



Red Hat Enterprise Linux 6

DM マルチパス機能

DM マルチパス機能の設定と管理

エディション 1

Red Hat Enterprise Linux 6 DM マルチパス機能

DM マルチパス機能の設定と管理
エディション 1

法律上の通知

Copyright © 2014 Red Hat, Inc. and others.

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 6 の Device-Mapper マルチパス機能の使用に関する情報を提供します。

目次

前書き	3
1. 対象読者	3
2. 関連ドキュメント	3
3. フィードバック	3
第1章 デバイスマッパーのマルチパス化	5
1.1. 新機能および変更された機能	5
1.1.1. Red Hat Enterprise Linux 6.0 の新機能および変更された機能	5
1.1.2. Red Hat Enterprise Linux 6.1 の新機能および変更された機能	6
1.1.3. Red Hat Enterprise Linux 6.2 の新機能および変更された機能	6
1.1.4. Red Hat Enterprise Linux 6.3 の新機能および変更された機能	6
1.1.5. Red Hat Enterprise Linux 6.4 の新機能および変更された機能	7
1.1.6. Red Hat Enterprise Linux 6.5 の新機能および変更された機能	7
1.1.7. Red Hat Enterprise Linux 6.6 の新機能および変更された機能	7
1.2. DM-MULTIPATH の概要	7
1.3. ストレージアレイサポート	10
1.4. DM-MULTIPATH コンポーネント	10
1.5. DM-MULTIPATH セットアップの概要	11
第2章 マルチパスデバイス	12
2.1. マルチパスデバイス識別子	12
2.2. クラスタ内で一貫したマルチパスデバイス名	12
2.3. マルチパスデバイスの属性	13
2.4. 論理ボリューム内のマルチパスデバイス	13
第3章 DM-MULTIPATH のセットアップ	15
3.1. DM-MULTIPATH のセットアップ	15
3.2. マルチパスデバイスの生成時にローカルディスクを無視する	16
3.3. ストレージデバイスの設定	18
3.4. INITRAMFS ファイルシステムでのマルチパス機能のセットアップ	19
第4章 DM-MULTIPATH 設定ファイル	20
4.1. 設定ファイルの概要	20
4.2. 設定ファイルのブラックリスト	21
4.2.1. WWID 単位のブラックリスト	22
4.2.2. デバイス名単位のブラックリスト	22
4.2.3. デバイスタイプ単位のブラックリスト	23
4.2.4. ブラックリスト例外	23
4.3. 設定ファイルのデフォルト	23
4.4. マルチパスデバイス設定属性	32
4.5. 設定ファイルデバイス	35
第5章 DM-MULTIPATH の管理とトラブルシューティング	42
5.1. オンラインマルチパスデバイスのサイズ変更	42
5.2. シングルパスデバイスからマルチパスデバイスへの ROOT ファイルシステムの移動	42
5.3. シングルパスデバイスからマルチパスデバイスへのスワップファイルシステムの移動	45
5.4. マルチパスデーモン	46
5.5. 多数の LUN に伴う問題	46
5.6. QUEUE_IF_NO_PATH 機能での問題	46
5.7. MULTIPATH コマンドの出力	46
5.8. MULTIPATH コマンドを使用したマルチパスクエリー	47
5.9. MULTIPATH コマンドのオプション	48
5.10. DMSETUP コマンドによるデバイスマッパーエントリーの判別	48

5.11. MULTIPATHD インタラクティブコンソールでのトラブルシューティング	49
付録A 改訂履歴	50
索引	53

前書き

本ガイドでは、Red Hat Enterprise Linux 6 リリースを対象とした Red Hat Enterprise Linux のデバイス マッパーマルチパス (DM-Multipath) 機能について解説します。

1. 対象読者

本ガイドは、Linux オペレーティングシステムを実行するシステムを管理するシステム管理者を対象としています。読者は Red Hat Enterprise Linux に精通している必要があります。

2. 関連ドキュメント

Red Hat Enterprise Linux の使用法に関する詳細は、以下の資料を参照してください。

- 『インストールガイド』 — Red Hat Enterprise Linux 6 のインストールに関する情報を記載しています。
- 『導入ガイド』 — Red Hat Enterprise Linux 6 の導入、設定、および管理に関する情報を記載しています。
- 『ストレージ管理ガイド』 — Red Hat Enterprise Linux 6 上のストレージデバイスおよびファイルシステムの効果的な管理方法について説明しています。

Red Hat Enterprise Linux 6 対応の Red Hat Cluster Suite に関する詳細は、以下の資料を参照してください。

- 『High Availability アドオン概要』 — Red Hat High Availability アドオンの全体的な概要を説明しています。
- 『クラスタの管理』 — High Availability アドオンのインストール、設定、および管理に関する情報を提供しています。
- 『論理ボリュームマネージャーの管理』 — LVM 論理ボリュームマネージャーについて説明しており、クラスタ環境における LVM の実行に関する情報も記載しています。
- 『Global File System 2: 設定と管理』 — Red Hat GFS2 (Red Hat Global File System 2) のインストール、設定、および保守に関する情報を提供しています。
- 『ロードバランサーの管理』 — 実サーバー群全体にわたって IP 負荷を分散するための Linux Virtual Servers (LVS) を提供する統合ソフトウェアコンポーネントセットのロードバランサーアドオンを利用したハイパフォーマンスのシステムとサービスの設定に関する情報を提供しています。
- 『リリースノート』 — Red Hat 製品の最新リリースについての情報を提供しています。

Red Hat Cluster Suite ドキュメントと他の Red Hat 文書は、HTML と PDF 形式により、オンラインで <https://access.redhat.com/site/documentation/> 入手できます。また Red Hat Enterprise Linux ドキュメント CD の RPM 形式でも取得可能です。

3. フィードバック

誤字/脱字を発見された場合、または本ガイドを改善するためのご意見/ご提案がございましたら、弊社にご連絡ください。その場合は、製品 **Red Hat Enterprise Linux 6** およびコンポーネント **doc-DM_Multipath** に対して、Bugzilla (<http://bugzilla.redhat.com/>) 内でご報告ください。バグ報告を提出される時には、以下のマニュアルの識別子を必ずご記入いただくようお願いします。

rh-DM_Multipath(EN)-6 (2014-10-8T15:15)

本ガイドを改善するためのご意見/提案をお持ちの場合は、可能な限り具体的にご説明いただくようお願いします。また、エラーを発見された場合は、弊社で対象箇所を容易に見つけることができるように、そのセクション番号と周辺の文章も含めてご報告いただくようお願いします。

第1章 デバイスマッパーのマルチパス化

デバイスマッパーのマルチパス化 (DM-Multipath) により、サーバーノードとストレージレイ間の複数の I/O パスを単一のデバイスに設定することができます。これらの I/O パスは、別々のケーブル、スイッチ、およびコントローラーを含むことができる物理 SAN 接続です。マルチパス化は I/O パスを集約し、その集約されたパスで構成される新しいデバイスを作成します。

この章では、Red Hat Enterprise Linux 6 の初期リリースに搭載された DM-Multipath の新機能について要約します。その後、DM-Multipath およびそのコンポーネントについての大まかな概要、さらに DM-Multipath のセットアップの概要について説明します。

1.1. 新機能および変更された機能

このセクションでは、Red Hat Enterprise Linux 6 の初回リリース以降に搭載された DM-Multipath の新機能および変更された機能について記載します。

1.1.1. Red Hat Enterprise Linux 6.0 の新機能および変更された機能

Red Hat Enterprise Linux 6.0 で更新/変更されたドキュメントと機能は以下のとおりです。

- Red Hat Enterprise Linux 6 リリースでは、基本的なフェイルオーバー設定用の DM-Multipath 初期セットアップ手順が変更され、DM-Multipath 設定ファイルの作成や `multipathd` 設定ユーティリティーを使用した DM-Multipath の有効化が可能になりました。このユーティリティーは、`device-mapper-multipath` モジュールのロード、`multipathd` デーモンの起動、および `chkconfig` の設定による再起動時のデーモンの自動的な起動を実行することができます。

新たなセットアップ手順についての情報は、「[DM-Multipath のセットアップ](#)」を参照してください。`multipathd` に関する詳細は、`multipathd(5)` の man ページを参照してください。

- Red Hat Enterprise Linux 6 リリースでは、マルチパスデバイスをセットアップする新しいモードを提供します。マルチパスデバイスは、`find_multipaths` 設定ファイルパラメーターを使用して設定されます。以前の Red Hat Enterprise Linux のバージョンでは、マルチパスは、明示的にブラックリスト化されていないすべてのパスに対してマルチパスデバイスの作成を常に試行していました。しかし、Red Hat Enterprise Linux 6 では、`find_multipaths` 設定パラメーターが `yes` に設定されている場合、マルチパスは以下にあげる 3 つの条件のいずれかを満たす場合のみにデバイスを作成します。
 - 同一の WWID を持つ、ブラックリスト化されていないパスが少なくとも 2 つある。
 - `multipath` コマンドでデバイスを指定することにより、ユーザーがデバイスの手動作成を強制的に実行する。
 - パスに以前に作成されたマルチパスデバイスと同一の WWID がある (そのマルチパスデバイスが同時に存在しない場合も同様)。`find_multipaths` パラメーターが設定されていない状態でマルチパスデバイスを作成した場合に従う手順についての説明は、「[設定ファイルのブラックリスト](#)」を参照してください。

この機能によって、大半のユーザーは、ブラックリストを編集することなく、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るようになります。

`find_multipaths` 設定パラメーターについての情報は、「[設定ファイルのデフォルト](#)」を参照してください。

- Red Hat Enterprise Linux 6 リリースは、次の I/O 操作に使用するパスを決定する `queue-length` と `service-time` の 2 つのパスセクターアルゴリズムを提供します。`queue-length` アルゴリズムは、パスの未処理の I/O の量を確認し、次に使用するパスを決定します。`service-time` ア

ルゴリズムは、パスの未処理の I/O の量と相対スループットを確認し、次に使用するパスを決定します。設定ファイル内のパスセクターパラメーターについての詳細は、[4章DM-Multipath 設定ファイル](#)を参照してください。

- Red Hat Enterprise Linux 6 リリースでは、優先度関数はコールアウトプログラムではなくなり、パスチェッカー関数のような動的共有オブジェクトになりました。`prio_callout` パラメーターは、`prio` パラメーターに置き換えられました。サポートされている `prio` 関数の詳細は、[4章DM-Multipath 設定ファイル](#)を参照してください。
- Red Hat Enterprise Linux 6 では、`multipath` コマンドの出力形式が変更されました。`multipath` コマンド出力についての詳細は、「[multipath コマンドの出力](#)」を参照してください。
- Red Hat Enterprise Linux 6 リリースでは、マルチパス `bindings` ファイルは、`/etc/multipath/bindings` にあります。
- Red Hat Enterprise Linux 6 リリースは、`multipath.conf` ファイル内に、`checker_timeout`、`fast_io_fail_tmo`、および `dev_loss_tmo` の 3 つの新たな `defaults` パラメーターを提供します。これらのパラメーターについての情報は、[4章DM-Multipath 設定ファイル](#)をご覧ください。
- マルチパス設定ファイル内の `user_friendly_names` オプションが `yes` に設定トされている場合、マルチパスデバイス名は、`mpathn` の形式になります。Red Hat Enterprise Linux 6 リリースでは、`n` はアルファベット 1 文字なので、マルチパスデバイス名は `mpatha` または `mpathb` となる可能性があります。以前のリリースでは、`n` は整数でした。

1.1.2. Red Hat Enterprise Linux 6.1 の新機能および変更された機能

Red Hat Enterprise Linux 6.1 で更新/変更されたドキュメントと機能は以下のとおりです。

- 本書に新たな章「[シングルパスデバイスからマルチパスデバイスへの root ファイルシステムの移動](#)」が追加されました。
- 本書に新たな章「[シングルパスデバイスからマルチパスデバイスへのスワップファイルシステムの移動](#)」が追加されました。

