# Red Hat Enterprise Linux 5
# DM Multipath

DM Multipath Configuration and Administration
Edition 3

# Red Hat Enterprise Linux 5 DM Multipath

DM Multipath Configuration and Administration
Edition 3

## Legal Notice

## Abstract

This book provides information on using the Device-Mapper Multipath feature of Red Hat Enterprise Linux 5

# Table of Contents

# Preface

This book describes the Device Mapper Multipath (DM-Multipath) feature of Red Hat Enterprise Linux for the Red Hat Enterprise Linux 5 release.

## 1. Audience

This book is intended to be used by system administrators managing systems running the Linux operating system. It requires familiarity with Red Hat Enterprise Linux.

## 2. Related Documentation

For more information about using Red Hat Enterprise Linux, refer to the following resources:

⫸ *Red Hat Enterprise Linux Installation Guide* — Provides information regarding installation of Red Hat Enterprise Linux 5.

⫸ *Red Hat Enterprise Linux Deployment Guide* — Provides information regarding the deployment, configuration and administration of Red Hat Enterprise Linux 5.

For more information about Red Hat Cluster Suite for Red Hat Enterprise Linux 5, refer to the following resources:

⫸ *Red Hat Cluster Suite Overview* — Provides a high level overview of the Red Hat Cluster Suite.

⫸ *Configuring and Managing a Red Hat Cluster* — Provides information about installing, configuring and managing Red Hat Cluster components.

⫸ *Logical Volume Manager Administration* — Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.

⫸ *Global File System: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS (Red Hat Global File System).

⫸ *Global File System 2: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS2 (Red Hat Global File System 2).

⫸ *Using GNBD with Global File System* — Provides an overview on using Global Network Block Device (GNBD) with Red Hat GFS.

⫸ *Linux Virtual Server Administration* — Provides information on configuring high-performance systems and services with the Linux Virtual Server (LVS).

⫸ *Red Hat Cluster Suite Release Notes* — Provides information about the current release of Red Hat Cluster Suite.

Red Hat Cluster Suite documentation and other Red Hat documents are available in HTML, PDF, and RPM versions on the Red Hat Enterprise Linux Documentation CD and online at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/

## 3. Feedback

If you spot a typo, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (http://bugzilla.redhat.com/bugzilla/). File the bug against the product **Red Hat Enterprise Linux 5** and against the component **Documentation-cluster**.

Be sure to mention the manual's identifier:

```
Bugzilla component: Documentation-cluster
Book identifier: DM_Multipath(EN)-5 (2014-6-30T15:35)
```

By mentioning this manual's identifier, we know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

```
Bugzilla component: Documentation-cluster
Book identifier: DM_Multipath(EN)-5 (2014-6-30T15:35)
```

# Chapter 1. Device Mapper Multipathing

Device Mapper Multipathing (DM-Multipath) allows you to configure multiple I/O paths between server nodes and storage arrays into a single device. These I/O paths are physical SAN connections that can include separate cables, switches, and controllers. Multipathing aggregates the I/O paths, creating a new device that consists of the aggregated paths.

## 1.1. New and Changed Features for Red Hat Enterprise Linux 5.7

Red Hat Enterprise Linux 5.7 includes the following documentation and feature updates and changes.

≫ The new configuration file parameters **`fast_io_fail_tmo`**, **`dev_loss_tmo`**, and **`file_timeout`** are documented in Chapter 4, *The DM-Multipath Configuration File*.

≫ The procedure for ensuring that multipath devices names are consistent across the nodes in a cluster has been expanded in Section 2.2, "Consistent Multipath Device Names in a Cluster".

## 1.2. New and Changed Features for Red Hat Enterprise Linux 5.8

Red Hat Enterprise Linux 5.8 includes the following documentation and feature updates and changes.

≫ The **`getuid_callout`** and **`prio_callout`** configuration file parameters now accept the **`%c`** wildcard. For information on the configuration file parameters, see Chapter 4, *The DM-Multipath Configuration File*.

≫ You can now define either the **`selector`** or **`path_selector`** parameter in the defaults section of the **`multipath.conf`** configuration file. Both parameters have the same meaning. For information on the multipath configuration defaults, see Table 4.1, "Multipath Configuration Defaults".

## 1.3. New and Changed Features for Red Hat Enterprise Linux 5.9

Red Hat Enterprise Linux 5.9 includes the following documentation and feature updates and changes.

≫ The **`user_friendly_names`** and **`flush_on_last_del`** multipath configuration parameters can now be defined in the **`multipaths`** and **`devices`** sections of the **`multipath.conf`** configuration file, as documented in Chapter 4, *The DM-Multipath Configuration File*.

## 1.4. Overview of DM-Multipath

DM-Multipath can be used to provide:

≫ Redundancy

DM-Multipath can provide failover in an active/passive configuration. In an active/passive configuration, only half the paths are used at any time for I/O. If any element of an I/O path (the cable, switch, or controller) fails, DM-Multipath switches to an alternate path.

≫ Improved Performance

DM-Multipath can be configured in active/active mode, where I/O is spread over the paths in a round-robin fashion. In some configurations, DM-Multipath can detect loading on the I/O paths and dynamically re-balance the load.

Figure 1.1, "Active/Passive Multipath Configuration with One RAID Device" shows an active/passive configuration with two I/O paths from the server to a RAID device. There are 2 HBAs on the server, 2 SAN switches, and 2 RAID controllers.
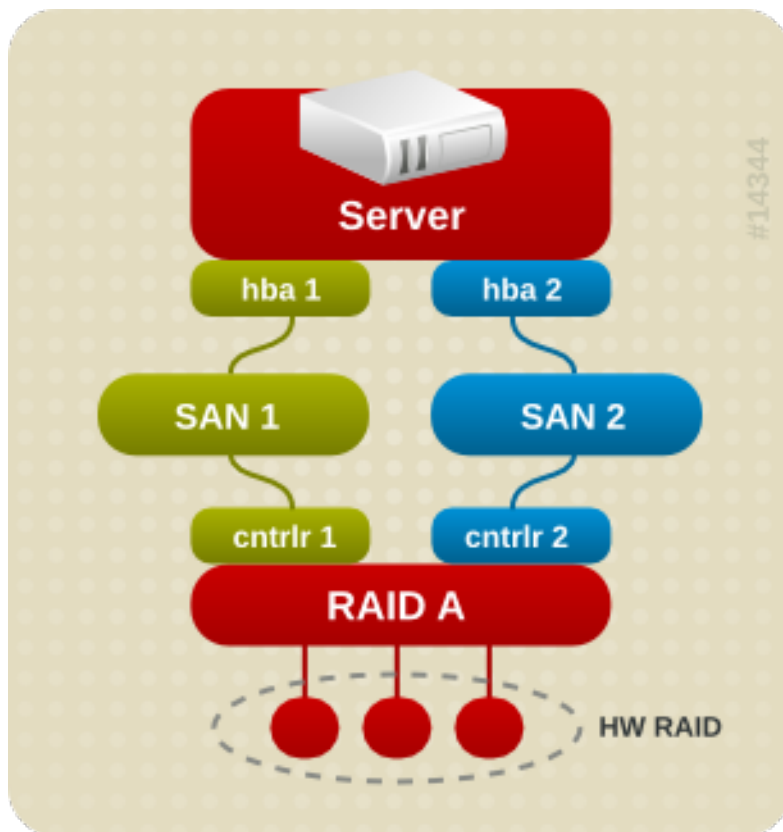


**Figure 1.1. Active/Passive Multipath Configuration with One RAID Device**

In this configuration, there is one I/O path that goes through hba1, SAN1, and controller 1 and a second I/O path that goes through hba2, SAN2, and controller2. There are many points of possible failure in this configuration:

> » HBA failure

> » FC cable failure

> » SAN switch failure

> » Array controller port failure

With DM-Multipath configured, a failure at any of these points will cause DM-Multipath to switch to the alternate I/O path.

