



# Red Hat Enterprise Linux 8

## Device Mapper Multipath の設定

Device Mapper Multipath 機能の使用



# Red Hat Enterprise Linux 8 Device Mapper Multipath の設定

---

Device Mapper Multipath 機能の使用

## 法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

The OpenStack<sup>®</sup> 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

本書は、Red Hat Enterprise Linux 8 で Device Mapper Multipath (DM-Multipath) 機能を設定および管理する方法を説明します。

## 目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
<b>第1章 DEVICE MAPPER MULTIPATH の概要</b>	<b>4</b>
1.1. さまざまな DM MULTIPATH 構成	4
1.2. DM MULTIPATH コンポーネント	7
<b>第2章 マルチパスデバイス</b>	<b>9</b>
2.1. マルチパスデバイス識別子	9
2.2. 論理ボリューム内のマルチパスデバイス	9
<b>第3章 DM MULTIPATH の設定</b>	<b>11</b>
3.1. DM-MULTIPATH の基本設定	11
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	12
3.3. 追加のストレージデバイスの設定	13
3.4. INITRAMFS ファイルシステムでマルチパスの設定	14
<b>第4章 DM MULTIPATH 設定ファイルの編集</b>	<b>15</b>
4.1. 設定ファイルの概要	15
4.2. DM MULTIPATH がデバイスタイムアウトの上書き	16
4.3. DM MULTIPATH のデバイスのブラックリスト	16
4.4. マルチパス設定ファイルのデフォルトの編集	19
4.5. 特定デバイスのマルチパス設定の編集	20
4.6. ストレージコントローラーのマルチパス設定の編集	21
4.7. すべてのデバイスへのマルチパス値の設定	22
<b>第5章 マルチパス化されたボリュームの管理</b>	<b>23</b>
5.1. MULTIPATH コマンド	23
5.2. オンラインのマルチパスデバイスのサイズ変更	25
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動	26
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動	27
5.5. DMSETUP コマンドでデバイスマッパーエントリーの特定	28
5.6. MULTIPATHD デーモンの管理	28
5.7. 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	29
<b>第6章 DM MULTIPATH のトラブルシューティング</b>	<b>30</b>
6.1. DM MULTIPATH トラブルシューティングチェックリスト	30
6.2. MULTIPATHD 対話式コンソールでトラブルシューティング	30
<b>第7章 EH_DEADLINE を使用したストレージエラーからの回復における最大時間の設定</b>	<b>32</b>
7.1. EH_DEADLINE パラメーター	32
7.2. EH_DEADLINE パラメーターの設定	33



## RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。改善点を報告する場合は、以下のように行います。

- 特定の文章に簡単なコメントを記入する場合は、以下の手順を行います。
  1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上端に **Feedback** ボタンがあることを確認してください。
  2. マウスカーソルで、コメントを追加する部分を強調表示します。
  3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
  4. 表示される手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
  1. [Bugzilla](#) の Web サイトにアクセスします。
  2. Component で **Documentation** を選択します。
  3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
  4. **Submit Bug** をクリックします。

## 第1章 DEVICE MAPPER MULTIPATH の概要

Device Mapper のマルチパス (DM Multipath) を使用すると、サーバーノードとストレージレイとの間の複数の I/O パスを 1 つのデバイスに設定できます。これらの I/O パスは、個別のケーブル、スイッチ、コントローラーを含むことができる物理的なストレージエリアネットワーク (SAN) 接続です。マルチパスは I/O パスを集約し、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。

DM Multipath は以下を提供します。

- 冗長性  
DM Multipath は、アクティブ/パッシブ構成でフェイルオーバーを提供できます。アクティブ/パッシブ構成では、パスは、I/O には常に半分しか使用されません。I/O パスの要素 (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えます。
- パフォーマンスの向上  
DM Multipath は、アクティブ/アクティブモードで構成できます。このモードでは、I/O はラウンドロビン方式でパスに分散されます。一部の構成では、DM Multipath は I/O パスの負荷を検出し、負荷を動的に再調整できます。

### 1.1. さまざまな DM MULTIPATH 構成

次に、DM Multipath の構成例をいくつか示します。

- [「1つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成」](#)
- [「2つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成」](#)
- [「1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成」](#)

#### 1.1.1.1 1つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成

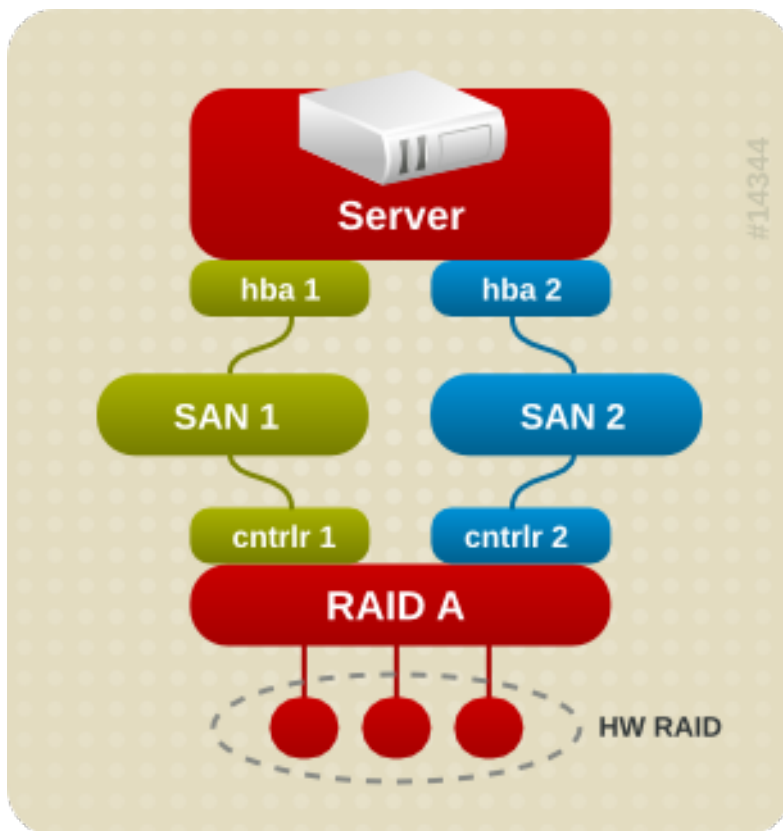
この構成では、サーバー上に 2 つのホストバスアダプター (HBA)、2 つの SAN スイッチ、および 2 つの RAID コントローラーがあります。この構成では、次のような障害が発生する可能性があります。