1.1.3. Red Hat Enterprise Linux 6.2 の新機能および変更された機能

Red Hat Enterprise Linux 6.2 で更新/変更されたドキュメントと機能は以下のとおりです。

- Red Hat Enterprise Linux 6.2 リリースでは、`multipath.conf` ファイルの `defaults`、`devices`、`multipaths` セクションに新しい `multipath.conf` パラメーター `rr_min_io_rq` を提供しています。`rr_min_io` パラメーターは Red Hat Enterprise Linux 6.2 では無効になりました。`rr_min_io_rq` 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。
- `dev_loss_tmo` 設定ファイルパラメーターは無限に設定できるようになりました。これは実際の `sysfs` 変数を 2147483647 秒または 68 年に設定します。このパラメーターについての詳細は、[4章DM-Multipath 設定ファイル](#)を参照してください。
- 「[シングルパスデバイスからマルチパスデバイスへの root ファイルシステムの移動](#)」に記載されている手順が更新されました。

1.1.4. Red Hat Enterprise Linux 6.3 の新機能および変更された機能

Red Hat Enterprise Linux 6.3 で更新/変更されたドキュメントと機能は以下のとおりです。。

- `queue_without_daemon` 設定ファイルパラメーターのデフォルト値は `no` に設定されるようになりました。
- `max_fds` 設定ファイルパラメーターのデフォルト値は `max` に設定されるようになりました。
- `user_friendly_names` 設定ファイルパラメーターは、`multipath.conf` 設定ファイルの `defaults`、`multipaths`、`devices` セクションで設定できるようになりました。
- `multipath.conf` 設定ファイルの `defaults` セクションは、新しい `hwtable_regex_match` パラメーターに対応します。

設定ファイル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。

1.1.5. Red Hat Enterprise Linux 6.4 の新機能および変更された機能

Red Hat Enterprise Linux 6.4 で更新/変更されたドキュメントと機能は以下のとおりです。

- `multipath.conf` 設定ファイルの `defaults` セクションと `devices` セクションは、新たに `retain_attached_hardware_handler` パラメーターと `detect_prio` パラメーターに対応しています。設定ファイルの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。
- 本書に「[initramfs ファイルシステムでのマルチパス機能のセットアップ](#)」のセクションが新たに加われました。

1.1.6. Red Hat Enterprise Linux 6.5 の新機能および変更された機能

Red Hat Enterprise Linux 6.5 で更新/変更されたドキュメントと機能は以下のとおりです。

- `multipath.conf` 設定ファイルの `defaults` セクションは、新たに `replace_wwide_whitespace` パラメーターと `reload_rewrite` パラメーターに対応しています。`multipath.conf` ファイルの `defaults` セクションについては、[表4.1「マルチパス設定デフォルト」](#) を参照してください。

1.1.7. Red Hat Enterprise Linux 6.6 の新機能および変更された機能

Red Hat Enterprise Linux 6.6 で更新/変更されたドキュメントと機能は以下のとおりです。

- `multipath.conf` 設定ファイルの `defaults` セクションは、新たに `force_sync` パラメーターに対応しています。`multipath.conf` ファイルの `defaults` セクションについては、[表4.1「マルチパス設定デフォルト」](#) を参照してください。
- `multipath` は、[表4.1「マルチパス設定デフォルト」](#) で説明されているように `-w` と `-W` に対応しています。

1.2. DM-MULTIPATH の概要

DM-Multipath を使用すると以下を実現できます。

- 冗長性

DM-Multipath はアクティブ/パッシブ設定でフェイルオーバーを提供することができます。アクティブ/パッシブ設定内では、I/O にはパスの半分だけが常に使用されます。I/O パスのいずれかのエレメント (ケーブル、スイッチ、またはコントローラー) に障害が発生すると、DM-Multipath は代替パスに切り替わります。

- パフォーマンスの向上

DM-Multipath は、アクティブ/アクティブモードに設定可能です。このモードでは、I/O はラウンドロビン式にパス群に分散されます。一部の設定では、DM-Multipath は I/O パス上のローディングを検出することができるため、ロードのバランスを動的に再調整します。

図1.1 「RAID デバイスが1つのアクティブ/パッシブマルチパス設定」は、サーバーから1つの RAID デバイスへの I/O パスが2つあるアクティブ/パッシブ設定を示しています。この設定では、サーバーに2つの HBA があり、2つの SAN スイッチ、2つの RAID コントローラーがあります。

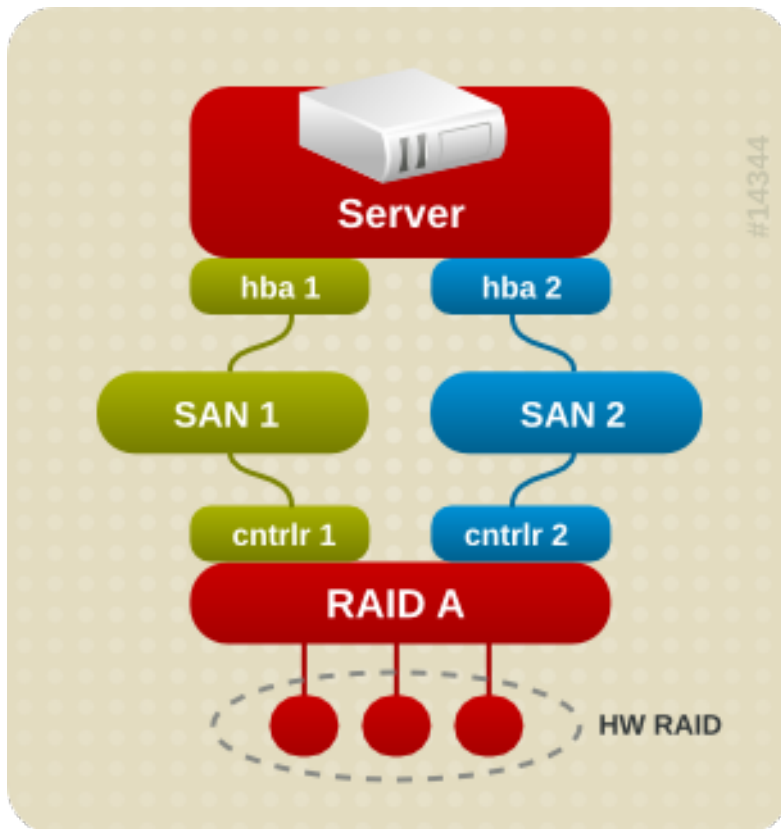


図1.1 RAID デバイスが1つのアクティブ/パッシブマルチパス設定

この設定では、hba1、SAN1、およびコントローラー1を通る1つ目のI/Oパスと、hba2、SAN2、およびコントローラー2を通る2つ目のI/O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、これらのいずれかのポイントで障害が発生した際に、DM-Multipath が代替 I/O パスに切り替わります。

図1.2 「RAID デバイスが2つのアクティブ/パッシブマルチパス設定」は、より複雑なアクティブ/パッシブ設定を示しています。サーバーに2つの HBA があり、2つの SAN スイッチ、RAID コントローラーが2つずつ付いた2つの RAID デバイスがあります。

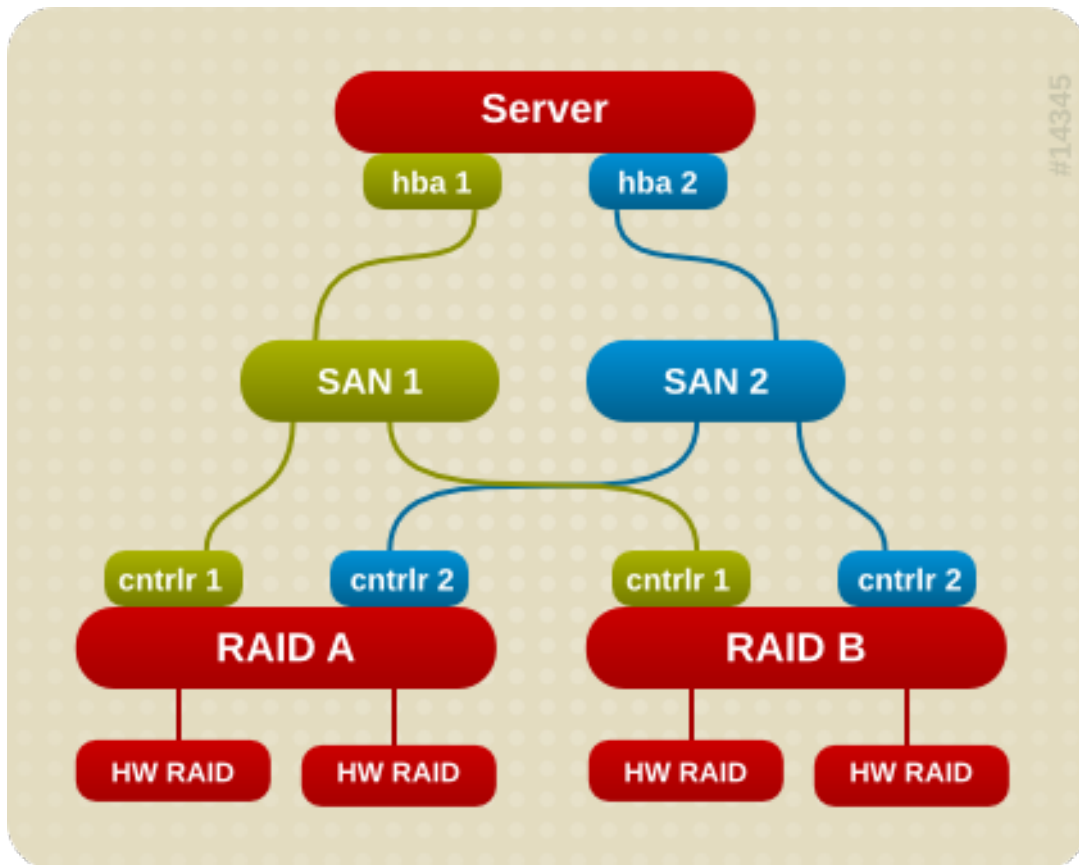


図1.2 RAID デバイスが2つのアクティブ/パッシブマルチパス設定

図1.2「RAID デバイスが2つのアクティブ/パッシブマルチパス設定」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「RAID デバイスが1つのアクティブ/パッシブマルチパス設定」に示す例と同様)。DM-Multipath が設定されていると、どちらかのRAID デバイスへのI/Oパスのいずれかのポイントで障害が発生した際に、DM-Multipath はそのデバイス用の代替I/Oパスに切り替わります。

図1.3「RAID デバイスが1つのアクティブ/アクティブマルチパス設定」は、サーバーに2つのHBAがあり、1つのSANスイッチ、2つのRAIDコントローラーがあるアクティブ/アクティブ設定を示しています。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 RAID デバイスが1つのアクティブ/アクティブマルチパス設定

1.3. ストレージアレイサポート

デフォルトでは、DM-Multipath は DM-Multipath に対応する最も一般的なストレージアレイのサポートを含んでいます。これらのサポートされたデバイスは `multipath.conf.defaults` ファイル内で見つけることができます。使用中のストレージアレイが DM-Multipath に対応しており、このファイル内にデフォルトで設定されていない場合は、それらを DM-Multipath 設定ファイル `multipath.conf` に追加する必要があります。DM-Multipath 設定ファイルについての情報は、[4章DM-Multipath 設定ファイル](#) を参照してください。

一部のストレージアレイでは、I/O エラーとパス切り替えに特別な処理が必要となります。これらには、別々のハードウェアハンドラーカーネルモジュールが必要です。

1.4. DM-MULTIPATH コンポーネント

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

表1.1 DM-Multipath コンポーネント

コンポーネント	説明
<code>dm_multipath</code> 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。
<code>mpathconf</code> ユーティリティ	デバイスマッパーのマルチパス化を設定し、有効化します。

