

Red Hat Enterprise Linux 4

DM Multipath

DM Multipath Configuration and Administration



Red Hat Enterprise Linux 4 DM Multipath

DM Multipath Configuration and Administration

Edition 1.0

Copyright © 2009 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, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

All other trademarks are the property of their respective owners.

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701

This book provides information on using the Device-Mapper Multipath feature of Red Hat Enterprise Linux 4.

| | |
|---|-----------|
| Preface | v |
| 1. Audience | v |
| 2. Related Documentation | v |
| 3. Feedback | vi |
| 4. Document Conventions | vi |
| 4.1. Typographic Conventions | vi |
| 4.2. Pull-quote Conventions | vii |
| 4.3. Notes and Warnings | viii |
| 1. Device Mapper Multipathing | 1 |
| 1.1. Overview of DM-Multipath | 1 |
| 1.2. Storage Array Support | 3 |
| 1.3. DM-Multipath Components | 3 |
| 1.4. DM-Multipath Setup Overview | 4 |
| 2. Multipath Devices | 5 |
| 2.1. Multipath Device Identifiers | 5 |
| 2.2. Consistent Multipath Device Names in a Cluster | 5 |
| 2.3. Multipath Device Attributes | 6 |
| 2.4. Multipath Devices in Logical Volumes | 6 |
| 3. Setting Up DM-Multipath | 7 |
| 3.1. Setting Up DM-Multipath | 7 |
| 3.2. Ignoring Local Disks when Generating Multipath Devices | 8 |
| 3.3. Adding Devices to the Multipathing Database | 10 |
| 4. The DM-Multipath Configuration File | 11 |
| 4.1. Configuration File Overview | 11 |
| 4.2. Configuration File Blacklist | 12 |
| 4.2.1. Blacklisting by WWID | 12 |
| 4.2.2. Blacklisting By Device Name | 12 |
| 4.3. Configuration File Defaults | 13 |
| 4.4. Multipaths Device Configuration Attributes | 16 |
| 4.5. Configuration File Devices | 17 |
| 5. DM-Multipath Administration and Troubleshooting | 21 |
| 5.1. Multipath Command Output | 21 |
| 5.2. Multipath Queries with multipath Command | 22 |
| 5.3. Multipath Command Options | 22 |
| 5.4. Determining Device Mapper Entries with the dmsetup Command | 22 |
| 5.5. Troubleshooting with the multipathd Interactive Console | 23 |
| 5.6. Resizing an Online Multipathed Device (RHEL 4.8 and later) | 23 |
| A. Revision History | 25 |
| Index | 27 |

Preface

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

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.
- *Red Hat Enterprise Linux Introduction to System Administration* — Provides introductory information for new Red Hat Enterprise Linux system administrators.
- *Red Hat Enterprise Linux System Administration Guide* — Provides more detailed information about configuring Red Hat Enterprise Linux to suit your particular needs as a user.
- *Red Hat Enterprise Linux Reference Guide* — Provides detailed information suited for more experienced users to reference when needed, as opposed to step-by-step instructions.
- *Red Hat Enterprise Linux Security Guide* — Details the planning and the tools involved in creating a secured computing environment for the data center, workplace, and home.

For more information about Red Hat Cluster Suite for Red Hat Enterprise Linux, 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.
- *Global File System: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS (Red Hat Global File System).
- *LVM Administrator's Guide: Configuration and Administration* — Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- *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 and PDF versions online at the following location:

<http://www.redhat.com/docs>

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/>) against the component **rh-cs**.

Be sure to mention the manual's identifier:

```
rh-MPIO(EN)-4.8 (2009-05-13T12:48)
```

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.

4. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*¹ set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

4.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to the first virtual terminal. Press **Ctrl+Alt+F1** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

¹ <https://fedorahosted.org/liberation-fonts/>

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh *john@example.com***.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount */home***.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

4.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo             echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

4.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.



Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

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. 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.