- HBA の障害
- ファイバーチャネルケーブルの障害
- 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 が構成されている場合は、これらのポイントのいずれかで障害が発生すると、DM Multipath は代替 I/O パスに切り替わります。図 1.1 [「1つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成」](#) では、サーバーから RAID デバイスへの 2 つの I/O パスを使用した構成を説明します。ここでは、**hba1**、**SAN1**、および **cntrlr1** を通る 1 つの I/O パスと、**hba2**、**SAN2**、および **cntrlr2** を通る別の I/O パスがあります。



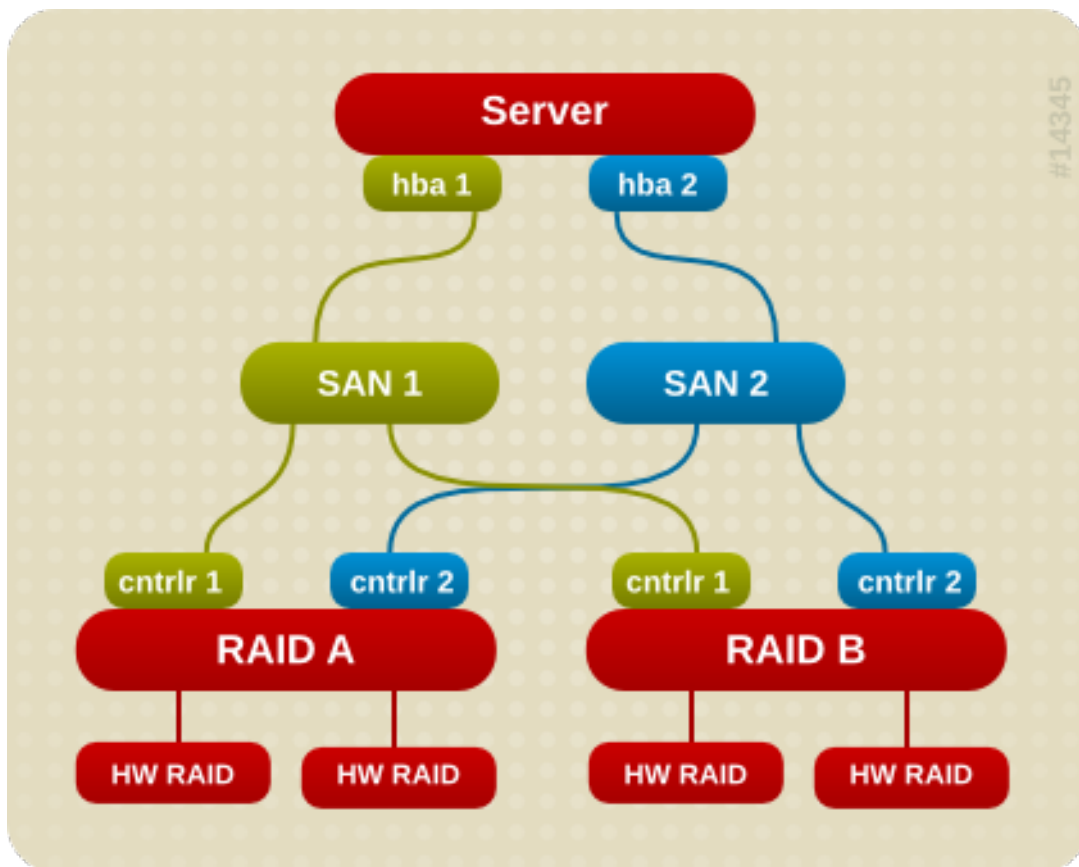
図1.11 つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成



### 1.1.2. 2つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成

この構成では、サーバー上に2つのHBA、2つのSANスイッチ、およびそれぞれ2つのRAIDコントローラーを備えた2つのRAIDデバイスがあります。DM Multipathが構成されている場合、いずれかの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すると、DM Multipathはそのデバイスの代替I/Oパスに切り替わります。図1.2「2つのRAIDデバイスを使用したアクティブ/パッシブのマルチパス構成」では、各RAIDデバイスへの2つのI/Oパスを使用した構成を説明します。ここでは、各RAIDデバイスへの2つのI/Oパスがあります。

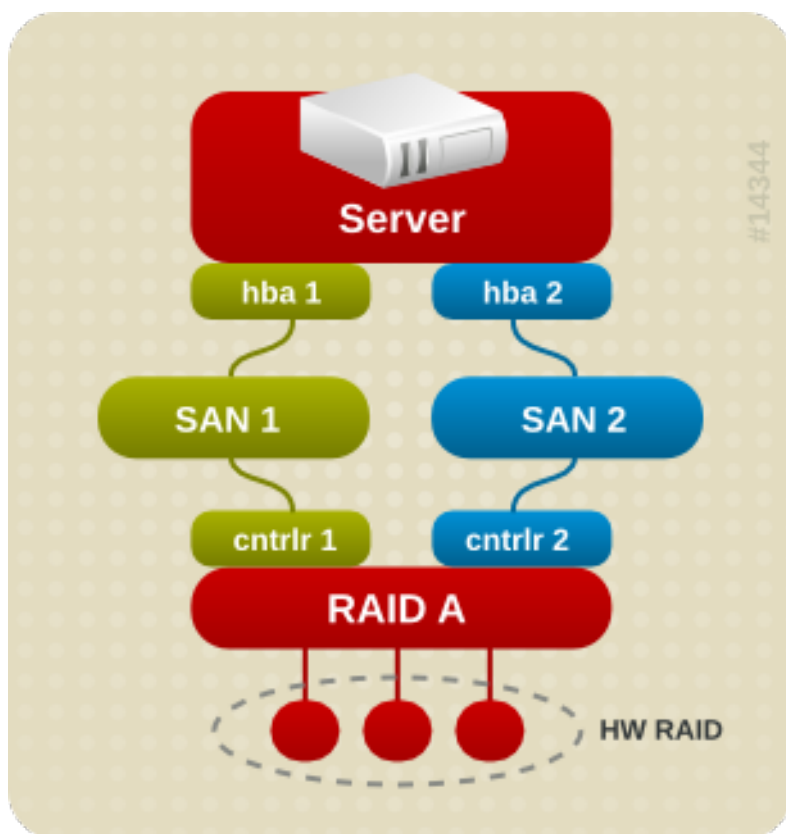
図1.2 2つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成