コンポーネント	説明
multipath コマンド	マルチパスデバイスを一覧表示し、設定します。通常は <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い際は <code>udev</code> プログラムで起動することも可能です。
multipathd デーモン	パスを監視します。パスの障害発生/復旧時にはパスグループスイッチを開始する場合があります。マルチパスデバイスへのインタラクティブな変更を行います。 <code>/etc/multipath.conf</code> ファイルに変更がある場合には再起動が必要です。
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-MP を持つ DOS ベースのパーティションにはこのコマンドを使用する必要があります。 <code>kpartx</code> はそれ自体のパッケージで提供されますが、 <code>device-mapper-multipath</code> パッケージは <code>kpartx</code> に依存しています。

1.5. DM-MULTIPATH セットアップの概要

DM-Multipath には、一般的なマルチパス設定に適したコンパイルされたデフォルト設定が含まれています。多くの場合 DM-multipath のセットアップ手順はシンプルです。

DM-Multipath で使用中のシステムを設定するための基本手順は以下のようになります。

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、マルチパス化を有効化します。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. 必要な場合は、`multipath.conf` 設定ファイルを編集して、デフォルト値を変更し、更新されたファイルを保存します。
4. マルチパスデーモンを起動します。

マルチパス設定のセットアップ手順の詳細は、[3章DM-Multipath のセットアップ](#)を参照してください。

第2章 マルチパスデバイス

DM-Multipath がない場合は、I/O パスが同じサーバーノードを同じストレージコントローラーに接続する場合でも、サーバーノードからストレージコントローラーまでの各パスは、システムにより別々のデバイスとして処理されます。DM-Multipath は、配下のデバイスの上に単一マルチパスデバイスを作成することにより、I/O パスを論理的に編成する方法を提供します。

2.1. マルチパスデバイス識別子

それぞれのマルチパスデバイスには World Wide Identifier (WWID) があり、これはグローバルに一意かつ不変であることが保証されます。デフォルトでは、マルチパスデバイス名は、その WWID に設定されます。別の方法として、マルチパス設定ファイルで `user_friendly_names` オプションを設定して、エイリアスを `mpathn` 形式のノードに一意の名前に設定することができます。

たとえば、ゾーニングされていない単一の FC スイッチを介して、2つのポートを持つストレージコントローラーに接続された2つの HBA を搭載するノードの場合、`/dev/sda`、`/dev/sdb`、`dev/sdc`、`/dev/sdd` の4つのデバイスを認識します。DM-Multipath は、マルチパス設定に従ってこれら4つの配下のデバイスへ I/O を再ルーティングする、一意の WWID を持つ単一デバイスを作成します。`user_friendly_names` 設定オプションが **yes** に設定されている場合、マルチパスデバイス名は `mpathn` に設定されます。

新規デバイスが DM-Multipath 管理下に置かれる場合、その新規デバイスは `/dev` ディレクトリー下の `/dev/mapper/mpathn` および `/dev/dm-n` の2つの別々の位置で確認することができます。

- `/dev/mapper` 内のデバイスは、ブートプロセス初期に作成されます。論理ボリュームを作成する場合などに、これらのデバイスを使用してマルチパス化したデバイスへアクセスします。
- `/dev/dm-n` の形式を持つすべてのデバイスは内部での使用のみを目的としており、通常は使用できません。

`user_friendly_names` 設定オプションを含むマルチパス設定デフォルトに関する情報は、「[設定ファイルのデフォルト](#)」を参照してください。

マルチパスデバイス名は、マルチパス設定ファイルの `multipaths` セクション内で `alias` オプションを使用することで、任意の名前に設定することもできます。マルチパス設定ファイルの `multipaths` セクションに関する情報は「[マルチパスデバイス設定属性](#)」を参照してください。

2.2. クラスタ内で一貫したマルチパスデバイス名

`user_friendly_names` 設定オプションが、**yes** に設定されている場合、マルチパスデバイス名はノードに一意となりますが、マルチパスデバイスを使用しているすべてのノード上で同じになる保証はありません。同様に、`multipath.conf` 設定ファイルの `multipaths` セクション内でデバイスの `alias` オプションを設定すると、その名前は、クラスタ内のすべてのノードで自動的に同じ名前にはなりません。これは、LVM を使用してマルチパスデバイスから論理デバイスを作成す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のデバイス名に一貫性を持たせる必要がある場合は、`user_friendly_names` オプションを **yes** に設定しないこと、およびデバイスにエイリアスを設定しないことを推奨します。`user_friendly_names` を **yes** に設定しない場合、またはデバイスのエイリアスを設定しない場合、デフォルトでは、デバイス名はそのデバイスの WWID となり、常に同じ名前になります。

ただし、クラスタ内のすべてのノードでシステム定義のユーザーフレンドリーな名前で一貫性を持たせるには、以下の手順を実行します。

1. 1つのマシン上ですべてのマルチパスデバイスをセットアップする。

2. 以下のコマンドを実行して、その他のマシン上でマルチパスデバイスをすべて無効にする。

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

3. 最初のマシンから、クラスター内のその他すべてのマシンに `/etc/multipath/bindings` ファイルをコピーする。
4. 以下のコマンドを実行して、クラスター内にあるその他のすべてのマシンで `multipathd` デモンを再度有効にする。

```
# service multipathd start
```

新たなデバイスを追加する場合には、この手順を繰り返す必要があります。

同様に、デバイスにエイリアスを設定する際にクラスター内のすべてのノードで一貫性を持たせるようにするには、同じ手順に従って、`/etc/multipath.conf` ファイルがクラスター内の各ノードで同一になることを確認してください。

1. 1つのマシン上で、`multipath.conf` ファイル内のマルチパスデバイスのエイリアスを設定する。
2. 以下のコマンドを実行して、その他のマシン上でマルチパスデバイスをすべて無効にする。

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

3. 最初のマシンから、クラスター内のその他すべてのマシンに `/etc/multipath.conf` ファイルをコピーする。
4. 以下のコマンドを実行して、クラスター内にあるその他のすべてのマシンで `multipathd` デモンを再度有効にする。

```
# service multipathd start
```

新たなデバイスを追加する場合には、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` や `alias` オプションに加え、マルチパスデバイスには多くの属性があります。特定のマルチパスデバイスに対してこの属性を変更するには、マルチパス設定ファイルの `multipaths` セクションに各デバイスのエントリーを作成してください。マルチパス設定ファイルの `multipaths` セクションについては、「[マルチパスデバイス設定属性](#)」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後は、LVM 物理ボリュームを作成する際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。たとえば、マルチパスデバイスの名前が `/dev/mapper/mpath0` の場合、以下のコマンドが `/dev/mapper/mpatha` を物理ボリュームとしてマークします。

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

結果として得られる LVM 物理デバイスは、他の LVM 物理デバイスを使用する場合と同じように、LVM ボリュームグループを作成する際に使用することができます。



注記

パーティションを設定済みのデバイス全体で LVM 物理ボリュームの作成を試行すると、**pvcreate** コマンドは失敗します。すべてのブロックデバイスにパーティションに対して何も指定されていない場合、Anaconda と Kickstart のインストールプログラムによって空のパーティションテーブルが作成されることに注意してください。1つの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しなければなりません。**kpartx -d** と **fdisk** のコマンドを使用して、既存のパーティションを削除することができます。システムに 2TB を超えるブロックデバイスがある場合には、**parted** コマンドを使用してパーティションを削除することができます。

配下の物理デバイスとしてアクティブ/パッシブのマルチパスアレイを使用する LVM 論理ボリュームを作成する場合、**lvm.conf** の中にフィルターを組み込み、マルチパスデバイスの内部にあるディスクを除外する必要があります。その理由は、I/O の受信時にアレイが自動的にアクティブパスをパッシブパスに変更する場合に、これらのデバイスがフィルターされていないと、LVM がパッシブパスをスキャンするたびにマルチパスはフェイルオーバーおよびフェイルバックしてしまうことがあるからです。パッシブパスをアクティブにする際にコマンドを必要とするアクティブ/パッシブアレイの場合、LVM はこの状態が発生すると警告メッセージを表示します。

LVM 設定ファイル (**lvm.conf**) 内ですべての 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 のセットアップ

この章では、DM-Multipath を設定するための手順を順を追って説明します。手順は以下のとおりです。

- 基本的な DM-Multipath のセットアップ
- ローカルディスクを無視する
- 設定ファイルへのデバイスの追加
- `initramfs` ファイルシステムでのマルチパスの起動

3.1. DM-MULTIPATH のセットアップ

ご使用のシステムに 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` パッケージが含ま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セットアップします。これによって、マルチパス設定ファイル `/etc/multipath.conf` が作成されます。

- `/etc/multipath.conf` ファイルがすでに存在する場合、`mpathconf` ユーティリティは、その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、`mpathconf` ユーティリティは、`/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` ファイルを開始ファイルとして使用します。
- `/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` ユーティリティが `/etc/multipath.conf` ファイルをゼロから作成します。

`/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ことによって、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。このコマンドによって、マルチパス設定が有効にされ、`multipathd` デーモンが起動します。

```
# 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、以下の手順に従って、基本的なフェイルオーバー設定用に 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
```

必要になる可能性のある `mpathconf` コマンドへの追加オプションについての情報を得るには、`mpathconf` の man ページを参照するか、または `--help` オプションを指定して `mpathconf` コマンドを実行してください。

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

Commands:
Enable: --enable
Disable: --disable
Set user_friendly_names (Default n): --user_friendly_names <y|n>
Set find_multipaths (Default n): --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>
chkconfig on/off multipathd (Default y): --with_chkconfig <y|n>
```

2. 必要な場合は、`/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してあるため、`/etc/multipath.conf` ファイルに明示的に設定する必要はありません。

`path_grouping_policy` のデフォルト値は `failover` に設定されています。そのため、この例では `/etc/multipath.conf` ファイルを編集する必要はありません。設定ファイル内の値のデフォルト以外の値に変更する方法についての情報は、[4章DM-Multipath 設定ファイル](#) を参照してください。

設定ファイルの初期デフォルトセクションは、マルチパスデバイス名が `mpathn` の形式になるようご使用のシステムを設定します。この設定がないと、マルチパスデバイス名はそのデバイスの WWID へとエイリアス化されます。

3. 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。
4. 以下のコマンドを実行します。

```
# service multipathd start
```

`user_friendly_name` の値が、設定ファイルで `yes` に設定されているため、マルチパスデバイスは `/dev/mapper/mpathn` として作成されます。デバイス名を別の任意のエイリアスに設定する方法についての情報は、[4章DM-Multipath 設定ファイル](#) を参照してください。

ユーザーフレンドリーな名前を使用したくない場合には、以下のコマンドを実行することができます。

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



注記

マルチパスデーモンの起動後にマルチパス設定ファイルを編集する必要がある場合、変更を有効にするには `service multipathd reload` コマンドを使用する必要があります。

3.2. マルチパスデバイスの生成時にローカルディスクを無視する

一部のマシンには、内蔵ディスク用にローカルの SCSI カードが搭載されています。これらのデバイスには、DM-Multipath は推奨されません。`find_multipaths` 設定パラメーターを `yes` に設定すると、これらのデバイスをブラックリスト化する必要はなくなります。`find_multipaths` 設定パラメーターについての情報は、「[設定ファイルのデフォルト](#)」を参照してください。

`find_multipaths` 設定パラメーターを `yes` に設定しない場合は、以下の手順に従って、マルチパス設定ファイルを修正し、マルチパスの設定時にローカルディスクを無視することができます。

1. どのディスクが内蔵ディスクかを判別し、それらをブラックリスト用にマークします。

この例では、`/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で最初に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` が表示されることに注意してください。

`multipath` コマンドの出力に関する詳細情報は、「[multipath コマンドの出力](#)」を参照してください。

```

# 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. デバイスマッパーがマルチパスマップ内の `/dev/sda` をマッピングできないようにするには、`/etc/multipath.conf` ファイルのブラックリストセクションを編集して、このデバイスを加えます。`devnode` タイプを使用しても `sda` デバイスをブラックリスト化することはできますが、これは `/dev/sda` が再起動後も同じである保証はないため、安全な手順ではありません。個別のデバイスをブラックリスト化するには、そのデバイスの WWID を使用します。

