムのマウント

Fusion ioMemory VSL は、ファイルシステムのマウントに標準的な `initrd` やカーネル同梱の手法(`/etc/fstab`)を使用していないため、自動ではマウントされません。自動で Fusion ioMemory デバイスをマウントするには：

1. ファイルシステムをマウントするコマンドを、通常どおり `/etc/fstab` に追加します。
2. 次の例のように、`'noauto'` オプションと `'0 0'` フラグを追加します(必要に応じてその他は変更してください)

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

(fioa は a、b、c など、使用している Fusion ioMemory デバイスの数によります。)

 `'noauto 0 0'` が指定されなかった場合、ブートが失敗することがあります。

`init` スクリプトによりドライバロード後にデバイスをマウントし、またデバイスをアンマウントしてからドライバをアンロードするには、オプションファイルにマウントポイントを追加します。

前の例にあるファイルシステムの場合、オプションは次のようになります：

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

udev の使用

`udev` を使用できる場合、ユーザーは Fusion ioMemory VSL ソフトウェアオプションを使用して、ブート時の自動ロードを抑止することができます。これをする場合、`/etc/modprobe.d/iomemory-vsl4.conf` の次の行を編集します：

```
# blacklist iomemory-vsl4
```

ロードを無効化するには、`"#"` を削除しファイルを保存します。

`Blacklist` コマンドは Linux 再起動で動作します。Fusion ioMemory VSL ドライバは `udev` によりロードされません。

ロードを再度有効化するには、`"#"` を戻しファイルを保存します。

udev と init スクリプトの両方

ユーザーはブート時のドライバロードを無効化することができ、そうすることで診断やトラブルシューティングを目的として自動 `attach` を抑止することができます。別のセクション「自動 `attach` の無効化」を参照してください。

代わりに、ブートローダーに次のカーネルパラメーターを追加することで、ioMemory VSL ドライバのロードを抑止することもできます：

```
iomemory-vsl4.disable=1
```

しかし、この手段ではドライバのすべての機能を停止し、実行可能なトラブルシューティングを減らすことになるため、推奨しません。

ブートローダーに次のパラメータを追加することで、自動 `attach` を無効化することもできます：

```
iomemory-vsl4.auto_attach=0
```

Fusion ioMemory デバイスとマルチパスストレージ

もし Fusion ioMemory デバイスをマルチパスストレージと共に使用している場合、Fusion ioMemory デバイスは blacklist し、device-mapper による dm-device の作成を抑止する必要があります。これは **dm-multipath** を有効化する前、かつドライバをロードする前に行われる必要があります。Fusion ioMemory デバイスが blacklist されていない場合、デバイスは busy 状態となり、atath、detach、ファームウェアのアップデートなどの操作が行えなくなります。

Fusion ioMemory デバイスを blacklist するには、/etc/multipath.conf ファイルを次のように編集します：




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

ドライバのアンロードの制御

Fusion ioMemory ドライバをアンロードする際、特別に考慮すべきことがあります。デフォルトでは、init スクリプトはファイルシステム上でオープンされているあらゆるプロセスを停止し、結果ファイルシステムをアンマウントできるようになります。この動作はオプションファイルの KILL_PROCS_ON_UMOUNT オプションにより制御されます。これらのプロセスが停止されない場合、ファイルシステムはアンマウントできません。これにより、ioMemory VSL ソフトウェアを正しくアンロードすることになります。

ファームウェアのアップグレード

Fusion ioMemory VSL ソフトウェアがロードされると、Fusion ioMemory デバイスのファームウェアが最新版であることを確認する必要があります。また必要に応じてアップデートする必要があります。この操作はコマンドラインユーティリティにより行うことができます。

-  Fusion ioMemory VSL ソフトウェアの版数に一致した、ファームウェアのアーカイブがダウンロードされていることを確認してください。
-  ファームウェアのダウングレードは行わないでください。追加の Fusion ioMemory デバイスをインストールする場合、すべてのデバイスを同一の最新版へアップグレードする必要があります。
-  **ゲスト OS のアップグレード**
Fusion ioMemory デバイスをゲスト OS 側で使用している場合(例えば VMDirectPathIO など)、ホストサーバのパワーサイクルを行う必要があります。仮想マシンの再起動では、ファームウェアは適用されません。

コマンドラインインターフェース

コマンドラインユーティリティの詳細は、「[コマンドラインユーティリティのリファレンス](#)」で説明しています。すべてのコマンドは root 権限が必要です。ファームウェアをアップグレードするには、次のステップを行います：

1. fio-status ユーティリティを実行し、表示を確認します。
 - デバイスが minimal mode であり、また理由がファームウェアが古いための場合、ファームウェアはアップグレードしなければなりません。

-
- デバイスが `minimal mode` ではないが、ファームウェアが最新版ではない場合、ファームウェアのアップグレードを推奨します。
2. ファームウェアが古い場合、`fio-update-iodrive` ユーティリティを使用してアップグレードを行います。

設定

Fusion ioMemory デバイスと Fusion ioMemory VSL ソフトウェアがインストールとロードされ、ファームウェアが最新版になっていると、必要に応じてデバイスやソフトウェアを設定することができます。この章では、いくつかの設定について紹介しています。

Fusion ioMemory VSL オプションの設定

この章では、Fusion ioMemory VSL ソフトウェアの設定について説明しています。より詳細については、「[モジュールパラメータの使用](#)」を参照してください。

一度だけの設定

Fusion ioMemory VSL ソフトウェアのオプションは、インストール時に `insmod` や `modprobe` コマンドで設定することができます。例えば、`auto_attach` オプションを 0 に設定するには、次のコマンドで行います：

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

このオプションは、Fusion ioMemory VSL ソフトウェアをロードした一度だけ有効になります。次のロード時には有効になりません。

固定的な設定

固定的な設定を行うには、`/etc/modprobe.d/iomemory-vsl4.conf` または類似したファイルに設定を行います。Fusion ioMemory デバイスの自動 attach を抑止するには、`iomemory-vsl4.conf` ファイルに次の行を追加します：

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

このオプションは、Fusion ioMemory VSL ソフトウェアをロードする際やブートの際に毎回有効になります。

PCIe Power Override の有効化

Fusion ioMemory PX600 デバイスは、標準である PCIe Gen2 x8 スロットからの 25W 以上の電力でも動作します。25W でも動作しますが、パフォーマンスはその分に制御されます。

さらに高いパフォーマンスで動作させるには、Fusion ioMemory デバイスをより高い電力で動作させる必要があります、PCIe からさらに高い電力を供給させる必要があります。

自動的な電力取得

いくつかの PCIe スロットは、追加の電力を供給できることがあります(一般的に最大 75W)。標準の 25W より高い電力の供給が可能な場合、またシステム BIOS が正しく電力情報を提供している場合、Fusion ioMemory デバイスは自動的に高い電力を取得します(電力あたりの最大パフォーマンスまで)。

Fusion ioMemory VSL ソフトウェアは、追加の電力取得についてシステムログで通知します。例えば、次の通知は電力制限を 75W にした場合です：

```
fioinf <device> 0000:11:00.0: PCIe Slot reported power limit: 75000mWatts  
fioinf <device> 0000:11:00.0: PCIe Adapter power limit: 75000mWatts  
fioinf <device> 0000:11:00.0: PCIe Adapter power Throttle point: 74750mWatts
```

最大電力の有効化

もし PCIe スロットがさらに追加の電力を供給できるが(システムのスペックを確認してください)、BIOS が正しく電力情報を提供しない場合、VSL モジュールのパラメータを使用して、PCIe 最大電力を変更することができます。

i このパラメータでは、PCIe スロットの 25W 上限を強制的に上書きします。このパラメータは、デバイスそれぞれに対して指定します(シリアル番号を使用します)。設定が有効になると、デバイスが使用する最大電力が使用されます。

o もし PCIe スロットが供給できる電力に限界があり、また設定により最大電力が有効になると、サーバのハードウェアが故障する可能性があります。十分に仕様を確認せず、ハードウェアが故障してしまった場合、動作の保証や修理は受け付けられません。

このパラメータを有効にする前に、サーバのドキュメントを参照し、PCIe スロットが電力供給に対応しているか確認してください。

次の事項について検討する必要があります：

- 複数の Fusion ioMemory デバイスが搭載されている場合、それらのすべての PCI スロットが最大電力の供給に対応していること、また合計しても最大電力を超えないことを確認してください。

i 例えば、マザーボードによっては、特定の PCIe スロットでのみ 75W を供給可能だが、最大値は複数スロットで共有していることがあります。このとき、強制的に最大電力を取得し、限界を超えた場合、サーバのハードウェアを破損する可能性があります。

- override パラメータが適切に有効化されると、システムへ固定的に設定されるため、デバイスが別の PCIe スロットへ移動した場合にも最大電力を取得します。これにより、別のスロットが破損する可能性があります。
- override パラメータは Fusion ioMemory VSL ソフトウェアは、ソフトウェア上の設定値であり、デバイス側には保存されません。デバイスが別のサーバへ移動した場合、デフォルトの 25W で動作します。

Override パラメータの有効化

シリアル番号の特定

このパラメータを有効にする前に、デバイスのシリアル番号を特定する必要があります。fio-status コマンドラインユーティリティを使用します。

i シリアル番号ラベル
または、デバイスに貼り付けられたシリアル番号ラベルを参照することもできます。ただし、この設定はソフトウェア上の設定で動作するもののため、fio-status コマンドを使ってソフトウェアでの確認を推奨します。

- fio-status: fio-status コマンドを実行します。次は表示例です：

```
fio-status
...
Adapter: ioMono ioMemory PX600-5200, Product Number:F14-005-5200-CS-0001, SN:1331G0009, FIO
SN:1331G0009
External Power: NOT connected
```

PCIe Power limit threshold: 24.75W

Connected ioMemory modules: fct0: Product Number:F14-005-5200-CS-0001, SN:1331G0009

この例では、1331G0009 がシリアル番号です。

- **fio-beacon**: 複数のデバイスが搭載されている場合、**fio-beacon** ユーティリティで物理的な搭載位置を特定することができます。


パラメータを設定する

モジュールのパラメータを設定するには、`/etc/modprobe.d/iomemory-vsl4.conf` ファイルを編集し、`external_power_override` パラメータを追加します。例：

```
options iomemory-vsl4 external_power_override=<SN-value>:<mW-value>
```