Figure 1.2, "Active/Passive Multipath Configuration with Two RAID Devices" shows a more complex active/passive configuration with 2 HBAs on the server, 2 SAN switches, and 2 RAID devices with 2 RAID controllers each.
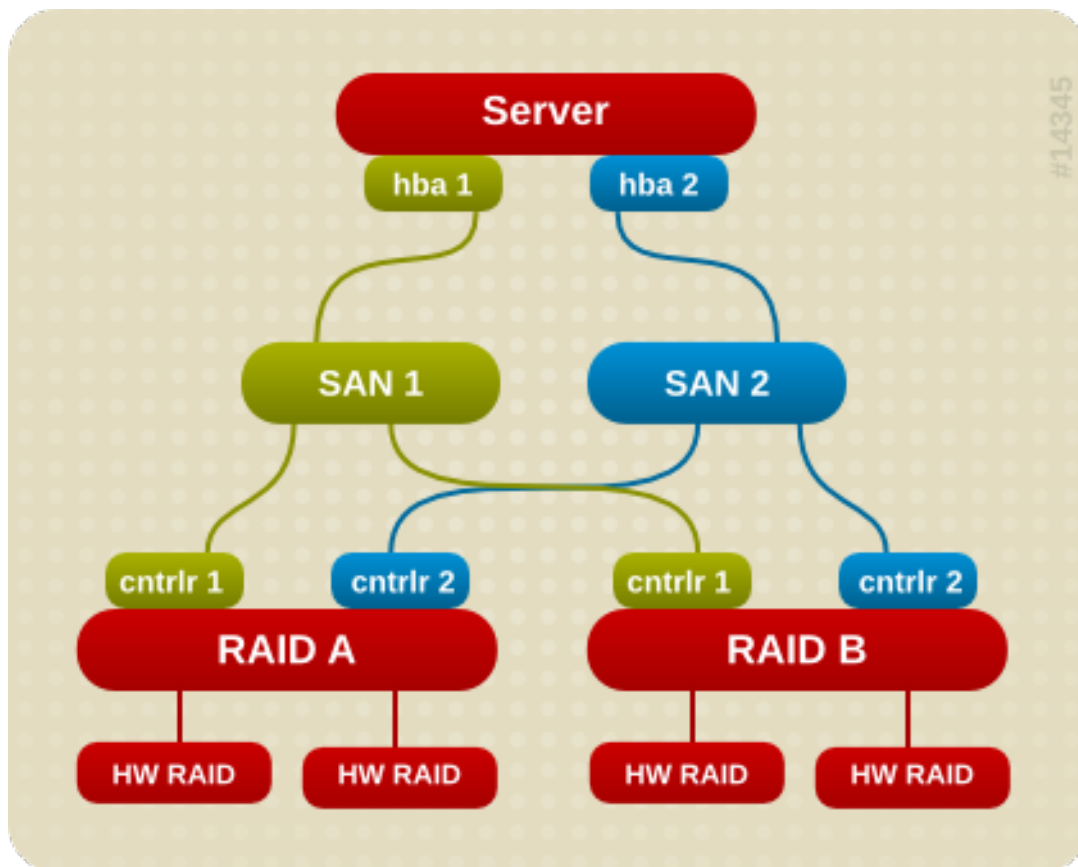
**Figure 1.2. Active/Passive Multipath Configuration with Two RAID Devices**

In the example shown in Figure 1.2, "Active/Passive Multipath Configuration with Two RAID Devices", there are two I/O paths to each RAID device (just as there are in the example shown in Figure 1.1, "Active/Passive Multipath Configuration with One RAID Device"). With DM-Multipath configured, a failure at any of the points of the I/O path to either of the RAID devices will cause DM-Multipath to switch to the alternate I/O path for that device.

Figure 1.3, "Active/Active Multipath Configuration with One RAID Device" shows an active/active configuration with 2 HBAs on the server, 1 SAN switch, and 2 RAID controllers. There are four I/O paths from the server to a storage device:

≫ hba1 to controller1

≫ hba1 to controller2

≫ hba2 to controller1

≫ hba2 to controller2

In this configuration, I/O can be spread among those four paths.

**Figure 1.3. Active/Active Multipath Configuration with One RAID Device**

## 1.5. Storage Array Support

By default, DM-Multipath includes support for the most common storage arrays that support DM-Multipath. The supported devices can be found in the **multipath.conf.defaults** file. If your storage array supports DM-Multipath and is not configured by default in this file, you may need to add them to the DM-Multipath configuration file, **multipath.conf**. For information on the DM-Multipath configuration file, see Chapter 4, *The DM-Multipath Configuration File*.

Some storage arrays require special handling of I/O errors and path switching. These require separate hardware handler kernel modules.

## 1.6. DM-Multipath Components

Table 1.1, "DM-Multipath Components". describes the components of DM-Multipath.

**Table 1.1. DM-Multipath Components**

| Component | Description |
| --- | --- |
| **dm-multipath** kernel module | Reroutes I/O and supports failover for paths and path groups. |
| **multipath** command | Lists and configures multipath devices. Normally started up with **/etc/rc.sysinit**, it can also be started up by a **udev** program whenever a block device is added or it can be run by the **initramfs** file system. |
| **multipathd** daemon | Monitors paths; as paths fail and come back, it may initiate path group switches. Provides for interactive changes to multipath devices. This must be restarted for any changes to the **/etc/multipath.conf** file. |

| Component | Description |
|---|---|
| `kpartx` command | Creates device mapper devices for the partitions on a device It is necessary to use this command for DOS-based partitions with DM-MP. The `kpartx` is provided in its own package, but the `device-mapper-multipath` package depends on it. |

## 1.7. DM-Multipath Setup Overview

DM-Multipath includes compiled-in default settings that are suitable for common multipath configurations. Setting up DM-multipath is often a simple procedure.

The basic procedure for configuring your system with DM-Multipath is as follows:

1. Install `device-mapper-multipath` rpm.

2. Edit the `multipath.conf` configuration file:

   ⯈ comment out the default blacklist

   ⯈ change any of the existing defaults as needed

   ⯈ save the configuration file

3. Start the multipath daemons.

4. Create the multipath device with the `multipath` command.

> **Important**
>
> Note that migrating the root file system of an existing Red Hat Enterprise Linux installation from single path storage to multipath storage is not supported. You must perform a new installation to move the root file system to a multipath storage device. Therefore you should plan your installation accordingly.

Detailed setup instructions for several example multipath configurations are provided in Chapter 3, *Setting Up DM-Multipath*.

# Chapter 2. Multipath Devices

Without DM-Multipath, each path from a server node to a storage controller is treated by the system as a separate device, even when the I/O path connects the same server node to the same storage controller. DM-Multipath provides a way of organizing the I/O paths logically, by creating a single multipath device on top of the underlying devices.

## 2.1. Multipath Device Identifiers

Each multipath device has a World Wide Identifier (WWID), which is guaranteed to be globally unique and unchanging. By default, the name of a multipath device is set to its WWID. Alternately, you can set the **user_friendly_names** option in the multipath configuration file, which sets the alias to a node-unique name of the form **mpath*n***.

For example, a node with two HBAs attached to a storage controller with two ports via a single unzoned FC switch sees four devices: **/dev/sda**, **/dev/sdb**, **dev/sdc**, and **/dev/sdd**. DM-Multipath creates a single device with a unique WWID that reroutes I/O to those four underlying devices according to the multipath configuration. When the **user_friendly_names** configuration option is set to **yes**, the name of the multipath device is set to **mpath*n***.

When new devices are brought under the control of DM-Multipath, the new devices may be seen in three different places under the **/dev** directory: **/dev/mapper/mpath*n***, **/dev/mpath/mpath*n***, and **/dev/dm-*n***.

- ≫ The devices in **/dev/mapper** are created early in the boot process. Use these devices to access the multipathed devices, for example when creating logical volumes.

- ≫ The devices in **/dev/mpath** are provided as a convenience so that all multipathed devices can be seen in one directory. These devices are created by the **udev** device manager and may not be available on startup when the system needs to access them. Do not use these devices for creating logical volumes or filesystems.

- ≫ Any devices of the form **/dev/dm-*n*** are for internal use only and should never be used.