`multipath -v2` コマンドへの出力で、`/dev/sda` デバイスの WWID は `SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1` であることに注意してください。このデバイスをブラックリスト化するには、`/etc/multipath.conf` ファイルに以下を含めます。

```

blacklist {
    wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}

```

3. `/etc/multipath.conf` ファイルを更新した後に、`multipathd` デーモンに対して、そのファイルをリロードするように手動で指示する必要があります。以下のコマンドによって、更新済みの `/etc/multipath.conf` ファイルがリロードされます。

```
# 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. デバイスが削除されたかどうかを確認するには、**multipath -ll** コマンドを実行して、現在のマルチパス設定を表示します。**multipath -ll** コマンドについての情報は、「[multipath コマンドを使用したマルチパスクエリー](#)」を参照してください。

ブラックリスト化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ためには、以下の例のように **multipath** コマンドを実行することができます。**-v** オプションを指定しなかった場合は、**multipath** コマンドは **v2** の詳細レベルにデフォルト設定されます。

```
# 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 にはデフォルトで DM-Multipath に対応する最も一般的なストレージレイのサポートが含まれています。サポートされているデバイスを含むデフォルトの設定値は **multipath.conf.defaults** ファイル内で見つけることができます。

デフォルトで既知のマルチパスデバイスとして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挿入します。

たとえば、HP Open-V シリーズに関する情報を追加するには、デバイス名が %n である場合のエントリは以下ようになります。

```
devices {
    device {
        vendor "HP"
        product "OPEN-V."
```

```
    getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"  
  }  
}
```

設定ファイルの **devices** セクションについての詳細は、「[設定ファイルデバイス](#)」を参照してください。

3.4. INITRAMFS ファイルシステムでのマルチパス機能のセットアップ

initramfs ファイルシステムでマルチパス化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使用して **dracut** コマンドを実行し、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することができます。

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

デフォルトでは、DM-Multipath はマルチパス化で最もよく使用する設定値を提供します。さらに DM-Multipath は、DM-Multipath 対応の最も一般的なストレージレイもサポートしています。そのデフォルト設定値とサポートされているデバイスは、`/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.defaults` ファイルで確認できます。

DM-Multipath のデフォルト設定値は `/etc/multipath.conf` 設定ファイルを編集して上書きすることができます。必要な場合は、デフォルトで対応していないストレージレイもこの設定ファイルに追加することができます。



注記

`initramfs` ファイルシステムで、設定されたマルチパス化を実行することができません。`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た場合、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。マルチパスの `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のマルチパス機能のセットアップ](#)」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。本書には、以下のトピックに関するセクションが含まれます。

- 設定ファイルの概要
- 設定ファイルのブラックリスト
- 設定ファイルのデフォルト
- 設定ファイルのマルチパス
- 設定ファイルのデバイス

マルチパス設定ファイル内では、ご使用の設定に必要なセクションのみか、または `multipath.conf.defaults` ファイルに設定しているデフォルト値を変更したいセクションのみを指定する必要があります。使用中の環境に関連していないか、またはデフォルト値を上書きする必要がないファイルのセクションがある場合、それらは初期ファイル内にあるため、コメントアウトしたままにできます。

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。

設定ファイルの注釈付きバージョンは `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.annotated` にあります。

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割されています。

blacklist

マルチパスでは考慮されない特定のデバイスの一覧表示

blacklist_exceptions

例外として指定されていない場合に、ブラックリストセクションのパラメーターに従ってブラックリスト化されることになるマルチパス候補の一覧表示

defaults

DM-Multipath の一般的なデフォルト設定

multipaths

個別のマルチパスデバイスの特性の設定です。これらの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内容を上書きします。

devices

個別ストレージコントローラー用の設定です。これらの値は設定ファイル内の **defaults** セクションで指定されている内容を上書きします。デフォルトでサポートされていないストレージレイを使用している場合は、そのレイの **devices** サブセクションを作成する必要がある場合があります。

システムがマルチパスデバイスの属性を判別する場合、システムは最初にマルチパス設定をチェックし、次にデバイスごとの設定をチェックしてから、マルチパスシステムのデフォルトをチェックします。

4.2. 設定ファイルのブラックリスト

マルチパス設定ファイルの **blacklist** セクションは、システムがマルチパスデバイスを設定する際に使用されないデバイスを指定します。ブラックリスト化されたデバイスはマルチパスデバイスにグループ化されません。

以前の Red Hat Enterprise Linux のリリースでは、マルチパスは明示的にブラックリスト化されていないすべてのパスを対象にマルチパスデバイスの作成を常に試行していました。しかし、Red Hat Enterprise Linux 6 では、**find_multipaths** 設定パラメーターを **yes** に設定すると、マルチパスは以下の3つの条件のいずれかを満たす場合にのみデバイスを作成します。

- 同一の WWID を持つ、ブラックリスト化されていないパスが少なくとも2つある。
- **multipath** コマンドでデバイスを指定することにより、ユーザーがデバイスの手動作成を強制的に実行する。
- パスに以前に作成されたマルチパスデバイスと同一の WWID がある (そのマルチパスデバイスが同時に存在しない場合も同様)。マルチパスデバイスが作成される時には常に、マルチパスがそのデバイスの WWID を記憶するので、同一の WWID を持つパスを認識するとすぐに、デバイスを自動的に再作成します。これによって、マルチパスブラックリストを編集することなく、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るようになります。

以前に **find_multipaths** パラメーターを使用せずにマルチパスデバイスを作成し、その後パラメーターを **yes** に設定した場合は、**/etc/multipath/wwids** ファイルからマルチデバイスとして作成したくないデバイスの WWID を削除しなければならない場合があります。以下は、**/etc/multipath/wwids** ファイルの例を示しています。WWID はスラッシュ (/) で囲まれています。

```
# 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:
/3600d0230000000000e13955cc3757802/
/3600d0230000000000e13955cc3757801/
/3600d0230000000000e13955cc3757800/
```

```

/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372 0E13955CC3757802/
/3600d0230000000000e13955cc3757803/

```

`find_multipaths` パラメーターを **yes** に設定すると、ブラックリスト化する必要があるのは、マルチパス化しない複数のパスがあるデバイスのみです。このため、通常はデバイスをブラックリスト化する必要はありません。

デバイスをブラックリスト化する必要がある場合には、以下の基準に従って行うことができます。

- WWID 単位。「[WWID 単位のブラックリスト](#)」の説明を参照。
- デバイス名の単位。「[デバイス名単位のブラックリスト](#)」の説明を参照。
- デバイスタイプ単位。「[デバイスタイプ単位のブラックリスト](#)」の説明を参照。

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化されます。詳細は「[デバイス名単位のブラックリスト](#)」を参照してください。

4.2.1. WWID 単位のブラックリスト

設定ファイルの `blacklist` セクションに `wwid` エントリーを持つ World-Wide Identification 単位でブラックリスト化する個別デバイスを指定することができます。

以下の例では、26353900f02796769 の WWID を持つデバイスをブラックリスト化する設定ファイル内の行を示しています。

```

blacklist {
    wwid 26353900f02796769
}

```

4.2.2. デバイス名単位のブラックリスト

設定ファイルの `blacklist` セクション内で `devnode` エントリーを指定することにより、デバイス名でデバイスタイプをブラックリスト化し、それらがマルチパスデバイスにグループ化されないようにすることができます。

以下の例は、すべての SCSI デバイスをブラックリスト化する設定ファイル内の行を示しています。これは、すべての `sd*` デバイスをブラックリスト化します。

```

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

```

設定ファイル内の `blacklist` セクションの `devnode` エントリーを使用して、特定タイプのすべてのデバイスでなく、ブラックリスト化する個別デバイスを指定できます。しかし、`udev` ルールで静的にマップされていない限り、その特定デバイスが再起動後に同一名となる保証はないため、この方法は推奨されません。たとえば、デバイス名は再起動後に `/dev/sda` から `/dev/sdb` に変わる可能性があります。

デフォルトでは、以下の `devnode` エントリー群はデフォルトブラックリストでコンパイルされます。これらのエントリー群によってブラックリスト化されるデバイスは、通常 DM-Multipath をサポートしません。これらのデバイス上でマルチパス化を有効にするには、「[ブラックリスト例外](#)」で説明されている様に、設定ファイルの `blacklist_exceptions` セクションでそれらを指定する必要があります。

■

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

4.2.3. デバイスタイプ単位のブラックリスト

device セクションを使用して設定ファイルの **blacklist** セクションに特定のデバイスタイプを指定できます。以下の例は、すべての IBM DS4200 デバイスと HP デバイスをブラックリスト化します。

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

4.2.4. ブラックリスト例外

設定ファイルの **blacklist_exceptions** セクションを使用して、デフォルトでブラックリスト化されているデバイスでのマルチパス化を有効にできます。

たとえば、多数のデバイスがあり、その中の 1 つのデバイス (3600d0230000000000e13955cc3757803 の WWID を持つ) のみをマルチパス化したい場合には、そのデバイスを除いた各デバイスを個別にブラックリスト化する代わりに、すべてをブラックリスト化してから、**/etc/multipath.conf** ファイルに以下の行を追加することによって、マルチパス化したいデバイスを許可することができます。

```
blacklist {
    wwid "*"
}

blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}
```

設定ファイル内の **blacklist_exceptions** セクションでデバイスを指定する場合、それらがブラックリストで指定されているのと同じ方法で例外を指定する必要があります。たとえば、そのブラックリスト化されたデバイスが WWID に関連付けられている場合でも、WWID 例外は **devnode** ブラックリストエントリーで指定されたデバイスへは適用できません。同様に **devnode** 例外は **devnode** エントリーのみに適用され、**device** 例外はデバイスエントリーにのみ適用されます。

4.3. 設定ファイルのデフォルト

/etc/multipath.conf 設定ファイルには、以下のように **user_friendly_names** パラメーターを **yes** に設定する **defaults** セクションが含まれています。

```
defaults {
    user_friendly_names yes
}
```