このパラメータの `<SN-value>:<mW-value>` は、コンマで区切られた値のペアで、シリアル番号と電力の最大値(ミリワット)を指定します。

例えば、`1149D0969:40000,1159E0972:40000,1331G0009:40000` と指定した場合、3つのデバイス(シリアル番号 `1149D0969, 1159E0972, 1331G0009`) について、最大約 40W の供給となります。


 パラメータの変更を適用するには、再起動またはリロードする必要があります。


デバイスを Swap として使用する


Fusion ioMemory デバイスを **Swap** として使用するには、`preallocate_memory` カーネルパラメータを指定する必要があります。このパラメータを設定するには、`/etc/modprobe.d/iomemory-vsl4.conf` へ行を追加します：

```
options iomemory-vsl4 preallocate_memory=1149D2717,1149D2717,1331G0009
```

- この場合、`1149D2717-1111,1149D2717-1111,1331G0009` は `fio-status` で参照できるシリアル番号です。

 アダプター(adapter)ではなく、Fusion ioMemory デバイスシリアル番号を提供する必要があることに注意してください。
`fio-format` を使用して、Fusion ioMemory デバイスを別のセクタサイズへ設定できます。4K セクタサイズのフォーマットが推奨されます—これにより、Fusion ioMemory VSL ソフトウェアのフットプリントを縮小します。代わりに、512B セクタにフォーマットすることもできます。しかし、512B セクタではより大きなメモリ消費となります。

 Fusion ioMemory デバイ스에メモリ割り当て予約を行い **swap** として使用するには、十分な RAM が必要です。Fusion ioMemory デバイスを `attach` しメモリ割り当て予約をした際、十分ではない RAM だった場合、プロセスの消失や不安定なシステムとなる可能性があります。

 `preallocate_memory` パラメータは、Fusion ioMemory VSL ソフトウェアをロードする際に読み込まれますが、メモリの割り当てはデバイスが `attach` された際に行われます。

メモリ割り当て予約量を設定

デバイスへ予約を有効化した場合、Fusion ioMemory VSL ソフトウェアは自動的にフォーマットされたセクタサイズに合わせてメモリを獲得します。もしセクタサイズが 4KiB より小さい場合(例えば 512B セクタ)、Fusion ioMemory VSL ソフトウェアはとて大量のメモリを獲得します(512B で予測される最大値)。

メモリの必要量は、製品の仕様書を参照ください。メモリが不足した場合、Fusion ioMemory デバイスやシステム全体が動作しません。

オペレーティングシステムは通常、4KiB ブロックの仮想メモリを使用するため、一般的には 4KiB セクタの場合のメモリ量を予約すれば十分です。Fusion ioMemory VSL ソフトウェアに 4KiB ブロックサイズをベースに予約させるには、次の 2 つの手段があります：

- デバイスを 4KiB セクタにフォーマットする `fiio-format` を使用して行うことができます。
- `preallocate_mb` パラメータを使用する このパラメータでは、Fusion ioMemory VSL ソフトウェアが予約するメモリ量を指定することができます。

❗ 確実に十分なメモリ量が指定されていることを確認してください。デバイスが `swap` として動作するためにメモリ量が不足(計算ミスや入力ミス)していた場合、システムのリソース不足によりクラッシュする可能性があります。

1. 各 Fusion ioMemory デバイスごとに、必要なメモリ量の最大値を特定します。
セクタサイズごとの必要なメモリ量は、仕様を参照してください。複数台のデバイスが搭載されている場合、合計して考慮する必要があります。
2. `/etc/modprobe.d/iomemory-vsl.conf` ファイルに値を設定します：

```
options iomemory-vsl preallocate_mb=<value>
```

<value> は MB 単位のメモリ量です。例えば 3500 と入力した場合、Fusion ioMemory VSL ソフトウェアは、デバイス 1 台あたりおよそ 3.5GB の RAM を予約します。

⚠ メモリ予約を効果的にするために、この値はデフォルトのメモリ消費量より大きくしなければなりません(`fiio-status -a` コマンドで参照できます)。

Logical Volume Manager を使用する

Logical Volume Manager (LVM) ボリュームグループ管理アプリケーションは、Fusion ioMemory デバイスのような mass storage device を制御します。次のように指定します：

1. `/etc/lvm/lvm.conf` 設定ファイルを探し、編集します。
2. 次の例のようなエントリを追加します：

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

パラメータ"16"は、デバイスがサポートしている最大パーティション数です。

ℹ LVM または MD を使用する場合、Fusion ioMemory VSL ドライバのロードには `udev` を使用しないでください。init スクリプトは、Fusion ioMemory デバイスの `detach` を試みる前に、LVM ボリュームや MD デバイスが `detach` されていることを確認します。

init スクリプトを使用している場合のファイルシステムのマウント

Fusion ioMemory VSL ソフトウェアのロードに init スクリプトを使用している場合、デバイスの設定が完了したら、別のセクションにある [init スクリプトを使用している場合のファイルシステムのマウント](#) の手順に従う必要があります。

Logical Volume Manager を使用して RAID を設定する

LVM で Fusion ioMemory デバイスを使う一番シンプルな方法は、そのブロックデバイス全体を使用することです (例: /dev/fioa1, /dev/fioa2 などではなく /dev/fioa)。この方法では、ブロックデバイスは事前にパーティション分割しておく必要がありません。次の例では、ブロックデバイス全体を使用しているものを想定しています。

デバイスを Stripe (RAID 0) または mirror (RAID 1) どちらを予定しているかに関わらず、最初の 2 ステップは同じです:

1. 最初に、次のように物理ボリュームを作成します:

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

2. 次に、物理ボリュームをボリュームグループへ追加します:

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

LVM で stripe ボリューム (RAID 0) を作成する

ボリュームグループを作成したら、そのボリュームグループの中にロジカルボリュームを作成することができます。この実例では、1 つだけ作成し、また iomemory_lv と名付けています:

1. 2 つの stripe のため、-i2 オプションを使用して作成します:

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

2. 新しく作成したボリュームへ、ファイルシステムを作成します:

```
$ mkfs.ext4 /dev/iomemory_vg/iomemory_lv
```

3. これをブート時に自動的にマウントするには、/etc/sysconfig/iomemory-vsl4 ファイルを編集しボリュームグループへのパスを追加します:

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

ロジカルボリュームではなく、ボリュームグループを追加する必要があることに注意してください。

LVM で mirror ボリューム (RAID 1) を作成する

ボリュームグループを作成したら、そのボリュームグループの中にロジカルボリュームを作成することができます。この実例では、1 つだけ作成し、また iomemory_lv と名付けています:

1. Mirror を linear ボリュームと 1 つのコピーとするため、-m1 オプションを使用して作成します:

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

最初の mirror synchronization の状況は、lvs コマンドを使用して確認できます。例えば:

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


i --corelog オプションを使用すると、LVM は in-memory リージョンを使用するか、または mirror leg の状態を記録します。これは in-memory であり再起動で消失することから、ブート時に mirror leg を再スキャンし再作成する必要があります。詳細は LVM のドキュメントを参照してください。

- 新しく作成したボリュームへ、ファイルシステムを作成します：

```
$ mkfs.ext4 /dev/iomemory_vg/iomemory_lv
```

- これをブート時に自動的にマウントするには、/etc/sysconfig/iomemory-vsl4 ファイルを編集しボリュームグループへのパスを追加します：

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

ロジカルボリュームではなく、ボリュームグループを追加する必要があることに注意してください。

Mdadm を使用して RAID を設定する

2 つ以上の Fusion ioMemory デバイスを RAID アレイとして設定することで、ソフトウェアベースの RAID ソリューションを使用することができます。

i RAID1/Mirror を使用していて 1 つのデバイスが故障した場合、RAID の再構築を始める前に、デバイスに fio-format を行う必要がある場合があります(交換していない正常なデバイス側ではありません)。この項目は例として、mdadm の一般的な RAID 設定を示しています。

w Linux kernel での RAID 5 実装は、高いデータレートで効果がありません。これは Linux kernel の問題です。代わりに RAID 10 やまたはサードパーティの RAID スタックを使用することを推奨します。

アレイのマウント

アレイの作成が完了したら(後述の例のような手順で)、/etc/sysconfig/iomemory-vsl4 ファイルを編集し次の例のようにアレイのパスを追加する必要があります。これにより、ブート時にアレイがマウントされます。

この例ではすべて、アレイを md0 という名前で作成しています。次は/etc/sysconfig/iomemory-vsl4 ファイルへ md0 を追加する場合の例です：

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

Fusion ioMemory VSL ソフトウェアのロードに init スクリプトを使用している場合、デバイスの設定が完了したら、別のセクションにある [init スクリプトを使用している場合のファイルシステムのマウント](#) の手順に従う必要があります。

RAID 0

stripe セットを、fioa と fiob という 2 つの Fusion ioMemory デバイスを使用して作成するには、次のコマンドを実行します：

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

アレイを固定する(再起動時にも検出)

Linux のいくつかのバージョンでは、設定ファイルは/etc/mdadm.conf ではなく、/etc/mdadm/mdadm.conf を使用します。

/etc/mdadm.conf を確認します。1 つまたは複数のデバイスが定義されている場合、“partitions” オプションが指定されていることを確認します。指定されていない場合、新しい DEVICE 行を追加し“partitions”を指定します：

```
DEVICE partitions
```

fio Fusion ioMemory デバイスを追加します：

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

/etc/mdadm.conf に何かアップデートが必要かどうかを確認するには、次のコマンドを発行します：

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

このコマンドの表示を確認し、現在の mdadm.conf を内容と、また/etc/mdadm.conf へ必要な行を追加します。

- ❶ 例えば、アレイが 2 つのデバイスで構成されている場合、コマンドの表示には mdadm.conf ファイルにはない 3 つの行が表示されます：1 つはアレイ、2 つはデバイス(1 デバイスあたり 1 行)。これらの行を mdadm.conf へ追加する必要があることに注意してください。

詳細については、お使いのディストリビューションの mdadm と mdadm.conf の man ページを参照してください。

これらの変更により、ほとんどのシステムでは RAID 0 アレイが再起動時に自動的に作成されます。しかし、もし/dev/md0 へのアクセスに問題があった場合には、次のコマンドを実行します：

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

init スクリプトを使用してドライバをロードしている場合、Fusion ioMemory VSL ドライバの udev による読み込みを無効化する必要があるかもしれません。詳細は [init スクリプトの使用](#) を参照してください。

