



Red Hat Enterprise Linux 7

DM Multipath

DM Multipath の設定と管理

Red Hat Enterprise Linux 7 DM Multipath

DM Multipath の設定と管理

Steven Levine

Red Hat Customer Content Services

slevine@redhat.com

法律上の通知

Copyright © 2017 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドは、Red Hat Enterprise Linux 7 で Device Mapper Multipath を使用する方法について説明しています。

目次

第1章 DEVICE MAPPER を使ったマルチパス設定	3
1.1. 新機能および変更された機能	3
1.2. DM MULTIPATH の概要	4
1.3. ストレージレイの対応	7
1.4. DM MULTIPATH のコンポーネント	7
1.5. DM MULTIPATH 設定の概要	8
第2章 マルチパスデバイス	9
2.1. マルチパスデバイス識別子	9
2.2. クラスタ内で整合性のあるマルチパスデバイス名の維持	9
2.3. マルチパスデバイスの属性	10
2.4. 論理ボリューム内のマルチパスデバイス	10
第3章 DM MULTIPATH の設定	12
3.1. DM MULTIPATH の設定	12
3.2. マルチパスデバイスの作成時にローカルのディスクを無視	13
3.3. ストレージデバイスの設定	15
3.4. INITRAMFS ファイルシステムでマルチパスの設定	16
第4章 DM MULTIPATH 設定ファイル	17
4.1. 設定ファイルの概要	17
4.2. 設定ファイルの BLACKLIST セクション	18
4.3. 設定ファイルの DEFAULTS セクション	21
4.4. 設定ファイルの MULTIPATHS セクション	31
4.5. 設定ファイルの DEVICES セクション	36
第5章 DM MULTIPATH の管理とトラブルシューティング	44
5.1. MULTIPATH HELPER で自動設定ファイルの生成	44
5.2. オンラインのマルチパスデバイスのサイズ変更	44
5.3. ROOT ファイルシステムを、シングルパスデバイスからマルチパスデバイスへ移行	44
5.4. SWAP ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行	46
5.5. マルチパスデーモン	46
5.6. 多数の LUN に伴う問題	47
5.7. QUEUE_IF_NO_PATH 機能に関する問題	47
5.8. MULTIPATH コマンドの出力	47
5.9. MULTIPATH コマンドを使ったマルチパスクエリー	48
5.10. MULTIPATH コマンドのオプション	49
5.11. DMSETUP コマンドを使ってデバイスマッパーのエントリーを特定	49
5.12. MULTIPATHD コマンド	50
5.13. 対話式コンソール MULTIPATHD を使用したトラブルシューティング	50
5.14. パッケージ削除時におけるマルチパスファイルのクリーンアップ	51
付録A 改訂履歴	52
索引	53

第1章 DEVICE MAPPER を使ったマルチパス設定

DM-Multipath (Device mapper multipath) を使用すると、複数のサーバーノードと複数のストレージレイとの間にある複数の I/O パスを、1つのデバイスに設定できます。I/O パスとは物理的な SAN 接続のことで、各ケーブルやスイッチ、コントローラーなどが含まれます。複数の I/O パスを集め、集めたパスで構成される新しいデバイスを1つ作成するのがマルチパス機能です。

この章では、Red Hat Enterprise Linux 7 の初期リリースの後に追加された DM-Multipath の機能についてまとめています。また、DM-Multipath および各コンポーネントについての高水準の概要、DM-Multipath の設定についての概要についても説明しています。

1.1. 新機能および変更された機能

本項では、Red Hat Enterprise Linux 7 の初期リリース以降の DM Multipath の新機能について説明します。

1.1.1. Red Hat Enterprise Linux 7.1 で新たに追加された機能および変更点

Red Hat Enterprise Linux 7.1 では、ドキュメントと機能が以下のように更新、変更されています。

- [表5.1「便利な multipath コマンドのオプション」](#) に、`multipath` コマンドの `-w` と `-W` のオプションに関する説明が加えられています。このオプションを使用すると `wwids` ファイルの管理が容易になります。
- `multipath.conf` ファイルの `features` パラメーターの `values` 引数に使用できる追加オプションの説明が、「[4章DM Multipath 設定ファイル](#)」に加えられています。
- [表4.1「マルチパス設定の defaults セクション」](#) に `force_sync` パラメーターの説明が加えられています。このパラメーターを「yes」に設定すると、パスチェッカーが非同期モードでは実行されなくなります。

さらに、ドキュメント全体にわたり、技術的な内容の若干の修正と明確化を行いました。

1.1.2. Red Hat Enterprise Linux 7.2 の新機能および変更された機能

Red Hat Enterprise Linux 7.2 では、ドキュメントと機能が以下のように更新、変更されています。

- 本書には新しい項「[Multipath Helper で自動設定ファイルの生成](#)」が加えられました。Multipath Helper アプリケーションを使用して、カスタムエイリアスとデバイスブラックリストのマルチパス設定や、個々のマルチパスデバイスの特性向け設定を作成できます。
- `multipath.conf` 設定ファイルの `defaults` セクションで、パラメーター `config_dir`、`new_bindings_in_boot`、`ignore_new_boot_devs`、`retrigger_tries`、および `retrigger_delays` がサポートされるようになりました。`multipath.conf` ファイルの `defaults` セクションについては、[表4.1「マルチパス設定の defaults セクション」](#) を参照してください。
- `multipath.conf` 設定ファイルの `defaults`、`devices`、および `multipaths` のセクションで、設定パラメーター `delay_watch_checks` および `delay_wait_checks` がサポートされるようになりました。これらの設定パラメーターの詳細については、「[4章DM Multipath 設定ファイル](#)」を参照してください。

さらに、ドキュメント全体にわたり、技術的な内容の若干の修正と明確化を行いました。

1.1.3. Red Hat Enterprise Linux 7.3 の新機能および変更された機能

Red Hat Enterprise Linux 7.3 では、ドキュメントと機能が以下のように更新、変更されています。

- **multipathd** コマンドが、"raw" 形式のバージョンでマルチパスデバイスとパスのステータスを表示する新しい形式のコマンドをサポートするようになりました。raw 形式ではヘッダーはプリントされず、フィールドはヘッダーと整合するようにコラムにパディングがされていません。代わりに、書式設定文字列に指定された通りにフィールドをプリントします。**multipathd** コマンドの詳細は、「[multipathd コマンド](#)」を参照してください。
- Red Hat Enterprise Linux 7.3 以降、デバイス設定で **prio "alua exclusive_pref_bit"** を指定すると、マルチパスが **pref** ビットセットのパスのみを含むパスグループを作成し、このパスグループに最高の優先度を与えるようになります。設定パラメーターの詳細は、「[4章DM Multipath 設定ファイル](#)」を参照してください。
- **multipath.conf** 設定ファイルの **defaults**、**devices**、および **multipaths** の各セクションで **skip_kpartx** 設定パラメーターがサポートされるようになりました。これらの設定パラメーターの詳細は、「[4章DM Multipath 設定ファイル](#)」を参照してください。

さらに、ドキュメント全体にわたり、技術的な内容の若干の修正と明確化を行いました。

1.1.4. Red Hat Enterprise Linux 7.4 の新機能および変更された機能

Red Hat Enterprise Linux 7.4 では、ドキュメントと機能が以下のように更新、変更されています。

- **multipath.conf** 設定ファイルの **defaults**、**devices**、および **multipaths** の各セクションで、**max_sectors_kb** 設定パラメーターをサポートするようになりました。設定パラメーターの詳細は、「[4章DM Multipath 設定ファイル](#)」を参照してください。
- **multipath.conf** 設定ファイルの **defaults** セクションおよび **devices** セクションで、**detect_path_checker** 設定パラメーターをサポートするようになりました。設定パラメーターの詳細は、「[4章DM Multipath 設定ファイル](#)」を参照してください。
- **multipath.conf** 設定ファイルの **defaults** セクションで、**remove_retries** パラメーターと **detect_path_checker** パラメーターをサポートするようになりました。**multipath.conf** ファイルの **defaults** セクションの詳細は、[表4.1「マルチパス設定の defaults セクション」](#) を参照してください。

1.2. DM MULTIPATH の概要

DM Multipath は以下のような目的に使用できます。

- 冗長性

DM-Multipath を active/passive に設定するとフェイルオーバーを行うことができます。active/passive に設定すると、常に全体の半分のパスだけが使用されます。I/O パス上のいずれかの構成要素(ケーブル、スイッチ、コントローラーなど)に障害が発生すると、DM-Multipath により別のパスへの切り替えが行われます。

- パフォーマンスの向上

DM-Multipath を active/active モードに設定すると、I/O はラウンドロビン式でパス全体に分散されます。I/O パスの負荷を検出し、動的に負荷分散の再調整を行うことができる設定もあります。

図1.1 「active/passive マルチパス設定 - RAID デバイスが1つ」に、サーバーから RAID デバイスへの I/O パスが2つある active/passive 設定を示します。サーバーには HBA、SAN スイッチ、RAID コントローラーが2つずつあります。

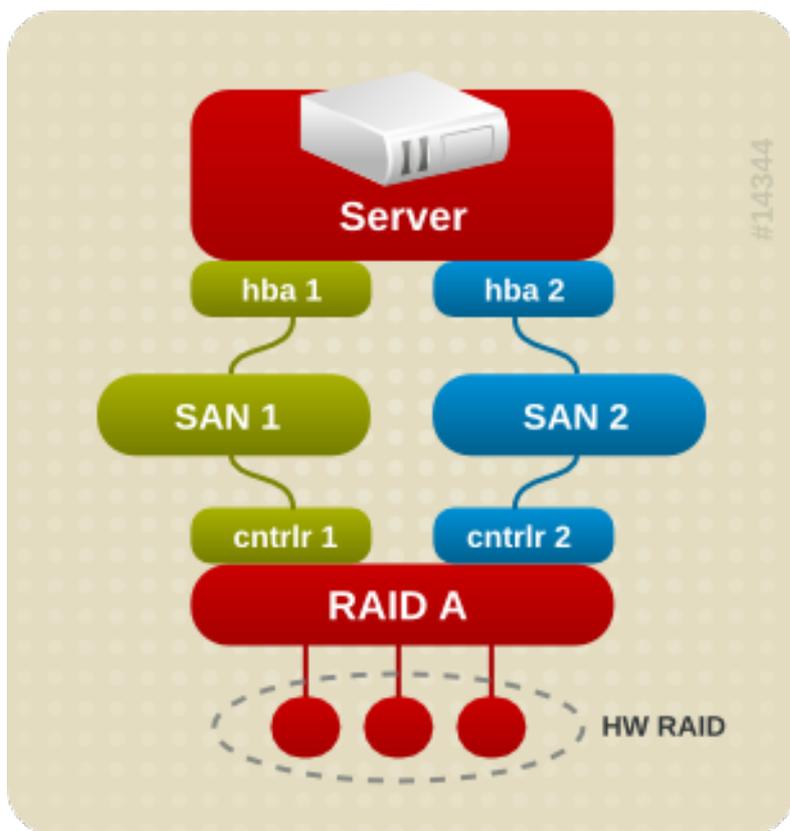


図1.1 active/passive マルチパス設定 - RAID デバイスが1つ

この設定では、hba1、SAN1、およびコントローラー1を通る I/O パスと、hba2、SAN2、およびコントローラー2を通る I/O パスの2つのパスがあります。この設定には、障害が発生する可能性のあるポイントが数多く存在します。

- HBA の障害
- FC ケーブルの障害
- SAN スイッチの障害
- アレイコントローラーポートの障害

DM-Multipath が設定されていると、このようなポイントで障害が発生したときに、DM-Multipath により代替 I/O パスへの切り替えが行われます。

図1.2 「active/passive マルチパス設定 - RAID デバイスが2つ」ではもう少し複雑な active/passive 設定を示します。サーバーには HBA、SAN スイッチがそれぞれ2つあり、RAID コントローラーがそれぞれ2つ付いた RAID デバイスが2つあります。

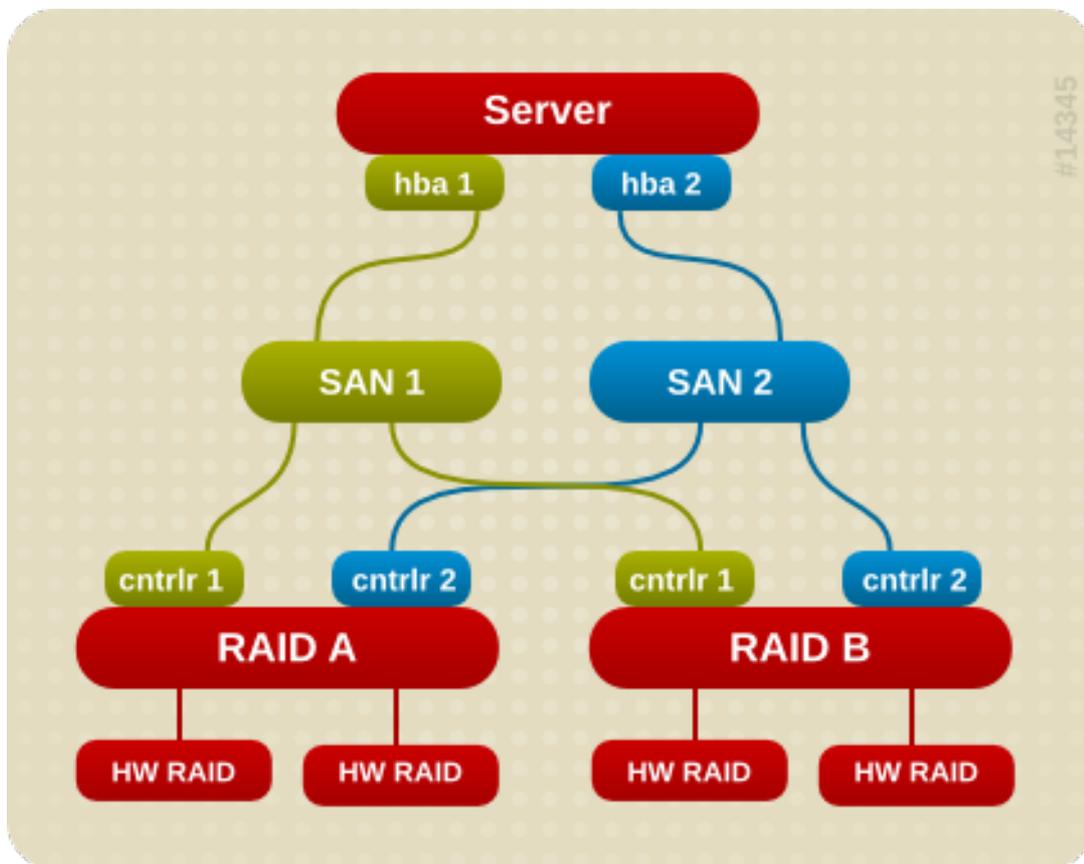


図1.2 active/passive マルチパス設定 - RAID デバイスが2つ

図1.2「active/passive マルチパス設定 - RAID デバイスが2つ」に示した例では、それぞれの RAID デバイスに2つの I/O パスがあります (図1.1「active/passive マルチパス設定 - RAID デバイスが1つ」に示した例と同様)。DM-Multipath が設定されていると、RAID デバイスへの I/O パスで障害が発生したときに、DM-Multipath が、I/O パスを別のパスに切り替えます。