For information on the multipath configuration defaults, including the **user_friendly_names** configuration option, see Section 4.3, "Configuration File Defaults".

You can also set the name of a multipath device to a name of your choosing by using the **alias** option in the **multipaths** section of the multipath configuration file. For information on the **multipaths** section of the multipath configuration file, see Section 4.4, "Multipaths Device Configuration Attributes".

## 2.2. Consistent Multipath Device Names in a Cluster

When the **user_friendly_names** configuration option is set to **yes**, the name of the multipath device is unique to a node, but it is not guaranteed to be the same on all nodes using the multipath device. Similarly, if you set the **alias** option for a device in the **multipaths** section of the **multipath.conf** configuration file, the name is not automatically consistent across all nodes in the cluster. This should not cause any difficulties if you use LVM to create logical devices from the multipath device, but if you require that your multipath device names be consistent in every node it is recommended that you not set the **user_friendly_names** option to **yes** and that you not configure aliases for the devices. By default, if you do not set **user_friendly_names** to **yes** or configure an alias for a device, a device name will be the WWID for the device, which is always the same.

If you want the system-defined user-friendly names to be consistent across all nodes in the cluster, however, you can follow this procedure:

1. Set up all of the multipath devices on one machine.

2. Disable all of your multipath devices on your other machines by running the following commands:

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

3. Copy the **bindings** file from the first machine to all the other machines in the cluster. By default, the location of this file is **/var/lib/multipath/bindings**. If **/var** is a separate partition on your system, however, you should change this value with the **bindings_file** option in the **defaults** section of the **multipath.conf** configuration file, as described in Section 4.3, "Configuration File Defaults". This file needs to be located on your root file system partition, for example:

```
bindings_file "/etc/multipath_bindings"
```

4. Re-enable the **multipathd** daemon on all the other machines in the cluster by running the following command:

```
# service multipathd start
```

If you add a new device, you will need to repeat this process.

Similarly, if you configure an alias for a device that you would like to be consistent across the nodes in the cluster, you should ensure that the **/etc/multipath.conf** file is the same for each node in the cluster by following the same procedure:

1. Configure the aliases for the multipath devices in the in the **multipath.conf** file on one machine.

2. Disable all of your multipath devices on your other machines by running the following commands:

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

3. Copy the **/etc/multipath.conf** file from the first machine to all the other machines in the cluster.

4. Re-enable the **multipathd** daemon on all the other machines in the cluster by running the following command:

```
# service multipathd start
```

When you add a new device you will need to repeat this process.

## 2.3. Multipath Device Attributes

In addition to the **user_friendly_names** and **alias** options, a multipath device has numerous attributes. You can modify these attributes for a specific multipath device by creating an entry for that device in the **multipaths** section of the multipath configuration file. For information on the **multipaths** section of the multipath configuration file, see Section 4.4, "Multipaths Device Configuration Attributes".

## 2.4. Multipath Devices in Logical Volumes

After creating multipath devices, you can use the multipath device names just as you would use a physical device name when creating an LVM physical volume. For example, if **/dev/mapper/mpath0** is the name of a multipath device, the following command will mark **/dev/mapper/mpath0** as a physical volume.

```
pvcreate /dev/mapper/mpath0
```

You can use the resulting LVM physical device when you create an LVM volume group just as you would use any other LVM physical device.

> **Note**
>
> If you attempt to create an LVM physical volume on a whole device on which you have configured partitions, the **pvcreate** command will fail. Note that the Anaconda and Kickstart installation programs create empty partition tables if you do not specify otherwise for every block device. If you wish to use the whole device rather than a partition, you must remove the existing partitions from the device. You can remove existing partitions with the **kpartx -d** and the **fdisk** commands. If your system has block devices that are greater that 2Tb, you can use the **parted** command to remove partitions.

When you create an LVM logical volume that uses active/passive multipath arrays as the underlying physical devices, you should include filters in the **lvm.conf** to exclude the disks that underlie the multipath devices. This is because if the array automatically changes the active path to the passive path when it receives I/O, multipath will failover and failback whenever LVM scans the passive path if these devices are not filtered. For active/passive arrays that require a command to make the passive path active, LVM prints a warning message when this occurs.

To filter all SCSI devices in the LVM configuration file (**lvm.conf**), include the following filter in the **devices** section of the file.

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

# Chapter 3. Setting Up DM-Multipath

This chapter provides step-by-step example procedures for configuring DM-Multipath. It includes the following procedures:

- Basic DM-Multipath setup

- Ignoring local disks

- Adding more devices to the configuration file

## 3.1. Setting Up DM-Multipath

Before setting up DM-Multipath on your system, ensure that your system has been updated and includes the **device-mapper-multipath** package.

Use the following procedure to set up DM-Multipath for a basic failover configuration.

1. Edit the **/etc/multipath.conf** file by commenting out the following lines at the top of the file. This section of the configuration file, in its initial state, blacklists all devices. You must comment it out to enable multipathing.

   ```
   blacklist {
           devnode "*"
   }
   ```

   After commenting out those lines, this section appears as follows.

   ```
   # blacklist {
   #         devnode "*"
   # }
   ```

2. The default settings for DM-Multipath are compiled in to the system and do not need to be explicitly set in the **/etc/multipath.conf** file.

   The default value of **path_grouping_policy** is set to **failover**, so in this example you do not need to change the default value. For information on changing the values in the configuration file to something other than the defaults, see Chapter 4, *The DM-Multipath Configuration File*.

   The initial defaults section of the configuration file configures your system that the names of the multipath devices are of the form **mpath**n; without this setting, the names of the multipath devices would be aliased to the WWID of the device.

3. Save the configuration file and exit the editor.

4. Execute the following commands:

   ```
   modprobe dm-multipath
   service multipathd start
   multipath -v2
   ```

   The **multipath -v2** command prints out multipathed paths that show which devices are multipathed. If the command does not print anything out, ensure that all SAN connections are set up properly and the system is multipathed.

For further information on the **multipath** command output, see Section 5.5, "Multipath Command Output".

5. Execute the following command to ensure sure that the multipath daemon starts on bootup:

```
chkconfig multipathd on
```

Since the value of **user_friendly_name** is set to **yes** in the configuration file the multipath devices will be created as **/dev/mapper/mpath**n. For information on setting the name of the device to an alias of your choosing, see Chapter 4, *The DM-Multipath Configuration File*.

## 3.2. Ignoring Local Disks when Generating Multipath Devices

Some machines have local SCSI cards for their internal disks. DM-Multipath is not recommended for these devices. The following procedure shows how to modify the multipath configuration file to ignore the local disks when configuring multipath.

1. Determine which disks are the internal disks and mark them as the ones to blacklist.

   In this example, **/dev/sda** is the internal disk. Note that as originally configured in the default multipath configuration file, executing the **multipath -v2** shows the local disk, **/dev/sda**, in the multipath map.

   For further information on the **multipath** command output, see Section 5.5, "Multipath Command Output".

```
[root@rh4cluster1 ~]# multipath -v2
create: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
[size=33 GB][features="0"][hwhandler="0"]
\_ round-robin 0
  \_ 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
[size=12 GB][features="0"][hwhandler="0"]
\_ round-robin 0
  \_ 2:0:0:0 sdb  8:16
  \_ 3:0:0:0 sdf  8:80

create: 3600a0b80001327510000009a436215ec
[size=12 GB][features="0"][hwhandler="0"]
\_ round-robin 0
  \_ 2:0:0:1 sdc  8:32
  \_ 3:0:0:1 sdg  8:96

create: 3600a0b80001327d800000070436216b3
[size=12 GB][features="0"][hwhandler="0"]
\_ round-robin 0
  \_ 2:0:0:2 sdd  8:48
  \_ 3:0:0:2 sdh  8:112

create: 3600a0b80001327510000009b4362163e
```

```
[size=12 GB][features="0"][hwhandler="0"]
\_ round-robin 0
  \_ 2:0:0:3 sde  8:64
  \_ 3:0:0:3 sdi  8:128
```

2. In order to prevent the device mapper from mapping **/dev/sda** in its multipath maps, edit the blacklist section of the **/etc/multipath.conf** file to include this device. Although you could blacklist the **sda** device using a **devnode** type, that would not be safe procedure since **/dev/sda** is not guaranteed to be the same on reboot. To blacklist individual devices, you can blacklist using the WWID of that device.