- ❶ SLES 11 では、ブート時にサービスが動作していることを確認するため、次のコマンドを実行する必要がある可能性があります：

```
chkconfig boot.md on chkconfig mdadm on
```

RAID 1

2 つの Fusion ioMemory デバイス fioa と fiob を使用して mirror セットを作成するには、次のコマンドを実行します：

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

- ❶ デバイスのフォーマットの直後に MD raid を作成する場合(raid0 以外)、-assume-clean を追加することで、RAID 作成の時間を短縮できます。

RAID 10

4 つの Fusion ioMemory デバイス fioa, fiob, fioc, fiod を使用して stripe mirror セットを作成するには、次のコマンドを実行します：

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

Discard (TRIM) のサポート

このバージョンの Fusion ioMemory VSL ソフトウェアでは、Discard (TRIM としても知られる)はデフォルトで有効になっています。

Discard は solid-state ストレージの問題に効果のあるものです。ユーザーがファイルを削除した場合、デバイスはその領域をスペアとして使用できるかどうかを気にしません。代わりに、デバイスはデータが有効であると仮定しなければなりません。


Discard は最近の多くのファイルシステムの機能です。これは、デバイスの論理セクタに有効が無いことを通知します。これにより、Wear-leveling ソフトウェアはその領域を、以降の書き込み操作に使用できるようになります。

Linux での Discard (TRIM)

Discard は Fusion ioMemory VSL ソフトウェアでデフォルトで有効になっています。しかし、Discard が正しく動作するには、Linux ディストリビューションがその機能をサポートしている必要があります、また Discard がファイルシステムのマウントポイントで有効になっている必要があります。TRIM コマンドは、discard マウントオプションがファイルシステムに追加された場合、または fstrim が定期的にマウントポイントで動作した場合に、Fusion ioMemory デバイスに送付されます。

別の言い方をすると、もし Linux ディストリビューションで Discard がサポートされており、またシステムでそれが有効になっている場合、Discard は Fusion ioMemory デバイスで正しく動作します。

Linux では、Discard はファイルシステム作成時に限らず、リクエストは kernel の Discard ioctl を使用してユーザースペースのアプリケーションにも発行されます。

 Kernel 2.6.33 またはそれ以前では、ext4 に既知の問題があり、Discard が有効になっているとデータが破損する可能性があります。多くのディストリビューションの Kernel ではすでに修正されています。お使いの Kernel 提供元に、正しく Discard がサポートされているか確認してください。

いくつかの Linux ディストリビューションでは、MD や LVM はファイルシステムから送付された Discard をデバイスに発行せず、これにより MD や LVM アレイの一部である Fusion ioMemory デバイスへも発行されない場合があります。詳細は、お使いのディストリビューションのドキュメントを参照してください。

RHEL 6.5 で、MD は Discard を RAID 0, RAID 1, RAID 10 でサポートしています。

パフォーマンスとチューニング

Fusion ioMemory デバイスは、広い帯域、高い Input/Output per Second (IOPS)、また特に顕著な低いレイテンシを提供します。

Fusion ioMemory デバイスが IOPS と低レイテンシを提供することで、デバイスのパフォーマンスはオペレーティングシステムや BIOS 設定により制限されてしまう可能性があります。これらの設定は、Fusion ioMemory デバイスの革新的な性能のために変更しチューニングする必要があるかもしれません。

Fusion ioMemory デバイスは基本的にこれらの設定には直接関係ないことから、このセクションでは性能を最適化するための一般的な設定内容を説明しています。

CPU Frequency Scaling の無効化

Dynamic Voltage や Frequency Scaling (DVFS) は、消費電力の縮小を目的に CPU の電圧や周波数を制御するための、電源管理の技術です。これらにより、CPU の電力や熱を調整しますが、代わりに CPU が低電源から高パフォーマンスのステートへ移行する際、パフォーマンスへの影響があります。

これらの電源抑制の技術は、I/O レイテンシや IOPS へ悪影響があることで知られています。パフォーマンスのチューニングを行う場合には、消費電力が上がったとしても、DVFS を縮小または無効化することで効果が得られることがあります。

DVFS が使用可能な場合、多くの場合は設定が可能で、またオペレーティングシステムや BIOS の設定の一部になっています。また、多くの場合はこれらの設定メニューのうち、Advanced Configuration and Power Interface (ACPI) セクションで見つけることができます。詳細はお使いのコンピュータのドキュメントを参照してください。

ACPI C-State を制限する

近年のプロセッサは、完全に使用されていない場合に、低電力モードへ切り替える機能を持っています。これらの idle state は ACPI C-state として知られています。C0 は通常でフルパワー、稼働中の state です。また高い C-state (C1, C2, C3 など) は低電力の state です。

ACPI C-state が電力を抑制している場合、I/O レイテンシや最大 IOPS へ悪影響があることがあります。高い C-state では、基本的により多くのプロセッサの機能が制限され電力が抑制され、またプロセッサが C0 state へ戻る際に時間がかかります。

最大性能のチューニングを行う場合には、消費電力が上がったとしても、C-state を制限するか無効化することで効果が得られることがあります。

ACPI C-State オプションを設定する

お使いのプロセッサが ACPI C-state に対応している場合、一般には BIOS インターフェース (Setup Utility など) を使ってこれを制限または無効にすることができます。ACPI C-state は Advanced Configuration and Power Interface (ACPI) メニューの一部であることが多いです。詳細はお使いのコンピュータのドキュメントを参照してください。

Linux での C-State

最近の Linux kernel では、もし BIOS により ACPI C-state が無効化されていても、ドライバにより有効化を試みる場合があります。この場合、次の kernel ブートオプションを追加することで (BIOS の設定に関わらず)、Linux 上で C-state を制限することができます：

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

この例では、最大で許された C-state は C0 (無効)となります。

NUMA Affinity の設定

NUMA (Non-Uniform Memory Access)アーキテクチャを使用可能なサーバでは、Fusion ioMemory デバイスのパフォーマンスを最大化するために特別なインストール手順が必要な場合があります。これは多くのマルチソケットのサーバが対象です。

いくつかの NUMA アーキテクチャーのサーバでは、システムブート時、BIOS は NUMA node を PCIe スロットに正しく関連付けません。正しくない割り当てでは、非効率な I/O ハンドリングとなりパフォーマンスの低下となります。

Fusion ioMemory VSL ソフトウェアは、自動的にデバイスに関連する NUMA node に割り当てます。しかし、それでもまだ NUMA node パラメータは最適ではなく、手動でデバイスへ割り当てる余地があります。

詳細は [NUMA の設定](#) を参照してください。

Interrupt Handler Affinity の設定

デバイスのレイテンシは、NUMA システムの割り込みの配置に影響されます。割り込みを I/O 発行するアプリケーションと同じ NUMA node に配置することを推奨します。もしその node の CPU がユーザーアプリケーションのタスクで満たされた場合、割り込みが別の node へ移動し、パフォーマンスに影響することがあります。

多くのオペレーティングシステムでは、割り込みを node 間で動的に配置することを試み、一般的に良い方向へ働きます。

Linux での IRQ Balancing

Linux では、動的な配置は IRQ Balancing と呼ばれます。IRQ balancer が動作しているかは、`/proc/interrupts` を参照することで確認できます。もし割り込みが適切にバランスされていない場合(多くのデバイス割り込みが一つの node にある)、または node が満たされている場合、IRQ balancer を停止し、手動で割り込みを配置し分散させることで、パフォーマンスを改善できる可能性があります。

- ❗ Fusion ioMemory VSL ソフトウェアをロードし Fusion ioMemory デバイスが attach された後に、IRQ Balancer を再起動することで、割り込みの affinity の問題を解消できることがあります。例えば、次のコマンドを実行します(お使いのディストリビューションに依存します)：

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

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

- ❗ もしそれでも affinity の問題を解消できない場合、手動でデバイス割り込みを特定の node に割り当てることを推奨します。

Linux での手動による割り込みの調整は、追加オプションとなっています。特定のデバイス割り込みを特定の node に割り当てるには、ご使用のオペレーティングシステムのドキュメントを参照してください。


デバイスの監視・管理

Fusion ioMemory デバイスを管理するために、ツールを提供しています。このツールにより、デバイスのエラー、警告、または問題の可能性を監視することができます。また、次のような機能を実行することもできます：

- ファームウェアのアップグレード
- ローレベルフォーマット
- attach または detach 操作
- デバイス状態の表示
- Swap や Paging の設定
- バグレポートの作成

管理ツールの使用

Fusion ioMemory デバイスを監視または管理するため、複数のツールを提供しています。それぞれのツールの詳細を確認し、最適なものをご使用ください。

 Fusion ioMemory VSL ソフトウェアはシステムログにいくつかのエラーメッセージを表示し、またトラブルシューティングのためには有用です。ただし、これは継続的な監視のためには適切ではありません。最適な結果のため、継続的な監視のためにはここに記載されているツールを使用してください。

- **コマンドラインユーティリティ:** このユーティリティは、Fusion ioMemory VSL ソフトウェアと一緒にインストールされ、ターミナル上で手動で動作します。fio-status ユーティリティはホスト上のデバイスの状態を提供します。その他のユーティリティは、その他の管理するための機能を提供します。詳細は [コマンドラインユーティリティのリファレンス](#) を参照してください。
- **Fujitsu ServerView RAID Manager:** このユーティリティでは、デバイスの状態表示、ログの表示や通知などの機能を提供します。詳細は ServerView RAID Manager のドキュメントを参照してください。

監視すべき状態の表示例

このセクションでは、監視可能な状態の例を示しています。これは基本的な紹介を意図しており、すべてのケースを示しているものではありません。これらの状態は、名前、状態、値など使用しているツールにより異なる場合があります。例えば、fio-status と ServerView RAID Manager では、名前が異なる可能性があります。

正しくこれらの状態を監視するためには、ご使用のツールのドキュメントを参照してください。

基本的なリファレンスとして、可能性のある状態・値は通常(**GREEN**)、注意・アラート (**YELLOW**)、エラー・警告 (**RED**) としています。

デバイスのステータス

すべての監視ツールは、以下のような Fusion ioMemory デバイスの状態を示します：

GREEN	attach 済み
YELLOW	detach 済み、ビジー(detach 中、attach 中、スキャン中、フォーマット中、アップデート中を含む)
RED	Minimal Mode、Powerloss Protect 無効