図1.3「active/active マルチパス設定 - RAID デバイスが1つ」では、サーバーに HBA が2つ、SAN スイッチが1つ、RAID コントローラーが2つある active/active 設定を示します。サーバーからストレージデバイスへの I/O パスは4つあります。

- hba1 から controller1 まで
- hba1 から controller2 まで
- hba2 から controller1 まで
- hba2 から controller2 まで

この設定では、I/O をこれらの4つのパスに分散することが可能です。

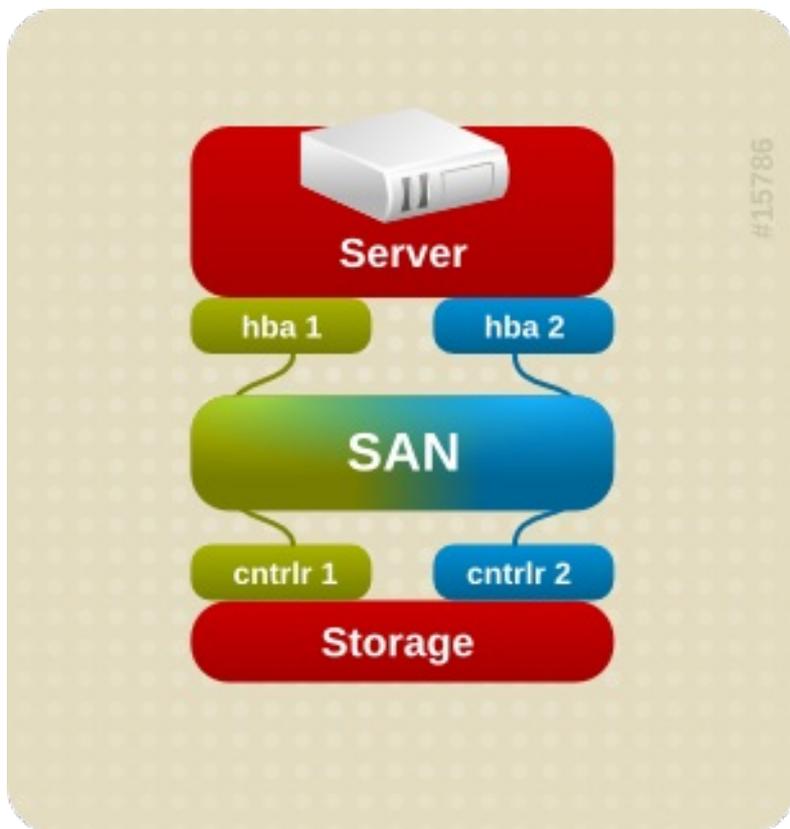


図1.3 active/active マルチパス設定 - RAID デバイスが1つ

1.3. ストレージアレイの対応

DM-Multipath に対応しているストレージアレイで最も一般的なものについては DM-Multipath の設定にデフォルトで含まれています。対応デバイスなどデフォルトの設定値については、以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

ストレージアレイが DM Multipath に対応していてもデフォルトの設定には含まれていない場合は、DM Multipath 設定ファイルの **multipath.conf** にそのストレージアレイを追加する必要があります。DM Multipath 設定ファイルの詳細は「[4章 DM Multipath 設定ファイル](#)」を参照してください。

ストレージアレイの中には、I/O エラーとパスの切り替えに特殊な処理を必要とする場合があります。こうしたストレージアレイの場合には、別途ハードウェアハンドラーカーネルモジュールが必要になります。

1.4. DM MULTIPATH のコンポーネント

表1.1 「DM Multipath のコンポーネント」 で、DM-Multipath のコンポーネントを説明します。

表1.1 DM Multipath のコンポーネント

コンポーネント	説明
dm_multipath カーネルモジュール	I/O を再ルーティングして、パス群とパスグループのフェイルオーバーをサポートします。

コンポーネント	説明
mpathconf ユーティリティ	デバイスマッパーのマルチパスを設定し、有効化します。
multipath コマンド	マルチパスデバイスを表示、設定します。通常、 <code>/etc/rc.sysinit</code> で起動しますが、ブロックデバイスを追加した場合は udev プログラムで起動することもできます。
multipathd デーモン	パスを監視します。パスに障害が発生し復旧した際、パスグループスイッチを起動できます。対話式でマルチパスデバイスに変更を加えることができます。 <code>/etc/multipath.conf</code> ファイルを変更した場合は、常にデーモンを再起動する必要があります。
kpartx コマンド	デバイス上のパーティション用にデバイスマッパーデバイスを作成します。DM Multipath の DOS ベースパーティションの場合はこのコマンドを使用する必要があります。 kpartx はそれ自体のパッケージで提供されますが、 device-mapper-multipath パッケージはそのパッケージに依存します。

1.5. DM MULTIPATH 設定の概要

DM Multipath には、一般的なマルチパス設定に適したデフォルト設定がコンパイルされています。DM Multipath をシステムに設定する基本的手順を以下に示します。

1. **device-mapper-multipath** rpm をインストールします。
2. **mpathconf** コマンドで設定ファイルを作成し、マルチパスを有効にします。設定ファイルを編集する必要がない場合は、このコマンドでマルチパスデーモンを起動することもできます。
3. 必要に応じて **multipath.conf** 設定ファイルを編集し、デフォルト値の変更、更新したファイルの保存を行います。
4. マルチパスデーモンを起動します。

マルチパス設定の手順については「[3章 DM Multipath の設定](#)」をご覧ください。

第2章 マルチパスデバイス

DM-Multipath を使用しない場合、サーバーノードからストレージコントローラーへの各パスは、システムでは別々のデバイスとして扱われます。I/O パスが同じサーバーノード、同じストレージコントローラーをつないでいても同様です。DM-Multipath は複数のデバイスをまとめて一つのマルチパスデバイスを作成し、複数の I/O のパスを論理的に整理する方法を提供します。

2.1. マルチパスデバイス識別子

マルチパスデバイスは、それぞれグローバルに固有で不変となる World Wide Identifier (WWID) を持っています。マルチパスデバイス名はデフォルトでは WWID が付けられます。マルチパス設定ファイル内に `user_friendly_names` オプションを設定すると、`mpathn` 形式の、ノード固有の名前にエイリアスを作成することができます。

例えば、HBA を 2 つ搭載したノードとポートが 2 つあるストレージコントローラーを、ゾーニングされていない 1 つの FC スイッチで接続した場合、`/dev/sda`、`/dev/sdb`、`dev/sdc`、`/dev/sdd` の 4 つのデバイスが認識されます。DM-Multipath はマルチパス設定に従ってこれら 4 つのデバイスへ I/O を再ルーティングする 1 つのデバイスを、固有の WWID を付けて作成します。`user_friendly_names` 設定オプションを `yes` に設定していた場合は、マルチパスデバイスの名前が `mpathn` になります。

新しいデバイスを DM-Multipath の管理下に置くと、`/dev` ディレクトリ配下の `/dev/mapper/mpathn` と `/dev/dm-n` の 2 ヶ所に表示されます。

- `/dev/mapper` のデバイスは起動プロセスの初期段階で作成され、論理ボリュームの作成時など、マルチパス設定したデバイスへのアクセスに使用されます。
- `/dev/dm-n` 形式のデバイスはすべて内部的な使用を目的としているため、管理者の方は絶対に使用しないでください。

`user_friendly_names` 設定オプションなど、マルチパス設定のデフォルト値については「[設定ファイルの defaults セクション](#)」を参照してください。

マルチパスデバイスに任意の名前を付けることもできます。マルチパス設定ファイルの `multipaths` セクションにある `alias` オプションを使用します。マルチパス設定ファイルの `multipaths` セクションについての詳細は「[設定ファイルの multipaths セクション](#)」を参照してください。

2.2. クラスタ内で整合性のあるマルチパスデバイス名の維持

`user_friendly_names` 設定オプションを `yes` に設定すると、マルチパスデバイスの名前が 1 ノードに対して固有となりますが、そのマルチパスデバイスを使用している全ノードで同じ名前になる保証はありません。同様に、`multipath.conf` 設定ファイルの `multipaths` セクションでデバイスに `alias` オプションを設定すると、その名前が自動的にクラスタ内の全ノードで同じにはなりません。LVM を使用してマルチパスデバイスから複数の論理デバイスを作成している場合は問題にならないはずですが、すべてのノードでマルチパスデバイス名を同じ名前にする必要がある場合は、`user_friendly_names` オプションを `yes` に設定しない、またデバイスにエイリアスを設定しないことをお勧めします。`user_friendly_names` を `yes` に設定したりデバイスにエイリアスを設定しなければ、デバイス名はデフォルトでそのデバイスの WWID になるため、整合性を維持することができます。

ただし、システムで定義される `user_friendly_names` を、クラスタ内の全ノードに対して固有にしたい場合には、以下の手順に従うと整合性を維持することができます。

1. すべてのマルチパスデバイスの設定をひとつのマシンで行います。

2. 次のコマンドを使って、他のマシンのマルチパスデバイスをすべて無効にします。

```
# service multipathd stop
# multipath -F
```

3. クラスタ内の 1 番目のマシンの `/etc/multipath/bindings` ファイルを他のすべてのマシンにコピーします。
4. 先ほど `multipathd` デーモンを無効にしたすべてのマシンで以下のコマンドを実行して、再度デーモンを有効にします。

```
# service multipathd start
```

新しいデバイスを追加する場合は、この手順を繰り返す必要があります。

同様に、エイリアスもクラスタ内のすべてのノードに対して固有となるよう設定する場合は、同じ手順でクラスタ内の各ノードの `/etc/multipath.conf` ファイルが同じになるようにします。

1. 1 つのマシンの `multipath.conf` ファイルで、マルチパスデバイスのエイリアスを設定します。
2. 次のコマンドを使って、他のマシンのマルチパスデバイスをすべて無効にします。

```
# service multipathd stop
# multipath -F
```

3. そのマシンの `/etc/multipath.conf` ファイルを、クラスタ内にある他のすべてのマシンにコピーします。
4. 先ほど `multipathd` デーモンを無効にしたすべてのマシンで以下のコマンドを実行して、再度デーモンを有効にします。

```
# service multipathd start
```

新しいデバイスを追加する場合は、この手順を繰り返す必要があります。

2.3. マルチパスデバイスの属性

`user_friendly_names` オプションや `alias` オプション以外にも、マルチパスデバイスには多くの属性があります。マルチパス設定ファイルの `multipaths` セクションに、特定のマルチパスデバイス用のエントリーを作成すると、そのデバイスの属性を変更することができます。マルチパス設定ファイルの `multipaths` セクションについては「[設定ファイルの multipaths セクション](#)」を参照してください。

2.4. 論理ボリューム内のマルチパスデバイス

マルチパスデバイスを作成したら、LVM 物理ボリュームを作成する時に物理デバイス名を使用するのと同じように、マルチデバイス名を使用することができます。例えば、マルチパスデバイスの名前が `/dev/mapper/mpath0` の場合は、以下のコマンドを実行すると `/dev/mapper/mpatha` が物理ボリュームとして認識されます。

```
pvcreate /dev/mapper/mpatha
```

上記のコマンドで得られた LVM 物理デバイスは、他の LVM 物理デバイスと同じように、LVM ボリュームグループを作成する時に使用することができます。



注記

複数のパーティションが設定されたデバイス全体に LVM 物理ボリュームを作成しようとすると、**pvcreate** コマンドは失敗します。特に指定しない限り、Anaconda や Kickstart インストールプログラムは、すべてのブロックデバイスに対して空のパーティションテーブルを作成するので注意してください。パーティションではなくデバイス全体を使用したい場合には、そのデバイスから既存のパーティションを削除する必要があります。既存パーティションの削除には、**kpartx -d** や **fdisk** のコマンドを使用できます。システムに 2TB 以上のブロックデバイスがある場合は、**parted** コマンドを使用してパーティションを削除します。

複数の物理デバイスで、active/passive のアレイを構成しマルチパスを使用する LVM 論理ボリュームを作成する場合は、`/etc/lvm/lvm.conf` ファイルにフィルターを追加して、このような複数デバイスで構成されマルチパスを使用するディスクを除外してください。フィルターを使って除外しないと、I/O の受信時に、active パスが自動的に passive パスに変更し、LVM でその passive パスがスキャンされるたびに、マルチパスによりフェイルオーバーとフェイルバックが繰り返されることとなります。passive パスを active パスに変更する際にコマンドを必要とする active/passive アレイの場合、このような状態が発生すると LVM で警告メッセージが表示されます。

LVM 設定ファイル (`lvm.conf`) ですべての SCSI デバイスをフィルターするには、そのファイルの **devices** セクションに、以下のフィルターを組み込みます。

```
filter = [ "r/block/", "r/disk/", "r/sd.*/", "a/*.*/" ]
```

第3章 DM MULTIPATH の設定

この章では、DM-Multipath を設定する以下のような手順を、順を追って説明していきます。

- DM-Multipath の基本設定
- ローカルディスクを無視する
- 設定ファイルにデバイスを追加する
- `initramfs` ファイルシステムでマルチパスを起動する

3.1. DM MULTIPATH の設定

システムに DM-Multipath をセットアップする前に、システムが更新されており、`device-mapper-multipath` パッケージがインストールされていることを確認してください。

`mpathconf` ユーティリティを使用してマルチパスを設定します。マルチパス設定ファイル `/etc/multipath.conf` が作成されます。

- `/etc/multipath.conf` ファイルが既に存在する場合は、`mpathconf` ユーティリティでファイルを編集します。
- `/etc/multipath.conf` ファイルが存在しない場合は、`/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` ファイルから作成します。
- `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` ファイルが存在しない場合は、`mpathconf` ユーティリティにより `/etc/multipath.conf` ファイルがゼロから作成されます。

`/etc/multipath.conf` ファイルを編集する必要がない場合は、以下のコマンドを実行すると、基本的なフェイルオーバー設定用の DM-Multipath をセットアップすることができます。このコマンドでマルチパス設定が有効になり、`multipathd` デーモンが起動されます。

```
# mpathconf --enable --with_multipathd y
```

`multipathd` デーモンを起動する前に `/etc/multipath.conf` ファイルを編集する必要がある場合は、以下の手順に従い基本的なフェイルオーバー設定用の DM-Multipath をセットアップします。

1. `--enable` オプションを指定して `mpathconf` コマンドを実行します。

```
# mpathconf --enable
```

`mpathconf` コマンドで利用できるその他のオプションについては `mpathconf` の man ページをご覧ください。か、`mpathconf` コマンドに `--help` オプションを指定して実行します。

```
# mpathconf --help
usage: /sbin/mpathconf <command>
```

Commands:

Enable: `--enable`

Disable: `--disable`

Set `user_friendly_names` (Default `y`): `--user_friendly_names <y|n>`

Set `find_multipaths` (Default `y`): `--find_multipaths <y|n>`

```
Load the dm-multipath modules on enable (Default y): --with_module
<y|n>
start/stop/reload multipathd (Default n): --with_multipathd <y|n>
```

- 必要に応じて `/etc/multipath.conf` ファイルを編集します。DM-Multipath 用のデフォルト設定はシステムにコンパイルされているため、`/etc/multipath.conf` ファイルで明示的に指定する必要はありません。

`path_grouping_policy` のデフォルト値は `failover` にセットされています。このため、この例ではデフォルト値を変更する必要はありません。設定ファイル内のデフォルト値の変更に関する詳細は「[4章 DM Multipath 設定ファイル](#)」をご覧ください。

設定ファイルの初期デフォルトセクションによりマルチパスデバイス名が `mpathn` の形式になるよう設定されます。この設定がないと、マルチパスデバイスの名前はそのデバイスの WWID にエイリアス設定されます。

- 必要に応じて設定ファイルを保存し、エディターを終了します。
- 以下のコマンドを実行します。

```
# service multipathd start
```

設定ファイルで、`user_friendly_names` の値を `yes` に設定したため、マルチパスデバイスが `/dev/mapper/mpathn` に作成されます。選択しているエイリアスにデバイス名を設定する方法は、「[4章 DM Multipath 設定ファイル](#)」を参照してください。

`user_friendly_name` を使用しない場合は、以下のコマンドを実行します。

```
# mpathconf --enable --user_friendly_names n
```



注記

マルチパスデーモンを起動してから、マルチパス設定ファイルの編集が必要になった場合は、その後 `service multipathd reload` コマンドを実行して、変更を反映させてください。

3.2. マルチパスデバイスの作成時にローカルのディスクを無視

内蔵ディスクにローカルの SCSI カードが搭載されているマシンがあります。これらのデバイスに DM-Multipath を使用することは推奨されません。`find_multipaths` 設定パラメーターを `yes` にセットすると、こうしたデバイスをブラックリストに指定する必要がなくなります。`find_multipaths` 設定パラメーターについては「[設定ファイルの defaults セクション](#)」を参照してください。

`find_multipaths` 設定パラメーターを `yes` にセットしない場合は、以下の手順に従ってマルチパス設定ファイルを修正すると、マルチパスの設定時にローカルのディスクを無視できます。

- どのディスクが内蔵ディスクかを確認し、そのディスクに、ブラックリストとして指定する印を付けます。

以下の例では `/dev/sda` が内蔵ディスクです。デフォルトのマルチパス設定ファイル内で元々設定されているように、`multipath -v2` を実行するとマルチパスマップ内にローカルディスク `/dev/sda` が表示されます。

multipath コマンドの出力に関する詳細は「[Multipath コマンドの出力](#)」をご覧ください。

```
# multipath -v2
create: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef
WINSYS,SF2372
size=33 GB features="0" hwhandler="0" wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 0:0:0:0 sda 8:0  [-----]

device-mapper ioctl cmd 9 failed: Invalid argument
device-mapper ioctl cmd 14 failed: No such device or address
create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16  undef ready  running
  ` - 3:0:0:0 sdf 8:80  undef ready  running

create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32  undef ready  running
  ` - 3:0:0:1 sdg 8:96  undef ready  running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48  undef ready  running
  ` - 3:0:0:2 sdg 8:112 undef ready  running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64  undef ready  running
  ` - 3:0:0:3 sdg 8:128 undef ready  running
```

2. デバイスマッパーによる `/dev/sda` のマルチパスマップへのマッピングを阻止するため、`/etc/multipath.conf` ファイルの `blacklist` セクションを編集してこのデバイスを追加します。`devnode` タイプで `sda` デバイスをブラックリストに指定することも可能ですが、`/dev/sda` が再起動後に同じく `/dev/sda` になる保証はないため安全な方法ではありません。デバイスを 1 つ 1 つブラックリストに指定する場合は、そのデバイスの WWID を使用します。

multipath -v2 コマンドに対する出力を見ると、`/dev/sda` デバイスの WWID は `SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1` になっています。このデバイスをブラックリストに指定する場合は `/etc/multipath.conf` ファイルに以下の記述を含めます。

```
blacklist {
    wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}
```

3. `/etc/multipath.conf` ファイルを更新したら、`multipathd` デーモンにファイルを再ロードするよう、手動で指示する必要があります。以下のコマンドで、更新した `/etc/multipath.conf` ファイルを再ロードします。

```
# service multipathd reload
```

4. 以下のコマンドを実行して、マルチパスデバイスを削除します。

```
# multipath -f SIBM-ESXSST336732LC_____F3ET0EP0Q000072428BX1
```

5. **multipath -ll** コマンドを実行して現在のマルチパス設定を表示させ、削除が正しく行われたかを確認します。**multipath -ll** コマンドについては「[multipath コマンドを使ったマルチパスクエリー](#)」を参照してください。

ブラックリストに指定されたデバイスが再度追加されていないことを確認する場合は、以下の例のように **multipath** コマンドを実行します。**-v** オプションを指定しないと、**multipath** コマンドのデフォルト詳細レベルは **v2** になります。

```
# multipath

create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16 undef ready running
  `- 3:0:0:0 sdf 8:80 undef ready running

create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready running
  `- 3:0:0:1 sdg 8:96 undef ready running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready running
  `- 3:0:0:2 sdg 8:112 undef ready running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready running
  `- 3:0:0:3 sdg 8:128 undef ready running
```

3.3. ストレージデバイスの設定

DM-Multipath に対応しているストレージレイで最も一般的なものについては、DM-Multipath の設定にデフォルトで含まれています。対応デバイスなど、デフォルトの設定値については、以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

既知のマルチパスデバイスとして、デフォルトではサポートされていないストレージデバイスを追加する必要がある場合は、**/etc/multipath.conf** ファイルを編集して適切なデバイス情報を記入します。

例えば、HP Open-V シリーズに関する情報を追加する場合は、以下のようなエントリーを記述します。この例では、すべてのパスに障害が発生した場合には、1 分間 (または 5 秒ごとの再試行で 12 回の再試行を行う) のキュー待ちを行うようデバイスを設定しています。

```
devices {
    device {
        vendor "HP"
        product "OPEN-V"
        no_path_retry 12
    }
}
```

設定ファイルの **devices** セクションについての詳細は「[設定ファイルの devices セクション](#)」をご覧ください。

3.4. INITRAMFS ファイルシステムでマルチパスの設定

initramfs ファイルシステムでマルチパスを設定することができます。マルチパスの設定後、以下のオプションを追加して **dracut** を実行すると、マルチパス設定ファイルで **initramfs** ファイルシステムを再構築できます。

```
# dracut --force --add multipath --include /etc/multipath
```

initramfs ファイルシステムからマルチパスを実行してマルチパス設定ファイルに変更を加える場合、変更を有効にするには **initramfs** ファイルシステムを再構築する必要があります。

第4章 DM MULTIPATH 設定ファイル

DM Multipath では、マルチパスでもっともよく使用する設定値がデフォルトで提供されています。また DM-Multipath に対応しているストレージレイで最も一般的なものについては、DM-Multipath の設定に含まれています。対応デバイスなど、デフォルトの設定値については、以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

DM-Multipath の設定値は、`/etc/multipath.conf` 設定ファイルを編集すると無効にできます。また、必要に応じて、対応はしていますがデフォルトでは設定ファイルに含まれていないストレージレイを、設定ファイルに追加することもできます。



注記

initramfs ファイルシステムでマルチパスの設定を実行することができません。**initramfs** ファイルシステムでマルチパスを実行して、マルチパス設定ファイルに変更を加えた場合は、**initramfs** ファイルシステムを再構築して変更を反映する必要があります。マルチパスで **initramfs** ファイルシステムを再構築する方法については「[initramfs ファイルシステムでマルチパスの設定](#)」を参照してください。

本章では、**multipath.conf** ファイルの解析や変更について説明します。以下のトピックに関する項が含まれます。

- 設定ファイルの概要
- 設定ファイルの blacklist セクション
- 設定ファイルの defaults セクション
- 設定ファイルの multipaths セクション
- 設定ファイルの devices セクション

マルチパス設定ファイルで指定するのは、使用状況に応じて必要とされるセクションや、デフォルト値の変更が必要なセクションのみです。使用環境には無関係なセクションや、デフォルト値を無効にする必要がないセクションは、初期ファイルに指定されているコメントアウトを削除する必要はありません。

設定ファイルでは、正規表現の記述構文を使用することができます。

設定ファイルの詳細については、**multipath.conf(5)** の man ページをご覧ください。

4.1. 設定ファイルの概要

マルチパス設定ファイルは以下のセクションに分かれています。

blacklist

マルチパス設定の対象として考慮しないデバイスの一覧。

blacklist_exceptions

ブラックリストの例外としてマルチパス設定の対象となるデバイスの一覧。この例外セクションに指定されていない場合は、`blacklist` セクションのパラメーターに従ってブラックリストに指定されません。

defaults

DM Multipath のデフォルトの全般設定。

multipaths

マルチパスデバイスの特性に関する個別設定。ここで指定する値は、設定ファイルの **defaults** と **devices** のセクションで指定されている値より優先されます。

devices

ストレージコントローラーの個別設定。ここで指定する値は、設定ファイル内の **defaults** セクションで指定されている値より優先されます。対応はしていますがデフォルトでは設定ファイルに含まれていないストレージアレイの場合は、そのアレイ用の **devices** サブセクションを作成する必要があります。

システムでマルチパスデバイスの属性が確認される場合は、`multipath` セクション、`devices` セクション、`defaults` セクションの順に設定がチェックされます。

4.2. 設定ファイルの **BLACKLIST** セクション

マルチパス設定ファイルの **blacklist** セクションには、システムがマルチパスデバイスを設定する際に使用しないデバイスを指定します。ブラックリストに指定したデバイスは、マルチパスデバイスには含まれません。

以前の Red Hat Enterprise Linux のリリースでは、明示的にブラックリストに指定されていないすべてのパスに、マルチパスデバイスの作成を試行していました。Red Hat Enterprise Linux 6 からは、`find_multipaths` 設定パラメーターを **yes** に設定すると、マルチパスによるデバイスの作成が、以下の 3 つの条件のうちのいずれかを満たす場合に限られます。

- ブラックリストに指定されておらず、同一の WWID を持つパスが少なくとも 2 つある場合。
- 手動で `multipath` コマンドを実行して、デバイスを指定して作成を強制している場合。
- 任意のパスが、以前に作成されたマルチパスデバイスと同一の WWID を持っている場合 (そのマルチパスデバイスがその時点で存在していない場合も含む)。マルチパスデバイスを作成すると、常にそのデバイスの WWID が記憶されるため、マルチパスは、同一の WWID を持つパスを認識すると、自動的にそのデバイスを再度作成します。これによって、マルチパスのブラックリストを編集しなくても、マルチパスが自動的に正しいパスを選択してマルチパスデバイスにすることができます。

`find_multipaths` パラメーターを使用しないでマルチパスデバイスを作成してから、パラメーターを **yes** に設定し直した場合は、`/etc/multipath/wwids` ファイルから、マルチパスデバイスとして作成しないデバイスの WWID を削除しなければならない可能性もあります。以下は、`/etc/multipath/wwids` ファイルの例を示しています。スラッシュ (/) で囲まれている部分が WWID です。

```
# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and
multipathd.
# You should not need to edit this file in normal circumstances.
```

```
#
# Valid WWIDs:
/3600d023000000000000e13955cc3757802/
/3600d023000000000000e13955cc3757801/
/3600d023000000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372 0E13955CC3757802/
/3600d023000000000000e13955cc3757803/
```

find_multipaths パラメーターを **yes** に設定すれば、複数のパスがあるにも関わらずマルチパスを設定しないデバイスだけを、ブラックリストに指定する必要があります。このため、一般的にはデバイスをブラックリストに指定する必要はありません。

デバイスをブラックリストを指定する必要がある場合は、以下の基準に従って行います。

- WWID で指定する場合は、「[WWID でブラックリストの指定](#)」を参照してください。
- デバイス名で指定する場合は、「[デバイス名でブラックリストの指定](#)」を参照してください。
- デバイスタイプで指定する場合は、「[デバイスタイプでブラックリストの指定](#)」を参照してください。

デフォルトでは、設定ファイルの初期ブラックリストセクションをコメントアウトした後でも、各種のデバイスタイプがブラックリストに指定されます。詳細は「[デバイス名でブラックリストの指定](#)」をご覧ください。

4.2.1. WWID でブラックリストの指定

設定ファイルの **blacklist** セクションで **wwid** エントリーを使用し、World-Wide IDentification でデバイスをブラックリストに指定できます。

以下の例では、WWID が 26353900f02796769 であるデバイスをブラックリストに指定しています。

```
blacklist {
    wwid 26353900f02796769
}
```

4.2.2. デバイス名でブラックリストの指定

設定ファイルの **blacklist** セクションで **devnode** エントリーを使用し、デバイス名でデバイスタイプをブラックリストに指定し、マルチパスデバイスに含まれないようにすることができます。

以下の例では、sd* のデバイスをすべてブラックリストに指定することで、すべての SCSI デバイスをブラックリストに指定しています。

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

blacklist セクションで **devnode** エントリーを使用すると、特定タイプに該当する全デバイスではなく、個々のデバイスをブラックリストに指定することができます。ただし、デバイスが **udev** ルールで静的にマッピングされていない限り、再起動後にそのデバイス名が同じ名前になる保証がないため、この方法は推奨できません。再起動すると、デバイス名が **/dev/sda** から **/dev/sdb** などに変わる可能性があります。

デフォルトでは、以下の **devnode** エントリーがデフォルトの **blacklist** にコンパイルされています。エントリーでブラックリストに指定しているデバイスは、通常 DM-Multipath に対応していません。これらのデバイスでマルチパスを有効にする場合は、「[ブラックリストから除外](#)」で説明しているように、設定ファイルの **blacklist_exceptions** セクションで除外の指定をする必要があります。

```
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^(td|ha)d[a-z]"
}
```

4.2.3. デバイスタイプでブラックリストの指定

設定ファイルの **blacklist** セクションに **device** セクションを使用し、特定のデバイスタイプを指定できます。以下の例では、IBM DS4200 デバイスと HP デバイスをすべてブラックリストに指定します。

```
blacklist {
    device {
        vendor "IBM"
        product "3S42"          #DS4200 Product 10
    }
    device {
        vendor "HP"
        product "*"
    }
}
```

4.2.4. ブラックリストから除外

設定ファイルの **blacklist_exceptions** セクションを使用して、デフォルトではブラックリストに指定されているデバイスで、マルチパスを有効にできます。