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

設定ファイルには、設定デフォルト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 {
#   udev_dir                /dev
#   polling_interval        5
#   path_selector           "round-robin 0"
#   path_grouping_policy    failover
#   getuid_callout          "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
# prio const
# path_checker directio
# rr_min_io 1000
# rr_weight uniform
# failback manual
# no_path_retry fail
# user_friendly_names no
#}
```

設定パラメーターのいずれかのデフォルト値を上書きするには、このテンプレートから該当する行を `defaults` セクションにコピーしてそのコメントを解除します。たとえば、`path_grouping_policy` パラメーターを上書きして `failover` のデフォルト値ではなく `multibus` にするには、そのテンプレートから該当する行を設定ファイルの始めにある `defaults` セクションにコピーして、以下のようにコメントを解除します。

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

表4.1「マルチパス設定デフォルト」は、`multipath.conf` 設定ファイルの `defaults` セクションに設定されている属性を説明しています。これらの値は、`multipath.conf` ファイルの `devices` と `multipaths` セクション内に指定されている属性で上書きされない限り、DM-Multipath によって使用されます。



注記

Red Hat Enterprise Linux 6.0 リリース以降、`mode`、`uid`、`gid` パラメーターは非推奨となっています。デバイスマッパーデバイスのパーミッション (マルチパスマッピングも含む) は、`udev` ルールを使って設定します。`/usr/share/doc/device-mapper-version` 内に `12-dm-permissions.rules` と呼ばれるテンプレートファイルがあります。これを使用して、`/etc/udev/rules.d` ディレクトリーに配置すると有効になります。

表4.1 マルチパス設定デフォルト

属性	説明
<code>polling_interval</code>	2つのパスチェックの間隔を秒単位で指定します。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。
<code>udev_dir</code>	<code>udev</code> デバイスノードが作成されるディレクトリーです。デフォルト値は、 <code>/dev</code> 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依存しており、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用のモードを定義します。このパラメーターが yes に設定されている場合、マルチパスは、ブラックリスト化されていないすべてのパスに対してデバイスの作成を試行するのではなく、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"> - 同一の WWID を持つ、ブラックリスト化されていないパスが少なくとも 2 つある。 - multipath コマンドでデバイスを指定することにより、ユーザーがデバイスの手動作成を強制的に実行する。 - パスに以前に作成されたマルチパスデバイスと同一の WWID がある。find_multipaths が設定された状態でマルチパスデバイスが作成される場合は常に、マルチパスがそのデバイスの WWID を記憶するため、同一の WWID を持つパスを認識するとすぐに、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 パラメーターが設定されていない状態でマルチパスデバイスを作成した場合に従うべき手順についての説明は、「設定ファイルのブラックリスト」を参照してください。 <p>デフォルト値は no です。</p>
verbosity	デフォルトの詳細度です。値が高いほど、詳細度が高くなります。有効なレベルは 0 と 6 の間です。デフォルト値は 2 です。
path_selector	<p>次の I/O 操作に使用するパスを決定するのに使うデフォルトアルゴリズムを指定します。使用できる値には以下が含まれます。</p> <ul style="list-style-type: none"> round-robin 0: パスグループ内のすべてのパスをループスルーを実行し、それぞれに同量の I/O を送ります。 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。 service-time 0: 各パスに対する未処理の I/O 全体のサイズを相対スループットで割ることによって求められるサービス時間が最も短いパスに、次の I/O 群を送ります。 <p>デフォルト値は round-robin 0 です。</p>

属性	説明
<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プリングポリシーを指定します。使用できる値は以下のようになります。</p> <p>fai lover: 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のすべてのパスが有効</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 fai lover です。</p>
<p>getuid_callout</p>	<p>一意のパス識別子を取得するためにコールアウトするデフォルトのプログラムと引数を指定します。絶対パスが必要です。</p> <p>デフォルト値は <code>/lib/udev/scsi_id --whitelisted --device=/dev/%n</code> です。</p>

属性	説明
prio	<p>パスの優先値を取得するためにコールアウトするデフォルトの関数を指定します。たとえば、SPC-3 内の ALUA ビットは、利用可能な prio 値を提供します。使用できる値は以下のとおりです。</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、パスの優先度を生成します。</p> <p>tpg_pref: 優先ポートのビットを使用し、SCSI-3 ALUA に基づいて、パスの優先度を生成します。</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: アクティブ/スタンバイモードの Compaq/HP コントローラーの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
<p>features</p>	<p>以下の形式を使用した、マルチパスデバイスのデフォルトの追加機能です。"<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>features に使用できる値には以下が含まれます。</p> <p>queue_if_no_path: これは no_path_retry を queue に設定することと同じです。この機能を使用する場合に発生する可能性のある問題についての情報は、「queue_if_no_path 機能での問題」を参照してください。</p> <p>retain_attached_hw_handler: (Red Hat Enterprise Linux リリース 6.4以降) このパラメーターが yes に設定されており、scsi 層がハードウェアハンドラーをパスデバイスにすでに接続している場合、マルチパスは、デバイスに対して、multipath.conf ファイルで指定されている hardware_handler を強制的に使用させることはありません。scsi 層がハードウェアハンドラーを接続していない場合、マルチパスは通常の方法で設定したハードウェアハンドラーを継続して使用します。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の間隔として <i>n</i> ミリ秒待機します。ここで、$0 \leq n \leq 60000$ です。</p>
<p>path_checker</p>	<p>パスの状態を判断するのに使用されるデフォルトのメソッドを指定します。使用できる値は以下のとおりです。</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 のクエリーを実行し、パスを判別します。</p> <p>hp_sw: アクティブ/スタンバイ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
fai lback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値は、アクティブパスを含む、最も優先度の高いパスグループに対する即時フェイルバックを指定します。</p> <p>manual の値は、即時フェイルバックを不可能にし、オペレーターの介入がある場合のみにフェイルバックが発生する可能性があるように指定します。</p> <p>followover の値は、パスグループの最初のパスがアクティブになる時点で自動フェイルバックが実行されるように指定します。これによって、他のノードがフェイルオーバーを要求した際に、ノードが自動的にフェイルバックしないようにします。</p> <p>「0」よりも大きい数値は、フェイルバックの遅延を秒単位で指定します。</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内で次のパスに切り替える前に、パスにルーティングする I/O 要求の数を指定します。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のみ有効です。これ以降の新しいシステムには 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内で次のパスに切り替える前に、要求ベースの device-mapper-multipath を使用して、パスにルーティングする I/O 要求の数を指定します。この設定は、現在のカーネルを実行しているシステム上で使用してください。2.6.31 より前のカーネルを実行しているシステムでは、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 に設定されている場合、次のパスを選択するために selector を呼び出す前に rr_min_io の要求をパスに送信する代わりに、送信する要求の数は rr_min_io に、prio 関数で決定されるパスの優先度を掛けることで求められます。uniform に設定されている場合、すべてのパスの重みは等しくなります。デフォルト値は uniform です。</p>
no_path_retry	<p>この属性の数値は、キューを無効にするまでに、システムが障害の発生したパスの使用を試行する回数を指定します。</p> <p>fail の値は、キューなしの即時の失敗を示します。</p> <p>queue の値は、パスが修復されるまでキューが停止されないことを示します。</p> <p>デフォルト値は 0 です。</p>

属性	説明
user_friendly_names	yes に設定されている場合、システムが <code>/etc/multipath/bindings</code> ファイルを使用して、永続的で一意のエイリアスを <code>mpathn</code> の形式でマルチパスへ割り当てるように指定します。 no に設定されている場合、システムが WWID をマルチパスのエイリアスとして使用する必要があることを指定します。いずれの場合も、ここでの指定は、設定ファイルの multipaths セクション内で指定するデバイス固有のエイリアスによって上書きされます。デフォルト値は no です。
queue_without_daemon	no に設定されている場合、 multipathd デーモンはシャットダウン時にすべてのデバイスのキューを無効にします。デフォルト値は no です。
flush_on_last_del	yes に設定されている場合、 multipathd デーモンは、デバイスへの最後のパスが削除された際にキューを無効にします。デフォルト値は no です。
max_fds	マルチパスおよび multipathd デーモンで開くことができるオープンファイル記述子の最大数を設定します。これは、 <code>ulimit -n</code> コマンドに相当します。Red Hat Enterprise Linux 6.3 リリースの時点で、デフォルト値は max であり、これは <code>/proc/sys/fs/nr_open</code> でシステムの上限として設定されます。これまでのリリースでは、この値が設定されていないと、オープンファイル記述子の最大数は呼び出し元のプロセスから取得されます。通常の値は 1024 です。その値が 1024 を超える場合は、万が一に備えてパスの最大値に 32 を加えた値に設定することをお勧めします。
checker_timeout	prioritizer および明示的なタイムアウトによって SCSI コマンドを数秒で発行するパスチェッカーに使用するタイムアウトです。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 リモートポート上での問題の検出後に、そのリモートポート上でデバイスへの I/O に失敗するまでの scsi 層の待機時間の秒数です。この値は <code>dev_loss_tmo</code> 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。デフォルト値は OS により決定されます。
dev_loss_tmo	FC リモートポート上での問題の検出後に、それがシステムから削除されるまでの scsi 層の待機時間の秒数です。無限に設定するには 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
hwtable_regex_match	<p>(Red Hat Enterprise Linux Release 6.3 以降) 組み込まれたデバイス設定を使ってマルチパスが設定ファイルの devices セクションからデバイス設定を統合する方法を制御します。</p> <p>multipath.conf ファイルの devices セクション内の各デバイス設定は、独自のデバイス設定を作成するか、または組み込まれたデバイス設定の 1 つを修正します。Red Hat Enterprise Linux 6.3 より前のバージョンでは、ユーザーのデバイス設定内のベンダー、製品、およびリビジョン文字列が組み込まれたデバイス設定のそれらの文字列に完全に一致する場合、組み込まれた設定はユーザー設定内のオプションにより修正されていました。そうでない場合は、ユーザーのデバイス設定は新しい設定として処理されていました。</p> <p>hwtable_regex_match が yes に設定されると、正規表現の一致が代わりに使用されます。ベンダー、製品、およびリビジョン文字列はすべて正規表現です。これらのオプションのユーザーデバイス設定値は、組み込まれたデバイス設定値と一致します。この一致は、実際のデバイスのベンダー製品とリビジョン文字列がデバイス設定の文字列と一致するのと同様に機能し、これによりデバイスに使用する必要のある設定を確認できます。ユーザーのデバイス設定が一致する場合は、組み込まれた設定はユーザー設定のオプションにより修正されます。そうでない場合は、ユーザーのデバイス設定は新しい設定として扱われます。</p>
retain_attached_hw_handler	<p>(Red Hat Enterprise Linux リリース 6.4 以降) このパラメーターが yes に設定されており、scsi 層がハードウェアハンドラーをパスデバイスにすでに接続している場合、マルチパスはデバイスに対して multipath.conf で指定される hardware_handler を強制的に使用させることはありません。scsi 層がハードウェアハンドラーを接続していない場合、マルチパスは通常の方法で設定したハードウェアハンドラーを継続して使用します。デフォルト値は no です。</p>
detect_prio	<p>(Red Hat Enterprise Linux リリース 6.4 以降) これが yes に設定されている場合、マルチパスは、まずデバイスが 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に prioritizer を判別します。デフォルト値は no です。</p>

属性	説明
<code>replace_wwid_whitespace</code>	(Red Hat Enterprise Linux リリース 6.5 以降) これが yes に設定されている場合、このパラメーターを使用するすべてのデバイスのデフォルトの <code>getuid_callout</code> は、 <code>--replace-whitespace</code> を含むように変更されます。
<code>reload_readwrite</code>	(Red Hat Enterprise Linux リリース 6.5 以降) これが yes に設定されている場合、 <code>multipathd</code> デーモンがパスデバイス変更イベントをリッスンし、デバイスが読み取り/書き込みになると、マルチパスデバイスをリロードします。
<code>force_sync</code>	(Red Hat Enterprise Linux Release 6.6 以降) これが「 yes 」に設定されている場合、パステッカーが非同期モードで実行されるのを防ぎます。

4.4. マルチパスデバイス設定属性

表4.2「マルチパス属性」は、それぞれの特定マルチパスデバイス用の `multipath.conf` 設定ファイルの `multipaths` セクションに設定できる属性を示しています。これらの属性は1つの指定されたマルチパスのみに適用されます。これらのデフォルトは DM-Multipath によって使用され、`multipath.conf` ファイルの `defaults` と `devices` セクションに設定されている属性を上書きします。

表4.2 マルチパス属性

属性	説明
<code>wwid</code>	<code>multipath</code> 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、 <code>multipath.conf</code> ファイルのこのセクションには必須です。
<code>alias</code>	<code>multipath</code> 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。 <code>user_friendly_names</code> を使用する場合には、この値を <code>mpathn</code> に設定しないでください。これによって、自動的に割り当てられたユーザーフレンドリーな名前と競合する可能性や、正しくないデバイスノード名が提供される可能性があります。

属性	説明
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ピングポリシーを指定します。使用できる値は以下のようになります。</p> <p>fai lover = 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のすべてのパスが有効</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>次の I/O 操作に使用するパスを決定するのに使うデフォルトアルゴリズムを指定します。使用できる値には以下が含まれます。</p> <p>round-robin 0: パスグループ内のすべてのパスをループスルーを実行し、それぞれに同量の 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: 各パスに対する未処理の I/O 全体のサイズを相対スループットで割ることによって求められるサービス時間が最も短いパスに、次の I/O 群を送ります。</p>
fai lback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値は、アクティブパスを含む、最も優先度の高いパスグループに対する即時フェイルバックを指定します。</p> <p>manual の値は、即時フェイルバックを不可能にし、オペレーターの介入がある場合のみにフェイルバックが発生する可能性があるように指定します。</p> <p>followover の値は、パスグループの最初のパスがアクティブになる時点で自動フェイルバックが実行されるように指定します。これによって、他のノードがフェイルオーバーを要求した際に、ノードが自動的にフェイルバックしないようにします。</p> <p>「0」よりも大きい数値は、フェイルバックの遅延を秒単位で指定します。</p>

属性	説明
prio	<p>パスの優先値を取得するためにコールアウトするデフォルトの関数を指定します。たとえば、SPC-3 内の ALUA ビットは、利用可能な prio 値を提供します。使用できる値は以下のとおりです。</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、パスの優先度を生成します。</p> <p>tpg_pref: 優先ポートのビットを使用し、SCSI-3 ALUA に基づいて、パスの優先度を生成します。</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: アクティブ/スタンバイモードの Compaq/HP コントローラーの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>
no_path_retry	<p>この属性の数値は、キューを無効にするまでに、システムが障害の発生したパスの使用を試行する回数を指定します。</p> <p>fail の値は、キューなしの即時の失敗を示します。</p> <p>queue の値は、パスが修復されるまでキューが停止されないことを示します。</p>
rr_min_io	<p>現在のパスグループ内で次のパスに切り替える前に、パスにルーティングする I/O 要求の数を指定します。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のみ有効です。それ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内で次のパスに切り替える前に、要求ベースの device-mapper-multipath を使用して、パスにルーティングする I/O 要求の数を指定します。この設定は、現在のカーネルを実行しているシステム上で使用してください。2.6.31 より前のカーネルを実行しているシステムでは、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
<code>rr_weight</code>	<code>priorities</code> に設定されている場合は、次のパスを選択するために <code>path_selector</code> を呼び出す前に <code>rr_min_io</code> の要求をパスに送信する代わりに、送信する要求の数は <code>rr_min_io</code> に、 <code>prio</code> 関数で決定されるパスの優先度を掛けることで求められます。 <code>uniform</code> に設定されている場合、すべてのパスの重みは等しくなります。
<code>flush_on_last_del</code>	<code>yes</code> に設定されている場合、マルチパスは、デバイスへの最後のパスが削除される際にキューを無効にします。
<code>user_friendly_names</code>	<code>yes</code> に設定されている場合、システムが <code>/etc/multipath/bindings</code> ファイルを使用して、永続的で一意のエイリアスを <code>mpathn</code> の形式でマルチパスに割り当てるように指定します。 <code>no</code> に設定されている場合は、システムが WWID をマルチパスのエイリアスとして使用するよう指定します。いずれの場合も、ここで指定される内容は設定ファイルの <code>multipaths</code> セクションで指定するデバイス固有のエイリアスで上書きされます。

以下の例では、2つの特定マルチパスデバイス用の設定ファイル内に指定されたマルチパス属性を示しています。1番目のデバイスには `3600508b4000156d70001200000b0000` の WWID と `yellow` のシンボリック名があります。

例にある2番目のマルチパスデバイスには `1DEC____321816758474` の WWID と `red` のシンボリック名があります。この例では、`rr_weight` 属性は `priorities` に設定されています。