デバイスが Minimal Mode となっている場合、ツールで Minimal Mode になっている理由を表示します。

必要なアクション

デバイスが **Minimal Mode** となっている場合、アクションはその理由に依存します。例えば、理由が古いファームウェアの場合、ファームウェアをアップデートする必要があります。

温度

Fusion ioMemory デバイスは適切な冷却が必要です。熱による破壊を予防するため、Fusion ioMemory VSL ソフトウェアは、オンボードコントローラが一定の温度に至った場合、書き込みパフォーマンスの抑制を開始します。またコントローラの温度が引き続き上昇した場合、コントローラの温度が最大動作温度まで至った場合、ソフトウェアはデバイスをシャットダウンします。

温度はデバイスより異なります。詳細は **Fusion ioMemory Hardware Installation Guide** を参照してください。



NAND の温度

Fusion ioMemory デバイスは NAND の温度も報告します。これも重要な温度情報です。こちらについても温度の閾値は **Fusion ioMemory Hardware Installation Guide** を参照してください。

必要なアクション

もし温度が閾値に近づきつつある場合、システムの冷却能力を向上させる必要があります。例えばシステムファンの速度を上げる、システム周辺の温度を下げる、書き込み負荷を下げる、またはデバイスを別のスロットに移動することが挙げられます。

Health Reserves のパーセンテージ

Fusion ioMemory デバイスは高い冗長性を持つストレージサブシステムで、複数レベルのコンポーネント故障から保護します。いずれのストレージサブシステムでも、コンポーネントの故障は起きえます。

継続的にデバイスの状態を監視するには、寿命に関しても注意する必要があります。次の表では、Health Reserve の状態を示します。

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

10%の状態では、一度の警告が発行されます。0%では、デバイスは使用できなくなります。3%で **write-reduced** モードへ移行します。1%でデバイスは **read-only** モードへ移行します。

詳細は、[デバイスの Health 情報を監視](#) を参照してください。

必要なアクション

デバイスが **0%** に近づきつつあると、より注意深く監視する必要があります。すぐにデバイスを交換できるよう、あらかじめ準備してください。

書き込み (Health Reserves) のステータス

Health Reserves のパーセンテージと関連して、管理ツールは書き込みの状態も通知します：

GREEN	デバイスは正常
--------------	---------

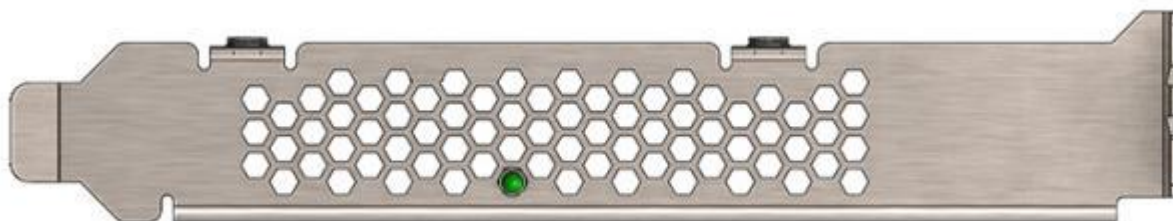
YELLOW	デバイスは write-reduced モードに近づいている
RED	デバイスは write-reduced モードまたは read-only モードになっており、最低限の flash を確保している

必要なアクション
 デバイスが 0% に近づくとつれて、より注意深く監視する必要があります。すぐにデバイスを交換できるよう、あらかじめ準備してください。

デバイスの LED Indicator

もしデバイスへの物理的なアクセスが可能な場合、ブラケットにある LED Indicator を使用することもできます。

次は LED の例を示しています：



次の表は、LED 表示の例を示しています：

LED	Indications	Notes
点灯	電源が入っておりドライバが動作している	
点滅 (高速)	読み込みまたは書き込み動作中	高速な点滅は、アクティビティを示し、また転送中のデータ量に依存します。空のセクタへのアクセスでは、点滅しない場合もあります。
点滅 (低速)	Beacon が動作中	動作させるには、 <code>fio-beacon</code> ユーティリティを使用してください。
消灯	電源オフ、ドライバがロードされていない、ドライバが動作していない	<code>fio-status</code> を参照しエラーがないか確認してください。

メンテナンス

このセクションでは、前のページで紹介されていない追加のメンテナンス機能を説明しています。

- ❗ すべてのコマンドは管理者権限が必要です。“root”でログインするか、`sudo` を使用してください。

Fusion ioMemory VSL ソフトウェアのアンインストール

- ❗ このバージョンの Fusion ioMemory VSL ソフトウェアは、PX600 デバイスにのみ対応しており、以前のデバイスには対応していません。以前のデバイス用のバージョンがインストールされている場合、あらかじめアンインストールする必要があります。手順は、以前のバージョンのドキュメントを参照してください。

アンインストールするパッケージの kernel 版数も指定する必要があります。次のコマンドでインストールされているドライバパッケージを参照してください：

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

表示例：

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

上記の例の版数を参照し、次のコマンドでアンインストールします(アンインストールする kernel 版数も指定する必要があります)：

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

ソフトウェアドライバのアンロード

ドライバをアンロードするには、次のコマンドを実行します：

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

Kernel のアップグレード

Fusion ioMemory VSL ソフトウェアがインストールされている環境で、kernel のアップグレードを計画している場合、次の順番で行う必要があります：

1. Fusion ioMemory VSL ドライバをアンロード
2. Fusion ioMemory VSL ソフトウェアをアンインストール
3. Kernel をアップグレード
4. 新しい kernel 用にコンパイルされた Fusion ioMemory VSL ソフトウェアをインストール

順番を誤った場合、ドライバのロードに問題が起きる可能性があります。

Fusion ioMemory VSL ソフトウェアの無効化

Fusion ioMemory VSL ソフトウェアは、デフォルトでオペレーティングシステムの開始と同時にロードされます。診断やトラブルシューティングを目的として、Fusion ioMemory VSL ソフトウェアを無効化することができます。

自動ロードを無効化するには、Fusion ioMemory VSL ソフトウェアをアンインストールするか、`/lib/modules/<kernel_version>` ディレクトリから移動してください。

Auto-Attach の無効化

Fusion ioMemory VSL ソフトウェアがインストールされていると、ロードされた際に自動的にすべてのデバイスが `attach` されるように設定されています。診断やトラブルシューティングを目的にこれを無効化したい場合は、次のように行います：

1. 次のファイルを編集する：

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

2. ファイルの以下の行から、コメントアウトを解除する：

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

3. ファイルを保存する。Auto-attach を再度有効にするには、上記の行を削除するか、コメントアウトするか、または次のように変更します：

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

予期しないシャットダウンの問題

電源消失やその他の理由により、予期しないシャットダウンが起きた場合、Fusion ioMemory デバイスへ次回起動時に `consistency check` を実行することができます。これは、完了には数分以上の時間がかかります。

Fusion ioMemory デバイスに書かれているデータは予期しないシャットダウンでも消失しませんが、デバイス上にある重要なデータ構造情報は正しくない可能性があります。この `consistency check` (または `rescan` と呼ばれる)は、データ構造を修復します。

Rescan の時間を改善する

デバイスの `rescan` は、Fusion ioMemory VSL ソフトウェアにより予期しないシャットダウンからの復帰時に実行され、またデバイスの最大容量に依存して長時間かかる場合もあります。

デフォルトの Fast Rescan

デフォルトでは、Fusion ioMemory デバイスは最適な `rescan` 時間に設定されています。デフォルトの `fast rescan` は、`“-R”` オプションを使って再フォーマットすることで無効化することができます。これを無効化することで、デバイスに確保されている `reserve` 領域が解放されます。

もしデフォルトの `fast rescan` 以外を使用する場合、以下のモジュールパラメータを使って `rescan` 時間を改善することができます。

モジュールパラメータを使った Faster Rescan

これらの 2 つのモジュールパラメータは、システムメモリ (RAM) を使用します。追加のメモリにより、`rescan` プロセスを高速に完了させ、シャットダウンからの復旧時間を短縮できます。このメモリ予約は一時的なもので、また `rescan` プロセスが完了すると解放されます。

これらのパラメータを使用することに決めた場合、最大の RAM を指定する必要があります。これをするには、このシナリオで使われる最大 RAM 容量を想定し、システム上の RAM を確認し、また最適なパラメータの内容を決定する必要があります。

モジュールパラメータの設定方法は、[モジュールパラメータの使用](#)を参照してください。

以下は2つのパラメータの比較です：

- **RMAP パラメータ**
 - **最速:** これにより、最速の rescan 時間となります。
 - **調整不可:** (すべて、または無し) このパラメータには十分な RAM が必要です。もし RAM 制限が少なすぎる場合、Fusion ioMemory VSL ソフトウェアは RMAP を使用せず、デフォルトの fast rescan プロセスに切り替わります。
 - **対象シナリオ:** このパラメータは十分な RAM がある場合に有効です。また、小さい容量、または少ない数の Fusion ioMemory デバイスに有効です。また、小さいランダム書き込みが多発する用途の場合に推奨します。
- **RSORT パラメータ**
 - **高速:** これにより、デフォルトの fast rescan よりも短い rescan 時間となります。
 - **調整可能:** このパラメータでは、Fusion ioMemory VSL ソフトウェアは、設定された限界値までのシステム RAM を使って rescan 時間の改善を行います。このとき、限界に達するとデフォルトの fast rescan プロセスに切り替わります。
 - **対象シナリオ:** このパラメータは、いずれのシナリオでも rescan 時間を改善します。特に多数または大容量の Fusion ioMemory デバイスが搭載されている場合に有効です。また、データベースが保存されている場合に推奨します。

RMAP パラメータ

rmap_memory_limit_MiB パラメータは、Fusion ioMemory VSL ソフトウェアが RMAP rescan プロセス中に使用するメモリの上限(in mebibytes)を指定します。このオプションは、搭載されているすべての Fusion ioMemory デバイスが動作するために十分なメモリがある場合にのみ使用する必要があります。十分なメモリがない場合、代わりに RSORT パラメータを使用してください。

このパラメータは多くのメモリが必要であることから、少ない数かつ小さい容量の Fusion ioMemory デバイスを使用している場合に有効ですが、最大の要件はシステムに搭載されているメモリ量と使用可能な限度です。