例えば、デバイスが多数あり、その中の1つのデバイス (以下の例では WWID が 3600d0230000000000e13955cc3757803 のデバイス) でのみマルチパスを有効にしたい場合は、有効にしたいデバイス以外のものを1つ1つブラックリストに指定するのではなく、一旦すべてのデバイスをブラックリストに指定してから、**/etc/multipath.conf** ファイルに以下の行を追加すれば、目的のデバイスにだけマルチパスを許可することができます。

```
blacklist {
    wwid "*"
}

blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}
```

設定ファイルの **blacklist_exceptions** セクションでデバイスを指定する場合は、**blacklist** で指定したのと同じ方法で指定しなければなりません。例えば、**blacklist** で **devnode** エントリーを使ってデバイスを指定している場合は、**exception** で除外指定に使用している WWID が正しくても適用されません。同様に **devnode** による除外は、**devnode** エントリーでブラックリストを指定したデバイスにしか適用されず、**device** による除外は、**device** エントリーでブラックリストを指定したデバイスにしか適用されません。

4.3. 設定ファイルの DEFAULTS セクション

`/etc/multipath.conf` 設定ファイルには `defaults` セクションがあり、以下のように `user_friendly_names` パラメーターを `yes` にセットすることができます。

```
defaults {
    user_friendly_names yes
}
```

上記は、`user_friendly_names` パラメーターのデフォルト値を上書きします。

設定ファイルに `defaults` のテンプレートが含まれています。このセクションは、以下のようにコメントアウトされています。

```
#defaults {
#    polling_interval          10
#    path_selector              "round-robin 0"
#    path_grouping_policy      multibus
#    uid_attribute              ID_SERIAL
#    prio                       alua
#    path_checker               readsector0
#    rr_min_io                  100
#    max_fds                    8192
#    rr_weight                  priorities
#    failback                   immediate
#    no_path_retry              fail
#    user_friendly_names       yes
#}
```

設定パラメーターのデフォルト値を上書きする場合は、このテンプレートの該当行を `defaults` セクションにコピーして、コメントを外します。例えば、`path_grouping_policy` パラメーターを、デフォルト値の `failover` から `multibus` に上書きするには、以下のように、テンプレートで該当行を見つけて設定ファイルの `defaults` セクションにコピーし、そのコメントを外します。

```
defaults {
    user_friendly_names    yes
    path_grouping_policy  multibus
}
```

`multipath.conf` 設定ファイルの `defaults` セクションに設定する属性を [表4.1「マルチパス設定の defaults セクション」](#) で説明します。`multipath.conf` ファイルの `devices` および `multipaths` セクションにおける属性指定で上書きされない限り、これらの値が DM-Multipath で使用されます。

表4.1 マルチパス設定の defaults セクション

属性	説明
<code>polling_interval</code>	パスチェックが行われる間隔を秒数で指定します。適正に機能するパスでは、チェックの間隔は $(4 * \text{polling_interval})$ まで徐々に増加します。デフォルト値は 5 です。

属性	説明
multipath_dir	動的共有オブジェクトが保管されるディレクトリーです。デフォルト値はシステムによって異なり、一般的には /lib/multipath になります。
find_multipaths	<p>マルチパスデバイスのセットアップモードを定義します。このパラメーターが yes に設定されていると、マルチパスは、ブラックリスト化されていないパスにはデバイス作成を試行しません。以下にあげる3つの条件のいずれかを満たす場合のみ、デバイスを作成します。</p> <ul style="list-style-type: none"> - ブラックリストに指定されておらず、同一の WWID を持つパスが少なくとも2つある場合 - ユーザーが手動で multipath コマンドを実行し、デバイスを指定して作成を強制している場合 - 任意のパスが、以前に作成されたマルチパスデバイスと同一の WWID を持っている場合。 find_multipaths を設定した状態でマルチパスデバイスを作成すると、そのデバイスの WWID が記憶されるため、マルチパスが同一の WWID を持つパスを認識するとすぐに、そのデバイスを自動的に再作成します。これによって、マルチパスのブラックリストを編集しなくても、マルチパスが自動的に正しいパスを選択してマルチパスデバイスにすることができます。 find_multipaths パラメーターを設定せずにマルチパスデバイスを既に作成している場合は、「設定ファイルの blacklist セクション」 の説明を参照してください。 <p>デフォルト値は no です。ただし、Red Hat Enterprise Linux 7 では、mpathconf が作成するデフォルトの multipath.conf ファイルにより、find_multipaths が有効になります。</p>
reassign_maps	デバイスマッパーマップの再割り当てを有効にします。このオプションを使用すると、 multipathd デーモンは、既存のデバイスマッパーのマップを、基本的なブロックデバイスではなく、常にマルチパスデバイスに向けてるように再マップします。使用可能な値は、 yes または no です。デフォルト値は yes です。
verbosity	デフォルトの詳細度です。値が高いほど詳細レベルが高くなります。使用できるレベルは0から6の間です。デフォルト値は 2 です。

属性	説明
<p>path_selector</p>	<p>次回の I/O 動作に使用するパスを決定する、デフォルトのアルゴリズムを指定します。使用できる値を以下に示します。</p> <p>round-robin 0: パスグループ内の全てのパスをループスルーし、それぞれのパスに同負荷の I/O を送ります。</p> <p>queue-length 0: 未処理の I/O 要求数が最も少ないパスに、次の I/O 群を送ります。</p> <p>service-time 0: 推定処理時間が最も短いパスに、次の I/O 群を送ります。推定処理時間は、各パスに対する未処理の I/O の合計サイズを、その相対スループットで割ることにより求められます。</p> <p>デフォルト値は round-robin 0 です。</p>
<p>path_grouping_policy</p>	<p>未指定のマルチパスに適用する、デフォルトのパスグルーピングポリシーを指定します。使用できる値を以下に示します。</p> <p>failover: 優先グループごとに 1 つのパス。</p> <p>multibus: 1 つの優先グループ内で有効なすべてのパス。</p> <p>group_by_serial: 検出されたシリアル番号ごとに 1 つの優先グループ。</p> <p>group_by_prio: パス優先値ごとに 1 つの優先グループ。優先度はグローバル、コントローラー別、またはマルチパスオプション別に指定されているコールアウトプログラムによって決定されます。</p> <p>group_by_node_name: ターゲットノード名ごとに 1 つの優先グループ。ターゲットノード名は <code>/sys/class/fc_transport/target*/node_name</code> 内で取得されます。</p> <p>デフォルト値は failover です。</p>

属性	説明
prio	<p>パスの優先値を得るために呼び出すデフォルトの関数を指定します。例えば、SPC-3 の ALUA ビットは悪用可能な prio 値を提供します。使用できる値を以下に示します。</p> <p>const: すべてのパスに優先度 1 を設定します。</p> <p>emc: EMC アレイのパス優先度を生成します。</p> <p>alua: SCSI-3 ALUA 設定に基づいてパス優先度を生成します。Red Hat Enterprise Linux 7.3 以降では、デバイス設定で prio "alua exclusive_pref_bit" を指定すると、マルチパスは pref ビットセットのパスのみを含むパスグループを作成し、このパスグループに最高の優先度を与えます。</p> <p>ontap: NetApp アレイのパスの優先度を生成します。</p> <p>rdac: LSI/Engenio RDAC コントローラーのパスの優先度を生成します。</p> <p>hp_sw: active/standby モードにおける Compaq/HP コントローラー用パスの優先度を生成します。</p> <p>hds: Hitachi HDS Modular ストレージアレイのパスの優先度を生成します。</p> <p>デフォルト値は const です。</p>

属性	説明
features	<p>マルチパスデバイスの追加デフォルト機能です。形式は "<i>number_of_features_plus_arguments feature1 ...</i>" となります。</p> <p>features に使用できる値:</p> <p>queue_if_no_path: no_path_retry を queue に設定するのと同じです。この機能の使用時に発生する可能性がある問題については「queue_if_no_path 機能に関する問題」を参照してください。</p> <p>retain_attached_hw_handler: SCSI 層によりハードウェアハンドラーがパスデバイスに接続されている状態でこのパラメーターを yes に設定した場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p> <p>pg_init_retries <i>n</i>: 失敗するまでパスグループの初期化を最高 <i>n</i> 回再試行します。ここで、$1 \leq n \leq 50$ です。</p> <p>pg_init_delay_msecs <i>n</i>: パスグループの初期化の再試行を、<i>n</i> ミリ秒間隔で待機します。ここで、$0 \leq n \leq 60000$ です。</p>
path_checker	<p>パスの状態を判断するのに使用するデフォルトのメソッドを指定します。使用できる値を以下に示します。</p> <p>readsector0: デバイスの最初のセクターを読み取ります。</p> <p>tur: デバイスに対して TEST UNIT READY コマンドを発行します。</p> <p>emc_clariion: EMC Clariion 固有の EVPD ページ 0xC0 の問い合わせを行い、パスを特定します。</p> <p>hp_sw: Active/Standby のファームウェアを搭載した HP ストレージアレイのパスの状態をチェックします。</p> <p>rdac: LSI/Engenio RDAC ストレージコントローラーのパスの状態をチェックします。</p> <p>directio: 直接 I/O を使用する最初のセクターを読み取ります。</p> <p>デフォルト値は directio です。</p>

属性	説明
failback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値を設定すると、active パスを含んでいる最も優先度の高いパスグループに直ちにフェイルバックします。</p> <p>manual の値を指定すると、フェイルバックはすぐに行われず、オペレーター介入がある場合のみにフェイルバックします。</p> <p>followover の値を指定すると、パスグループの最初のパスが active になったときに自動フェイルバックを実行します。これにより、別のノードがフェイルオーバーを要求しているときは、ノードが自動的にフェイルバックしなくなります。</p> <p>「0」以上の数値で、フェイルバックの秒単位の遅延を指定します。</p> <p>デフォルト値は manual です。</p>
rr_min_io	<p>現在のパスグループで、次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。この設定は、2.6.31 より前のカーネルを実行しているシステムにのみ適用されます。2.6.31 以降のシステムでは、rr_min_io_rq を使用してください。デフォルト値は 1000 です。</p>
rr_min_io_rq	<p>現在のパスグループで、次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。要求ベースの device-mapper-multipath を使用しています。現在のカーネルを実行しているシステムで使用してください。2.6.31 より前のバージョンのカーネルを実行しているシステムの場合は rr_min_io を使用してください。デフォルト値は 1 です。</p>
rr_weight	<p>priorities に設定すると、次のパスを選択する selector を呼び出すまでに送信する要求数の決定に rr_min_io を使用しません。送信する要求数は、rr_min_io に、prio 関数で決定されるパスの優先度を掛けて決定します。uniform に設定すると、全てのパスの重みが等しくなります。デフォルト値は uniform です。</p>
no_path_retry	<p>障害の発生したパスの使用を何回試行してからキュー待ちを無効にするか、その回数を指定します。</p> <p>fail を指定すると、キュー待ちはせず直ちに失敗します。</p> <p>queue を指定すると、パスが修復されるまでキュー待ちは停止しません。</p> <p>デフォルト値は 0 です。</p>

属性	説明
user_friendly_names	yes に設定すると、 <code>/etc/multipath/bindings</code> ファイルを使った永続的で固有となるエイリアスが、 <code>mpathn</code> の形式でマルチパスに割り当てられます。 no に設定すると、WWID がマルチパスのエイリアスに使用されます。ただし、設定ファイルの multipaths セクションで指定されるデバイス固有のエイリアスの方が、ここでの設定よりも優先されることとなります。デフォルト値は no です。
queue_without_daemon	no に設定すると、 <code>multipathd</code> デーモンにより、シャットダウン時にすべてのデバイスのキュー待ちが無効になります。デフォルト値は no です。
flush_on_last_del	yes に設定すると、 <code>multipathd</code> デーモンにより、デバイスへの最後のパスが削除された時点でキュー待ちが無効になります。デフォルト値は no です。
max_fds	マルチパスおよび <code>multipathd</code> デーモンで開くことが可能な、オープンファイル記述子の最大数をセットします。これは、 <code>ulimit -n</code> コマンドに相当します。Red Hat Enterprise Linux 6.3 リリース以降、デフォルト値は max になっています。つまり、 <code>/proc/sys/fs/nr_open</code> に記述されたシステムの上限に設定されます。これより前のリリースでは、値を設定しないと、オープンファイル記述子の最大数は呼び出しのプロセスが取得し、通常は 1024 です。この数が 1024 を超える場合は、念のため、パスの最大数に 32 を加えた数を設定してください。
checker_timeout	明示的なタイムアウトで、SCSI コマンドを発行するパス checker および prioritizer に使用するタイムアウト (秒) です。デフォルト値は <code>sys/block/sdx/device/timeout</code> から取得します。
fast_io_fail_tmo	FC リモートポートで問題を検出してから、そのリモートポートでデバイスへの I/O に失敗するまでの scsi 層の待機時間 (秒) です。この値は <code>dev_loss_tmo</code> よりも小さくする必要があります。これを off に設定すると、タイムアウトが無効になります。デフォルト値は OS により決定されます。
dev_loss_tmo	FC リモートポートで問題を検出してから、その問題がシステムから削除されるまでの scsi 層の待機時間 (秒) です。無限に設定する場合は、2147483647 秒または 68 年に設定します。デフォルト値は OS により決定されます。

属性	説明
hw_string_match	<p>multipath.conf ファイルの devices セクション内の各デバイス設定から独自のデバイス設定を作成するか、組み込みデバイス設定の1つを修正します。hw_string_match を yes に設定すると、ユーザーのデバイス設定内のベンダー、製品、およびリビジョンの文字列が、組み込みのデバイス設定の文字列と完全に一致する場合には、組み込みの設定がユーザー設定オプションで修正されます。完全に一致しない場合、ユーザーのデバイス設定は新しい設定として処理されます。hw_string_match を no に設定すると、文字列による一致ではなく正規表現による一致が使用されます。</p> <p>hw_string_match パラメーターは、デフォルトで no に設定されます。</p>
retain_attached_hw_handler	<p>SCSI 層で、ハードウェアハンドラーがパスデバイスに接続されている状態で、このパラメーターを yes に設定した場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層でハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p>
detect_prio	<p>yes に設定すると、最初にデバイスで ALUA に対応しているかをチェックします。対応している場合は alua prioritizer をデバイスに割り当てます。デバイスが ALUA に対応していない場合は通常通り prioritizer の特定を行います。デフォルト値は no です。</p>
uid_attribute	<p>固有のパス識別子を与えます。デフォルト値は ID_SERIAL です。</p>
force_sync	<p>(Red Hat Enterprise Linux Release 7.1 以降) 「yes」に設定すると、パスチェッカーは非同期モードで実行しません。つまり、一度に実行されるチェッカーは1つのみになります。これは、複数の multipathd チェッカーが並行して実行されていて、CPU に大きなプレッシャーをかけている場合に便利です。デフォルト値は no です。</p>
delay_watch_checks	<p>(Red Hat Enterprise Linux Release 7.2 以降) 0 よりも大きい値に設定すると、multipathd デーモンは、指定されたチェック回数の間、有効なパスを監視します。監視されているパスが再び失敗すると、パスが次に有効になったときに、delay_wait_checks で指定された連続的なチェック回数の間保持されるまで使用されません。これにより、信用できないパスがオンラインに戻ったときにすぐに使用されることを防ぐことができます。デフォルト値は no です。</p>
delay_wait_checks	<p>(Red Hat Enterprise Linux Release 7.2 以降) 0 よりも大きい値に設定した場合は、オンラインに戻ったデバイスが delay_watch_checks で指定されたチェック回数内に再び失敗し、次にオンラインに戻るときに、デバイスはマークされ、遅延が発生します。デバイスは、delay_wait_checks で指定されたチェック回数を経過するまで使用されません。デフォルト値は no です。</p>