   Note that in the output to the **multipath -v2** command, the WWID of the **/dev/sda** device is SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1. To blacklist this device, include the following in the **/etc/multipath.conf** file.

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

3. After you have updated the **/etc/multipath.conf** file, you must manually tell the **multipathd** daemon to reload the file. The following command reloads the updated **/etc/multipath.conf** file.

   ```
   service multipathd reload
   ```

4. Run the following commands:

   ```
   multipath -F
   multipath -v2
   ```

   The local disk or disks should no longer be listed in the new multipath maps, as shown in the following example.

   ```
   [root@rh4cluster1 ~]# multipath -F
   [root@rh4cluster1 ~]# multipath -v2
   create: 3600a0b80001327d80000006d43621677
   [size=12 GB][features="0"][hwhandler="0"]
   \_ round-robin 0
     \_ 2:0:0:0 sdb  8:16
     \_ 3:0:0:0 sdf  8:80

   create: 3600a0b80001327510000009a436215ec
   [size=12 GB][features="0"][hwhandler="0"]
   \_ round-robin 0
     \_ 2:0:0:1 sdc  8:32
     \_ 3:0:0:1 sdg  8:96

   create: 3600a0b80001327d800000070436216b3
   [size=12 GB][features="0"][hwhandler="0"]
   \_ round-robin 0
     \_ 2:0:0:2 sdd  8:48
     \_ 3:0:0:2 sdh  8:112

   create: 3600a0b80001327510000009b4362163e
   ```

```
[size=12 GB][features="0"][hwhandler="0"]
\_ round-robin 0
  \_ 2:0:0:3 sde  8:64
  \_ 3:0:0:3 sdi  8:128
```

## 3.3. Adding Devices to the Multipathing Database

By default, DM-Multipath includes support for the most common storage arrays that support DM-Multipath. The default configuration values, including supported devices, can be found in the **multipath.conf.defaults** file.

If you need to add a storage device that is not supported by default as a known multipath device, edit the **/etc/multipath.conf** file and insert the appropriate device information.

For example, to add information about the HP Open-V series the entry looks like this:

```
devices {
      device {
              vendor "HP"
              product "OPEN-V."
              getuid_callout "/sbin/scsi_id -g -u -p0x80 -s /block/%n"
      }
}
```

For more information on the **devices** section of the configuration file, see Section 4.5, "Configuration File Devices".

# Chapter 4. The DM-Multipath Configuration File

By default, DM-Multipath provides configuration values for the most common uses of multipathing. In addition, DM-Multipath includes support for the most common storage arrays that support DM-Multipath. The default configuration values and the supported devices can be found in the **/usr/share/doc/device-mapper-multipath-0.4.7/multipath.conf.defaults** file.

You can override the default configuration values for DM-Multipath by editing the **/etc/multipath.conf** configuration file. If necessary, you can also add a storage array that is not supported by default to the configuration file. This chapter provides information on parsing and modifying the **multipath.conf** file. It contains sections on the following topics:

- » Configuration file overview

- » Configuration file blacklist

- » Configuration file defaults

- » Configuration file multipaths

- » Configuration file devices

In the multipath configuration file, you need to specify only the sections that you need for your configuration, or that you wish to change from the default values specified in the **multipath.conf.defaults** file. If there are sections of the file that are not relevant to your environment or for which you do not need to override the default values, you can leave them commented out, as they are in the initial file.

The configuration file allows regular expression description syntax.

An annotated version of the configuration file can be found in **/usr/share/doc/device-mapper-multipathd-0.4.7/multipath.conf.annotated**.

## 4.1. Configuration File Overview

The multipath configuration file is divided into the following sections:

**blacklist**

Listing of specific devices that will not be considered for multipath. By default all devices are blacklisted. Usually the default blacklist section is commented out.

**blacklist_exceptions**

Listing of multipath candidates that would otherwise be blacklisted according to the parameters of the blacklist section.

**defaults**

General default settings for DM-Multipath.

**multipaths**

Settings for the characteristics of individual multipath devices. These values overwrite what is specified in the **defaults** and **devices** sections of the configuration file.

**devices**

Settings for the individual storage controllers. These values overwrite what is specified in the **defaults** section of the configuration file. If you are using a storage array that is not supported by default, you may need to create a **devices** subsection for your array.

When the system determines the attributes of a multipath device, first it checks the multipath settings, then the per devices settings, then the multipath system defaults.

## 4.2. Configuration File Blacklist

The **blacklist** section of the multipath configuration file specifies the devices that will not be used when the system configures multipath devices. Devices that are blacklisted will not be grouped into a multipath device.

By default, all devices are blacklisted, since the following lines appear in the initial configuration file.

```
blacklist {
        devnode "*"
}
```

To enable multipathing on all of the devices that are supported by default, comment out those lines, as described in Section 3.1, "Setting Up DM-Multipath".

After commenting out the universal blacklist, you can specify general device types and individual devices to blacklist. You can blacklist devices according to the following criteria:

» By WWID, as described in Section 4.2.1, "Blacklisting by WWID"

» By device name, as described in Section 4.2.2, "Blacklisting By Device Name"

» By device type, as described in Section 4.2.3, "Blacklisting By Device Type"

By default, a variety of device types are blacklisted, even after you comment out the initial blacklist section of the configuration file. For information, see Section 4.2.2, "Blacklisting By Device Name".

### 4.2.1. Blacklisting by WWID

You can specify individual devices to blacklist by their World-Wide IDentification with a **wwid** entry in the **blacklist** section of the configuration file.

The following example shows the lines in the configuration file that would blacklist a device with a WWID of 26353900f02796769.

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

### 4.2.2. Blacklisting By Device Name

You can blacklist device types by device name so that they will not be grouped into a multipath device by specifying a **devnode** entry in the **blacklist** section of the configuration file.

The following example shows the lines in the configuration file that would blacklist all SCSI devices, since it blacklists all sd* devices.

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

```
}
```

You can use a **devnode** entry in the **blacklist** section of the configuration file to specify individual devices to blacklist rather than all devices of specific type; this is not recommended, however. Unless it is statically mapped by **udev** rules, there is no guarantee that a specific device will have the same name on reboot. For example, a device name could change from **/dev/sda** to **/dev/sdb** on reboot.

By default, the following **devnode** entries are compiled in the default blacklist; the devices that these entries blacklist do not generally support DM-Multipath. To enable multipathing on any of these devices, you would need to specify them in the **blacklist_exceptions** section of the configuration file, as described in Section 4.2.4, "Blacklist Exceptions".

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

## 4.2.3. Blacklisting By Device Type

You can specify specific device types in the **blacklist** section of the configuration file with a **device** section. The following example blacklists all IBM DS4200 and HP devices.

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

## 4.2.4. Blacklist Exceptions

You can use the **blacklist_exceptions** section of the configuration file to enable multipathing on devices that have been blacklisted by default. For example, by default, devices with vendor = "IBM" and product = "S/390.*" are blacklisted since usually DASD devices are used for local storage. If you do need to enable multipathing on these devices, uncomment the following lines in the configuration file:

```
blacklist_exceptions {
        device {
                vendor  "IBM"
                product "S/390.*"
        }
}
```

When specifying devices in the **blacklist_exceptions** section of the configuration file, you must specify the exceptions in the same way they were specified in the blacklist. For example, a WWID exception will not apply to devices specified by a **devnode** blacklist entry, even if the blacklisted device is associated with that WWID. Similarly, **devnode** exceptions apply only to **devnode** entries, and **device** exceptions apply only to device entries.

## 4.3. Configuration File Defaults

The **/etc/multipath.conf** configuration file includes a **defaults** section that sets the
**user_friendly_names** parameter to **yes**, as follows.

```
defaults {
        user_friendly_names yes
}
```

This overwrites the default value of the **user_friendly_names** parameter.