### 1.1.3.1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成

この構成では、サーバー上に HBA が 2 つ、SAN スイッチが 2 つ、および RAID コントローラーが 2 つあります。図1.3「1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成」では、サーバーからストレージデバイスへの 2 つの I/O パスを使用した構成を説明します。ここで、I/O は、これら 2 つのパスに分散できます。

図1.3 1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成



## 1.2. DM MULTIPATH コンポーネント

表1.1 「DM Multipath のコンポーネント」では、DM Multipath 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
<b>dm_multipath</b> カーネルモジュール	I/O を再ルーティングし、パスとパスグループのフェールオーバーに対応します。
<b>mpathconf</b> ユーティリティ	デバイスマッパーマルチパスを構成して有効にします。
<b>multipath</b> コマンド	マルチパスデバイスを一覧表示して構成します。これは、ブロックデバイスが追加されるたびに <b>udev</b> により実行され、デバイスがマルチパスデバイスの一部であるかどうかを判断します。
<b>multipathd</b> デモン	マルチパスデバイスを自動的に作成および削除し、パスを監視します。パスが失敗して戻ってくると、マルチパスデバイスが更新される場合があります。マルチパスデバイスへのインタラクティブな変更を許可します。 <b>/etc/multipath.conf</b> ファイルに変更がある場合は、サービスを再読み込みします。

<b>kpartx</b> コマンド	デバイス上のパーティションのデバイスマッパーデバイスを作成します。このコマンドは、マルチパスデバイスが作成され、その上にパーティションデバイスが作成されると、 <b>udev</b> により自動的に実行されます。 <b>kpartx</b> コマンドは独自のパッケージで提供されますが、 <b>device-mapper-multipath</b> パッケージはそれに依存しています。
<b>mpathpersist</b>	マルチパスデバイスに <b>SCSI-3</b> 永続予約を設定します。このコマンドは、 <b>sg_persist</b> が SCSI デバイスに対して行う方法と似ていますが、マルチパスデバイスのすべてのパスで永続的な予約の設定を処理します。 <b>multipathd</b> と調整して、後で追加されるパスに予約が正しく設定されるようにします。この機能を使用するには、 <b>reservation_key</b> 属性を <b>/etc/multipath.conf</b> ファイルで定義する必要があります。定義しないと、 <b>multipathd</b> デーモンは、新しく検出されたパスまたは復元されたパスの永続的な予約を確認しません。

#### 関連情報

- [man ページの multipath](#)
- [man ページの multipathd](#)
- [/etc/multipath.conf ファイル](#)

## 第2章 マルチパスデバイス

DM Multipath を使用しない場合は、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している場合でも、システムは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を個別のデバイスとして扱います。DM Multipath は、基礎となるデバイスの上に1つの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

新しいデバイスが DM Multipath の制御下にある場合に、これらのデバイスは `/dev/mapper/` ディレクトリーおよび `/dev/` ディレクトリーに作成されます。`/dev/dm-X` という形式のデバイスは内部使用専用であるため、管理者が直接使用するものではありません。

マルチパスデバイス名:

- **user\_friendly\_names** 構成オプションが **no** に設定されている場合は、マルチパスデバイスの名前が World Wide Identifier (WWID) に設定されます。デフォルトでは、マルチパスデバイスの名前は WWID に設定されます。デバイス名は `/dev/mapper/WWID` になります。また、`/dev/dm-X` という名前の `/dev/` ディレクトリーにも作成されます。
- または、`/etc/multipath.conf` ファイルで、**user\_friendly\_names** オプションを **yes** に設定できます。これにより、**multipath** セクションの **alias** が、**mpathN** 形式のノード固有の名前に設定されます。デバイス名は、`/dev/mapper/mpathN` および `/dev/dm-X` になります。ただし、マルチパスデバイスを使用するすべてのノードでデバイス名が同じであるとは限りません。同様に、`/etc/multipath.conf` ファイルで **alias** オプションを設定した場合は、クラスター内のすべてのノードで自動的に名前が一致しません。

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、これにより問題が発生することはありません。すべてのノードでマルチパスデバイス名の一貫性を保つために、Red Hat は、**user\_friendly\_names** オプションを無効にすることが推奨されます。

たとえば、ゾーンに分けられていない1つの FC スイッチにより、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 を持つノードは、`/dev/sda`、`/dev/sdb`、`/dev/sdc`、および `/dev/sdd` の4つのデバイスを認識します。DM Multipath は、マルチパス構成に従って、I/O を基本となるこれらの4つのデバイスにルーティングしなおす一意の WWID を持つシングルデバイスを作成します。

**user\_friendly\_names** オプションおよび **alias** オプションの他に、マルチパスデバイスには他の属性もあります。`/etc/multipath.conf` ファイルの **multipaths** セクションに、デバイスのエントリーを作成することにより、特定のマルチパスデバイスのこれらの属性を変更できます。