属性	説明
ignore_new_boot_devs	(Red Hat Enterprise Linux Release 7.2 以降) yes に設定すると、ブートの初期段階でノードが initramfs ファイルシステムに存在しているときに、WWID が /etc/multipath/wwids の initramfs copy に存在していないデバイスは、マルチパスによって作成されません。この機能は、インストール中のブートに使用できます (マルチパスが、 udev ルールによって最初に現れたときに要求しなかったデバイスに、設定を自ら試行する場合)。このパラメーターは yes または no に設定できます。設定しない場合のデフォルト値は no です。
retrigger_tries 、 retrigger_delay	(Red Hat Enterprise Linux Release 7.2 以降) retrigger_tries パラメーターと retrigger_delay パラメーターは、 udev が元の uevents の処理に完全に失敗し、マルチパスでデバイスを使用できなくなった場合に、 multipathd が uevents を再トリガーするために使用されます。 retrigger_tries パラメーターは、デバイスが完全にセットアップされていない場合にマルチパスが uevent の再トリガーを試行する回数を設定します。 retrigger_delay パラメーターは、再試行の間隔 (秒) を設定します。これら両方のオプションではゼロ以上の数値を使用できます。 retrigger_tries パラメーターをゼロに設定すると、再試行が無効になります。 retrigger_delay パラメーターをゼロに設定すると、パステッカーの次のループで uevent が再発行されます。 retrigger_tries パラメーターが設定されていない場合は、デフォルトで 3 に設定されます。 retrigger_delay パラメーターが設定されていない場合は、デフォルトで 10 に設定されます。
new_bindings_in_boot	(Red Hat Enterprise Linux Release 7.2 以降) new_bindings_in_boot パラメーターは、通常のファイルシステムのバインディングファイルによりすでに提供されている user_friendly_name を、マルチパスが initramfs ファイルシステムで提供しないようにするために使用されます (initramfs ファイルシステムが再作成された場合にのみ、 initramfs ファイルシステムの user_friendly_names バインディングが同期されるため発生する可能性がある問題)。このパラメーターを no に設定すると、マルチパスは initramfs ファイルシステムに新しいバインディングを作成しません。デバイスで /etc/multipath/bindings の initramfs コピーにバインディングがない場合、マルチパスは WWID をエイリアスとして使用し、 user_friendly_name を提供しません。ブートの後半で、ノードが通常のファイルシステムをマウントした後で、マルチパスはデバイスに user_friendly_name を提供します。このパラメーターは yes または no に設定できます。設定しない場合は、デフォルトで no に設定されます。

属性	説明
config_dir	(Red Hat Enterprise Linux Release 7.2 以降) "" 以外の値に設定すると、マルチパスは、このディレクトリーで ".conf" で終わるファイルをアルファベット順に検索し、まるで設定情報が /etc/multipath.conf ファイルに存在するかのよう、それらのファイルから設定情報を読み取ります。これにより、マシンごとに固有の設定ファイル以外に、マシン間で共有する 1 つの主要設定を持つことができます。 config_dir パラメーターは、"" または完全修飾ディレクトリー名のいずれかにする必要があります。このパラメーターは主要な /etc/multipath.conf ファイルでのみ設定でき、 config_dir ファイルに指定されたファイルには設定できません。デフォルト値は /etc/multipath/conf.d です。
deferred_remove	yes に設定すると、multipathd は、最終パスデバイスを削除するときに、通常の削除ではなく据置き削除を行います。これにより、通常の削除が行われて失敗し、multipathd デバイスが使用中である場合に、そのデバイスは、最終ユーザーがデバイスを終了したときに自動的に削除されます。デフォルト値は no です。
log_checker_err	once に設定すると、multipathd は、最初のパスチェッカーエラーを詳細レベル 2 でログに記録します。以降のすべてのエラーは、デバイスが復元されるまで詳細レベル 3 でログに記録されます。 always に設定されている場合、multipathd は、常に詳細レベル 2 でパスチェッカーエラーをログに記録します。デフォルト値は always です。
skip_kpartx	(Red Hat Enterprise Linux リリース 7.3 およびそれ以降では) yes に設定すると、 kpartx は自動的にデバイス上でパーティションを作成しません。これにより、デバイスにパーティションテーブルがある場合でも、ユーザーはパーティションを作成せずにマルチパスデバイスを作成することができます。このオプションのデフォルト値は no です。
max_sectors_kb	(Red Hat Enterprise Linux リリース 7.4 およびそれ以降では) マルチパスデバイスの初回アクティベート時の前に、 max_sectors_kb デバイスキューパラメーターを、マルチパスデバイスのすべての基礎的なパスで指定された値に設定します。マルチパスデバイスの作成時には、デバイスはパスデバイスから max_sectors_kb 値を継承します。手動でこの値をマルチパスデバイス向けに高めたり、パスデバイス向けにこの値を低くすると、マルチパスデバイスはパスデバイスが許可するよりも大きな I/O 操作を作成する場合があります。 max_sectors_kb パラメーターを使用すると、パスデバイス上にマルチパスデバイスを作成する前に容易にこれらの値が設定でき、無効なサイズの I/O 操作が渡されることを回避できます。このパラメーターを設定しないと、パスデバイスではデバイスドライバがこれを設定し、マルチパスデバイスがパスデバイスからこれを継承します。
remove_retries	(Red Hat Enterprise Linux Release 7.4 以降) マルチパスが、使用中のデバイスを削除するのを試行する回数を設定します。一度試行してから次の試行が行われるまで、マルチパスは 1 秒スリープ状態となります。デフォルトの値は 0 で、マルチパスは削除を試行しません。

属性	説明
disable_changed_wwids	(Red Hat Enterprise Linux Release 7.4 以降) yes に設定した場合は、マルチパスデバイスを構成しているパスデバイスの WWID が変更すると、パスの WWID がマルチパスデバイスの WWID に復元されるまで、マルチパスがパスデバイスへのアクセスを無効にします。デフォルト値は no で、パスの WWID が変更したかどうかを確認しません。
detect_path_checker	(Red Hat Enterprise Linux Release 7.4 以降) yes に設定すると、デバイスが ALUA に対応しているかどうかをマルチパスデバイスが確認します。対応している場合は、 tur パスチェッカーが自動的に使用されます。対応していない場合は、通常通りに path_checker が選択されます。デフォルト値は no です。

4.4. 設定ファイルの MULTIPATHS セクション

multipath.conf 設定ファイルの **multipaths** セクションで各特定マルチパスデバイスに対して設定できる属性を表4.2「Multipaths の属性」に示します。属性が適用されるのは、指定したマルチパスデバイスのみです。デフォルト値が DM-Multipath によって使用され、**multipath.conf** ファイルの **defaults** と **devices** セクションに設定した属性より優先されます。

表4.2 Multipaths の属性

属性	説明
wwid	multipath 属性を適用するマルチパスデバイスの WWID を指定します。このパラメーターは、 multipath.conf ファイルのこのセクションに必須となります。
alias	multipath 属性が適用されるマルチパスデバイスのシンボリック名を指定します。 user_friendly_names を使用する場合には、この値を mpathn に設定しないでください。設定すると、自動的に割り当てられる user_friendly_names と競合してしまうため、誤ったデバイスノード名が与えられる可能性があります。
path_grouping_policy	<p>未指定のマルチパスに適用する、デフォルトのパスグループポリシーを指定します。使用できる値を以下に示します。</p> <p>failover = 優先グループごとに1つのパス</p> <p>multibus = 1つの優先グループで有効なすべてのパス</p> <p>group_by_serial = 検出されたシリアル番号ごとに1つの優先グループ</p> <p>group_by_prio = パス優先値ごとに1つの優先グループ</p> <p>group_by_node_name = ターゲットノード名ごとに1つの優先グループ</p>

属性	説明
path_selector	<p>次回の I/O 動作に使用するパスを決定する、デフォルトのアルゴリズムを指定します。使用できる値を以下に示します。</p> <p>round-robin 0: パスグループ内の全てのパスをループスルーし、それぞれのパスに同負荷の I/O を送ります。</p> <p>queue-length 0: 未処理の I/O 要求数が最も少ないパスに、次の I/O 群を送ります。</p> <p>service-time 0: 推定処理時間が最も短いパスに、次の I/O 群を送ります。推定処理時間は、各パスに対する未処理の I/O の合計サイズを、その相対スループットで割ることにより求められます。</p>
failback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値を設定すると、active パスを含んでいる最も優先度の高いパスグループに直ちにフェイルバックします。</p> <p>manual の値を指定すると、フェイルバックはすぐに行われず、オペレーター介入がある場合のみにフェイルバックします。</p> <p>followover の値を指定すると、パスグループの最初のパスが active になったときに自動フェイルバックを実行します。これにより、別のノードがフェイルオーバーを要求しているときは、ノードが自動的にフェイルバックしなくなります。</p> <p>「0」以上の数値で、フェイルバックの秒単位の遅延を指定します。</p>

属性	説明
prio	<p>パスの優先値を得るために呼び出すデフォルトの関数を指定します。例えば、SPC-3 の ALUA ビットは悪用可能な prio 値を提供します。使用できる値を以下に示します。</p> <p>const: すべてのパスに優先度 1 を設定します。</p> <p>emc: EMC アレイのパス優先度を生成します。</p> <p>alua: SCSI-3 ALUA 設定に基づいてパス優先度を生成します。Red Hat Enterprise Linux 7.3 以降では、デバイス設定で prio "alua exclusive_pref_bit" を指定すると、マルチパスは pref ビットセットのパスのみを含むパスグループを作成し、このパスグループに最高の優先度を与えます。</p> <p>ontap: NetApp アレイのパスの優先度を生成します。</p> <p>rdac: LSI/Engenio RDAC コントローラーのパスの優先度を生成します。</p> <p>hp_sw: active/standby モードにおける Compaq/HP コントローラー用パスの優先度を生成します。</p> <p>hds: Hitachi HDS Modular ストレージアレイのパスの優先度を生成します。</p>
features	<p>マルチパスデバイスの追加デフォルト機能です。形式は <code>"number_of_features_plus_arguments feature1 ..."</code> となります。</p> <p>features に使用できる値:</p> <p>queue_if_no_path: no_path_retry を queue に設定するのと同じです。この機能の使用時に発生する可能性がある問題については 「queue_if_no_path 機能に関する問題」 を参照してください。</p> <p>retain_attached_hw_handler: SCSI 層によりハードウェアハンドラーがパスデバイスに接続されている状態でこのパラメーターを yes に設定した場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p> <p>pg_init_retries <i>n</i>: 失敗するまでパスグループの初期化を最高 <i>n</i> 回再試行します。ここで、$1 \leq n \leq 50$ です。</p> <p>pg_init_delay_msecs <i>n</i>: パスグループの初期化の再試行を、<i>n</i> ミリ秒間隔で待機します。ここで、$0 \leq n \leq 60000$ です。</p>

属性	説明
----	----

no_path_retry	<p>障害の発生したパスの使用を何回試行してからキュー待ちを無効にするか、その回数を指定します。</p> <p>fail を指定すると、キュー待ちはせず直ちに失敗します。</p> <p>queue を指定すると、パスが修復されるまでキュー待ちは停止しません。</p>
rr_min_io	<p>現在のパスグループで、次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。この設定は、2.6.31 より前のカーネルを実行しているシステムにのみ適用されます。2.6.31 以降のシステムには、rr_min_io_rq を使用してください。デフォルト値は 1000 です。</p>
rr_min_io_rq	<p>現在のパスグループで、次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。要求ベースの device-mapper-multipath を使用しています。現在のカーネルを実行しているシステムで使用してください。2.6.31 より前のバージョンのカーネルを実行しているシステムの場合は rr_min_io を使用してください。デフォルト値は 1 です。</p>

属性	説明
rr_weight	priorities に設定すると、次のパスを選択する path_selector を呼び出すまでに送信する要求数の決定に rr_min_io を使用しません。送信する要求数は、 rr_min_io に、 prio 関数で決定されるパスの優先度を掛けて決定します。 uniform に設定すると、全てのパスの重みが等しくなります。
flush_on_last_del	yes に設定すると、デバイスへの最後のパスが削除された時点でキュー待ちを無効にします。
user_friendly_names	yes に設定すると、 /etc/multipath/bindings ファイルを使った永続的で固有となるエイリアスが、 mpathn の形式でマルチパスに割り当てられます。 no に設定すると、WWID がマルチパスのエイリアスに使用されます。ただし、設定ファイルの multipaths セクションで指定されるデバイス固有のエイリアスの方が、ここでの設定よりも優先されることとなります。
delay_watch_checks	(Red Hat Enterprise Linux Release 7.2 以降) 0 よりも大きい値に設定した場合、 multipathd デーモンは、指定しているチェック回数の間、有効なパスを監視します。監視されているパスが再び失敗すると、パスが次に有効になったときに、 delay_wait_checks で指定した連続的なチェック回数の間保持されるまで使用されません。これにより、信用できないパスがオンラインに戻ったときにすぐに使用されることを防ぐことができます。
delay_wait_checks	(Red Hat Enterprise Linux Release 7.2 以降) 0 よりも大きい値に設定すると、オンラインに戻ったデバイスが、 delay_watch_checks で指定されたチェック回数内に再び失敗し、次にオンラインに戻るときに、デバイスはマークされ、遅延が発生します。デバイスは、 delay_wait_checks で指定されたチェック回数を経過するまで使用されません。
deferred_remove	yes に設定すると、 multipathd は最終パスデバイスが削除される際に通常の削除ではなく遅延された削除を行います。これにより、通常の削除が行われ、削除に失敗したときに multipathd デバイスが使用中である場合、デバイスは最終ユーザーがデバイスを終了したときに自動的に削除されます。
skip_kpartx	(Red Hat Enterprise Linux Release 7.3 以降) yes に設定すると、 kpartx は自動的にデバイスにパーティションを作成しません。これにより、デバイスにパーティションテーブルがある場合でも、パーティションを作成せずにマルチパスデバイスを作成することができます。