このパラメータは、Fusion ioMemory デバイス容量のブロック数あたり、4.008 bytes の RAM が必要です。

1. 最初に、各デバイスでフォーマットされているブロックの数を特定します。
 - a. この情報は、`fiio-format` ユーティリティを使ってデバイスをフォーマットしている際に参照できません。
 - b. または、デバイス容量とセクタサイズを使って想定することもできます。
次の例では、1000GB デバイスで 512B セクタの場合の、簡単な想定例です：

$$1000\text{GB} * 1000\text{MB}/\text{GB} * 1000\text{KB}/\text{MB} * 1000\text{B}/\text{KB} * 1 \text{ Block}/512\text{B} = 1,953,125,000 \text{ Blocks}$$

2. ブロックを 4.008 bytes で積算(また MiB に計算)し、必要なメモリ量を特定します。
 - a. この例は、1.95 billion ブロックの場合です：

$$1,953,125,000 \text{ Blocks} * 4.008\text{B}/\text{Block} * 1\text{KiB}/1024\text{B} * 1\text{MiB}/1024\text{KiB} = \sim 7465\text{MiB of RAM}$$

- b. この例では、1000GB かつ 512B セクタフォーマットの Fusion ioMemory デバイスのために、7465 MiB のメモリが必要で、また RMAP パラメータは 7500 に設定します。



デフォルト値

RMAP パラメータはデフォルトでは 3100 に設定されています。このデフォルトの低い値に設定されていることで、rescan プロセスはシステムのすべての RAM を使用しません。

- RMAP 値が Fusion ioMemory デバイスのブロック数に対して少なすぎる場合、Fusion ioMemory VSL ソフトウェアは RMAP プロセスを使用せず、デフォルトの fast rescan プロセスを使用します。
 - もし RMAP パラメータのためにメモリが十分でない場合、RSORT パラメータの使用を検討してください。
3. モジュールパラメータを、確認した値に設定します。設定方法は [モジュールパラメータの使用](#) を参照してください。

RSORT パラメータ

rsort_memory_limit_MiB パラメータは、RSORT rescan プロセスを実行中に Fusion ioMemory VSL ソフトウェアが使用するメモリ (RAM) の限界を設定します。RSORT rescan プロセスはデフォルトの rescan プロセスより高速で、またデータベースのデータストアとして使用されているデバイスの rescan に対して推奨します。

このパラメータでメモリ制限が提供されると、Fusion ioMemory VSL ソフトウェアは、rescan が完了するか、またはメモリ限界まで使用されるまで、RSORT プロセスが動作します。もしプロセスがメモリ不足になった場合、それはデフォルトの fast rescan プロセスに切り替わります。しかし、このプロセスの使用を最適化するため、対象となる RAM 使用量を計算し、その結果により限界を設定することが重要です。上限を高く設定することに不利益はありませんが、RSORT プロセスは必要な分(かつ上限まで)だけの RAM を使用します。

このターゲットは、書き込み extent あたり 32 bytes となります。例えば、お使いのデータベースが 16KB の書き込みを行う場合、Fusion ioMemory デバイス容量 16KB あたり 1 つの書き込み extent となります。

書き込み extent あたりのブロック数

RSORT プロセスの見積もりを行うとき、書き込み extent あたりのブロック数を見る必要があります。RSORT プロセスの rescan 時間は、書き込み extent あたり 8 ブロック以上設定されている場合に、デフォルトの fast rescan プロセスより高速になります。例えば、もし Fusion ioMemory デバイスが 512B セクタサイズにフォーマットされている場合(KB あたり 2 ブロック)、またお使いのデータベースが 8KB chunk で書き込みを行っている場合、お使いのデータベースは書き込み extent あたり 16 ブロックの書き込みを行っており、RSORT による rescan 時間の改善が見込めます。

- 最初に、各デバイスでフォーマットされているブロックの数を特定します。
 - この情報は、`fiio-format` ユーティリティを使ってデバイスをフォーマットしている際に参照できません。
 - または、デバイス容量とセクタサイズを使って想定することもできます。
次の例では、1000GB デバイスで 512B セクタの場合の、簡単な想定例です(KB あたり 2 セクタ)：

$$1000\text{GB} * 1000\text{MB}/\text{GB} * 1000\text{KB}/\text{MB} * 1000\text{B}/\text{KB} * 1 \text{ Block}/512\text{B} = 1,953,125,000 \text{ Blocks}$$


- デバイスにある合計の書き込み extent 数を特定するため、ブロックあたりの書き込み extent を、ブロック数で除算します。
 - この例では、1.95 billion ブロックの場合です。16KB の書き込み extent を想定しています(512B セクタで 32 ブロックの書き込み)：

$$1,953,125,000 \text{ Blocks} * 1 \text{ Write Extent}/32 \text{ Blocks} = 61,035,156 \text{ Writes}$$

- このパラメータでのメモリ量を特定するため、書き込み数を 32 byte RAM で積算します(また MiB に変換します)。
 - この例では、61 million の書き込み extent がありました：

$$61,035,156 \text{ Writes} * 32\text{B}/\text{Write} * 1\text{KiB}/1024\text{B} * 1\text{MiB}/1024\text{KiB} = \sim 1863\text{MiB of RAM}$$

-
- b. この例では、512B セクタフォーマットされた 1000GB の Fusion ioMemory デバイスあたり、RSORT 上限をシステムメモリの 2300 MiB に設定します。


 **デフォルト値**

RSORT パラメータは、デフォルトで 0M となっており、上限は 100000 (100 GiB) となります。


4. モジュールパラメータを、確認した値に設定します。設定方法は [モジュールパラメータの使用](#) を参照してください。

付録 A - コマンドラインユーティリティのリファレンス

Fusion ioMemory VSL ソフトウェアのインストールパッケージには、多くのコマンドラインユーティリティが含まれており、デフォルトでは `/usr/bin` にインストールされます。これらにより、デバイスへ複数のアクセス、診断、操作を行うことができます。

 この表に含まれるものの他にも `/usr/bin` ディレクトリには追加のユーティリティがインストールされている場合があります。これらのユーティリティは、Fusion ioMemory VSL ユーティリティで使用されるよう依存関係があり、またユーザーはカスタマーサポートから指定されない限り直接使用するべきではありません。



ユーティリティ	用途
<code>fio-attach</code>	Fusion ioMemory デバイスを OS 上に表示します。
<code>fio-beacon</code>	Fusion ioMemory デバイスの LED を点灯します。
<code>fio-bugreport</code>	問題のトラブルシューティングのために、詳細レポートを用意します。
<code>fio-detach</code>	一時的に Fusion ioMemory デバイスを OS から取り外します。
<code>fio-format</code>	Fusion ioMemory に low-level フォーマットを実行します。
<code>fio-pci-check</code>	主に Fusion ioMemory デバイスに対して、PCI bus tree を確認します。
<code>fio-status</code>	デバイスの情報を表示します。
<code>fio-sure-erase</code>	デバイスからデータを消去します。
<code>fio-update-iodrive</code>	Fusion ioMemory デバイスのファームウェアをアップデートします。

 すべてのユーティリティに `-h (Help)` と `-v (Version)` オプションが用意されています。また、これらのオプションが実行された場合、ユーティリティを実行すると情報を表示して終了します。

fio-attach

説明

Fusion ioMemory デバイスを `attach` し、オペレーティングシステムに表示します。これにより `/dev` の中に `fiox` (`x` は `a`、`b`、`c` など) のブロックデバイスを作成します。その後、Fusion ioMemory デバイスへパーティションを作成しフォーマット、または RAID アレイを構成できます。このコマンドは、動作中に進捗バーとパーセンテージを表示します。


-  ほとんどのケースでは、Fusion ioMemory VSL ソフトウェアは、ロード時に自動的にデバイスを検出し `attach` します。このため、`fio-attach` を実行する必要があるのは、手動で `fio-detach` を実行した後か、Fusion ioMemory VSL ソフトウェアの `auto_attach` パラメータを `0` に設定している場合です。
-  Fusion ioMemory デバイスが `Minimal Mode` になっている場合、`Minimal Mode` の原因が取り除かれるまで、`auto-attach` は無効化されます。

構文

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

<device>はデバイス node の名前(/dev/fctx)となり、また x はデバイス番号: 0、1、2 などです。例えば、/dev/fct0 はシステムで検出されている 1 つ目の Fusion ioMemory デバイスとなります。デバイス番号は fio-status に表示されます。

複数の Fusion ioMemory デバイスを指定することもできます。例えば/dev/fct1 /dev/fct2 は 2 つ目と 3 つ目の Fusion ioMemory デバイスとなります。

オプション	説明
-f	metadata rescan を強制します。これにより追加の時間が必要になりますが、通常は不要です。  カスタマーサポートに指示された場合にのみ使用してください。
-c	Clean な場合にのみ attach します。
-q	Quiet: 進捗とパーセンテージの表示を無効化します。
-Q	Quiet: 進捗の表示のみを無効化します。

fiio-beacon

説明

Fusion ioMemory デバイスの LED を点灯させ、位置を特定します。最初に Fusion ioMemory デバイスを detach してから、fiio-beacon を実行します。

構文

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

<device>はデバイス node の名前(/dev/fctx)となり、また x はデバイス番号: 0、1、2 などです。例えば、/dev/fct0 はシステムで検出されている 1 つ目の Fusion ioMemory デバイスとなります。デバイス番号は fio-status に表示されます。

オプション	説明
-0	Off: (ゼロ) beacon を消灯します。
-1	On: LED beacon を点灯します(ゆっくり点滅)。
-p	デバイスの PCI bus ID を表示します。

fiio-bugreport

説明

問題のトラブルシューティングのための詳細なレポートを用意します。結果のファイルは/tmp ディレクトリに保存され、またこのユーティリティが実行された日時も記録されます。

例:

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

構文

```
fio-bugreport
```

備考

このユーティリティは、デバイスの現在の状態を取得します。カスタマーサポートは、トラブルシューティングの際の情報としてこのファイルの取得と送付を依頼する場合があります。

表示例

```
-bash-3.2# fio-bugreport
BUGREPORT_VERSION 2.0
Running fio-read-lebmap /dev/fct0
Running fio-read-lebmap -m /dev/fct0
Running fio-get-erase-count /dev/fct0
Running fio-get-erase-count -b /dev/fct0
Running fio-kinfo -i /dev/fct0
Running fio-kinfo -i --driver-data /dev/fct0
Running dc-status -v
Running dc-status -b -fj -U
Collecting /proc/buddyinfo
Collecting /proc/cgroups
Collecting /proc/cmdline ...
Collecting /etc/init.d/iomemory-vsl4
Collecting /etc/modprobe.d/iomemory-vsl4.conf
Collecting /etc/sysconfig/iomemory-vsl4
Building tar file... Please attach the bugreport tar file
/tmp/fio-bugreport-20090921.173256-sdv9ko.tar.bz2
to your support case, including steps to reproduce the problem.
If you do not have an open support case for this issue, please open a support case with a problem description and then
attach this file to your new case.
```