```

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. 設定ファイルデバイス

表4.3「デバイス属性」は、`multipath.conf` 設定ファイルの `devices` セクションにあるそれぞれの個別ストレージデバイス用に設定できる属性を示しています。デバイスを含むパス用の `multipath.conf` ファイルの `multipaths` セクションで指定される属性によって上書きされない限り、これらの属性は DM-Multipath によって使用されます。これらの属性は `multipath.conf` ファイルの `defaults` セクションに設定されている属性を上書きします。

マルチパス化に対応する多くのデバイスは、マルチパス設定にデフォルトで含まれています。デフォルト

トでサポートされているデバイスの値は `multipath.conf.defaults` ファイルに一覧表示されます。おそらくこれらのデバイスの値を変更する必要はありませんが、変更する場合は、デフォルト値を上書きするデバイス用の設定ファイルにエントリーを1つ含めます。デバイスの `multipath.conf.defaults` ファイルからデバイス設定のデフォルトをコピーし、変更したい値を上書きすることができます。

デフォルトで自動的に設定されていないデバイスを設定ファイルのこのセクションに追加するには、`vendor` と `product` のパラメーターを設定する必要があります。これらの値は、`/sys/block/device_name/device/vendor` と `/sys/block/device_name/device/model` で見つけることができます。ここで、`device_name` は、以下の例にあるようにマルチパス化の対象となるデバイスです。

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

指定する追加のパラメーターは、個々のデバイスによって異なります。デバイスがアクティブ/アクティブの場合は、通常追加のパラメーターを設定する必要はありません。`path_grouping_policy` は `multibus` に設定することもできます。設定する必要のある他のパラメーターには、表4.3「デバイス属性」で説明されているように `no_path_retry` と `rr_min_io` があります。

デバイスがアクティブ/パッシブである場合に、I/O を持つパス群が自動的にパッシブパスに切り替わる場合は、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てから、作動することをテストする必要があります (そうしない場合、デバイスはフェイルオーバーを続けます)。これは、`path_checker` がほとんど常に `tur` に設定されることを意味します。この方法は「Test Unit Ready」コマンドをサポートするすべての SCSI デバイスで機能します。大半のデバイスは、このコマンドをサポートしていません。

表4.3 デバイス属性

属性	説明
<code>vendor</code>	<code>device</code> 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。
<code>product</code>	<code>device</code> 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。
<code>revision</code>	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。
<code>product_blacklist</code>	製品別にデバイスをブラックリスト化するために使用する正規表現を指定します。

属性	説明
hardware_handler	<p>パスグループの切り替え、または I/O エラーの処理時にハードウェア特有の動作を実行するのに使用するモジュールを指定します。使用できる値は以下のとおりです。</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。使用できる値は以下のようになります。</p> <p>fai lover = 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のすべてのパスが有効</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>
getuid_callout	<p>一意のパス識別子を取得するためにコールアウトするデフォルトのプログラムと引数を指定します。絶対パスが必要です。</p>
path_selector	<p>次の I/O 操作に使用するパスを決定するのに使うデフォルトアルゴリズムを指定します。使用できる値には以下が含まれます。</p> <p>round-robin 0: パスグループ内のすべてのパスをループスルーを実行し、それぞれに同量の 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: 各パスに対する未処理の I/O 全体のサイズを相対スループットで割ることによって求められるサービス時間が最も短いパスに、次の I/O 群を送ります。</p>

属性	説明
path_checker	<p>パスの状態を判断するのに使用されるデフォルトのメソッドを指定します。使用できる値は以下のとおりです。</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 のクエリーを実行し、パスを判別します。</p> <p>hp_sw: アクティブ/スタンバイ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>以下の形式を使用した、マルチパスデバイスのデフォルトの追加機能です。"<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>features に使用できる値には以下が含まれます。</p> <p>queue_if_no_path: これは no_path_retry を queue に設定することと同じです。この機能を使用する場合に発生する可能性のある問題についての情報は、「queue_if_no_path 機能での問題」 を参照してください。</p> <p>retain_attached_hw_handler: (Red Hat Enterprise Linux リリース 6.4 以降) このパラメーターが yes に設定されており、scsi 層がハードウェアハンドラーをパスデバイスにすでに接続している場合、マルチパスはデバイスに対して、multipath.conf で指定されている hardware_handler を強制的に使用させることはありません。scsi 層がハードウェアハンドラーを接続していない場合、マルチパスは通常の方法で設定したハードウェアハンドラーを継続して使用します。</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の間隔として <i>n</i> ミリ秒待機します。ここで、$0 \leq n \leq 60000$ です。</p>

属性	説明
prio	<p>パスの優先値を取得するためにコールアウトするデフォルトの関数を指定します。たとえば、SPC-3 内の ALUA ビットは、利用可能な prio 値を提供します。使用できる値は以下のとおりです。</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、パスの優先度を生成します。</p> <p>tpg_pref: 優先ポートのビットを使用し、SCSI-3 ALUA に基づいて、パスの優先度を生成します。</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: アクティブ/スタンバイモードの Compaq/HP コントローラーの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>
fai lback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値は、アクティブパスを含む、最も優先度の高いパスグループに対する即時フェイルバックを指定します。</p> <p>manual の値は、即時フェイルバックを不可能にし、オペレーターの介入がある場合のみにフェイルバックが発生する可能性があるように指定します。</p> <p>followover の値は、パスグループの最初のパスがアクティブになる時点で自動フェイルバックが実行されるように指定します。これによって、他のノードがフェイルオーバーを要求した際に、ノードが自動的にフェイルバックしないようにします。</p> <p>「0」よりも大きい数値は、フェイルバックの遅延を秒単位で指定します。</p>

属性	説明
rr_weight	priorities に設定されている場合は、次のパスを選択するために path_selector を呼び出す前に rr_min_io の要求をパスに送信する代わりに、送信する要求の数は rr_min_io に、 prio 関数で決定されるパスの優先度を掛けることで求められます。 uniform に設定されている場合、すべてのパスの重みは等しくなります。
no_path_retry	<div data-bbox="598 454 1428 568" style="border: 1px solid black; padding: 5px;"> <p>この属性の数値は、キューを無効にするまでに、システムが障害の発生したパスの使用を試行する回数を指定します。</p> </div> <div data-bbox="598 568 1428 651" style="border: 1px solid black; padding: 5px;"> <p>fail の値は、キューなしの即時の失敗を示します。</p> </div> <div data-bbox="598 651 1428 766" style="border: 1px solid black; padding: 5px;"> <p>queue の値は、パスが修復されるまでキューが停止されないことを示します。</p> </div>
rr_min_io	現在のパスグループ内で次のパスに切り替える前に、パスにルーティングする I/O 要求の数を指定します。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のみ有効です。それ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。
rr_min_io_rq	現在のパスグループ内で次のパスに切り替える前に、要求ベースの device-mapper-multipath を使用して、パスにルーティングする I/O 要求の数を指定します。この設定は、現在のカーネルを実行しているシステム上で使用してください。2.6.31 より前のカーネルを実行しているシステムでは、 rr_min_io を使用してください。デフォルト値は 1 です。
fast_io_fail_tmo	FC リモートポート上での問題の検出後に、そのリモートポート上でデバイスへの I/O に失敗するまでの scsi 層の待機時間の秒数です。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。
dev_loss_tmo	FC リモートポート上での問題の検出後に、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。
flush_on_last_del	yes に設定されている場合、 multipathd デーモンは、デバイスへの最後のパスが削除される際にキューを無効にします。
user_friendly_names	yes に設定されている場合、システムが /etc/multipath/bindings ファイルを使用して、マルチパスへの永続的で一意のエイリアスを mpathn の形式で割り当てるように指定します。 no に設定されている場合、システムが WWID をマルチパスのエイリアスとして使用するよう指定します。いずれの場合も、ここで指定される内容は、設定ファイルの multipaths セクションで指定するデバイス特有のエイリアスで上書きされます。デフォルト値は no です。

属性	説明
retain_attached_hw_handler	(Red Hat Enterprise Linux リリース 6.4 以降) このパラメーターが yes に設定されており、scsi 層がハードウェアハンドラーをパスデバイスにすでに接続している場合、マルチパスはデバイスに対して、 multipath.conf で指定されている hardware_handler を強制的に使用させることはありません。scsi 層がハードウェアハンドラーを接続していない場合、マルチパスは通常の方法で設定したハードウェアハンドラーを継続して使用します。
detect_prio	(Red Hat Enterprise Linux リリース 6.4 以降) これが yes に設定されている場合、マルチパスは、まずデバイスが ALUA に対応しているかチェックされます。対応している場合、 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に prioritizer を判別します。

以下の例は、マルチパス設定ファイル内の **device** エントリーを示しています。

```
# }
# 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。本章には、以下のトピックを扱ったセクションが含まれます。

- オンラインマルチパスデバイスのサイズ変更
- シングルパスデバイスからマルチパスデバイスへの root デバイスの移動
- シングルパスデバイスからマルチパスデバイスへのスワップデバイスの移動
- マルチパスデーモン
- 多数の LUN に伴う問題
- queue_if_no_path 機能での問題
- `multipath` コマンドの出力
- `multipath` コマンドを使用したマルチパスクエリー
- `multipath` コマンドのオプション
- `dmsetup` コマンドを使用したマルチパスクエリー
- `multipathd` インタラクティブコンソールでのトラブルシューティング

5.1. オンラインマルチパスデバイスのサイズ変更

オンライン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、以下の手順を使用します。

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、そのデバイスの `rescan` ファイルに 1 と書き込むと、SCSI ドライバーが、以下のコマンドにあるように再スキャンを行います。