属性	説明
max_sectors_kb	(Red Hat Enterprise Linux Release 7.4 以降) マルチパスデバイスの初回アクティベートの前に、 max_sectors_kb デバイスキューパラメーターを、マルチパスデバイスの基本的なすべてのパスで指定されている値に設定します。マルチパスデバイスの作成時には、デバイスは、パスデバイスから max_sectors_kb 値を継承します。手動で、この値をマルチパスデバイス向けに高くしたり、パスデバイス向けに低くすると、マルチパスデバイスは、パスデバイスが許可するよりも大きな I/O 操作を作成する場合があります。 max_sectors_kb パラメーターを使用すると、パスデバイス上にマルチパスデバイスを作成する前に容易にこれらの値が設定でき、無効なサイズの I/O 操作が渡されることを回避できます。ユーザーがこのパラメーターを設定しないと、パスデバイスはデバイスドライバーにこれを設定させ、マルチパスデバイスがパスデバイスからこれを継承します。

設定ファイルで、2つの特定マルチパスデバイスに対して `multipath` 属性を指定している例を以下に示します。1番目のデバイスの WWID は **3600508b4000156d70001200000b0000** で、シンボリック名 **yellow** があります。

2番目のマルチパスデバイスの WWID は **1DEC_____321816758474** で、シンボリック名 **red** があります。この例では、`rr_weight` 属性に **priorities** を設定しています。

```

multipaths {
    multipath {
        wwid                3600508b4000156d70001200000b0000
        alias                yellow
        path_grouping_policy multibus
        path_selector        "round-robin 0"
        failback              manual
        rr_weight             priorities
        no_path_retry        5
    }
    multipath {
        wwid                1DEC_____321816758474
        alias                red
        rr_weight             priorities
    }
}

```

4.5. 設定ファイルの **DEVICES** セクション

`multipath.conf` 設定ファイルの `devices` セクションでストレージデバイスごとに設定できる属性を [表4.3「devicesの属性」](#) に示します。そのデバイスを含むパスでは、`multipath.conf` ファイルの `multipaths` セクションに指定されている属性で上書きされない限り、この属性が使用されます。この属性は、`multipath.conf` ファイルの `defaults` セクションに設定されている属性より優先されます。

マルチパスに対応しているデバイスの多くは、デフォルトでマルチパス設定に含まれています。対応デバイスなど、デフォルトの設定値については、以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

おそらく、こうしたデバイスの値を修正する必要はありませんが、必要が生じた場合は、設定ファイルにエントリーを追加すればデフォルト値を上書きすることができます。**multipathd show config** コマンドが表示する、目的のデバイスのデフォルト値をコピーして、変更したい値に書き換えることができます。

設定ファイルのこのセクションに、デフォルトで自動的に設定されないデバイスを追加する場合は、**vendor** と **product** のパラメーターを設定する必要があります。値は **/sys/block/device_name/device/vendor** と **/sys/block/device_name>/device/model** で検索できます。*device_name* は、マルチパス設定対象のデバイスです。以下に例を示します。

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

追加で指定するパラメーターは、使用するデバイスによって異なります。デバイスが **active/active** なら通常、追加でパラメーターを設定する必要はありません。**path_grouping_policy** を **multibus** に設定する必要が生じることがあるかもしれませんが、その他に設定が必要となる可能性があるパラメーターには、**no_path_retry** や **rr_min_io** があります。これらについては表4.3「**devices** の属性」で説明しています。

デバイスが **active/passive** であるにも関わらず、I/O を持つパスが自動的に **passive** パスに切り替えられる場合は、チェッカーの関数を、I/O をパスに送信しない関数に変更し、それが正しく動作するか検証する必要があります(これを行わないとデバイスはフェイルオーバーし続けます)。つまり、ほとんどの場合は、**path_checker** を **tur** に設定するということになります。この設定は、Test Unit Ready コマンドに対応するすべての SCSI デバイスで機能します。Test Unit Ready コマンドは大半のデバイスで対応しています。

パスの切り替えに特殊なコマンドを必要とするデバイスにマルチパスを設定するには、ハードウェアハンドラーカーネルモジュールが必要になります。現在、利用可能なハードウェアハンドラーは **emc** です。このハンドラーが目的のデバイスに使用できない場合、そのデバイスへのマルチパス設定はできない可能性があります。

表4.3 **devices** の属性

属性	説明
vendor	device 属性が適用されるストレージデバイスのベンダー名 (例: COMPAQ) を指定します。
product	device 属性が適用されるストレージデバイスの製品名 (例: HSV110 (C)COMPAQ) を指定します。
revision	ストレージデバイスの製品リビジョン識別子を指定します。
product_blacklist	製品別にデバイスをブラックリスト指定するのに使用する正規表現を指定します。

属性	説明
alias_prefix	デフォルトの "mpath" の代わりに、このデバイスタイプに使用する user_friendly_names 接頭辞。
hardware_handler	<p>パスグループの切り替えや I/O エラーの処理時に、ハードウェア固有の動作を行うのに使用するモジュールを指定します。使用できる値を以下に示します。</p> <p>1 emc: EMC ストレージアレイ用のハードウェアハンドラーです。</p> <p>1 alua: SCSI-3 ALUA アレイ用のハードウェアハンドラーです。</p> <p>1 hp_sw: Compaq/HP コントローラー用のハードウェアハンドラーです。</p> <p>1 rdac: LSI/Engenio RDAC コントローラー用のハードウェアハンドラーです。</p>
path_grouping_policy	<p>未指定のマルチパスに適用する、デフォルトのパスグループリングポリシーを指定します。使用できる値を以下に示します。</p> <p>failover = 優先グループごとに 1 つのパス</p> <p>multibus = 1 つの優先グループで有効なすべてのパス</p> <p>group_by_serial = 検出されたシリアル番号ごとに 1 つの優先グループ</p> <p>group_by_prio = パス優先値ごとに 1 つの優先グループ</p> <p>group_by_node_name = ターゲットノード名ごとに 1 つの優先グループ</p>
path_selector	<p>次回の I/O 動作に使用するパスを決定する、デフォルトのアルゴリズムを指定します。使用できる値を以下に示します。</p> <p>round-robin 0: パスグループ内の全てのパスをループスルーし、それぞれのパスに同負荷の I/O を送ります。</p> <p>queue-length 0: 未処理の I/O 要求数が最も少ないパスに、次の I/O 群を送ります。</p> <p>service-time 0: 推定処理時間が最も短いパスに、次の I/O 群を送ります。推定処理時間は、各パスに対する未処理の I/O の合計サイズを、その相対スループットで割ることにより求められます。</p>

属性	説明
path_checker	<p>パスの状態を判断するのに使用するデフォルトのメソッドを指定します。使用できる値を以下に示します。</p> <p>readsector0: デバイスの最初のセクターを読み取ります。</p> <p>tur: デバイスに対して TEST UNIT READY を発行します。</p> <p>emc_clariion: EMC Clariion 固有の EVPD ページ 0xC0 の問い合わせを行い、パスを特定します。</p> <p>hp_sw: Active/Standby のファームウェアを搭載した HP ストレージアレイのパスの状態をチェックします。</p> <p>rdac: LSI/Engenio RDAC ストレージコントローラーのパスの状態をチェックします。</p> <p>directio: 直接 I/O を使用する最初のセクターを読み取ります。</p>
features	<p>マルチパスデバイスの追加デフォルト機能です。形式は "<i>number_of_features_plus_arguments feature1 ...</i>" となります。</p> <p>features に使用できる値:</p> <p>queue_if_no_path: no_path_retry を queue に設定すると同じです。この機能の使用時に発生する可能性がある問題については「queue_if_no_path 機能に関する問題」を参照してください。</p> <p>retain_attached_hw_handler: SCSI 層によりハードウェアハンドラーがパスデバイスに接続されている状態で、このパラメーターを yes に設定した場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p> <p>pg_init_retries <i>n</i>: 失敗するまでパスグループの初期化を最高 <i>n</i> 回再試行します。ここで、$1 \leq n \leq 50$ です。</p> <p>pg_init_delay_msecs <i>n</i>: パスグループの初期化の再試行を、<i>n</i> ミリ秒間隔で待機します。ここで、$0 \leq n \leq 60000$ です。</p>

属性	説明
prio	<p>パスの優先値を得るために呼び出すデフォルトの関数を指定します。例えば、SPC-3 の ALUA ビットは悪用可能な prio 値を提供します。使用できる値を以下に示します。</p> <p>const: すべてのパスに優先度 1 を設定します。</p> <p>emc: EMC アレイのパス優先度を生成します。</p> <p>alua: SCSI-3 ALUA 設定に基づいてパス優先度を生成します。Red Hat Enterprise Linux 7.3 以降では、デバイス設定で prio "alua exclusive_pref_bit" を指定すると、マルチパスは pref ビットセットのパスのみを含むパスグループを作成し、このパスグループに最高の優先度を与えます。</p> <p>ontap: NetApp アレイのパスの優先度を生成します。</p> <p>rdac: LSI/Engenio RDAC コントローラーのパスの優先度を生成します。</p> <p>hp_sw: active/standby モードにおける Compaq/HP コントローラー用パスの優先度を生成します。</p> <p>hds: Hitachi HDS Modular ストレージアレイのパスの優先度を生成します。</p>
failback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値を設定すると、active パスを含んでいる最も優先度の高いパスグループに直ちにフェイルバックします。</p> <p>manual の値を指定すると、フェイルバックはすぐに行われず、オペレーター介入がある場合のみにフェイルバックします。</p> <p>followover の値を指定すると、パスグループの最初のパスが active になったときに自動フェイルバックを実行します。これにより、別のノードがフェイルオーバーを要求しているときは、ノードが自動的にフェイルバックしなくなります。</p> <p>「0」以上の数値で、フェイルバックの秒単位の遅延を指定します。</p>
rr_weight	<p>priorities に設定すると、次のパスを選択する path_selector を呼び出すまでに送信する要求数の決定に rr_min_io を使用しません。送信する要求数は、rr_min_io に、prio 関数で決定されるパスの優先度を掛けて決定します。uniform に設定すると、全てのパスの重みが等しくなります。</p>

属性	説明
no_path_retry	<p>障害の発生したパスの使用を何回試行してからキュー待ちを無効にするか、その回数を指定します。</p> <p>fail を指定すると、キュー待ちはせず直ちに失敗します。</p> <p>queue を指定すると、パスが修復されるまでキュー待ちは停止しません。</p>
rr_min_io	<p>現在のパスグループで、次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。この設定は、2.6.31 より前のカーネルを実行しているシステムにのみ適用されます。2.6.31 以降のシステムには、rr_min_io_rq を使用してください。デフォルト値は 1000 です。</p>
rr_min_io_rq	<p>現在のパスグループで、次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。要求ベースの device-mapper-multipath を使用しています。現在のカーネルを実行しているシステムで使用してください。2.6.31 より前のバージョンのカーネルを実行しているシステムの場合は rr_min_io を使用してください。デフォルト値は 1 です。</p>
fast_io_fail_tmo	<p>FC リモートポートで問題を検出してから、そのリモートポートでデバイスへの I/O が失敗するまでの scsi 層の待機時間 (秒) です。この値は dev_loss_tmo よりも小さくすることをお勧めします。これを off に設定すると、タイムアウトが無効になります。</p>
dev_loss_tmo	<p>FC リモートポート上で問題を検出してから、システムから削除されるまでの scsi 層の待機時間 (秒) です。この値を無制限に設定するには 2147483647 秒または 68 年に設定します。</p>
flush_on_last_del	<p>yes に設定すると、デバイスへの最後のパスが削除された時点で、multipathd デーモンがキュー待ちを無効にします。</p>
user_friendly_names	<p>yes に設定すると、/etc/multipath/bindings ファイルを使った永続的で固有となるエイリアスが、mpathn の形式でマルチパスに割り当てられます。no に設定すると、WWID がマルチパスのエイリアスに使用されます。ただし、設定ファイルの multipaths セクションで指定されるデバイス固有のエイリアスの方が、ここでの設定よりも優先されることとなります。デフォルト値は no です。</p>
retain_attached_hw_handler	<p>SCSI 層で、ハードウェアハンドラーがパスデバイスに接続されている状態で、このパラメーターを yes に設定した場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層でハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p>

属性	説明
detect_prio	yes に設定すると、デバイスで ALUA に対応しているかを最初にチェックします。対応している場合は、 alua prioritizer をデバイスに割り当てます。デバイスが ALUA に対応していない場合は、通常通り prioritizer の特定を行います。
uid_attribute	固有のパス識別子を与えます。
delay_watch_checks	(Red Hat Enterprise Linux Release 7.2 以降) 0 よりも大きい値に設定した場合、 multipathd デーモンは、指定しているチェック回数の間、有効なパスを監視します。監視されているパスが再び失敗すると、パスが次に有効になったときに、 delay_wait_checks で指定した連続的なチェック回数の間保持されるまで使用されません。これにより、信用できないパスがオンラインに戻ったときにすぐに使用されることを防ぐことができます。
delay_wait_checks	(Red Hat Enterprise Linux Release 7.2 以降) 0 よりも大きい値に設定すると、オンラインに戻ったデバイスが、 delay_watch_checks で指定されたチェック回数内に再び失敗し、次にオンラインに戻るときに、デバイスはマークされ、遅延が発生します。デバイスは、 delay_wait_checks で指定されたチェック回数を経過するまで使用されません。
deferred_remove	yes に設定すると、 multipathd は最終パスデバイスが削除されるときに通常の削除ではなく遅延された削除を行います。これにより、通常の削除が行われ、削除に失敗したときに multipathd デバイスが使用中である場合、デバイスは最終ユーザーがデバイスを終了したときに自動的に削除されます。
skip_kpartx	(Red Hat Enterprise Linux Release 7.3 以降) yes に設定すると、 kpartx は自動的にデバイスにパーティションを作成しません。これにより、デバイスにパーティションテーブルがある場合でも、パーティションを作成せずにマルチパスデバイスを作成することができます。
max_sectors_kb	(Red Hat Enterprise Linux Release 7.4 以降) マルチパスデバイスの初回アクティベートの前に、 max_sectors_kb デバイスキューパラメーターを、マルチパスデバイスの基本的なすべてのパスで指定されている値に設定します。マルチパスデバイスの作成時には、デバイスは、パスデバイスから max_sectors_kb 値を継承します。手動で、この値をマルチパスデバイス向けに高くしたり、パスデバイス向けに低くすると、マルチパスデバイスは、パスデバイスが許可するよりも大きな I/O 操作を作成する場合があります。 max_sectors_kb パラメーターを使用すると、パスデバイス上にマルチパスデバイスを作成する前に容易にこれらの値が設定でき、無効なサイズの I/O 操作が渡されることを回避できます。ユーザーがこのパラメーターを設定しないと、パスデバイスはデバイスドライバーにこれを設定させ、マルチパスデバイスがパスデバイスからこれを継承します。

属性	説明
detect_path_checker	(Red Hat Enterprise Linux Release 7.4 以降) yes に設定すると、マルチパスにより、デバイスが ALUA に対応しているかどうかを確認されます。対応している場合は、 tur パスチェッカーが自動的に使用されます。対応していない場合は、通常通りに path_checker が選択されます。

マルチパス設定ファイルの **device** エントリーの例を以下に示します。

```
# }
# device {
#   vendor    "COMPAQ  "
#   product   "MSA1000      "
#   path_grouping_policy multibus
#   path_checker tur
#   rr_weight priorities
# }
#}
```