The configuration file includes a template of configuration defaults. This section is commented out, as follows.

```
#defaults {
#       udev_dir                /dev
#       polling_interval        10
#       selector                "round-robin 0"
#       path_grouping_policy    multibus
#       getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
#       prio_callout            /bin/true
#       path_checker            readsector0
#       rr_min_io               100
#       max_fds                 8192
#       rr_weight               priorities
#       failback                immediate
#       no_path_retry           fail
#       user_friendly_names     yes
#}
```

To overwrite the default value for any of the configuration parameters, you can copy the relevant line from
this template into the **defaults** section and uncomment it. For example, to overwrite the
**path_grouping_policy** parameter so that it is **multibus** rather than the default value of **failover**,
copy the appropriate line from the template to the initial **defaults** section of the configuration file, and
uncomment it, as follows.

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

Table 4.1, "Multipath Configuration Defaults" describes the attributes that are set in the **defaults** section of
the **multipath.conf** configuration file. These values are used by DM-Multipath unless they are overwritten
by the attributes specified in the **devices** and **multipaths** sections of the **multipath.conf** file.

**Table 4.1. Multipath Configuration Defaults**

| Attribute | Description |
| --- | --- |
| **udev_dir** | Specifies the directory where **udev** device nodes are created. The default value is **/udev**. |
| **verbosity** | (Red Hat Enterprise Linux 5.3 and later) Specifies the verbosity level of the command. It can be overridden by the **-v** command line option. The default value is 2. |

| Attribute | Description |
|-----------|-------------|
| **polling_interval** | Specifies the interval between two path checks in seconds. The default value is 5. |
| **selector, path_selector** | Specifies the default algorithm to use in determining what path to use for the next I/O operation. The default value is **round-robin 0**. **selector** and **path_selector** have the same meaning; the **path_selection** attribute is supported in Red Hat Enterprise Linux 5.8 and later. |
| **path_grouping_policy** | Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include:<br><br>**failover** = 1 path per priority group<br><br>**multibus** = all valid paths in 1 priority group<br><br>**group_by_serial** = 1 priority group per detected serial number<br><br>**group_by_prio** = 1 priority group per path priority value<br><br>**group_by_node_name** = 1 priority group per target node name<br><br>The default value is **failover**. |
| **getuid_callout** | Specifies the default program and arguments to call out to obtain a unique path identifier. An absolute path is required.<br><br>This parameter accepts the following wildcard values:<br><br>**%n**: replaced with the device name as it appears in sysfs.<br><br>**%c**: replaced with the device name as it appears in **/dev** (this is different than the sysfs name for CCISS devices). (Red Hat Enterprise Linux 5.8 and later)<br><br>**%d**: replaced with *major_nr:minor_nr*<br><br>**%h**: replaced with **host_nr**:**bus_nr**:**target_nr**:**lun_nr**.<br><br>The default value of **getuid_callout** is **/sbin/scsi_id -g -u -s /block/%n**. |
| **prio_callout** | Specifies the default program and arguments to call out to obtain a path weight. Weights are summed for each path group to determine the next path group to use in case of failure. "none" is a valid value. This parameter accepts the same wildcard values as **getuid_callout**. |
| **path_checker** | Specifies the default method used to determine the state of the paths. Possible values include: **readsector0**, **rdac**, **tur**, **cciss_tur**, **hp_tur** (Red Hat Enterprise Linux 5.5 and later), **emc_clariion**, **hp_sw**, and **directio**. The default value is **readsector0**. |

| Attribute | Description |
|---|---|
| **features** | The default extra features of multipath devices, using the format: "*number_of_features_plus_arguments feature1* ..." Possible values for features include:<br><br>**queue_if_no_path**, which is the same as setting **no_path_retry** to **queue**. For information on issues that may arise when using this feature, see Section 5.4, "Issues with queue_if_no_path feature".<br><br>**pg_init_retries** *n*: Retry path group initialization up to *n* times before failing where *n* is between 1 and 50, inclusive.<br><br>**pg_init_delay_secs** *n*: Wait *n* seconds between path group initialization retries where *n* is between 0 and 50, inclusive. |
| **rr_min_io** | Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. The default value is **1000**. |
| **max_fds** | (Red Hat Enterprise Linux 5.2 and later) Sets the maximum number of open file descriptors for the **multipathd** process. In Red Hat Enterprise Linux 5.3, this option allows a value of **max**, which sets the number of open file descriptors to the system maximum. As of the Red Hat Enterprise Linux 5.9 release, the default value is **max**, which sets this to the system limit from **/proc/sys/fs/nr_open**. For earlier releases, if this is not set the maximum number of open file descriptors is taken from the calling process; it is usually 1024. To be safe, this should be set to the maximum number of paths plus 32, if that number is greater than 1024. |
| **rr_weight** | If set to **priorities**, then instead of sending **rr_min_io** requests to a path before calling **selector** or **path_selector** to choose the next path, the number of requests to send is determined by **rr_min_io** times the path's priority, as determined by the **prio_callout** program. Currently, there are priority callouts only for devices that use the **group_by_prio** path grouping policy, which means that all the paths in a path group will always have the same priority.<br><br>If set to **uniform**, all path weights are equal. The default value is **uniform**. |
| **failback** | Specifies path group failback.<br><br>A value of 0 or **immediate** specifies that as soon as there is a path group with a higher priority than the current path group the system switches to that path group.<br><br>A numeric value greater than zero specifies deferred failback, expressed in seconds.<br><br>A value of **manual** specifies that failback can happen only with operator intervention.<br><br>The default value is **manual**. |

| Attribute | Description |
|---|---|
| `no_path_retry` | A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queueing.<br><br>A value of **fail** indicates immediate failure, without queueing.<br><br>A value of **queue** indicates that queueing should not stop until the path is fixed. |
| `flush_on_last_del` | (Red Hat Enterprise Linux 5.3 and later) If set to **yes**, the **multipathd** daemon will disable queueing when the last path to a device has been deleted. The default value is **no**. |
| `queue_without_daemon` | (Red Hat Enterprise Linux 5.3 and later) If set to **no**, the **multipathd** daemon will disable queueing for all devices when it is shut down. As of Red Hat Enterprise Linux release 5.9, the default value is **no**; for earlier releases, the default value is **yes**. |
| `user_friendly_names` | If set to **yes**, specifies that the system should using the **bindings** file to assign a persistent and unique alias to the multipath, in the form of **mpath*n***. The default location of the **bindings** file is **/var/lib/multipath/bindings**, but this can be changed with the **bindings_file** option. If set to **no**, specifies that the system should use use the WWID as the alias for the multipath. In either case, what is specified here will be overridden by any device-specific aliases you specify in the **multipaths** section of the configuration file. The default value is **no**. |
| `bindings_file` | (Red Hat Enterprise Linux 5.2 and later) The location of the **bindings** file that is used with the **user_friendly_names** option. The default value is **/var/lib/multipath/bindings**. |
| `mode` | (Red Hat Enterprise Linux 5.3 and later) The mode to use for the multipath device nodes, in octal. The default value is determined by the process. |
| `uid` | (Red Hat Enterprise Linux 5.3 and later) The user ID to use for the multipath device nodes. You must use the numeric user ID. The default value is determined by the process. |
| `gid` | (Red Hat Enterprise Linux 5.3 and later) The group ID to use for the multipath device nodes. You must use the numeric group ID. The default value is determined by the process. |
| `checker_timeout` | (Red Hat Enterprise Linux 5.5 and later) The timeout value to use for path checkers that issue SCSI commands with an explicit timeout, in seconds. The default value is taken from **sys/block/sd*x*/device/timeout**. |
| `pg_prio_calc` | (Red Hat Enterprise Linux 5.6 and later) If set to **avg**, sets the priority of path groups to the average of the priorities of the included paths. If set to **sum**, sets the priority of path groups to the sum of the priorities of the included paths. As of Red Hat Enterprise Linux release 5.9, the default value is **avg**; for earlier releases, the default value is **sum**. |
| `log_checker_err` | (Red Hat Enterprise Linux 5.6 and later) If set to **once**, a path checker error is logged once at logging level 2 and any later errors are logged at level 3 until the device is restored. If set to **always**, every path checker error is logged at logging level 2. The default value is **always**. |

| Attribute | Description |
|---|---|
| `fast_io_fail_tmo` | (Red Hat Enterprise Linux 5.7 and later) The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before failing I/O to devices on that remote port. This value should be smaller than the value of `dev_loss_tmo`. Setting this to `off` will disable the timeout. The default value is determined by the OS. |
| `dev_loss_tmo` | (Red Hat Enterprise Linux 5.7 and later) The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before removing it from the system. The default value is determined by the OS. |
| `file_timeout` | (Red Hat Enterprise Linux 5.7 and later) The number of seconds multipath will wait for a necessary file to appear while setting up a multipath device. The default value is `90`. |