#### 関連情報

- man ページの **multipath**
- man ページの **multipath.conf**
- `/etc/multipath.conf` ファイル
- [「DM Multipath コンポーネント」](#)

### 2.2. 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、LVM 物理ボリュームを作成するときに物理デバイス名を使用すると同じように、マルチパスデバイス名を使用できます。たとえば、`/dev/mapper/mpatha` がマルチパスデバイス名である場合、`pvcreate /dev/mapper/mpatha` コマンドは、`/dev/mapper/mpatha` を

物理ボリュームとしてマークします。

他の LVM 物理デバイスを使用するのと同じように、LVM ボリュームグループを作成するときに、作成された LVM 物理デバイスを使用できます。



### 注記

パーティションを設定したデバイス全体に 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。Anaconda および Kickstart の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特に指定しない限り、空のパーティションテー  
ブルを作成します。パーティションを作成する代わりにデバイス全体を使用する場合は、  
デバイスから既存のパーティションを削除します。**kpartx -d** デバイスコマンドと **fdisk**  
ユーティリティーを使用して、既存のパーティションを削除できます。システムに 2Tb  
を超えるブロックデバイスがある場合は、**parted** ユーティリティーを使用してパー  
ティションを削除します。

アクティブ/パッシブ マルチパスアレイを基礎となる物理デバイスとして使用する LVM 論理ボリューム  
を作成する場合は、必要に応じて、`/etc/lvm/lvm.conf` ファイルにフィルターを追加して、マルチパス  
デバイスに下線を引くディスクを除外できます。これは、I/O の受信時にアレイがアクティブパスを  
パッシブパスに自動的に変更する場合に、このようなデバイスにフィルターが設定されていないと、  
LVM がパッシブパスをスキャンするたびにマルチパスがフェールオーバーおよびフェールバックする  
ためです。パッシブパスをアクティブにするコマンドを必要とする **アクティブ/パッシブ** アレイでこの  
問題が発生すると、LVM が警告メッセージを出力します。

`/etc/lvm/lvm.conf` ファイル内のすべての **sd** デバイスをフィルタリングするには、**filter = [ "r/block/",  
"r/disk/", "r/sd./", "a./" ]** を、そのファイルを **devices** セクションに追加します。

### 関連情報

- man ページの **lvm.conf**
- [「DM Multipath コンポーネント」](#)

## 第3章 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。

### 3.1. DM-MULTIPATH の基本設定

**mpathconf** ユーティリティーを使用して DM Multipath 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** ファイルが存在する場合は、**mpathconf** ユーティリティーで編集します。
- **/etc/multipath.conf** ファイルが存在しない場合は、**mpathconf** ユーティリティーにより **/etc/multipath.conf** ファイルが新たに作成されます。

**mpathconf** ユーティリティーの詳細は、man ページの **mpathconf(8)** を参照してください。

**/etc/multipath.conf** ファイルを編集する必要がない場合は、以下の **mpathconf** コマンドを実行すると、基本的なフェイルオーバー設定の DM Multipath をセットアップできます。このコマンドでマルチパス設定ファイルが有効になり、**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** コマンドに必要な追加オプションの詳細は、man ページの **mpathconf(8)** を参照するか、**--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 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>
```

2. 必要に応じて **/etc/multipath.conf** ファイルを編集します。DM Multipath 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。

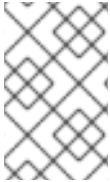
**path\_grouping\_policy** のデフォルト値は **failover** に設定されているため、この例では **/etc/multipath.conf** ファイルを編集する必要はありません。

設定ファイル内で初めてのデフォルトセクションはシステムを設定するため、マルチパスデバイスの名前の形式は **/dev/mapper/mpathn** になります。この設定がないと、デバイスの WWID がマルチパスデバイスの名前になります。ユーザーフレンドリー名を使用しない場合は、次のコマンドを実行します。

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

- 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。

```
# systemctl start multipathd.service
```



### 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る場合は、**systemctl reload multipathd.service** コマンドを実行して変更を反映する必要があります。

## 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

ローカルの SCSI カードが内蔵ディスクに搭載されているマシンがあります。DM Multipath をこのようなデバイスで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 **on** に設定すると、このような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** 設定パラメーターと、このパラメーターに設定する値の意味に関する詳細は、man ページの **multipath.conf(5)** を参照してください。

**find\_multipaths** 設定パラメーターを **on** 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編集すると、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。

この例では、**multipath** コマンドで **-d** オプションを指定し、マルチパスデバイスを作成しないドライ実行であることを示しています。

```
# multipath -v2 -d
: 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 [-----]
   `--

: 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

: 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

: 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

: 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、マルチパスマップで `/dev/sda` をマッピングしないようにするには、`/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できますが、`/dev/sda` が再起動後に同じ `/dev/sda` になる保証はないため安全な方法ではありません。個々のデバイスを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` コマンドの出力を見ると、`/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追加します。

```

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に、手動で指示する必要があります。次のコマンドで、更新した `/etc/multipath.conf` ファイルを再読み込みします。

```
# systemctl reload multipathd.service
```

### 3.3. 追加の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アレイで最も一般的なものは、DM-Multipath 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は、次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
```

デフォルトでは対応していないストレージデバイスを既知のマルチパスデバイスとして追加する必要がある場合は、`/etc/multipath.conf` ファイルを編集し、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にデバイスを設定します。

```

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