例えば、bug report ファイルの名前は `fiobugreport-20090921.173256-sdv9ko.tar.bz2` となり、次のことを示します：

- 日付：20090921 (YYYY:MM:DD)
- 時間：173256, or 17:32:56
- 補足：sdv9ko.tar.bz2

fio-detach

説明

Fusion ioMemory デバイスを detach し、関連する fctx の Fusion ioMemory ブロックデバイスを OS 上から取り除きます。fio-detach ユーティリティは、detach 操作を実行する前に、デバイスがすべての read/write アクティビティが完了するまで待ちます。デフォルトでは、このコマンドは、動作中に進捗バーとパーセンテージを表示しません。

 このユーティリティを使用する前に、detach しようとしているデバイスがマウントや使用されていないことを確認してください。

構文

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

<device>はデバイス node の名前(/dev/fctx)となり、また x はデバイス番号: 0、1、2 などです。例えば、/dev/fct0 はシステムで検出されている 1 つ目の Fusion ioMemory デバイスとなります。デバイス番号は fio-status に表示されます。

複数の Fusion ioMemory デバイスを指定することもできます。例えば/dev/fct1 /dev/fct2 は 2 つ目と 3 つ目の Fusion ioMemory デバイスとなります。または、例えば/dev/fct*のように、ワイルドカードを使用してすべての Fusion ioMemory デバイスを指定することもできます。

オプション	説明
-q	Quiet: 進捗とパーセンテージの表示を無効化します。
-Q	Quiet: 進捗の表示のみを無効化します。

備考

Fusion ioMemory デバイスの detach の試行は、失敗し、またデバイスがビジーであると表示されることがあります。これは一般的に、Fusion ioMemory デバイスがソフトウェア RAID ボリュームの一部である場合、マウントされている場合、または他のプロセスがデバイスを Open している場合に発生します。

ツール fuser、mount、または lsof により、デバイスが Open されているかなどを特定することができます。

fiio-firmware

説明

このユーティリティは、システムの問題を調査する際に役立ちます。このユーティリティは、Fusion ioMemory VSL ソフトウェアをロードせずに、Fusion ioMemory デバイスのファームウェアのバージョンとリビジョンを報告します。デバイスのファームウェアバージョンが示されることで、使用すべき Fusion ioMemory VSL ソフトウェアのバージョンを特定できます。

出力は CSV フォーマットとなり、各 Fusion ioMemory デバイスごとの行になります：

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

構文

```
fiio-firmware [options]
```

オプション	説明
-f	x.y.z のファームウェアバージョンのみを表示します。
-n	列のヘッダー表示を無効化します。
-p	次回起動時に表示される、pending 状態のファームウェアを表示します。
-r	ファームウェアリビジョンのみを表示します。

fio-format

説明

- ❗ Fusion ioMemory デバイスはフォーマット済みの状態で出荷されるため、デバイスの論理サイズやブロックサイズを変更したい場合を除き、通常は **fio-format** を実行する必要はありません。データを確実に消去したい場合は、**fio-sure-erase** を使用してください。

Fusion ioMemory デバイスへ Low-level フォーマットを実行します。デフォルトでは、実行中に進捗とパーセンテージを表示します。

- ⚠ このユーティリティはデバイス上のすべてのデータを削除するため、注意してください。フォーマットを実行するかどうか、確認が表示されます。

- ❗ 大きなブロック(セクタ)サイズを使用するとき、例えば 4096 bytes などでは、Fusion ioMemory VSL ソフトウェアによる最大メモリ消費量を劇的に縮小することができます。しかし、いくつかのアプリケーションは 512 byte セクタに対応していない場合があります。
-s または -o オプションを使用しなかった場合、デバイスサイズは推奨されるデフォルト値となります。使用した場合、-s または -o オプションと同時にサイズまたはパーセンテージも提示される必要があります。

- ⚠ フォーマット中の電源不良を予防するため、システムへ追加の電源バックアップを使用することを推奨します。

構文

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

<device>はデバイス node の名前(/dev/fctx)となり、また x はデバイス番号: 0、1、2 などです。例えば、/dev/fct0 はシステムで検出されている 1 つ目の Fusion ioMemory デバイスとなります。デバイス番号は **fio-status** に表示されます。

オプション	説明
-b <size B K>	ブロック(セクタ)サイズを設定します。バイト単位または kibibyte 単位(base 2)で指定します。512b または 4KiB セクタのみがサポートされています。例: -b 512B または -b 4K (512B の中の B はオプション)。 Fusion ioMemory PX600 デバイスは 4KiB セクタサイズで出荷されます。もしセクタサイズを指定しなかった場合、ユーティリティはデフォルトの 4KiB セクタサイズにフォーマットします。
-f	フォーマットを強制的に実行し、確認や警告をバイパスします。このオプションは、 fio-format が正しく動作しない場合など稀に必要になります。("Are you sure?"の確認メッセージは、-y オプションを指定しない限り表示されます)
-q	Quiet: 進捗とパーセンテージの表示を無効化します。
-Q	Quiet: 進捗の表示のみを無効化します。

オプション	説明
-s <size M G T %>	デバイス容量を、指定されたサイズ(TB, GB, MB)、または最大容量に対するパーセンテージに設定します： <ul style="list-style-type: none"> • T : terabytes (TB) の値にフォーマットします • G : gigabytes (GB) の値にフォーマットします • M : megabytes (MB) の値にフォーマットします • % : パーセンテージ、例えば 70% にフォーマットします(%記号は必須です)
-R	予期しないシャットダウン時の Fast rescan を無効化し、reserve 領域を解放します。
-y	すべての質問に、自動で“yes”と回答します(確認をバイパスします)。

その後、Fusion ioMemory デバイスを再度 attach する必要があります。

fio-pci-check

説明

主に Fusion ioMemory デバイスに関連する、PCI bus tree のエラーを確認します。このユーティリティは、それぞれの Fusion ioMemory デバイスの現状を表示します。また、標準的な PCI Express のエラー情報を表示し、また State をリセットします。

- ❗ 最初に fio-pci-check を実行した場合に、いくつかの correctable error が表示されるのは通常のことです。次以降の実行では 1 つまたは 2 つのエラーとなることも通常のことです。

構文

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

オプション	説明
-d <value>	1 = リンク無効; 0 = リンク再開(非推奨)
-e	PCIe error reporting を有効にする。
-f	システム上のすべてのデバイスをスキャンする。
-n	Config space への書き込みを実行しない。これによりエラーが消去されることを防ぎます。
-o	Fusion ioMemory デバイスの PCIe link 設定を最適化します。read request サイズが小さい場合、最大に設定します。
-r	リンクの retrain を強制します。
-v	Verbose: ハードウェアについてのより詳細を表示します。

fio-status

説明

搭載されているデバイスの詳細情報を表示します。このユーティリティは、**fctx** または **fioc** のどちらのデバイスにも実行できます。このユーティリティは **Fusion ioMemory VSL** ドライバに依存します。もしドライバがロードされていない場合、最低限の **status** 情報が表示されます。

fioc-status はエラー情報、例えば **Minimal Mode**、**Read-only mode**、**write-reduced mode** などのアラートを表示し、またその理由も表示します。

構文

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

<device>はデバイス **node** の名前(/dev/fctx)となり、また **x** はデバイス番号: **0**、**1**、**2** などです。例えば、**/dev/fctx0** はシステムで検出されている 1 つ目の **Fusion ioMemory** デバイスとなります。デバイス番号は **fioc-status** に表示されます。

もし<device>が指定されなかった場合、**fioc-status** はシステム上のすべての **Fusion ioMemory** デバイスの情報を表示します。もし **Fusion ioMemory VSL** ドライバがロードされていない場合、このパラメータは無視されます。

オプション	説明
-a	各デバイスのすべての情報を表示します。
-e	各デバイスのエラーや警告を表示します。このオプションは、問題の診断に使用され、その他の情報、例えばフォーマットサイズなどは隠されます。
-c	Count : 搭載されている Fusion ioMemory デバイスの数のみを表示します。
-d	基本的な情報と、データの読み取りと書き込み量(lifetime data volumes)を表示します。このオプションは、 -a オプションが使用されている場合は不要です。
-fj	Format JSON : 表示を JSON 形式で作成します。
-fx	Format XML : 表示を XML 形式で作成します。
-u	無効なフィールドを表示します。 -fj または -fx の場合にのみ有効です。
-U	無効なフィールドと理由の詳細を表示します。 -fj または -fx の場合にのみ有効です。 <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p>i いくつかの fioc-status フィールドは、使用しているオペレーティングシステムやデバイスによっては存在しません。例えば、古いフィールドは、新しい Fusion ioMemory デバイスには存在しません。</p> </div>
-F<field>	特定のフィールドの値のみ表示します(フィールド名は次のオプションを参照してください)。デバイスが指定されている必要があります。複数の -F オプションを指定することもできます。
-l	-F でアクセスできるそれぞれのフィールドを一覧表示します。
-L	システム上のすべての Fusion ioMemory デバイスを一覧表示します。

基本情報: オプションが指定されていないければ、**fioc-status** は次の基本情報を表示します：

- システムにインストールされているデバイスの数と種類
- **Fusion ioMemory VSL** ソフトウェアバージョン

Adapter information:

- Adapter type
- Product number
- Product UUID
- PCIe power limit threshold (存在する場合)
- 接続されている Fusion ioMemory デバイス

Block device information:

- Attach status
- Product name
- Product number
- Serial number
- PCIe address と slot
- ファームウェア バージョン
- デバイスのサイズと最大容量
- Internal temperature (Fusion ioMemory VSL がロードされた後の average と maximum)、摂氏単位
- Health status: healthy, nearing wearout, write-reduced or read-only
- Reserve capacity (パーセンテージ)
- Warning capacity threshold (パーセンテージ)

Data Volume Information: -d オプションが使用された場合、基本情報に加えて、次の data volume information が表示されます :