## 4.4. Multipaths Device Configuration Attributes

Table 4.2, "Multipath Attributes" shows the attributes that you can set in the `multipaths` section of the `multipath.conf` configuration file for each specific multipath device. These attributes apply only to the one specified multipath. These defaults are used by DM-Multipath and override attributes set in the `defaults` and `devices` sections of the `multipath.conf` file.

**Table 4.2. Multipath Attributes**

| Attribute | Description |
|---|---|
| `wwid` | Specifies the WWID of the multipath device to which the `multipath` attributes apply. |
| `alias` | Specifies the symbolic name for the multipath device to which the `multipath` attributes apply. |
| `path_grouping_policy` | Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include:<br><br>`failover` = 1 path per priority group<br><br>`multibus` = all valid paths in 1 priority group<br><br>`group_by_serial` = 1 priority group per detected serial number<br><br>`group_by_prio` = 1 priority group per path priority value<br><br>`group_by_node_name` = 1 priority group per target node name |
| `prio_callout` | Specifies the default program and arguments to call out to obtain a path weight. Weights are summed for each path group to determine the next path group to use in case of failure. "none" is a valid value. |
| `path_selector` | Specifies the default algorithm to use in determining what path to use for the next I/O operation. |

| Attribute | Description |
|---|---|
| **failback** | Specifies path group failback.<br><br>A value of 0 or **immediate** specifies that as soon as there is a path group with a higher priority than the current path group the system switches to that path group.<br><br>A numeric value greater than zero specifies deferred failback, expressed in seconds.<br><br>A value of **manual** specifies that failback can happen only with operator intervention. |
| **rr_weight** | If set to **priorities**, then instead of sending **rr_min_io** requests to a path before calling **selector** to choose the next path, the number of requests to send is determined by **rr_min_io** times the path's priority, as determined by the **prio_callout** program. Currently, there are priority callouts only for devices that use the **group_by_prio** path grouping policy, which means that all the paths in a path group will always have the same priority.<br><br>If set to **uniform**, all path weights are equal. |
| **no_path_retry** | A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queueing.<br><br>A value of **fail** indicates immediate failure, without queueing.<br><br>A value of **queue** indicates that queueing should not stop until the path is fixed. |
| **user_friendly_names** | If set to **yes**, specifies that the system should use the **/etc/multipath/bindings** file to assign a persistent and unique alias to the multipath, in the form of **mpath***n*. If set to **no**, specifies that the system should use use the WWID as the alias for the multipath. In either case, what is specified here will be overridden by any device-specific aliases you specify with the **alias** attribute for a specific multipath device. |
| **flush_on_last_del** | (Red Hat Enterprise Linux 5.3 and later) If set to **yes**, the **multipathd** daemon will disable queueing when the last path to a device has been deleted. |
| **rr_min_io** | Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. |
| **mode** | (Red Hat Enterprise Linux 5.3 and later) The mode to use for the multipath device nodes, in octal. The default value is determined by the process. |
| **uid** | (Red Hat Enterprise Linux 5.3 and later) The user ID to use for the multipath device nodes. You must use the numeric user ID. The default value is determined by the process. |
| **gid** | (Red Hat Enterprise Linux 5.3 and later) The group ID to use for the multipath device nodes. You must use the numeric group ID. The default value is determined by the process. |

The following example shows multipath attributes specified in the configuration file for two specific multipath devices. The first device has a WWID of **3600508b4000156d70001200000b0000** and a symbolic name of **yellow**.