```

### 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できます。マルチパスの設定後、次のオプションを指定して **dracut** コマンドを実行すると、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
```

**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の設定ファイルを変更した場合は、**initramfs** ファイルシステムを再構築して変更を有効にする必要があります。

## 第4章 DM MULTIPATH 設定ファイルの編集

DM Multipath では、マルチパスで最も一般的に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 に対応する最も一般的なストレージレイが DM Multipath に含まれています。対応しているデバイスなどのデフォルトの設定値は、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 のデフォルトの設定値は、`/etc/multipath.conf` 設定ファイルを編集するとオーバーライドできます。必要な場合は、デフォルトでは対応し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行うことができます。**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の設定ファイルを変更した場合は、**initramfs** ファイルシステムを再構築して変更を有効にする必要があります。

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ファイルのセクションや、デフォルト値を無効にする必要がないファイルのセクションでは、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できます。

設定ファイルの詳細は、man ページの **multipath.conf(5)** を参照してください。

### 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

この例外セクションに指定されていなければ、**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るマルチパスの候補になる一覧。

#### 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 **overrides**、**devices**、および **defaults** 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。デフォルトでは対応していないストレージレイを使用している場合は、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。

#### overrides

すべてのデバイスに適用される設定。ここで指定する値は、設定ファイルの **devices** セクションおよび **defaults** セクションで指定されている値より優先されます。

システムがマルチパスデバイスの属性を決定するとき、まずマルチパス設定をチェックし、次にデバイス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確認します。

## 4.2. DM MULTIPATH がデバイスタイムアウトの上書き

**recovery\_tmo sysfs** オプションは、特定の iSCSI デバイスのタイムアウトを制御します。次のオプションは、システム全体の **recovery\_tmo** 値を上書きします。

- **replacement\_timeout** 設定オプションは、システム全体で全 iSCSI デバイスの **recovery\_tmo** 値を上書きします。
- DM Multipath が管理するすべての iSCSI デバイスで、DM Multipath の **fast\_io\_fail\_tmo** オプションは、システム全体の **recovery\_tmo** 値を上書きします。  
DM Multipath の **fast\_io\_fail\_tmo** オプションは、ファイバーチャネルデバイスの **fast\_io\_fail\_tmo** オプションを上書きします。

DM Multipath の **fast\_io\_fail\_tmo** オプションは **replacement\_timeout** よりも優先します。Red Hat では、**replacement\_timeout** を使用して、DM Multipath が管理するデバイスの **recovery\_tmo** を上書きすることは推奨しません。これは、**multipathd** サービスが再読み込みを行うと、DM Multipath が常に **recovery\_tmo** をリセットするためです。

## 4.3. DM MULTIPATH のデバイスのブラックリスト

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。

**find\_multipaths** 設定パラメーターを **off** に設定すると、マルチパスは明示的にブラックリストに指定していないパスに対して常にマルチパスデバイスを作成しようとします。**find\_multipaths** 設定パラメーターを **on** に設定すると、以下の 3 つの条件のいずれかが満たされた場合に限り、マルチパスによりデバイスが作成されます。

- 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。
- ユーザーが **multipath** コマンドでデバイスを指定して、デバイスの作成を手動で強制する場合。
- 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、同一の WWID を持つパスが検出されると、そのデバイスが自動的に再作成されます。これにより、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せずにマルチパスデバイスを作成してから、パラメーターを **on** に設定した場合は、**/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。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:
/3600d0230000000000e13955cc3757802/
```

```

/3600d0230000000000e13955cc3757801/
/3600d0230000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372    0E13955CC3757802/
/3600d0230000000000e13955cc3757803/

```

`find_multipaths` には、`on` と `off` 以外に以下の値を設定することもできます。

- **strict** - マルチパスは、マルチパス化されていなかったパスを許可しないため、そのパスは `/etc/multipath/wwids` ファイルに記載されていません。
- **smart** - マルチパスは常に `udev` のブラックリストに指定されていないデバイスを許可しますが、`find_multipaths_timeout` パラメーターで設定されたタイムアウト期間内に `multipathd` がデバイスを作成しないとそのクレームをデバイス上で解放します。`find_multipaths_timeout` パラメーターの詳細は、man ページの `multipath.conf(5)` を参照してください。

`find_multipaths` の組み込みデフォルト値は `off` です。ただし、`mpathconf` により作成されたデフォルトの `multipath.conf` ファイルは、`find_multipaths` の値を `on` に設定します。

`find_multipaths` に設定できる値の詳細は、man ページの `multipath.conf(5)` を参照してください。

`find_multipaths` パラメーターを `on` に設定した場合は、マルチパス化しない複数のパスを持つデバイスのみをブラックリストに指定する必要があります。そのため、通常はデバイスをブラックリストに指定する必要がありません。

デバイスをブラックリストに指定する必要がある場合は、WWID、デバイス名、デバイスタイプ、プロパティ、およびプロトコルでデバイスをブラックリストに指定できます。各デバイスで、この5つの基準は「property、devnode、device、protocol、wwid」の順に評価されます。いずれかの基準によりデバイスがブラックリストに指定されていることが判明した場合は、`multipathd` により処理から除外され、それ以降の基準は評価されません。各基準において、デバイスがホワイトリストとブラックリストの両方に一致した場合は、ホワイトリストが優先されます。

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](#)」を参照してください。

#### 4.3.1. 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ます。

```

blacklist {
    wwid 26353900f02796769
}

```

#### 4.3.2. 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。たとえば、システムが再起動するとデバイス名が **/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** エントリーがデフォルトのブラックリストにコンパイルされています。このエントリーによりブラックリストに指定されるデバイスは、通常 DM Multipath をサポートしません。このようなデバイスでマルチパスを有効にするには、「[ブラックリストの例外](#)」の説明に従って、設定ファイルの **blacklist\_exceptions** セクションにデバイス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|ha)d[a-z]"
}
```

#### 4.3.3. 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** セクションに特定のデバイスタイプを指定するには、**device** セクションを使用します。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 ".*"
    }
}
```

#### 4.3.4. udev プロパティでブラックリストの指定

**multipath.conf** 設定ファイルの **blacklist** セクションおよび **blacklist\_exceptions** セクションは **property** パラメーターに対応します。このパラメーターを使用すると、ユーザーは特定タイプのデバイスをブラックリストに指定できます。**property** パラメーターは、デバイスの **udev** 環境変数名と照合される正規表現の文字列を取ります。