```
# echo 1 > /sys/block/device_name/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 -k 'resize map mpatha'
```

5. ファイルシステムのサイズを変更します (LVM や 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
```

5.2. シングルパスデバイスからマルチパスデバイスへの ROOT ファイルシステムの移動

シングルパスデバイス上にシステムをインストールしている場合、root ファイルシステムに別のパスを後で追加するには、root ファイルシステムをマルチパス化されたデバイスに移動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デバイスに移動する手順を説明します。

`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後で、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し、マルチパスモジュールをロードしてから、`multipathd` の `chkconfig` を `on` に設定します。

```
# mpathconf --enable
```

`mpathconf` コマンドを使用したマルチパスの設定に関する詳細は、「[DM-Multipath のセットアップ](#)」を参照してください。

2. 「[設定ファイルのブラックリスト](#)」で説明されているように、`/etc/multipath.conf` ファイルの `blacklist` と `blacklist_exceptions` のセクションを編集します。
3. 設定ファイルが正しくセットアップされていることを確認するには、`/sbin/multipath` コマンドで `-v3` のオプションを使用して、マルチパスデーモンがご使用の root デバイス上でマルチパスデバイスの作成を試行したかどうかを確認します。root デバイスが使用中であるため、このコマンドは失敗しますが、このコマンドの出力のパス一覧に root デバイスが表示されるはずで

コマンド出力で、以下のような形式の行を探します。

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

たとえば、root ファイルシステムが `sda` またはそのパーティションの 1 つに設定されている場合、以下のような行が出力に表示されます。

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

出力の最後では、マルチパスデバイスに割り当てられた root デバイスを確認できます。

```
time | devname: ownership set to mpathdev
```

たとえば、出力は以下のように表示されます。

```
Jun 14 06:48:21 | sda: ownership set to mpatha
```

マルチパスデバイスを作成するコマンドが失敗したことを示す、以下の形式の行も確認できます。

```
time | mpathdev: domap (0) failure for create/reload map
```

上記の例では、コマンド出力に以下の行が表示されます。

```
Jun 14 06:48:21 | mpatha: domap (0) failure for create/reload map
```

4. **multipath** で **initramfs** ファイルシステムを再構築するには、以下のオプションと共に **dracut** コマンドを実行します。

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

5. ご使用の **root** 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されている場合、**fstab** ファイルを編集して、適切なマルチパスデバイス名に変更する必要がある場合があります。ご使用の **root** デバイスが LVM デバイスであるか、または UUID などでマウントされている場合、この手順は不要となります。

1. **-v3** のオプションを指定して **/sbin/multipath** コマンドを実行するステップ 3 で説明した手順を用いて、**root** デバイスの WWID を判別します。
2. **/etc/multipath.conf** ファイル内に **root** デバイス用のエイリアスを設定します。

```
multipaths {
    multipath {
        wwid WWID_of_root_device
        alias rootdev
    }
}
```

3. **/etc/fstab** を編集して、**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。

たとえば、**/etc/fstab** ファイル内に以下のようなエントリーがあった場合:

```
/dev/sda1 / ext4 defaults 1 1
```

そのエントリーを以下のように変更します。

```
/dev/mapper/rootdev / ext4 defaults 1 1
```

/etc/fstab ファイルを編集する必要がある場合は、**/etc/grub/grub.conf** ファイルを編集し、**root** パラメーターを **root=/dev/sda1** から **root=/dev/mapper/rootdev** に変更する必要もあります。

以下の例は、編集前の **grub.conf** ファイルのエントリーを示しています。

```
title Red Hat Enterprise Linux FoundationServer (2.6.32-71.24.1.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.24.1.el6.x86_64 ro root=/dev/sda1 rd_NO_LUKS
rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us
    console=ttyS0,115200n8 crashkernel=auto
    initrd /initramfs-2.6.32-71.24.1.el6.x86_64.img
```

以下の例は、編集後の **grub.conf** ファイルのエントリーを示しています。

```
title Red Hat Enterprise Linux FoundationServer (2.6.32-71.24.1.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.24.1.el6.x86_64 ro root=/dev/mapper/rootdev
rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16
    KEYTABLE=us console=ttyS0,115200n8 crashkernel=auto
    initrd /initramfs-2.6.32-71.24.1.el6.x86_64.img
```


- 6. マシンをシャットダウンします。
- 7. マシンに対して他のパスが可視となるように、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 (/) がマルチパス化されたデバイス上にあることを確認します。

5.3. シングルパスデバイスからマルチパスデバイスへのスワップファイルシステムの移動

デフォルトでは、スワップ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の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のに特別な手順は必要ありません。ただし、ご使用のスワップデバイスが LVM ボリュームではない場合で、デバイス名でマウントされる場合には、**fstab** ファイルを編集して、適切なマルチパスデバイス名に変更する必要がある場合があります。

1. **-v3** オプションを指定して **/sbin/multipath** コマンドを実行し、スワップデバイスの WWID 番号を判別します。このコマンドの出力のパス一覧には、スワップデバイスが表示されます。

コマンド出力で、スワップデバイスを示す以下のような形式の行を探します。

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

たとえば、ご使用のスワップファイルシステムが **sda** またはそのパーティションの 1 つに設定されている場合、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** ファイル内でスワップ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** を編集して、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。

たとえば、**/etc/fstab** ファイル内に以下のようなエントリーがあった場合:

```
/dev/sda2 swap                ext4    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                ext4    defaults    0 0
```

5.4. マルチパスデーモン

マルチパス設定の実装において問題があることが判明した場合は、3章DM-Multipath のセットアップに説明されているように、マルチパスデーモンが稼働していることを確認する必要があります。マルチパス化されたデバイスを使用するためには、`multipathd` デーモンが稼働していなければなりません。

5.5. 多数の LUN に伴う問題

多数の LUN がノードに追加される際に、マルチパス化されたデバイス群を使用すると `udev` デバイスマネージャーがデバイス群からデバイスノードを作成する時間が大幅に延長される可能性があります。この問題が発生した場合は、`/etc/udev/rules.d/40-multipath.rules` 内の以下の行を削除することによって修正することができます。

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

この行により、ブロックデバイスがノードに追加されるたびに `udev` デバイスマネージャーは `multipath` を実行することになります。この行が削除されている場合でも、`multipathd` デーモンは依然としてマルチパス化されたデバイスを自動的に作成し、`multipath` はマルチパス化された root ファイルシステムを伴うノードの起動プロセスで呼び出されます。唯一の変更点は、`multipathd` デーモンが稼働していない場合は、マルチパス化されたデバイスは自動的に作成されない点にあります。これは大多数のマルチパスユーザーには問題とならないはずです。

5.6. QUEUE_IF_NO_PATH 機能での問題

`features "1 queue_if_no_path"` が `/etc/multipath.conf` ファイルに指定されていると、I/O を発行するいずれのプロセスも、1 つまたは複数のパスが復元されるまでハングします。これを回避するには、`/etc/multipath.conf` ファイルに `no_path_retry N` パラメーターを設定します (ここで、*N* は、システムがパスを再試行する回数です)。

`no_path_retry` パラメーターを設定する際には、`/etc/multipath.conf` ファイルから `features "1 queue_if_no_path"` オプション削除します。ただし、多くの SAN デバイスと同様に、`features "1 queue_if_no_path"` オプションがコンパイル時のデフォルトとして設定されているマルチパス化されたデバイスを使用している場合には、`features "0"` を明示的に追加してこのデフォルトを上書きする必要があります。これを行うには、`/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.defaults` から、デバイスの既存のデバイスセクションを `/etc/multipath.conf` にコピーし、ご使用のニーズに合わせて編集することができます。

`features "1 queue_if_no_path"` オプションを使用する必要がある際に上記の問題に直面した場合は、`dmsetup` コマンドを使用して特定の LUN (すべてのパスが利用できないもの) 用のランタイム時のポリシーを編集します。たとえば、マルチパスデバイス `mpath2` 上のポリシーを `"queue_if_no_path"` から `"fail_if_no_path"` に変更したい場合は、以下のコマンドを実行します。

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

指定する必要があるのは、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。

5.7. MULTIPATH コマンドの出力

マルチパスデバイスを作成、修正、または一覧表示する際には、現在のデバイスセットアップを表示できます。その形式は以下のようになります。

各マルチパスデバイスの場合:

■

```
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
```

各パスグループの場合:

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

各パスの場合:

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

たとえば、マルチパスコマンドの出力は以下のように表示される可能性があります。

```
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
```

パスが有効で I/O の準備ができている場合は、パスのステータスは **ready** または **ghost** となります。パスが有効でない場合は、ステータスは **faulty** または **shaky** となります。このパスステータスは `/etc/multipath.conf` ファイルに定義されているポーリング間隔に従って `multipathd` デーモンによって定期的に更新されます。

dm ステータスはパスステータスと似ていますが、カーネルの観点からは異なります。dm ステータスには、2つの状態が含まれます。**faulty** に類似した **failed** と、他のすべてのパス状態をカバーする **active** です。1つのデバイスのパス状態と dm 状態は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。ステータスが **offline** の場合は、この SCSI デバイスが無効にされていることを意味します。



注記

マルチパスデバイスが作成されたり、修正されたりする時点では、パスグループのステータス、dm のデバイス名、書き込みパーミッション、および dm のステータスは不明です。また、それらの機能も常に正しいとは限りません。

5.8. MULTIPATH コマンドを使用し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して、現在のマルチパス設定を表示することができます。`-l` オプションは `sysfs` 内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と、その他のすべての使用可能な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場合、`multipath` コマンドの `-v` オプションを使用して指定できる3つの詳細レベルがあります。`-v0` を指定すると、出力なしとなります。`-v1` を指定すると、作成または更新されたマルチパス名のみが出力され、`kpartx` などの他のツールに提供することができます。`-v2` を指定すると、すべての検知されたパス、マルチパスおよびデバイスマップが表示されます。

以下の例は、`multipath -l` コマンドの出力を示しています。

```
# multipath -l
```

```
3600d023000000000e13955cc3757800 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
```

以下の例は、`multipath -ll` コマンドの出力を示しています。

```
# multipath -ll
3600d023000000000e13955cc3757801 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
3600d023000000000e13955cc3757803 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
```

5.9. MULTIPATH コマンドのオプション

表5.1「便利な `multipath` コマンドのオプション」では、便利な `multipath` コマンドのオプションをいくつか説明しています。

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、およびシステム上の他のすべての使用可能な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>	不要なすべてのマルチパスデバイスを削除します。
<code>-w device</code>	(Red Hat Enterprise Linux 6.6 以降) <code>wwids</code> ファイルから指定されたデバイスの <code>wwid</code> を削除します。
<code>-W</code>	(Red Hat Enterprise Linux 6.6 以降) 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。

5.10. DMSETUP コマンドによるデバイスマッパーエントリーの判別

`dmsetup` コマンドを使用して、マルチパス化されたデバイスに一致するデバイスマッパーエントリーを見つけることができます。

以下のコマンドは、すべてのデバイスマッパーデバイスとそれらのメジャー番号とマイナー番号を表示します。マイナー番号によって dm デバイスの名前が決定します。たとえば、マイナー番号 3 は マルチパス化されたデバイス `/dev/dm-3` に対応しています。

```
# 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)
```

5.11. MULTIPATHD インタラクティブコンソールでのトラブルシューティング

`multipathd -k` コマンドは、`multipathd` デーモンへのインタラクティブインターフェースです。このコマンドを実行すると、インタラクティブな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ます。インタラクティブなコマンドを入力するか、または `CTRL-D` を入力して終了できます。