The second multipath device in the example has a WWID of **1DEC_____321816758474** and a symbolic name of **red**. In this example, the **rr_weight** attributes is set to **priorities**.

```
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. Configuration File Devices

Table 4.3, "Device Attributes" shows the attributes that you can set for each individual storage device in the **devices** section of the **multipath.conf** configuration file. These attributes are used by DM-Multipath unless they are overwritten by the attributes specified in the **multipaths** section of the **multipath.conf** file for paths that contain the device. These attributes override the attributes set in the **defaults** section of the **multipath.conf** file.

Many devices that support multipathing are included by default in a multipath configuration. The values for the devices that are supported by default are listed in the **multipath.conf.defaults** file. You probably will not need to modify the values for these devices, but if you do you can overwrite the default values by including an entry in the configuration file for the device that overwrites those values. You can copy the device configuration defaults from the **multipath.conf.defaults** file for the device and override the values that you want to change.

To add a device to this section of the configuration file that is not configured automatically by default, you need to set the **vendor** and **product** parameters. You can find these values by looking at **/sys/block/*device_name*/device/vendor** and **/sys/block/*device_name*/device/model** where *device_name* is the device to be multipathed, as in the following example:

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

The additional parameters to specify depend on your specific device. If the device is active/active, you will usually not need to set additional parameters. You may want to set **path_grouping_policy** to **multibus**. Other parameters you may need to set are **no_path_retry** and **rr_min_io**, as described in Table 4.3, "Device Attributes".

If the device is active/passive, but it automatically switches paths with I/O to the passive path, you need to change the checker function to one that does not send I/O to the path to test if it is working (otherwise, your device will keep failing over). This almost always means that you set the **path_checker** to **tur**; this works for all SCSI devices that support the Test Unit Ready command, which most do.

If the device needs a special command to switch paths, then configuring this device for multipath requires a hardware handler kernel module. The current hardware handlers are **emc** and **rdac**. If these are not sufficient for your device, you may not be able to configure the device for multipath.

**Table 4.3. Device Attributes**

| Attribute | Description |
|---|---|
| **vendor** | Specifies the vendor name of the storage device to which the **device** attributes apply, for example **COMPAQ**. |
| **product** | Specifies the product name of the storage device to which the **device** attributes apply, for example **HSV110 (C)COMPAQ**. |
| **path_grouping_policy** | Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include:<br><br>**failover** = 1 path per priority group<br><br>**multibus** = all valid paths in 1 priority group<br><br>**group_by_serial** = 1 priority group per detected serial number<br><br>**group_by_prio** = 1 priority group per path priority value<br><br>**group_by_node_name** = 1 priority group per target node name |
| **getuid_callout** | Specifies the default program and arguments to call out to obtain a unique path identifier. An absolute path is required.<br><br>This parameter accepts the following wildcard values:<br><br>**%n**: replaced with the device name as it appears in sysfs.<br><br>**%c**: replaced with the device name as it appears in **/dev** (this is different than the sysfs name for CCISS devices). (Red Hat Enterprise Linux 5.8 and later)<br><br>**%d**: replaced with *major_nr:minor_nr*<br><br>**%h**: replaced with *host_nr*:*bus_nr*:*target_nr*:*lun_nr*. |
| **prio_callout** | Specifies the default program and arguments to call out to obtain a path weight. Weights are summed for each path group to determine the next path group to use in case of failure. "none" is a valid value. This parameter accepts the same wildcard values as **getuid_callout**. |
| **path_checker** | Specifies the default method used to determine the state of the paths. Possible values include **readsector0**, **rdac**, **tur**, **cciss_tur**, **hp_tur**, **emc_clariion**, **hp_sw**, and **directio**. |
| **path_selector** | Specifies the default algorithm to use in determining what path to use for the next I/O operation. |

| Attribute | Description |
|---|---|
| **features** | The default extra features of multipath devices, using the format: "*number_of_features_plus_arguments feature1* ..." Possible values for features include:<br><br>**queue_if_no_path**, which is the same as setting **no_path_retry** to **queue**. For information on issues that may arise when using this feature, see Section 5.4, "Issues with queue_if_no_path feature".<br><br>**pg_init_retries** *n*: Retry path group initialization up to *n* times before failing where *n* is between 1 and 50, inclusive.<br><br>**pg_init_delay_secs** *n*: Wait *n* seconds between path group initialization retries where *n* is between 0 and 50, inclusive. |
| **hardware_handler** | Specifies a module that will be used to perform hardware specific actions when switching path groups or handling I/O errors. Possible values include **0**, **1 emc**, and **1 rdac**. The default value is **0**. |
| **rr_weight** | If set to **priorities**, then instead of sending **rr_min_io** requests to a path before calling **selector** to choose the next path, the number of requests to send is determined by **rr_min_io** times the path's priority, as determined by the **prio_callout** program. Currently, there are priority callouts only for devices that use the **group_by_prio** path grouping policy, which means that all the paths in a path group will always have the same priority.<br><br>If set to **uniform**, all path weights are equal. |
| **no_path_retry** | A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queueing.<br><br>A value of **fail** indicates immediate failure, without queueing.<br><br>A value of **queue** indicates that queueing should not stop until the path is fixed. |
| **failback** | Specifies path group failback.<br><br>A value of 0 or **immediate** specifies that as soon as there is a path group with a higher priority than the current path group the system switches to that path group.<br><br>A numeric value greater than zero specifies deferred failback, expressed in seconds.<br><br>A value of **manual** specifies that failback can happen only with operator intervention. |
| **rr_min_io** | Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. |

| Attribute | Description |
|---|---|
| `fast_io_fail_tmo` | (Red Hat Enterprise Linux release 5.7 and later) The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before failing I/O to devices on that remote port. This value should be smaller than the value of **dev_loss_tmo**. Setting this to **off** will disable the timeout. |
| `dev_loss_tmo` | (Red Hat Enterprise Linux release 5.7 and later) The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before removing it from the system. |
| `flush_on_last_del` | (Red Hat Enterprise Linux 5.3 and later) If set to **yes**, the **multipathd** daemon will disable queueing when the last path to a device has been deleted. |
| `user_friendly_names` | If set to **yes**, specifies that the system should use the **/etc/multipath/bindings** file to assign a persistent and unique alias to the multipath, in the form of **mpath*n***. If set to **no**, specifies that the system should use the WWID as the alias for the multipath. In either case, what is specified here will be overridden by any device-specific aliases you specify in the **multipaths** section of the configuration file. |
| `product_blacklist` | Specifies a regular expression used to blacklist devices by product. |

The following example shows a **device** entry in the multipath configuration file.

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

# Chapter 5. DM-Multipath Administration and Troubleshooting

This chapter will provide information on administering DM-Multipath on a running system. It includes sections on the following topics:

≫ Resizing an Online Multipath Device

≫ The Multipath Daemon

≫ Issues with Large Number of LUNs

≫ Issues with queue_if_no_path feature

≫ Multipath Command Output

≫ Multipath Queries with multipath Command

≫ Multipath Command Options

≫ Multipath Queries with dmsetup Command

≫ Troubleshooting with the multipathd Interactive Console

## 5.1. Resizing an Online Multipath Device

If you need to resize an online multipath device, use the following procedure.

1. Resize your physical device.

2. Use the following command to find the paths to the LUN:

   ```
   # multipath -l
   ```

3. Resize your paths. For SCSI devices, writing a 1 to the **rescan** file for the device causes the SCSI driver to rescan, as in the following command:

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

4. Resize your multipath device by running the multipathd resize command:

   ```
   # multipathd -k'resize map mpath0'
   ```

5. Resize the filesystem (assuming no LVM or DOS partitions are used):

   ```
   # resize2fs /dev/mapper/mpath0
   ```

For further information on resizing an online LUN, see the *Online Storage Reconfiguration Guide.*

## 5.2. The Multipath Daemon

If you find you have trouble implementing a multipath configuration, you should ensure that the multipath daemon is running, as described in Chapter 3, *Setting Up DM-Multipath*. The **multipathd** daemon must be running in order to use multipathed devices.

## 5.3. Issues with Large Number of LUNs

When a large number of LUNs are added to a node, using multipathed devices can significantly increase the time it takes for the **udev** device manager to create device nodes for them. If you experience this problem, you can correct it by deleting the following line in **/etc/udev/rules.d/40-multipath.rules**:

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

This line causes the **udev** device manager to run **multipath** every time a block device is added to the node. Even with this line removed, the **multipathd** daemon will still automatically create multipathed devices, and **multipath** will still be called during the boot process for nodes with multipathed root file systems. The only change is that multipathed devices will not be automatically created when the **multipathd** daemon is not running, which should not be a problem for the vast majority of multipath users.

## 5.4. Issues with queue_if_no_path feature

If **features "1 queue_if_no_path"** is specified in the **/etc/multipath.conf** file, then any process that issues I/O will hang until one or more paths are restored. To avoid this, set the **no_path_retry** *N* parameter in the **/etc/multipath.conf** file (where *N* is the number of times the system should retry a path).

When you set the **no_path_retry** parameter, remove the **features "1 queue_if_no_path"** option from the **/etc/multipath.conf** file as well. If, however, you are using a multipathed device for which the **features "1 queue_if_no_path"** option is set as a compiled-in default, as it is for many SAN devices, you must explicitly add **features "0"** to override this default. You can do this by copying the existing devices section for your device from **/usr/share/doc/device-mapper-multipath-0.4.7/multipath.conf.defaults** into **/etc/multipath.conf** and editing it to suit your needs.

If you need to use the **features "1 queue_if_no_path"** option and you experience the issue noted here, use the **dmsetup** command to edit the policy at runtime for a particular LUN (that is, for which all the paths are unavailable). For example, if you want to change the policy on the multipath device **mpath2** from **"queue_if_no_path"** to **"fail_if_no_path"**, execute the following command.

```
dmsetup message mpath2 0 "fail_if_no_path"
```

Note that you must specify the **mpath*n*** alias rather than the path.

## 5.5. Multipath Command Output

When you create, modify, or list a multipath device, you get a printout of the current device setup. The format is as follows.

For each multipath device:

```
action_if_any: alias (wwid_if_different_from_alias) [size][features]
[hardware_handler]
```

For each path group:

```
\_ scheduling_policy [path_group_priority_if_known]
[path_group_status_if_known]
```

For each path:

```
\_ host:channel:id:lun devnode major:minor [path_status] [dm_status_if_known]
```

For example, the output of a multipath command might appear as follows:

```
mpath1 (3600d0230003228bc000339414edb8101) [size=10 GB][features="0"]
[hwhandler="0"]
\_ round-robin 0 [prio=1][active]
 \_ 2:0:0:6 sdb 8:16 [active][ready]
\_ round-robin 0 [prio=1][enabled]
 \_ 3:0:0:6 sdc 8:64 [active][ready]
```

If the path is up and ready for I/O, the status of the path is **ready** or **active**. If the path is down, the status is **faulty** or **failed**. The path status is updated periodically by the **multipathd** daemon based on the polling interval defined in the **/etc/multipath.conf** file.

The dm status is similar to the path status, but from the kernel's point of view. The dm status has two states: **failed**, which is analogous to **faulty**, and **active** which covers all other path states. Occasionally, the path state and the dm state of a device will temporarily not agree.

> **Note**
>
> When a multipath device is being created or modified, the path group status and the dm status are not known. Also, the features are not always correct. When a multipath device is being listed, the path group priority is not known.

## 5.6. Multipath Queries with multipath Command

You can use the **-l** and **-ll** options of the **multipath** command to display the current multipath configuration. The **-l** option displays multipath topology gathered from information in **sysfs** and the device mapper. The **-ll** option displays the information the **-l** displays in addition to all other available components of the system.

When displaying the multipath configuration, there are three verbosity levels you can specify with the **-v** option of the **multipath** command. Specifying **-v0** yields no output. Specifying **-v1** outputs the created or updated multipath names only, which you can then feed to other tools such as **kpartx**. Specifying **-v2** prints all detected paths, multipaths, and device maps.

The following example shows the output of a **multipath -l** command.