以下の例は、udev プロパティ **ID\_ATA** を持つすべてのデバイスをブラックリストに指定します。

```
blacklist {
    property "ID_ATA"
}
```

#### 4.3.5. デバイスプロトコルでブラックリストの指定

**protocol** セクションのある設定ファイルの **blacklist** セクションに、マルチパス化から除外されるデバ

イスのプロトコルを指定できます。マルチパスが認識するプロトコル文字列は `scsi:fc`、`scsi:spi`、`scsi:ssa`、`scsi:sbp`、`scsi:srp`、`scsi:iscsi`、`scsi:sas`、`scsi:adt`、`scsi:ata`、`scsi:unspec`、`ccw`、`cciss`、`nvme`、および `undef` です。パスが使用しているプロトコルを表示するには、**`multipathd show paths format "%d %P"`** コマンドを実行します。

以下の例は、未定義のプロトコルまたは未知の SCSI トランスポートタイプを持つすべてのデバイスをブラックリストに指定します。

```
blacklist {
    protocol "scsi:unspec"
    protocol "undef"
}
```

#### 4.3.6. ブラックリストの例外

設定ファイルの **`blacklist_exceptions`** セクションを使用すると、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。

たとえば、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する場合は、有効にする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します。これにより、必要なデバイスにのみマルチパスを許可できます。

```
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。たとえば、`blacklist` で **`devnode`** エントリーを使用し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。

**`property`** パラメーターの挙動は、他の **`blacklist_exception`** パラメーターとは異なります。このパラメーターを設定した場合は、一致する **`udev`** 変数がデバイスに必要なになります。この変数がないと、デバイスはブラックリストに指定されます。このパラメーターを使用すると、USB スティックやローカルハードドライブなど、マルチパスが無視する必要のある SCSI デバイスをブラックリストに登録できます。合理的にマルチパス化できる SCSI デバイスのみを許可するには、以下の例のようにこのパラメーターを **`SCSI_IDENT_ID_WWN`** に設定します。

```
blacklist_exceptions {
    property "(SCSI_IDENT_ID_WWN)"
}
```

## 4.4. マルチパス設定ファイルのデフォルトの編集

**`/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`** に設定できます。

```
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。たとえば、**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** セクションで設定される属性の詳細は、man ページの **multipath.conf(5)** を参照してください。**multipath.conf** ファイルの **devices** セクション、**multipaths** セクション、または **overrides** セクションに指定された属性により上書きされた場合を除き、この値は DM Multipath により使用されます。

## 4.5. 特定デバイスのマルチパス設定の編集

**multipath.conf** 設定ファイルの **multipaths** セクションにある属性は、指定された1つのマルチパスにのみ適用されます。このデフォルトは DM Multipath により使用され、**multipath.conf** ファイルの **overrides** セクション、**defaults** セクション、および **devices** セクションに設定された属性を上書きします。

**multipath.conf** 設定ファイルの **defaults** セクションで設定される属性の詳細は、man ページの **multipath.conf(5)** を参照してください。

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つ目のデバイスの WWID は **3600508b4000156d70001200000b0000** で、シンボリック名は **yellow** です。

2つ目のマルチパスデバイスの WWID は **1DEC\_321816758474** で、シンボリック名は **red** です。**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
  }
}

```

## 4.6. ストレージコントローラーのマルチパス設定の編集

**multipath.conf** 設定ファイルの **devices** セクションは、各ストレージデバイスの属性を設定します。デバイスを含むパスの **multipath.conf** ファイルの **multipaths** セクションまたは **overrides** セクションに指定された属性により上書きされた場合を除き、これらの値は DM Multipath により使用されます。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た属性を上書きします。

**multipath.conf** 設定ファイルの **devices** セクションで設定される属性の詳細は、man ページの **multipath.conf(5)** を参照してください。

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は、次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、変更する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、デフォルトでは自動で設定されないデバイスを追加する場合は、**vendor** パラメーターおよび **product** パラメーターを設定する必要があります。以下の例のように、この値は **/sys/block/device\_name/device/vendor** および **/sys/block/device\_name/device/model** で見つけることができます。**device\_name** はマルチパス化するデバイスになります。

```

# 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り異なります。デバイスが **active/active** の場合は通常、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** に設定することもできます。この他に、設定が必要となる可能性があるパラメーターは **no\_path\_retry** と **rr\_min\_io** です。

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ど

うかを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は、そのデバイスにマルチパスを設定できない可能性があります。

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
# }
# }
```

## 4.7. すべてのデバイスへのマルチパス値の設定

**multipath.conf** 設定ファイルの **overrides** セクションでは、すべてのデバイスの設定値を設定できます。たとえば、すべてのデバイスで **no\_path\_retry** を **fail** に設定できます。このセクションは、**multipath.conf** 設定ファイルの **devices** セクションおよび **defaults** セクションの両方で対応しているすべての属性に対応します。これは、**vendor**、**product**、および **revision** 以外の **devices** セクション属性になります。このデバイスを含むパスの **multipath.conf** ファイルで、**multipaths** セクションに指定された属性により上書きされた場合を除き、この属性はすべてのデバイスの DM Multipath により使用されます。この属性は、**multipath.conf** ファイルの **devices** セクションおよび **defaults** セクションに設定された属性を上書きします。

**multipath.conf** 設定ファイルの **devices** セクションおよび **defaults** セクションで設定される属性の詳細は、man ページの **multipath.conf(5)** を参照してください。

## 第5章 マルチパス化されたボリュームの管理

DM Multipath は、マルチパスボリュームの管理に使用できるさまざまなツールとコマンドを提供します。

### 5.1. MULTIPATH コマンド

**multipath** コマンドは、デバイスの複数のパスを検出および結合するために使用されます。マルチパス化されたデバイスの管理に使用できるさまざまなオプションを提供します。

表5.1「便利な **multipath** コマンドオプション」では、**multipath** コマンドの便利なオプションの一部を説明します。

表5.1 便利な **multipath** コマンドオプション

オプション	説明
<b>-l</b>	<b>sysfs</b> およびデバイスマッパーから取得した現在のマルチパス設定を表示します。
<b>-ll</b>	<b>sysfs</b> およびデバイスマッパーと、システムで利用可能なその他の全コンポーネントから取得した現在のマルチパス設定を表示します。
<b>-f device</b>	指定したマルチパスデバイスを削除します。
<b>-F</b>	不要なマルチパスデバイスをすべて削除します。
<b>-w device</b>	<b>wwids</b> ファイルから指定デバイスの <b>wwid</b> を削除します。
<b>-W</b>	現在のマルチパスデバイスのみが含まれるように、 <b>wwids</b> ファイルをリセットします。

#### 5.1.1. Multipath コマンドの出力

マルチパスデバイスの作成、修正、一覧表示を行うと、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。