第5章 DM MULTIPATH の管理とトラブルシューティング

この章では、稼働中のシステムで DM-Multipath を管理する方法について説明します。

5.1. MULTIPATH HELPER で自動設定ファイルの生成

Red Hat Enterprise Linux で、Multipath Helper アプリケーションを使用して、マルチパスデバイス用の基本的な設定を生成できます。このアプリケーションでは、カスタムエイリアスとデバイスブラックリストのマルチパス設定や、個々のマルチパスデバイスの特性向け設定を作成できます。作業が完了したら、アプリケーションで、選択した設定パラメーターを含むインストールスクリプトを生成し、確認のための `multipath.conf` 設定ファイルが提供されます。

Multipath Helper アプリケーションは、<https://access.redhat.com/labsinfo/multipathhelper> にあります。

5.2. オンラインのマルチパスデバイスのサイズ変更

オンラインのマルチパスデバイスのサイズを変更する必要がある場合は、以下の手順に従ってください。

1. 物理デバイスのサイズを変更します。
2. 以下のコマンドを実行して、LUN へのパスを検索します。

```
# multipath -l
```

3. パスのサイズを変更します。SCSI デバイスの場合、デバイスの `rescan` ファイルに 1 と書き込むと、SCSI ドライバーによる再スキャンが行われます。以下にコマンドの例を示します。

```
# echo 1 > /sys/block/path_device/device/rescan
```

それぞれのパスデバイスに対してこのコマンドを実行します。たとえば、パスデバイスが `sda`、`sdb`、`sde`、および `sdf` の場合は、以下のコマンドを実行します。

```
# echo 1 > /sys/block/sda/device/rescan
# echo 1 > /sys/block/sdb/device/rescan
# echo 1 > /sys/block/sde/device/rescan
# echo 1 > /sys/block/sdf/device/rescan
```

4. `multipathd resize` コマンドを実行して、マルチパスデバイスのサイズを変更します。

```
# multipathd resize map multipath_device
```

5. ファイルシステムのサイズを変更します (LVM または DOS のパーティションが使用されていないことを前提とします)。

```
# resize2fs /dev/mapper/mpatha
```

5.3. ROOT ファイルシステムを、シングルパスデバイスからマルチパスデバイスへ移行

シングルパスのデバイスにシステムをインストールしてから、別のパスを root ファイルシステムに追加する場合は、root ファイルシステムをマルチパスのデバイスに移行する必要があります。本セクションでは、シングルパスからマルチパスのデバイスへの移行手順を説明します。

device-mapper-multipath パッケージがインストールされていることを確認した上で、以下の手順を実行します。

1. 以下のコマンドを実行して **/etc/multipath.conf** 設定ファイルを作成、マルチパスモジュールをロードして **multipathd** の **chkconfig** を **on** に設定します。

```
# mpathconf --enable
```

mpathconf コマンドを使ったマルチパスの設定に関する詳細は「[DM Multipath の設定](#)」を参照してください。

2. **find_multipaths** 設定パラメーターが **yes** に設定されていない場合は、「[設定ファイルの blacklist セクション](#)」の説明に従い **/etc/multipath.conf** ファイルの **blacklist** セクションと **blacklist_exceptions** セクションを変更してください。
3. 検出され次第、root デバイスにマルチパスデバイスを構築させるため、次のコマンドを実行します。また、このコマンドを実行することで、パスが1つしかない場合でも必ずマルチパスデバイスが構築されるようになります。

```
# multipath -a root_devname
```

例えば root デバイスが **/dev/sdb** の場合は次のコマンドを実行します。

```
# multipath -a /dev/sdb  
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

4. 設定ファイルの設定が正しく行われたことを確認するため、**multipath** コマンドを実行して、次のような行が出力されていることを検索します。この行は、コマンドがマルチパスデバイスの作成に失敗したことを示しています。

```
date wwid: ignoring map
```

例えばデバイスの WWID が 3600d02300069c9ce09d41c4ac9c53200 なら次のような行が出力に表示されます。

```
# multipath  
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

5. **dracut** コマンドに以下のオプションを付けて実行し、**multipath** 付きの **initramfs** ファイルシステムを再ビルドします。

```
# dracut --force -H --add multipath
```

6. マシンをシャットダウンします。
7. マシンから他のパスが見えるように、FC スイッチを設定します。
8. マシンを起動します。

9. root ファイルシステム (/) が、マルチパス設定したデバイス上にあることを確認します。

5.4. SWAP ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行

デフォルトでは、swap デバイスは論理ボリュームとして設定されます。論理ボリュームグループを構成する物理ボリューム上でマルチパスを設定している限り、これらのデバイスをマルチパスデバイスとして設定する特別な手順は必要ありません。ただし、swap デバイスが LVM ボリュームではなく、デバイス名でマウントする場合には、**fstab** ファイルに、適切なマルチパスデバイス名を設定しないといけない場合があります。

1. **-v3** オプションを付けて **/sbin/multipath** コマンドを実行し、swap デバイスの WWID 番号を確認します。コマンドの出力のパス一覧に swap デバイスが表示されます。

swap デバイスを示す、次のような行のコマンド出力を探します。

```
WWID H:B:T:L devname MAJOR:MINOR
```

例えば、swap ファイルシステムを **sda** か、そのパーティションの 1 つに設定している場合は、以下のような行が表示されます。

```
==== paths list ====
...
1ATA      WDC WD800JD-75MSA3          WD-WMAM9F
1:0:0:0 sda 8:0
...
```

2. **/etc/multipath.conf** ファイルに、swap デバイスのエイリアスを設定します。

```
multipaths {
    multipath {
        wwid WWID_of_swap_device
        alias swapdev
    }
}
```

3. **/etc/fstab** で、root デバイスへの古いデバイスパスを、マルチパスデバイスに置き換えます。

例えば **/etc/fstab** ファイルに、以下のようなエントリーがあった場合、

```
/dev/sda2 swap                                swap    defaults        0 0
```

そのエントリーを以下のように変更します。

```
/dev/mapper/swapdev swap        swap    defaults        0 0
```

5.5. マルチパスデーモン

マルチパス設定の実装に問題があることが判明した場合は、「[3章 DM Multipath の設定](#)」に説明しているように、マルチパスデーモンが実行しているか確認してください。マルチパスが設定されているデバイスを使用するには、**multipathd** デーモンを実行しておく必要があります。

5.6. 多数の LUN に伴う問題

1 ノードに LUN を多数追加した場合は、マルチパスを設定したデバイスを使用すると、**udev** デバイスマネージャーによるデバイスノードの作成時間が大幅に増大する場合があります。この問題が発生した場合は、`/etc/udev/rules.d/40-multipath.rules` で以下の行を削除すると修正できます。

```
KERNEL!="dm-[0-9]*", ACTION=="add", PROGRAM==" /bin/bash -c '/sbin/lsmode | /bin/grep ^dm_multipath'", RUN+="/sbin/multipath -v0 %M:%m"
```

ブロックデバイスがノードに追加される度に、**udev** デバイスマネージャーが **multipath** を実行する原因となるのがこの行です。この行を削除しても、**multipathd** デーモンにより、マルチパスを設定したデバイスが自動的に作成され、マルチパスを設定した root ファイルシステムで、ノードの起動中に **multipath** が呼び出されます。唯一異なるのは、**multipathd** デーモンが実行されていない時は、マルチパスを設定したデバイスが自動的に作成されないという点です。これで、多くの場合は問題とはなくなるはずですが。

5.7. QUEUE_IF_NO_PATH 機能に関する問題

features "1 queue_if_no_path" でマルチパスデバイスを設定した場合、1 つ以上のパスが復帰するまで、I/O を発行するプロセスがすべてハングします。これを回避するために、`/etc/multipath.conf` ファイルに **no_path_retry N** パラメーターを設定します (**N** はシステムでパスを再試行する回数)。

features "1 queue_if_no_path" オプションを使用しなければならない状況で上記の問題が発生した場合は、**dmsetup** コマンドを使って、特定の LUN (どのパスも利用できないもの) 用のランタイム時ポリシーを編集します。例えば、マルチパスデバイス **mpathc** のポリシーを **"queue_if_no_path"** から **"fail_if_no_path"** に変更する場合は以下のコマンドを実行します。

```
dmsetup message mpathc 0 "fail_if_no_path"
```

指定するのは、パスではなく **mpathn** のエイリアスであることに注意してください。

5.8. MULTIPATH コマンドの出力

マルチパスデバイスの作成、修正、一覧表示を行う際、現在のデバイス設定が表示されます。形式を以下に示します。

マルチパスデバイス

```
action_if_any: alias (wwid_if_different_from_alias)
dm_device_name_if_known vendor,product size=size features='features'
hwhandler='hardware_handler' wp=write_permission_if_known
```

パスグループ

```
-+- policy='scheduling_policy' prio=prio_if_known
status=path_group_status_if_known
```

パス

```
`- host:channel:id:lun devnode major:minor dm_status_if_known path_status
online_status
```

マルチパスコマンドの出力例を以下に示します。

```
3600d023000000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `-- 6:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   `-- 7:0:0:0 sdf 8:80 active ready running
```

パスの準備が整うと、パスの状態が **ready** または **ghost** になります。パスに障害が発生すると、状態は **faulty** または **shaky** になります。パスの状態は、`/etc/multipath.conf` ファイルに定義されているポーリング間隔に応じ、`multipathd` デーモンによって定期的に更新されます。

dm の状態は、パスの状態と似ていますが、カーネルの観点からは異なります。dm の状態は、**failed** (**faulty** に類似) と、それ以外の状態を示す **active** の 2 種類があります。任意のデバイスのパス状態と dm 状態が、一時的に一致しない場合があります。

online_status に使用できる値は **running** と **offline** です。**offline** の状態とは、この SCSI デバイスが無効になっているということです。



注記

マルチパスデバイスを作成中または修正中の場合、パスグループの状態、dm のデバイス名、書き込みパーミッション、dm の状態などは不明です。また、機能も常に正しいとは限りません。

5.9. MULTIPATH コマンドを使ったマルチパスクエリー

`multipath` コマンドで **-l** と **-ll** のオプションを使用すると、現在のマルチパス設定を表示します。**-l** オプションは、**sysfs** の情報から収集したマルチパストポロジーと、デバイスマッパーを表示します。**-ll** オプションは、**-l** オプションが表示する情報の他に、使用できる他のすべてのシステムコンポーネントを表示します。

マルチパス設定を表示する際に、`multipath` コマンドの **-v** オプションで指定できる詳細レベルが 3 つあります。**-v0** は出力がありません。**-v1** は、作成または更新されたマルチパス名だけを出力します。マルチパス名は、**kpartx** など他のツールに与えることができます。**-v2** は検知されたすべてのパス、マルチパス、デバイスマップを表示します。

`multipath -l` コマンドの出力を以下に示します。

```
# multipath -l
3600d023000000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `-- 6:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   `-- 7:0:0:0 sdf 8:80 active ready running
```

`multipath -ll` コマンドの出力を以下に示します。

```
# multipath -ll
3600d023000000000000e13955cc3757801 dm-10 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
```

```
|-+- policy='round-robin 0' prio=1 status=enabled
|  `- 19:0:0:1 sdc 8:32  active ready  running
`-+- policy='round-robin 0' prio=1 status=enabled
   `- 18:0:0:1 sdh 8:112 active ready  running
3600d023000000000000e13955cc3757803 dm-2 WINSYS,SF2372
size=125G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   |- 19:0:0:3 sde 8:64  active ready  running
   `- 18:0:0:3 sdj 8:144 active ready  running
```

5.10. MULTIPATH コマンドのオプション

役立つと思われる `multipath` コマンドのオプションを [表5.1「便利な multipath コマンドのオプション」](#) に示します。

表5.1 便利な `multipath` コマンドのオプション

オプション	説明
<code>-l</code>	<code>sysfs</code> から収集した現在のマルチパス設定とデバイスマッパーを表示します。
<code>-ll</code>	<code>sysfs</code> から収集した現在のマルチパス設定、デバイスマッパー、使用可能な全システムコンポーネントを表示します。
<code>-f device</code>	指定されたマルチパスデバイスを削除します。
<code>-F</code>	不要なマルチパスデバイスをすべて削除します。
<code>-w device</code>	指定したデバイスの <code>wwid</code> を、 <code>wwids</code> ファイルから削除します。
<code>-W</code>	現在のマルチパスデバイスのみを組み込むように、 <code>wwids</code> ファイルをリセットします。

5.11. DMSETUP コマンドを使ってデバイスマッパーのエントリーを特定

`dmsetup` コマンドを使って、マルチパスを設定したデバイスに一致するデバイスマッパーのエントリーを検索できます。

以下のコマンドでは、すべてのデバイスマッパーデバイスとそのメジャー番号、マイナー番号を表示します。dm デバイス名は、マイナー番号で特定できます。例えば、マイナー番号 3 は、マルチパス設定したデバイス `/dev/dm-3` になります。

```
# dmsetup ls
mpathd (253:4)
mpathep1 (253:12)
mpathfp1 (253:11)
mpathb (253:3)
mpathgp1 (253:14)
mpathhp1 (253:13)
mpatha (253:2)
```

```

mpathh (253:9)
mpathg (253:8)
VolGroup00-LogVol01 (253:1)
mpathf (253:7)
VolGroup00-LogVol00 (253:0)
mpathe (253:6)
mpathbp1 (253:10)
mpathd (253:5)

```

5.12. MULTIPATHD コマンド

multipathd コマンドを使用して、**multipathd** デーモンを管理できます。利用可能な **multipathd** コマンドについては、**multipathd(8)** man ページを参照してください。

一部の **multipathd** コマンドには、後にワイルドカードが付いた **format** オプションを含むものがあります。次のコマンドを実行すると、使用できるワイルドカード一覧を表示できます。

```
# multipathd show wildcards
```

Red Hat Enterprise Linux リリース 7.3 の時点では、**multipathd** コマンドは、"raw" 形式のバージョンでマルチパスデバイスとパスのステータスを表示する、新しい形式のコマンドをサポートしています。raw 形式では、ヘッダーはプリントされず、フィールドはヘッダーと整合するようにコラムにパディングがされていません。代わりに、書式設定文字列に指定された通りにフィールドをプリントします。このため、この出力はスクリプトに使いやすくなっています。**multipathd show wildcards** コマンドを使用すると、書式設定文字列に使用されたワイルドカードを表示することができます。

以下の **multipathd** コマンドは、**multipathd** が監視しているマルチパスデバイスを、通常および raw 形式でマルチパスワイルドカードのある書式設定文字列を使用して表示します。

```
list|show maps|multipaths format $format
list|show maps|multipaths raw format $format
```

以下の **multipathd** コマンドは、**multipathd** が監視しているパスを、通常および raw 形式でマルチパスワイルドカードのある書式設定文字列を使用して表示します。

```
list|show paths format $format
list|show paths raw format $format
```