- Physical bytes written
- Physical bytes read

All Information: -a オプションが使用された場合、基本情報に加えて、すべての情報が表示されます :

Adapter information:

- Manufacturer number
- Part number
- Date of manufacture
- Power loss protection status
- PCIe bus voltage (平均、最小、最大)
- PCIe bus current (平均、最大)
- PCIe bus power (平均、最大)
- PCIe power limit threshold (W(ワット))
- PCIe slot available power (W(ワット))
- PCIe negotiated link information (レーン数とスループット)
- Product UUID

Block device information:


- Manufacturer's code
- Manufacturing date
- Vendor と sub-vendor information
- Format status と sector information (デバイスが attach されている場合)
- Controller ID と Low-level format GUID


- PCIe slot available power
- PCIe negotiated link information
- Card temperature, 摂氏単位
- Internal voltage (平均、最大)
- Auxiliary voltage (平均、最大)
- 正常なブロック、データ、metadata の割合
- Lifetime data volume の統計
- RAM 使用量


Error Mode Information: Fusion ioMemory VSL ソフトウェアが Minimal Mode、Read-only、Write-reduced モードの状態で fio-status が実行された場合、表示には次の違いがあります：

- Attach 状態は"Status unknown: Driver is in MINIMAL MODE:"となる
- Minimal Mode の理由が表示される ("Firmware is out of date. Update firmware."など)
- "Geometry and capacity information not available."と表示される
- Media health 情報は表示されない

fio-sure-erase

 最適な結果を得るため、システム上にクリアしたくない Fusion ioMemory デバイスが搭載されている場合は、このユーティリティを使用しないでください。まず、誤ってデータをクリアしたくないデバイスを取り外してください。このユーティリティによりデータが削除された場合、復旧することはできません。

 このユーティリティを使用する前に、大切なデータはすべてバックアップしてください。

 デバイスが Read-only mode の場合、fio-sure-erase を実行する前に、fio-format ユーティリティでフォーマットしてください。デバイスが Minimal mode の場合、fio-sure-erase はデバイスのデータを消去できません。ファームウェアのアップデートにより、Minimal mode を解決できることがあります。

fio-sure-erase を実行するためには、ブロックデバイスは detach されている必要があります。

説明

fio-sure-erase は、Fusion ioMemory デバイスから安全にデータを削除するためのユーティリティです。データの削除や破壊は、“Clear”と”Purge”のレベルで構成され、次の標準に従っています：

1. DOD 5220.22-M - Flash EPROM の標準に従う
2. NIST SP800-88 - Flash EPROM の標準に従う


証明の詳細については、次のページを参照してください。 <http://www.fusionio.com/overviews/fusionsureerase/>


Clear と Purge のサポートについては、以下を参照してください。

構文

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

<device>はデバイス `node` の名前(/dev/fctx)となり、また `x` はデバイス番号: 0、1、2 などです。例えば、/dev/fct0 はシステムで検出されている 1 つ目の Fusion ioMemory デバイスとなります。デバイス番号は `fio-status` に表示されます。

オプション	説明
-p	Clear の代わりに Purge: 消去の後に書き込みを行います。Purge の詳細は、下を参照してください。 <div style="border: 1px solid orange; background-color: #ffff00; padding: 5px; margin: 5px 0;"> デバイスの Purge には、デバイスのサイズと Purge すべき量に合わせて、完了するまで時間がかかります。</div>
-y	確認なし: ユーティリティを実行するために、yes または no の入力を不要にします。
-t	現在のフォーマットパラメータを確保せず、デバイス情報やセクタサイズをデフォルトに戻します。
-q	Quiet: ステータスバーを表示しません。

 オプションを指定せずに `fio-sure-erase` を実行した場合、Clear が実行されます。詳細は以下を参照してください。

ユーティリティが完了すると、それぞれのメモリ上のブロックは 1 bit または 0 bit に統一されます。

Clear のサポート

“Clear”は `fio-sure-erase` (オプションなし)のデフォルトで、すべての NAND メディアに対して完全な low-level の消去(すべてのセルを“1”)を行います。

この操作に必要な Metadata (media event log, erase counts, physical bytes read/written, performance, thermal history) は破壊されませんが、ユーザー領域の metadata を破壊します。

以下は、Clear 操作で実行されるステップを説明しています：

1. すべてのアクセス可能なブロックに対して、統合された map を作成します(これにより `fio-sure-erase` は、以前に除外された不良ブロックも含め、それぞれのブロックの位置を把握できるようになります)。
2. それぞれのブロックに、消去サイクルを実行します(すべてのセルは“1”で統一されます)。
3. 不良ブロックの map を復旧します。
4. デバイスをフォーマットします(ユーティリティがヘッダーも含めすべてクリアするため、デバイスを再利用できるようにします)。

Purge のサポート


“Purge”は、-p オプションと共に `fio-sure-erase` を使用した場合に実行されます。Purge は、まず最初にすべての NAND media (除外されたブロックを含む)を 1 つの文字(すべてのセルは論理的“0”)で上書きし、その後、完全なチップからのすべての media の消去(すべてのセルは“1”)を行います。

この操作に必要な Metadata (media event log, erase counts, physical bytes read/written, performance, thermal history) は破壊されませんが、ユーザー領域の metadata を破壊します。

以下は、Purge 操作で実行されるステップを説明しています：


1. すべてのアクセス可能なブロックに対して、統合された **map** を作成します(これにより **fiore-erase** は、以前に除外された不良ブロックも含め、それぞれのブロックの位置を把握できるようになります)。
2. それぞれのブロックに、書き込みサイクルを実行します(すべてのセルは"0"で統一されます)。
3. それぞれのブロックに、消去サイクルを実行します(すべてのセルは"1"で統一されます)。
4. 不良ブロックの **map** を復旧します。
5. デバイスをフォーマットします(ユーティリティがヘッダーも含めすべてクリアするため、デバイスを再利用できるようにします)。

fiore-update-iodrive

 アップグレードを実施する前に、大切なデータはバックアップしておいてください。


説明

Fusion ioMemory デバイスのファームウェアをアップデートします。このユーティリティはすべての Fusion ioMemory デバイスのため PCIe bus をスキャンし、それらのアップデートを行います。それぞれのデバイスについて進捗バーとパーセンテージが表示されます。

 ファームウェアのアップデートの最中は、必ず電源が切れないことが重要で、最悪のケースではデバイスが故障する可能性があります。UPS が準備されていない場合は、ファームウェアのアップデートの前に追加することを検討してください。

複数段階に渡ってファームウェアアップデートを順次実行する場合、それぞれのアップデートをするごとに Fusion ioMemory VSL ドライバをロードすることが重要です。そうしなければ、デバイス上のフォーマットが正しく変更されず、データが消失することがあります。

ファームウェアのダウングレードは行わないでください。それによりデータが消失したり、ワランティが失効します。

 デフォルトのアクション(-d または-s オプションなし)では、すべての Fusion ioMemory デバイスをファームウェアアーカイブのファイル **fiore-firmware-fusion_<version>-<date>.fff** に更新します。事前に、すべてのデバイスに対してアップデートを実行してもよいか確認してください。疑問がある場合は、-p (Pretend) オプションを使用すると、事前に予定を確認できます。

1 つまたは特定のデバイスをアップデートしたい場合：

- Fusion ioMemory VSL ドライバをロードし、-d オプションを使用してデバイス番号を指定してください。

オンラインでのファームウェアアップデート

このユーティリティでは、互換性のある Fusion ioMemory デバイスをアップデートします。このユーティリティは、デバイスがファームウェアのアップデートをサポートしているかを確認します(attach された状態で)。

ファームウェアアップデートは、システムを再起動するまで有効になりません。もしすでにファームウェアをアップデートしたが、再起動していない場合は、**fiore-status** ユーティリティは次のように表示します：



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

もしデバイスが互換性がない場合、`fiio-update-iodrive` ユーティリティは、デバイスがサポートされていない旨のエラーを応答します。この場合、まずデバイスを `detach` した後に、再度ユーティリティを実行します。どちらのケースでもアップデートされたファームウェアを有効にするためには再起動が必要です。

構文

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

<firmware-path>はファームウェアアーカイブファイル `fiio-firmware-fusion_<version>-<date>.fff` への完全パスを指定します。もし `.fff` ファームウェアアーカイブファイルを直接入手した場合、そのファイルを使用します。もしファームウェアをパッケージとしてダウンロードしインストールした場合、デフォルトのパスは `/usr/share/fiio/firmware` となります。このパラメータは必須です。

オプション	説明
-d	指定されたデバイス(<code>fctx</code> の形式、 <code>x</code> は <code>fiio-status</code> に表示されるデバイス番号)をアップデートします。このオプションが指定されなかった場合、すべてのデバイスがアップデートされます。 <div style="background-color: yellow; padding: 5px;"> このオプションは注意して使用してください。誤った Fusion ioMemory デバイスをアップデートすると、デバイスが破損することがあります。</div>
-f	強制アップデートを行います(カスタマーサポートから指定された場合のみ使用してください)。もし Fusion ioMemory VSL ドライバがロードされていない場合、このオプションは追加で <code>-s</code> オプションが必要です。 <div style="background-color: yellow; padding: 5px;"> このオプションは注意して使用してください。デバイスが破損することがあります。</div>
-l	アーカイブ内にあるファームウェアを一覧表示します。
-p	Pretend: どんなアップデートが行われるかを表示します。しかし、ファームウェアはアップデート行われません。
-c	デバイスにあるロックをクリアします。
-q	Quiet: 進捗とパーセンテージの表示を無効化します。
-Q	Quiet: 進捗の表示のみを無効化します。
-y	すべての警告メッセージに自動で <code>"yes"</code> と回答します。
-s	指定されたスロットにあるデバイスをアップデートします。 <code>"**"</code> のワイルドカードを使用して複数指定することもできます。スロットは、次の PCIe フォーマットで特定できます(<code>lspci</code> の表示となります)： <div style="border: 1px solid black; padding: 5px; margin-top: 5px;">[[[<domain>]:]<bus>:][<slot>][. [<func>]]</div>

もしこのセクションに、別のページからたどり着いた場合、元のページに戻ってください。

付録 B - デバイスの Health 情報を監視

この章では、Fusion ioMemory デバイスの Health 情報を監視し、データを保護しデバイスの寿命を確認する方法をご案内しています。

Health 情報の法則

Fusion ioMemory VSL ソフトウェアは、あらかじめ用意された閾値を使用してブロックの交替を管理します。fio-status ユーティリティは health indicator を表示し、100 で開始され 0 に向かってカウントダウンします。閾値を超えた場合、いくつかのアクションが実行されます。