```
3600d0230000000000e13955cc3757800 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
```

パスが有効になり、I/O が可能になると、パスのステータスが **ready** または **ghost** になります。パスが停止している場合は、ステータスが **faulty** または **shaky** になります。パスのステータスは、`/etc/multipath.conf` ファイルに定義されているポーリング間隔に基づき、`multipathd` デーモンにより定期的に更新されます。

その他の可能なパスのステータス値は以下のとおりです。

- **i/o pending** - チェッカーがアクティブにこのパスを確認し、ステータスはまもなく更新されません。
- **i/o timeout - faulty** と同じです。タイムアウトする前にチェッカーが成功または失敗のいずれかを返さなかったことをユーザーに通知します。
- **removed** - システムからパスが削除され、マルチパスデバイスからまもなく削除されます。**faulty** と同様に処理されます。
- **wild** - 内部エラーまたは設定の問題によって、`multipathd` はパスチェッカーを実行できませんでした。マルチパスがパス上の多くのアクションをスキップすることを除いて、**faulty** とほぼ同じです。
- **unchecked** - パスが今検出された、割り当てられたパスチェッカーがない、パスチェッカーにエラーが発生したなどの理由で、このパスでパスチェッカーが実行していません。これは **wild** と同じように処理されます。
- **delayed** - パスチェッカーはパスが有効であると返したにも関わらず、パスが複数回失敗し、パスを遅延するようにマルチパスが設定されているため、マルチパスがパスの復元を遅らせています。

dm のステータスはパスのステータスと似ていますが、視点はカーネルのものになります。dm のステータスが **active** の場合は、パスのステータスが **ready** および **ghost** になります。**pending** パスのステータスと同等の dm ステータスはありません。その他のパスのステータスはすべて、dm のステータスでは **failed** となります。dm のステータスは、パスチェッカーが完了するまで現在のステータスを保持します。

**online\_status** に使用できる値は **running** と **offline** です。**offline** は、この SCSI デバイスが無効になっていることを表します。



#### 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、dm のステータスなどが不明です。また、機能も常に正しいとは限りません。

## 5.1.2. マルチパス設定の表示

**multipath** コマンドで **-l** オプションおよび **-ll** オプションを使用すると、現在のマルチパス設定を表示できます。**-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ほかに、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とき、**multipath** コマンドの **-v** オプションを使用すると詳細レベルを指定できます。**-v0** を指定すると何も出力されません。**-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** などの別のツールに提供できます。**-v2** を指定すると、検出されたパス、マルチパス、およびデバイスマップがすべて出力されます。より詳細な情報を出力する場合は、**-v3**、**-v4**、または **-v5** を指定することもできます。

以下は、**multipath -l** コマンドの出力例になります。

```
# multipath -l
3600d02300000000000e13955cc3757800 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
```

以下は、**multipath -ll** コマンドの出力例になります。

```
# multipath -ll
3600d02300000000000e13955cc3757801 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
3600d02300000000000e13955cc3757803 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
```

## 5.2. 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. 次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 デバイスの場合は、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。以下にコマンド例を示します。

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

各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、次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動

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. 次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読み込んで **multipathd** の **chkconfig** を **on** に設定します。

```
# mpathconf --enable
```

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、「[DM Multipath のデバイスのブラックリスト](#)」の説明に従って、**/etc/multipath.conf** ファイルの **blacklist** セクションおよび **blacklist\_exceptions** セクションを変更します。
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と、パスが1つしかない場合でも必ず **find\_multipaths** がデバイスを許可するようになります。

```
# multipath -a root_devname
```

たとえば、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、以下の形式の行の出力を見つけます。これは、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
```

たとえば、デバイスの WWID が **3600d02300069c9ce09d41c4ac9c53200** の場合は、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. **multipath** で **initramfs** ファイルシステムを再構築するため、以下のオプションを使用して **dracut** コマンドを実行します。

**# 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、マルチパスを設定したデバイス上にあるかどうかを確認します。

## 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動

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でマルチパスを設定している限り、このような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、**/etc/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** オプションを指定して **/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**

たとえば、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** ファイルで、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。たとえば、**/etc/fstab** ファイルに、以下のようなエントリーがあるとします。

```
/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. DMSETUP コマンドでデバイスマッパーエントリーの特定

**dmsetup** コマンドを使用すると、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、そのメジャー番号および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。たとえば、マイナー番号 3 は、マルチパスを設定したデバイス **/dev/dm-3** に対応します。

```
# 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)
```

## 5.6. MULTIPATHD デーモンの管理

**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** コマンドは、man ページの **multipathd(8)** を参照してください。

次のコマンドは、**multipathd show maps** コマンドの出力の標準デフォルト形式を示しています。

```
# multipathd show maps
name sysfs uuid
mpathc dm-0 360a98000324669436c2b45666c567942
```

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の一覧を表示できます。