以下のコマンドは、**multipathd show maps** での raw 形式と非 raw 形式の違いを表示します。raw 形式ではヘッダーがなく、コラム間は単一スペースのみであることを注意してください。

```

# multipathd show maps format "%n %w %d %s"
name      uuid                               sysfs vend/prod/rev
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN

# multipathd show maps raw format "%n %w %d %s"
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN

```

5.13. 対話式コンソール MULTIPATHD を使用したトラブルシューティング

multipathd -k コマンドは、**multipathd** デーモンに対する対話式のインターフェースです。このコマンドを実行すると対話式のマルチパスコンソールが立ち上がります。このコマンドの実行後に

help を入力すると使用可能なコマンドの一覧が表示され、対話式コマンドを入力することができます。**CTRL-D** を入力すると終了します。

multipathd 対話式コンソールを使用すると、システムで発生している問題のトラブルシューティングを行うことができます。例えば、以下のようなコマンドを順番に入力すると、デフォルト値などのマルチパス設定を表示して、その後コンソールを終了します。

```
# multipathd -k
> > show config
> > CTRL-D
```

次の順序でコマンドを実行し、**multipath.conf** に対する変更がすべて反映されるようにします。

```
# multipathd -k
> > reconfigure
> > CTRL-D
```

次の順序でコマンドを実行し、パスチェッカーが正しく動作していることを確認します。

```
# multipathd -k
> > show paths
> > CTRL-D
```

5.14. パッケージ削除時におけるマルチパスファイルのクリーンアップ

device-mapper-multipath rpm ファイルを削除して

も、**/etc/multipath.conf**、**/etc/multipath/bindings**、および **/etc/multipath/wwids** の各ファイルは削除されないことに注意してください。これらのファイルは、**device-mapper-multipath** パッケージの次回インストール時に手動で削除する必要があります。

付録A 改訂履歴

改訂 4.0-6.3 翻訳ファイルを XML ソースバージョン 4.0-6 と同期	Sun Feb 11 2018	Terry Chuang
改訂 4.0-6.2 翻訳ファイルを XML ソースバージョン 4.0-6 と同期	Sun Feb 11 2018	Terry Chuang
改訂 4.0-6.1 翻訳ファイルを XML ソースバージョン 4.0-6 と同期	Sun Sep 24 2017	Terry Chuang
改訂 4.0-6 7.4 GA 公開用バージョン。	Wed Jul 19 2017	Steven Levine
改訂 4.0-3 7.4 ベータ公開用ドキュメントの準備。	Wed May 10 2017	Steven Levine
改訂 0.4-5 7.3 バージョンでの小規模なアップデート。	Wed Nov 16 2016	Steven Levine
改訂 0.4-3 7.3 GA 公開用バージョン。	Mon Oct 24 2016	Steven Levine
改訂 0.4-1 7.3 ベータ公開用ドキュメントの準備。	Mon Aug 15 2016	Steven Levine
改訂 0.3-6 7.2 GA 公開用ドキュメントの準備。	Mon Nov 9 2015	Steven Levine
改訂 0.3-3 7.2 ベータ公開用ドキュメントの準備。	Wed Aug 19 2015	Steven Levine
改訂 0.2-7 7.1 GA リリース向けバージョン。	Mon Feb 16 2015	Steven Levine
改訂 0.2-6 7.1 Beta リリース向けバージョン。	Thu Dec 11 2014	Steven Levine
改訂 0.1-22 7.0 GA リリース向けバージョン。	Mon Jun 2 2014	Steven Levine
改訂 0.1-10 7.0 Beta 更新。	Wed Apr 9 2014	Steven Levine
改訂 0.1-3 7.0 pre-Beta バージョン。	Tue Nov 26 2013	Steven Levine
改訂 0.1-1 Red Hat Enterprise Linux 6 バージョンのドキュメントからブランチ作成	Wed Jan 16 2013	Steven Levine

索引

シンボル

[/etc/multipath.conf](#) パッケージ, [DM Multipath](#) の設定

ストレージアレイ

追加する, [ストレージデバイスの設定](#), [設定ファイルの devices セクション](#)

[ストレージアレイの対応](#), [ストレージアレイの対応](#)

デバイス

追加する, [ストレージデバイスの設定](#)

[デバイス名](#), [マルチパスデバイス識別子](#)

[フェイルオーバー](#), [DM Multipath](#) の概要

ブラックリスト

[WWID](#), [WWID](#) でブラックリストの指定

[デバイス名](#), [デバイス名でブラックリストの指定](#)

[設定ファイル](#), [設定ファイルの blacklist セクション](#)

[マルチパスデバイス](#), [マルチパスデバイス](#)

[LVM 論理ボリューム](#), [論理ボリューム内のマルチパスデバイス](#)

[論理ボリューム](#), [論理ボリューム内のマルチパスデバイス](#)

[マルチパスデバイスのサイズ変更](#), [オンラインのマルチパスデバイスのサイズ変更](#)

[マルチパスデーモン \(multipathd\)](#), [マルチパスデーモン](#)

[マルチパス設定した root ファイルシステム](#), [root ファイルシステムを、シングルパスデバイスからマルチパスデバイスへ移行](#)

[マルチパス設定した swap ファイルシステム](#), [swap ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行](#)

[ローカルのディスク、無視する](#), [マルチパスデバイスの作成時にローカルのディスクを無視](#)

[新機能および変更された機能](#), [新機能および変更された機能](#)

概要

[新機能および変更された機能](#), [新機能および変更された機能](#)

設定

[DM Multipath](#), [DM Multipath](#) の設定

設定する

[DM Multipath](#), [DM Multipath](#) の設定

設定ファイル

[alias](#) パラメーター, [設定ファイルの multipaths セクション](#)

[alias_prefix](#) パラメーター, [設定ファイルの devices セクション](#)

[checker_timeout](#) パラメーター, [設定ファイルの defaults セクション](#)

[deferred_remove](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

delay_wait_checks パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

delay_watch_checks パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

detect_path_checker パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [devices](#) セクション

dev_loss_tmo パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [devices](#) セクション

disable_changed_wwids パラメーター, 設定ファイルの [defaults](#) セクション

failback パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

fast_io_fail_tmo パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [devices](#) セクション

features パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

flush_on_last_del パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

force_sync パラメーター, 設定ファイルの [defaults](#) セクション

hardware_handler パラメーター, 設定ファイルの [devices](#) セクション

hw_string_match パラメーター, 設定ファイルの [defaults](#) セクション

ignore_new_boot_devs パラメーター, 設定ファイルの [defaults](#) セクション

log_checker_err パラメーター, 設定ファイルの [defaults](#) セクション

max_fds パラメーター, 設定ファイルの [defaults](#) セクション

max_sectors_kb パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

new_bindings_in_boot パラメーター, 設定ファイルの [defaults](#) セクション

no_path_retry パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

path_checker パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [devices](#) セクション

path_grouping_policy パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

path_selector パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション, 設定ファイルの [devices](#) セクション

polling-interval パラメーター, 設定ファイルの [defaults](#) セクション

prio パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [devices](#) セクション

product パラメーター, 設定ファイルの [devices](#) セクション

product_blacklist パラメーター, 設定ファイルの [devices](#) セクション

queue_without_daemon パラメーター, 設定ファイルの [defaults](#) セクション

reassign_maps パラメーター, 設定ファイルの [defaults](#) セクション

remove_retries パラメーター, 設定ファイルの [defaults](#) セクション

retain_attached_hw_handler パラメーター, 設定ファイルの [defaults](#) セクション, 設定ファイルの [multipaths](#) セクション

retrigger_delay パラメーター, 設定ファイルの [defaults](#) セクション

retrigger_tries パラメーター, 設定ファイルの [defaults](#) セクション

revision パラメーター, [設定ファイルの devices セクション](#)

rr_min_io パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

rr_weight パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

skip_kpartx パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

uid_attribute パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

user_friendly_names パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

vendor パラメーター, [設定ファイルの devices セクション](#)

verbosity パラメーター, [設定ファイルの defaults セクション](#)

wwid パラメーター, [設定ファイルの multipaths セクション](#)

ブラックリスト, [設定ファイルの blacklist セクション](#)

概要, [設定ファイルの概要](#)

A

active/active 設定

図, [DM Multipath の概要](#)

定義, [DM Multipath の概要](#)

active/passive 設定

図, [DM Multipath の概要](#)

定義, [DM Multipath の概要](#)

alias パラメーター, [設定ファイルの multipaths セクション](#)

設定ファイル, [マルチパスデバイス識別子](#)

alias_prefix パラメーター, [設定ファイルの devices セクション](#)

B

blacklist

デバイスタイプ, [デバイスタイプでブラックリストの指定](#)

デフォルトのデバイス, [デバイス名でブラックリストの指定](#)

blacklist_exceptions セクション

multipath.conf ファイル, [ブラックリストから除外](#)

C

checker_timeout パラメーター, [設定ファイルの defaults セクション](#)

configuration file

config_dir parameter, [設定ファイルの defaults セクション](#)

[detect_prio](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

[config_dir](#) parameter, [設定ファイルの defaults セクション](#)

D

defaults セクション

[multipath.conf](#) ファイル, [設定ファイルの defaults セクション](#)

[deferred_remove](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[delay_wait_checks](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[delay_watch_checks](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[detect_path_checker](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

[detect_prio](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

[dev/mapper](#) ディレクトリー, [マルチパスデバイス識別子](#)

[device-mapper-multipath](#) パッケージ, [DM Multipath の設定](#)

devices

追加する, [設定ファイルの devices セクション](#)

devices セクション

[multipath.conf](#) ファイル, [設定ファイルの devices セクション](#)

[dev_loss_tmo](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

[disable_changed_wwids](#) パラメーター, [設定ファイルの defaults セクション](#)

DM Multipath

および LVM, [論理ボリューム内のマルチパスデバイス](#)

コンポーネント, [DM Multipath のコンポーネント](#)

デバイス, [マルチパスデバイス](#)

デバイス名, [マルチパスデバイス識別子](#)

フェイルオーバー, [DM Multipath の概要](#)

冗長性, [DM Multipath の概要](#)

定義, [Device Mapper を使ったマルチパス設定](#)

概要, [DM Multipath の概要](#)

設定, [DM Multipath の設定](#)

設定、概要, [DM Multipath 設定の概要](#)

設定する, [DM Multipath の設定](#)

設定ファイル, [DM Multipath 設定ファイル](#)

[dm-n](#) デバイス, [マルチパスデバイス識別子](#)

dmsetup コマンド、デバイスマッパーのエントリーを特定する, [dmsetup コマンドを使ってデバイスマッパーのエントリーを特定](#)

dm_multipath カーネルモジュール, [DM Multipath のコンポーネント](#)

F

failback パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

fast_io_fail_tmo パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

features パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

flush_on_last_del パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

force_sync パラメーター, [設定ファイルの defaults セクション](#)

H

hardware_handler パラメーター, [設定ファイルの devices セクション](#)

hw_string_match パラメーター, [設定ファイルの defaults セクション](#)

I

ignore_new_boot_devs パラメーター, [設定ファイルの defaults セクション](#)

initramfs

マルチパスを起動する, [initramfs ファイルシステムでマルチパスの設定](#)

K

kpartx コマンド, [DM Multipath のコンポーネント](#)

L

log_checker_err パラメーター, [設定ファイルの defaults セクション](#)

LVM 論理ボリューム

マルチパスデバイス, [論理ボリューム内のマルチパスデバイス](#)

lvm.conf ファイル, [論理ボリューム内のマルチパスデバイス](#)

M

max_fds パラメーター, [設定ファイルの defaults セクション](#)

max_sectors_kb パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

mpathconf コマンド, [DM Multipath のコンポーネント](#)

Multipath Helper, [Multipath Helper](#) で自動設定ファイルの生成

multipath コマンド, [DM Multipath のコンポーネント](#)

multipath コマンドの

オプション, [multipath コマンドのオプション](#)

クエリー, [multipath コマンドを使ったマルチパスクエリー](#)

出力, [Multipath コマンドの出力](#)

[multipath.conf](#) のファイル

[blacklist_exceptions](#) セクション, [ブラックリストから除外](#)

[multipath.conf](#) ファイル, [ストレージレイの対応](#), [DM Multipath 設定ファイル](#)

[defaults](#) セクション, [設定ファイルの defaults セクション](#)

[devices](#) セクション, [設定ファイルの devices セクション](#)

[multipaths](#) セクション, [設定ファイルの multipaths セクション](#)

[multipathd](#)

コマンド, [対話式コンソール multipathd を使用したトラブルシューティング](#)

対話式コンソール, [対話式コンソール multipathd を使用したトラブルシューティング](#)

[multipathd start](#) コマンド, [DM Multipath の設定](#)

[multipathd](#) デーモン, [DM Multipath のコンポーネント](#)

[multipaths](#) セクション

[multipath.conf](#) ファイル, [設定ファイルの multipaths セクション](#)

N

[new_bindings_in_boot](#) パラメーター, [設定ファイルの defaults セクション](#)

[no_path_retry](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

P

[path_checker](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

[path_grouping_policy](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[path_selector](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[polling_interval](#) パラメーター, [設定ファイルの defaults セクション](#)

[prio](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

[product](#) パラメーター, [設定ファイルの devices セクション](#)

[product_blacklist](#) パラメーター, [設定ファイルの devices セクション](#)

Q

[queue_without_daemon](#) パラメーター, [設定ファイルの defaults セクション](#)

R

[reassign_maps](#) パラメーター, [設定ファイルの defaults セクション](#)

[remove_retries](#) パラメーター, [設定ファイルの defaults セクション](#)

[retain_attached_hw_handler](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

[retrigger_delay](#) パラメーター, [設定ファイルの defaults セクション](#)

retrigger_tries パラメーター, 設定ファイルの **defaults** セクション

revision パラメーター, 設定ファイルの **devices** セクション

root ファイルシステム, **root** ファイルシステムを、シングルパスデバイスからマルチパスデバイスへ移行

rr_min_io パラメーター, 設定ファイルの **defaults** セクション, 設定ファイルの **multipaths** セクション

rr_weight パラメーター, 設定ファイルの **defaults** セクション, 設定ファイルの **multipaths** セクション, 設定ファイルの **devices** セクション

S

skip_kpartxr パラメーター, 設定ファイルの **defaults** セクション, 設定ファイルの **multipaths** セクション, 設定ファイルの **devices** セクション

swap ファイルシステム, **swap** ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行

U

uid_attribute パラメーター, 設定ファイルの **defaults** セクション, 設定ファイルの **devices** セクション

user_friendly_names パラメーター, マルチパスデバイス識別子, 設定ファイルの **defaults** セクション, 設定ファイルの **multipaths** セクション, 設定ファイルの **devices** セクション

V

vendor パラメーター, 設定ファイルの **devices** セクション

verbosity パラメーター, 設定ファイルの **defaults** セクション

W

World Wide Identifier (WWID), マルチパスデバイス識別子

wwid パラメーター, 設定ファイルの **multipaths** セクション