```
# multipath -l
mpath1 (3600d0230003228bc000339414edb8101)
[size=10 GB][features="0"][hwhandler="0"]
\_ round-robin 0 [prio=1][active]
 \_ 2:0:0:6 sdb 8:16 [active][ready]
\_ round-robin 0 [prio=1][enabled]
 \_ 3:0:0:6 sdc 8:64 [active][ready]
```

## 5.7. Multipath Command Options

Table 5.1, "Useful `multipath` Command Options" describes some options of the `multipath` command that you may find useful.

**Table 5.1. Useful `multipath` Command Options**

| Option | Description |
| --- | --- |
| `-l` | Display the current multipath configuration gathered from `sysfs` and the device mapper. |
| `-ll` | Display the current multipath configuration gathered from `sysfs`, the device mapper, and all other available components on the system. |
| `-f device` | Remove the named multipath device. |
| `-F` | Remove all unused multipath devices. |

## 5.8. Determining Device Mapper Entries with the dmsetup Command

You can use the `dmsetup` command to find out which device mapper entries match the multipathed devices.

The following command displays all the device mapper devices and their major and minor numbers. The minor numbers determine the name of the dm device. For example, a minor number of 3 corresponds to the multipathed device **/dev/dm-3**.

```
# dmsetup ls
mpath2  (253, 4)
mpath4p1       (253, 12)
mpath5p1       (253, 11)
mpath1  (253, 3)
mpath6p1       (253, 14)
mpath7p1       (253, 13)
mpath0  (253, 2)
mpath7  (253, 9)
mpath6  (253, 8)
VolGroup00-LogVol01     (253, 1)
mpath5  (253, 7)
VolGroup00-LogVol00     (253, 0)
mpath4  (253, 6)
mpath1p1       (253, 10)
mpath3  (253, 5)
```

## 5.9. Troubleshooting with the multipathd Interactive Console

The `multipathd -k` command is an interactive interface to the `multipathd` daemon. Entering this command brings up an interactive multipath console. After entering this command, you can enter `help` to get a list of available commands, you can enter a interactive command, or you can enter `CTRL-D` to quit.

The `multipathd` interactive console can be used to troubleshoot problems you may be having with your system. For example, the following command sequence displays the multipath configuration, including the defaults, before exiting the console.

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

```
> > CTRL-D
```

The following command sequence ensures that multipath has picked up any changes to the
**multipath.conf**,

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

Use the following command sequence to ensure that the path checker is working properly.

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

# Appendix A. Revision History

| | | |
|---|---|---|
| **Revision 9.0-10** | **Mon Sep 8 2014** | **Steven Levine** |

Version for 5.11 GA release

| | | |
|---|---|---|
| **Revision 9.0-7** | **Mon Jul 7 2014** | **Steven Levine** |

Resolves #1117042
Documents updated features parameter of configuration defaults.

| | | |
|---|---|---|
| **Revision 9.0-6** | **Mon Jun 30 2014** | **Steven Levine** |

Beta release of Red Hat Enterprise Linux 5.11

| | | |
|---|---|---|
| **Revision 9.0-5** | **Tue Jun 17 2014** | **Steven Levine** |

Resolves: # 1114667
Adds note indicating that migrating the root filesystem from single path storage to multipath storage is not supported and requires a new installation.

| | | |
|---|---|---|
| **Revision 8.0-3** | **Mon Sep 30 2013** | **Steven Levine** |

Version for Red Hat Enterprise Linux 5.10 GA release

| | | |
|---|---|---|
| **Revision 8.0-1** | **Wed Jul 10 2013** | **Steven Levine** |

Beta release of Red Hat Enterprise Linux 5.10

| | | |
|---|---|---|
| **Revision 7.0-7** | **Fri Jan 4 2013** | **Steven Levine** |

Version for Red Hat Enterprise Linux 5.9 GA release

| | | |
|---|---|---|
| **Revision 7.0-5** | **Thu Aug 23 2012** | **Steven Levine** |

Beta release of Red Hat Enterprise Linux 5.9

Corrects small capitalization error.

| | | |
|---|---|---|
| **Revision 7.0-2** | **Thu Aug 9 2012** | **Steven Levine** |

Fixes error in Multipath Attributes table.

| | | |
|---|---|---|
| **Revision 7.0-1** | **Tue Jul 31 2012** | **Steven Levine** |

Resolves: #839007
Documents support for user_friendly_names and flush_on_last_del in multipaths and devices section of multipath configuration file. Updates documentation of default values.

| | | |
|---|---|---|
| **Revision 6.0-3** | **Thu Feb 16 2012** | **Steven Levine** |

Release for GA of Red Hat Enterprise Linux 5.8

| | | |
|---|---|---|
| **Revision 6.0-2** | **Thu Dec 15 2011** | **Steven Levine** |

Beta release of Red Hat Enterprise Linux 5.8

| | | |
|---|---|---|
| **Revision 6.0-1** | **Wed Nov 16 2011** | **Steven Levine** |

Resolves: #741463
Documents new parameter support in Red Hat Enterprise Linux 5.8.

Resolves: #753942
Corrects description of **`multipath -F`** parameter.

| **Revision 5.0-1** | **Thu Jul 21 2011** | **Steven Levine** |
| --- | --- | --- |

Initial revision for Red Hat Enterprise Linux 5.7 release

Resolves: #703610
Documents new configuration file parameters.

Resolves: #510333
Clarifies procedure for maintaining consistent device names across a cluster.

| **Revision 4.0-1** | **Thu Dec 23 2010** | **Steven Levine** |
| --- | --- | --- |

Initial revision for Red Hat Enterprise Linux 5.6 release

Resolves: #661532
Updates configuration file documentation to include Red Hat Enterprise Linux 5.6 options and parameters.

Resolves: #654525
Corrects small typos in document.

| **Revision 3.0-1** | **Wed Mar 10 2010** | **Steven Levine** |
| --- | --- | --- |

Initial revision for Red Hat Enterprise Linux 5.5 release

Resolves: #561167
Updates configuration file documentation to include Red Hat Enterprise Linux 5.5 options and parameters.

Resolves: #541836
Fixes minor typos.

Resolves: #541836
Fixes minor typo.

| **Revision 2.0-1** | **Tue Aug 18 2009** | **Steven Levine** |
| --- | --- | --- |

Initial revision for Red Hat Enterprise Linux 5.4 release

Resolves: #466970
Adds note clarifying that if a multipathed device contains partitions, you cannot create a logical volume using the whole device without removing the partitions.

Resolves #513330
Elaborates documentation to address issues with queue_if_no_path features option.

Resolves #510707
Adds a procedure for resizing online multipath devices.

Resolves #498986
Fixes minor typo.

Resolves #498988
Fixes minor typo.

| **Revision 1.0-1** | **Thu Jan 29 2009** |
|---|---|

First version

# Index

## Symbols

**/etc/multipath.conf package, Setting Up DM-Multipath**

## A

**active/active configuration**
- definition, Overview of DM-Multipath
- illustration, Overview of DM-Multipath

**active/passive configuration**
- definition, Overview of DM-Multipath
- illustration, Overview of DM-Multipath

**alias parameter , Multipaths Device Configuration Attributes**
- configuration file, Multipath Device Identifiers

## B

**bindings_file parameter, Configuration File Defaults**

**blacklist**
- configuration file, Configuration File Blacklist
- default devices, Blacklisting By Device Name
- device name, Blacklisting By Device Name
- device type, Blacklisting By Device Type
- in configuration file, Setting Up DM-Multipath
- WWID, Blacklisting by WWID

**blacklist_exceptions section**
- multipath.conf file, Blacklist Exceptions

## C

## D