```
# multipathd show wildcards
```

**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 **format** コマンドに対応しています。raw 形式ではヘッダーは出力されず、フィールドがパディングされていないため、列とヘッダーが調整されていません。代わりに、フィールドは format 文字列の指定どおりに出力します。このため、出力はスクリプトで使いやすくなります。**multipathd show wildcards** コマンドを使用すると、format 文字列で使用されるワイルドカードを表示できます。



次の **multipathd** コマンドは、format 文字列でマルチパスのワイルドカードを使用し、**multipathd** が監視しているマルチパスデバイスを通常の raw 形式で表示します。

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

以下の **multipathd** コマンドは、format 文字列でマルチパスのワイルドカードを使用し、**multipathd** が監視しているパスを通常の形式と raw 形式で表示します。

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

以下のコマンドは、**multipathd show maps** の出力を、raw 形式と非 raw 形式で示した場合の相違点を表示します。**raw** 形式ではヘッダーがなく、単一のスペースのみが列の間にあることに注意してください。

```
# 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
```

## 5.7. 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。このファイルは、**device-mapper-multipath** パッケージの後続のインストール時に手動で削除する必要があります。

## 第6章 DM MULTIPATH のトラブルシューティング

マルチパス設定の実装に問題がある場合は、さまざまな項目を確認できます。

### 6.1. DM MULTIPATH トラブルシューティングチェックリスト

マルチパス設定の反応が遅かったり、機能しない場合は、以下の問題が原因として考えられます。

#### 6.1.1. マルチパスデーモンが実行していない

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](#)の説明に従って、マルチパスデーモンが実行していることを確認する必要があります。

マルチパスが設定されたデバイスを使用するには、**multipathd** デーモンが実行している必要があります。

#### 6.1.2. 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は、1つ以上のパスが復旧するまで、I/O を実行するプロセスがすべてハングします。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 (**N** はシステムでパスを再試行する回数に置き換えます)。

**features "1 queue\_if\_no\_path"** オプションを使用する必要があり、ここで説明する問題が発生した場合は、特定の LUN に対して起動時にキューイングポリシーを無効にできます (すべてのパスが利用できません)。次のコマンドは、特定デバイスのキューイングを無効にします。

```
multipathd disablequeueing map device
```

次のコマンドは、すべてのデバイスのキューイングを無効にします。

```
multipathd disablequeueing maps
```

デバイスのキューイングを無効にしたら、**multipathd** が再起動または再読み込みされるか、以下のコマンドのいずれかを実行するまで、そのデバイスは無効になります。

次のコマンドは、特定デバイスのキューイングを以前の値にリセットします。

```
multipathd restorequeueing map device
```

次のコマンドは、すべてのデバイスのキューイングを以前の値にリセットします。

```
multipathd restorequeueing maps
```

### 6.2. MULTIPATHD 対話式コンソールで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。このコマンドを実行してから **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。たとえば、次のコマンドシーケンスを実行すると、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
```

次のコマンドシーケンスを実行すると、**multipath.conf** への変更が確実に反映されます。

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

次のコマンドシーケンスを実行し、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
```

## 第7章 EH\_DEADLINE を使用したストレージエラーからの回復における最大時間の設定

障害が発生した SCSI デバイスを復旧するのに許容できる最大時間を設定できます。この設定は、ストレージハードウェアが不具合により応答しなくなっても、I/O 応答時間を保証します。

### 7.1. EH\_DEADLINE パラメーター

SCSI エラー処理 (EH) メカニズムは、障害が発生した SCSI デバイスでエラーからの復旧の実行を試行します。SCSI ホストオブジェクト `eh_deadline` パラメーターでは、復旧時間の最大量を設定できます。設定した時間が過ぎると、SCSI EH は、ホストバスアダプター (HBA) 全体を停止してリセットします。

`eh_deadline` を使用すると、以下のいずれかの時間を短縮できます。

- エラーのあるパスのシャットオフ
- パスの切り替え
- RAID スライスの無効化



#### 警告

`eh_deadline` が過ぎると、SCSI EH は HBA をリセットします。これは、エラーが発生しているものだけでなく、HBA 上のすべてのターゲットパスに影響します。一部の冗長パスがその他の理由により利用できない場合は、I/O エラーが発生する可能性があります。`eh_deadline` は、すべてのターゲットに完全な冗長マルチパス設定がある場合にのみ有効にします。

#### `eh_deadline` が便利なシナリオ

多くの場合、`eh_deadline` を有効にする必要はありません。`eh_deadline` は、ファイバーチャネル (FC) スイッチやターゲットポートでリンクの損失が発生した場合や、HBA が Registered State Change Notification (RSCN) を受信しない場合など、特定のシナリオで便利です。このような場合、I/O 要求やエラーからの復旧コマンドは、エラーに遭遇することなく、すべてタイムアウトになります。この環境で `eh_deadline` を設定すると、リカバリー時間に上限が課せられます。これにより、DM Multipath により、利用できる別のパスで不具合の発生した I/O の再試行が可能になります。

以下の条件では、`eh_deadline` 機能はこれ以上のメリットをもたらしません。その理由は、DM Multipath の再試行を可能にする I/O とエラー復旧コマンドがすぐに失敗するためです。

- RSCN が有効になっている場合
- HBA が利用できなくなっているリンクを登録しない場合

#### 可能な値

`eh_deadline` の値は秒単位で指定されます。

デフォルト設定は `off` で、時間制限が無効になり、すべてのエラー復旧が行われるようになります。

## 7.2. EH\_DEADLINE パラメーターの設定

この手順では、SCSI を復旧する最大時間を制限する **eh-daedline** パラメーターの値を設定します。

### 手順

- **eh\_deadline** は、以下のいずれかの方法で設定できます。

#### sysfs

`/sys/class/scsi_host/host*/eh_deadline` ファイルに秒数を書き込みます。

#### カーネルパラメーター

すべての SCSI HBA のデフォルト値は **scsi\_mod.eh\_deadline** カーネルパラメーターを使用して設定します。

### 関連情報

- [How to set eh\\_deadline and eh\\_timeout persistently, using a udev rule](#)