`multipathd` インタラクティブコンソールは、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に使用することができます。たとえば、以下のコマンドのシーケンスは、デフォルトを含む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
```

以下のコマンドのシーケンスは、マルチパスが `multipath.conf` への変更を取得していることを確認します。

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

以下のコマンドのシーケンスを使用して、パスチェッカーが正しく機能することを確認します。

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

付録A 改訂履歴

改訂 7.0-6.2 翻訳および校閲完了	Wed May 6 2015	Aiko Sasaki
改訂 7.0-6.1 翻訳ファイルを XML ソースバージョン 7.0-6 と同期	Wed May 6 2015	Aiko Sasaki
改訂 7.0-6 RHEL 6 スプラッシュページを更新し、 <code>sort_order</code> を実装	Tue Dec 16 2014	Steven Levine
改訂 7.0-4 6.6 GA リリース向けバージョン	Thu Aug 7 2014	Steven Levine
改訂 7.0-3 6.6 ベータリリース向けバージョン	Thu Aug 7 2014	Steven Levine
改訂 7.0-1 バグを修正: #1117038 multipath コマンドの新規の <code>-w</code> と <code>-W</code> オプションについて文書化。 バグを修正: #1116946 multipath.conf ファイルの features エントリーのすべての使用可能な値について文書化。 バグを修正: #969080 若干の編集による明確化。 バグを修正: #1081264 force_sync パラメーターについて文書化。	Mon Jul 7 2014	Steven Levine
改訂 6.0-7 レビューサーバーベータビルドを修正するための事務的な更新	Wed Apr 10 2014	Steven Levine
改訂 6.0-6 6.5 GA リリース向けのバージョン	Wed Nov 13 2013	Steven Levine
改訂 6.0-5 6.5 ベータリリース向けバージョン	Fri Sep 27 2013	Steven Levine
改訂 6.0-3 バグを修正: #987144、#987671 multipath.conf ファイルの新パラメーター <code>reload_rewrite</code> および <code>replace_wwid_whitespace</code> を文書化。 バグを修正: #911630、#969022 若干の誤字/誤植を修正	Fri Sep 27 2013	Steven Levine
改訂 6.0-1 RHEL 6.5 の新しいデフォルトパラメーターを追加。	Tue Sep 10 2013	Steven Levine
改訂 5.0-9 6.4 GA リリース向けバージョン	Mon Feb 18 2013	Steven Levine
改訂 5.0-7 6.4 ベータリリース向けバージョン	Mon Nov 26 2012	Steven Levine
改訂 5.0-6	Tue Nov 13 2012	Steven Levine

バグを修正: #856829

新しい **retain_attached_hardware_handler** と **detect_prio** 設定パラメーターについて文書化。

バグを修正: #815925

initramfs ファイルシステムでのマルチパス化のセットアップ手順を文書化。

バグを修正: #836333

誤字/誤植を修正。

バグを修正: #856826

path_selector パラメーターの更新を反映するために、**multipath.conf** 抜粋例を更新。

改訂 4.0-3	Fri Jun 15 2012	Steven Levine
6.3 GA リリース向けバージョン		
改訂 4.0-2	Wed Apr 25 2012	Steven Levine
バグを修正: #815925		
dmsetup ls の例の出力形式を更新。		
改訂 4.0-1	Wed Mar 28 2012	Steven Levine
バグを修正: #786399		
若干の誤字/誤植を修正。		
バグを修正: #771420		
Red Hat Enterprise Linux 6.3 の DM-Multipath 新機能を文書化。		
改訂 3.0-3	Thu Dec 1 2011	Steven Levine
Red Hat Enterprise Linux 6.2 の GA リリース		
バグを修正: #753899		
multipath -F オプションの説明を修正。		
改訂 3.0-2	Fri Oct 7 2011	Steven Levine
バグを修正: #743767		
誤字の修正と若干の問題を明確化。		
改訂 3.0-1	Mon Sep 19 2011	Steven Levine
Red Hat Enterprise Linux 6.2 Beta リリース向けの初回バージョン。		
バグを修正: #707638		
Red Hat Enterprise Linux 6.2 の DM-Multipath 新機能を文書化。		
バグを修正: #715457		
SCSI デバイスのフィルター例を修正。		
バグを修正: #623450		
root デバイスをマルチパスボリュームに移動する手順を更新。		
バグを修正: #725374、#738051		
若干の誤字/誤植を修正。		
改訂 2.0-1	Thu May 19 2011	Steven Levine

Red Hat Enterprise Linux 6.1 向けの初回改訂。

バグを修正: #623450

root デバイスおよびスワップデバイスをシングルパスデバイスからマルチパスデバイスに移動する新たな手順を追加。

バグを修正: #693948

デバイス属性表の若干のエラーを修正。

バグを修正: #694683

若干の誤字を修正。

バグを修正: #702721

旧版の /dev/mpath への言及を削除。

改訂 1.0-1

Wed Nov 10 2010

Steven Levine

Red Hat Enterprise Linux 6 リリースの初版。

索引

シンボル

/etc/multipath.conf パッケージ, [DM-Multipath のセットアップ](#)

アクティブ/アクティブ設定

図, [DM-Multipath の概要](#)

定義, [DM-Multipath の概要](#)

アクティブ/パッシブ設定

図, [DM-Multipath の概要](#)

定義, [DM-Multipath の概要](#)

ストレージアレイ

追加, [ストレージデバイスの設定](#)

ストレージアレイの

追加, [設定ファイルデバイス](#)

ストレージアレイサポート, [ストレージアレイサポート](#)

スワップファイルシステム, [シングルパスデバイスからマルチパスデバイスへのスワップファイルシステムの移動](#)

デバイスの

追加, [ストレージデバイスの設定](#), [設定ファイルデバイス](#)

デバイス名, [マルチパスデバイス識別子](#)

フィードバック

本マニュアルに関するお問い合わせ先情報, [フィードバック](#)

フェイルオーバー, [DM-Multipath の概要](#)

ブラックリスト

WWID, [WWID 単位のブラックリスト](#)

デバイスタイプ, [デバイスタイプ単位のブラックリスト](#)

デバイス名, [デバイス名単位のブラックリスト](#)

デフォルトデバイス, [デバイス名単位のブラックリスト](#)

設定ファイル, [設定ファイルのブラックリスト](#)

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](#)

マルチパス化された root ファイルシステム, [シングルパスデバイスからマルチパスデバイスへの root ファイルシステムの移動](#)

マルチパス化されたスワップファイルシステム, [シングルパスデバイスからマルチパスデバイスへのスワップファイルシステムの移動](#)

ローカルディスクを無視する, [マルチパスデバイスの生成時にローカルディスクを無視する](#)

[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](#) のセットアップ

設定ファイル

[alias](#) パラメーター, [マルチパスデバイス設定属性](#)

[checker_timeout](#) パラメーター, [設定ファイルのデフォルト](#)

[dev_loss_tmo](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[failback](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[fast_io_fail_tmo](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[features](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[flush_on_last_del](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[getuid_callout](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[hardware_handler](#) パラメーター, [設定ファイルデバイス](#)

[hwtable_regex_match](#) パラメーター, [設定ファイルのデフォルト](#)

[max_fds](#) パラメーター, [設定ファイルのデフォルト](#)

[no_path_retry](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[path_checker](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[path_grouping_policy](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[path_selector](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[polling-interval](#) パラメーター, [設定ファイルのデフォルト](#)

[prio](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[product](#) パラメーター, [設定ファイルデバイス](#)

[product_blacklist](#) パラメーター, [設定ファイルデバイス](#)

[queue_without_daemon](#) パラメーター, [設定ファイルのデフォルト](#)

[revision](#) パラメーター, [設定ファイルデバイス](#)

[rr_min_io](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#)

[rr_weight](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[udev_dir](#) パラメーター, [設定ファイルのデフォルト](#)

[user_friendly_names](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[vendor](#) パラメーター, [設定ファイルデバイス](#)

[verbosity](#) パラメーター, [設定ファイルのデフォルト](#)

[wwid](#) パラメーター, [マルチパスデバイス設定属性](#)

ブラックリスト, [設定ファイルのブラックリスト](#)
概要, [設定ファイルの概要](#)

A

alias パラメーター, [マルチパスデバイス設定属性](#)
設定ファイル, [マルチパスデバイス識別子](#)

B

blacklist_exceptions セクション
multipath.conf ファイル, [ブラックリスト例外](#)

C

checker_timeout パラメーター, [設定ファイルのデフォルト](#)
configuration file
detect_prio パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#)
force_sync パラメーター, [設定ファイルのデフォルト](#)
retain_attached_hw_handler パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#)

D

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

detect_prio パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#)
dev/mapper directory, [マルチパスデバイス識別子](#)
device-mapper-multipath パッケージ, [DM-Multipath のセットアップ](#)
devices セクション
multipath.conf ファイル, [設定ファイルデバイス](#)

dev_loss_tmo パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

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 コンポーネント](#)
セットアップ, [DM-Multipath のセットアップ](#)
セットアップの概要, [DM-Multipath セットアップの概要](#)
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, [デバイスマッパーのマルチパス化](#)
概要, [DM-Multipath の概要](#)

設定, [DM-Multipath のセットアップ](#)

設定ファイル, [DM-Multipath 設定ファイル](#)

[dm-n devices](#), [マルチパスデバイス識別子](#)

[dmsetup](#) コマンドによるデバイスマッパーエントリーの判別, [dmsetup](#) コマンドによるデバイスマッパーエントリーの判別

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

F

[failback](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[fast_io_fail_tmo](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[features](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[flush_on_last_del](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[force_sync](#) パラメーター, [設定ファイルのデフォルト](#)

G

[getuid_callout](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

H

[hardware_handler](#) パラメーター, [設定ファイルデバイス](#)

[hwtable_regex_match](#) パラメーター, [設定ファイルのデフォルト](#)

I

[initramfs](#)

マルチパスの起動, [initramfs](#) ファイルシステムでのマルチパス機能のセットアップ

K

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

L

[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, [設定ファイルのデフォルト](#)

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

[multipath](#) コマンド, [DM-Multipath 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](#) コマンドを使用し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, [ブラックリスト例外](#)

[multipath.conf](#) ファイル, [ストレージレイサポート](#), [DM-Multipath 設定ファイル](#)

[defaults](#) セクション, [設定ファイルのデフォルト](#)

[devices](#) セクション, [設定ファイルデバイス](#)

[multipaths](#) セクション, [マルチパスデバイス設定属性](#)

[multipath.conf.annotated](#) ファイル, [DM-Multipath 設定ファイル](#)

[multipath.conf.defaults](#) ファイル, [ストレージレイサポート](#), [DM-Multipath 設定ファイル](#)

[multipathd](#)

[インタラクティブコンソール](#), [multipathd インタラクティブコンソールでのトラブルシューティング](#)

コマンド, [multipathd インタラクティブコンソールでのトラブルシューティング](#)

[multipathd start](#) コマンド, [DM-Multipath のセットアップ](#)

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

[multipaths](#) セクション

[multipath.conf](#) ファイル, [マルチパスデバイス設定属性](#)

N

[no_path_retry](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

P

[path_checker](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[path_grouping_policy](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[path_selector](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

[polling_interval](#) パラメーター, [設定ファイルのデフォルト](#)

[prio](#) パラメーター, [設定ファイルのデフォルト](#), [設定ファイルデバイス](#)

[product](#) パラメーター, [設定ファイルデバイス](#)

[product_blacklist](#) パラメーター, [設定ファイルデバイス](#)

Q

[queue_without_daemon](#) パラメーター, [設定ファイルのデフォルト](#)

R

[retain_attached_hw_handler](#) パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#)

[revision](#) パラメーター, [設定ファイルデバイス](#)

[root](#) ファイルシステム, [シングルパスデバイスからマルチパスデバイスへの root ファイルシステムの移動](#)

rr_min_io パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#)

rr_weight パラメーター, [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

U

udev_dir パラメーター, [設定ファイルのデフォルト](#)

user_friendly_names パラメーター, [マルチパスデバイス識別子](#), [設定ファイルのデフォルト](#), [マルチパスデバイス設定属性](#), [設定ファイルデバイス](#)

V

vendor パラメーター, [設定ファイルデバイス](#)

verbosity パラメーター, [設定ファイルのデフォルト](#)

W

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

wwid パラメーター, [マルチパスデバイス設定属性](#)