10%の health 閾値では、一度の警告が発行されます。アラームを受領するには、次の章を参照してください。

0%では、デバイスは不具合があると判断されます。3%で write-reduced mode に移行し、寿命の延長とデータの保護を試みます。この状態では、Fusion ioMemory デバイスは、書き込み性能は抑制されますは、それ以外は通常どおり動作します。

1%以下になると、デバイスは read-only mode に移行し、Fusion ioMemory デバイスへのどんな書き込みもエラーとなります。ファイルシステムによっては、read-only のブロックデバイスをマウントするため、特殊なオプションを指定し、また read-only でマウントされる必要があります。

例えば Linux では、ext3 は“-o ro,noload”が使用されます。“noload”オプションでは、ファイルシステムのジャーナルは更新しません。

Read-only mode は、デバイスからデータを取り出す最後の機会です、その後にデバイスを継続して使用することで寿命となる可能性があります。

Fusion ioMemory デバイスは failure mode に移行する場合があります。この場合、デバイスはオフライン状態でアクセスできません。この原因は、内部の致命的な不良、誤ったファームウェアのアップデート手順、またはデバイスの寿命です。

- ❶ 本製品は「有寿命部品」となっております。詳細は弊社のホームページを参照ください。
<http://jp.fujitsu.com/platform/server/primergy/support/parts/>
- ❷ 複数の Fusion ioMemory デバイスが搭載されている場合、デバイスのモードはそれぞれのデバイスで別々に制御されます。

Health 監視の手法

fio-status -a: fio-status ユーティリティ(-a オプションつき)の表示に、health のパーセンテージやデバイスの状態が表示されます。以下は表示例です。

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

fio-status ユーティリティでは、次のような Health 状態が表示されます：

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

ソフトウェア RAID と Health 監視

ソフトウェア RAID スタックは一般的に、古くからあるストレージメディアに対して、故障を検出しリカバリする設計になっています。Fusion ioMemory デバイスは、可能な限り緩やかに不良へ移行するよう試みるため、この不良メカニズムは多くのソフトウェア RAID スタックと互換性があります。


RAID グループ内の Fusion ioMemory デバイスは、次のようなケースで不良となります： a) デバイスが **write-reduced** 状態で遅延している場合、b) 高負荷な書き込みワークロードに含まれている場合。これらの場合、最悪はデバイスは RAID グループから除外される可能性があります。デバイスが **read-only** モードの場合も、書き込み I/O が発行されると RAID グループから除外されます。致命的な故障も、従来のストレージデバイスと同様に制御されます。

付録 C - モジュールパラメータの使用

次の表では、`/etc/modprobe.d/iomemory-vsl4.conf` を編集し値を指定することで設定可能なモジュールパラメータを説明しています。

それぞれの設定ファイルのモジュールパラメータは、`iomemory-vsl4` のオプションとして使用されます。ファイル `/etc/modprobe.d/iomemory-vsl4.conf` はいくつかのパラメータ例を、コメントアウトされた形で含んでいます。これらのコメントアウトを解除して使用することもできます。

- i** これらの変更は、有効になるためには Fusion ioMemory VSL ソフトウェアがロードされる前に完了する必要があります。

モジュールパラメータ	デフォルト (最小, 最大)	説明
<code>auto_attach</code>	1 (0, 1)	1 (デフォルト) = ドライバをロードする時、常にデバイスを <code>attach</code> する。 0 = ドライバをロードする時、デバイスを <code>attach</code> しない。
<code>disable_msi</code>	0 (0, 1)	MSI-X を含む Message Signal Interrupts を無効にする。
<code>disable_msix</code>	0 (0, 1)	MSI-X のみを無効にする。
<code>external_power_override</code>	デバイス選択なし	選択したデバイスが PCIe slot からすべての電源を取得する。 <div style="background-color: yellow; padding: 5px; border: 1px solid black;"> 注意して使用してください。詳細は PCIe Power Override の使用 を参照してください。</div>
<code>fio_dev_wait_timeout_secs</code>	30	ドライバをロードする際、 <code>/dev/fio*</code> ファイルの表示を待つ時間を、秒単位で指定します。Udev を使用していないシステムでは、これは 0 に設定されタイムアウトを無効化し、不要なドライバロード時の待ち時間を避けることができます。
<code>force_minimal_mode</code>	0	1 = デバイスを強制的に Minimal Mode にします。 0 = デバイスを強制的に Minimal Mode にしません。
<code>numa_node_override</code>	選択なし	<affinity specification>の組み合わせのリストで、システム上のデバイスの <code>affinity</code> を指定します。それぞれの組み合わせの中ではイコール(“=”)で分割され、また組み合わせごとにはコロン(“:”)で分割されます。<affinity specification>の組み合わせは次の構文となります： <device-id>=<node-number> 詳細は、 NUMA の設定 を参照してください。

モジュールパラメータ	デフォルト (最小, 最大)	説明
preallocate_memory	デバイス選 択なし	選択されたデバイスに対し、 swap としての使用など、あらかじめ必要なメモリを確保します。このパラメータの <value>は、デバイスのシリアル番号をコンマ区切りでリストしたものです。
preallocate_mb	値の選択なし	このパラメータの<value>は、MB 単位のシステムメモリ量で、Fusion ioMemory VSL ソフトウェアがそれぞれの Fusion ioMemory デバイスに対して、あらかじめ確保します。例えば、3500 という値を指定した場合、Fusion ioMemory VSL ソフトウェアは、Fusion ioMemory デバイスごとにおよそ 3.5GB の RAM を確保します。詳細は デバイスを Swap として使用する を参照してください。
rmap_memory_limit_MiB	3100 (0, 100000)	予期しないシャットダウン後の rescan 時間を改善するための、システムメモリ (RAM) の両を MiB 単位で指定します。詳細は Rescan の時間を改善する を参照してください。
rsort_memory_limit_MiB	0 (0, 100000)	予期しないシャットダウン後の rescan 時間を改善するための、システムメモリ (RAM) の両を MiB 単位で指定します。詳細は Rescan の時間を改善する を参照してください。
use_workqueue	0 (0 or 3)	Linux のみ: 3 = OS 標準の I/O evelators を使用します; 0 = バイパスします

- external_power_override と preallocate_memory 以外のパラメータはグローバルで、コンピューター上のすべての Fusion ioMemory デバイスに適用されます。

付録 D - NUMA の設定

⚠ 不要なパラメータ

NUMA node パラメータにより、NUMA node の affinity を設定することができますが、多くの場合は不要です。Fusion ioMemory VSL ソフトウェアは自動的に、Fusion ioMemory デバイスと NUMA node の割り込み affinity を作成します。しかし、手動でこれらのパラメータで NUMA node の設定を変えたい場合に、この付録を参照します。

NUMA アーキテクチャーについて

NUMA (Non-Uniform Memory Access) アーキテクチャーが提供されているサーバでは、特別なインストール手順を使って Fusion ioMemory デバイスのパフォーマンスを最大化できる可能性があります。これは、多くの複数ソケットのサーバが対象です。

いくつかの NUMA アーキテクチャーのサーバでは、システムのブート時、BIOS は PCIe slot を適切な NUMA node に割り当てません。正しくないマッピングでは、非効率な I/O ハンドリングと極端な性能低下となる可能性があります。

numa_node_override パラメータの使用

このパラメータを使用し、デバイスを特定の NUMA node に割り当てることができます。

⚠ この例では、最終的な変更した affinity 設定を示しています。通常、システムごとに仕様を確認しながら調査し、またそれに合わせて Fusion ioMemory デバイスと PCI slot を指定する必要があります。ご使用のシステムに依存した調査と設定が必要です。

お使いのシステムに設定すべき値は、システムの構成に依存して、下に示す例とは異なることがあります。正しい設定を作成するには、`fio-status` を使って全てのデバイスを一覧表示し<device-id>(下記)を特定してください。その後、`numa_node_override` パラメータのテンプレートを編集し、お使いのシステム構成に合わせてください。

Device ID を特定する

次のフォーマットで<device-id>を提示する必要があります：

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

多くの場合、domain は 0000 ですが、サーバに依存します。または、`fio-status` の表示にも Device ID が表示されます。例えば：

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

この例では、device ID は 0000:04:00.0 と 0000:15:00.0 となり、また domain 0000 です。

❌ PCI device ID の bus number は、システム上の PCI デバイス構成を変更した場合に、変わる可能性があります。例えば、ネットワークカードまたは別の Fusion ioMemory デバイスを追加した場合です。Device ID が変わった場合、この設定も更新する必要があります。

numa_node_override パラメータ

NUMA アーキテクチャーに Fusion ioMemory デバイスを設定するには、iomemory-vs14 ファイルを編集し numa_node_override parameter を使用する必要があります。

numa_node_override パラメータは、システム上のデバイスの<affinity specification>組み合わせのリストになっています。組み合わせの中ではイコール(=)で分割され、また組み合わせのアイテムごとにはコロン(:)で分割されます。

構文:

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

それぞれの <affinity specification> は次の構文となります :

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

簡単な例 :

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

この場合、次のように作成されます :

<device-id>	Node/Group	Processor Affinity
0000:04:00.0	node 1	node 1 のすべてのプロセッサ
0000:1d:00.0	node 0	node 0 のすべてのプロセッサ
0000:05:00.0	node 2	node 2 のすべてのプロセッサ
0000:1e:00.0	node 3	node 3 のすべてのプロセッサ

拡張設定

ご使用のサーバが複数の NUMA node を持っており、また複数の Fusion ioMemory デバイスが搭載されている場合、Fusion ioMemory デバイスは複数の node に分散されていることを確認する必要があります。

デバイスと node は、電氣的に近い PCIe slot との組み合わせに設定されることが最適であることから(ご使用のサーバと、その NUMA アーキテクチャーに関して、深い理解が必要)、単純にすべてのデバイスの node affinity を node にまたがって分散させることで、性能が改善できることがあります。

お客様サポート

次の Web サイトでサポート対象の OS を確認し、最新ドライバをダウンロードできます。

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

日本市場の場合は以下の URL をご使用ください。

<http://jp.fujitsu.com/platform/server/primergy/>

常に最新版をご使用いただくことを推奨します。