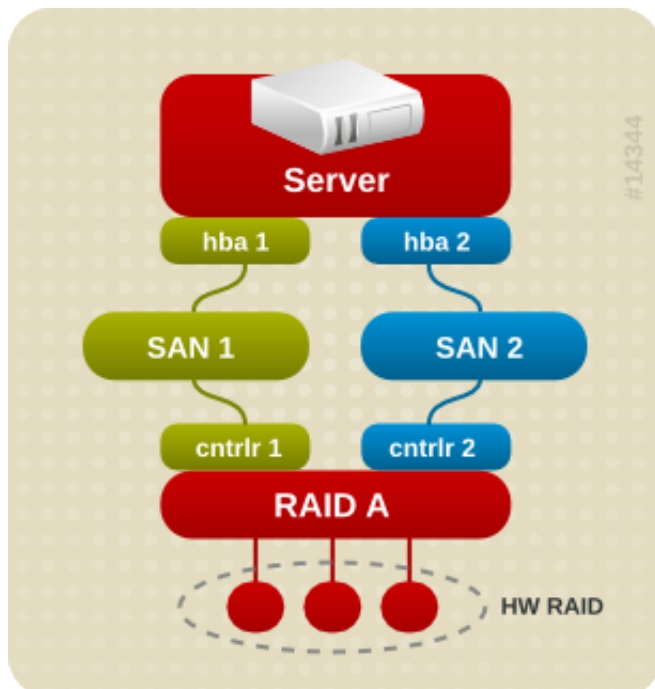


Figure 1.1. Active/Passive Multipath Configuration with One RAID Device

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.

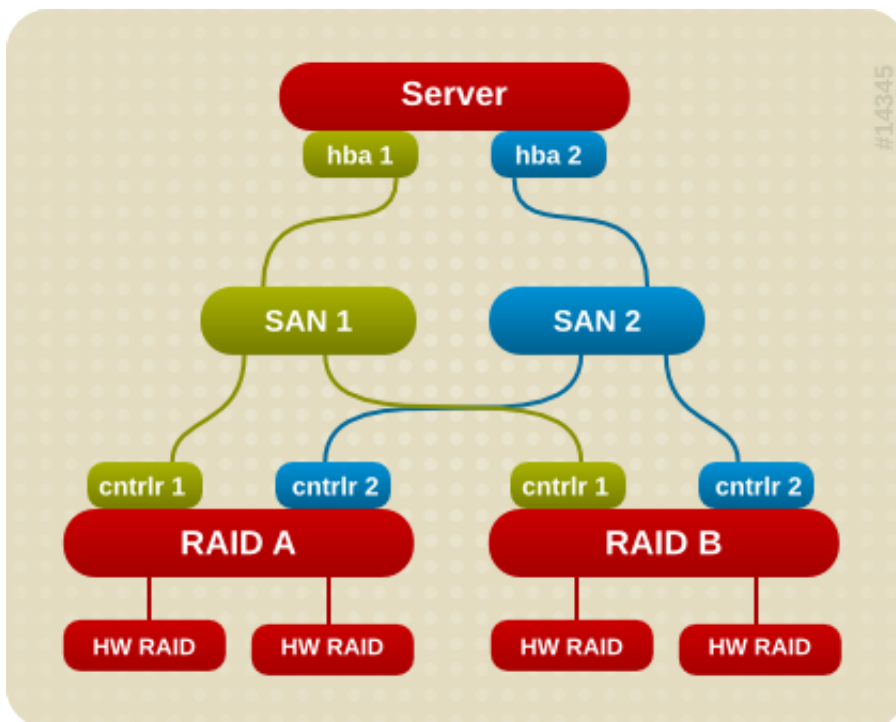


Figure 1.2. Active/Passive Multipath Configuration with Two RAID Devices

As in the example shown in *Figure 1.1, “Active/Passive Multipath Configuration with One RAID Device”*, there are two I/O paths to each 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.

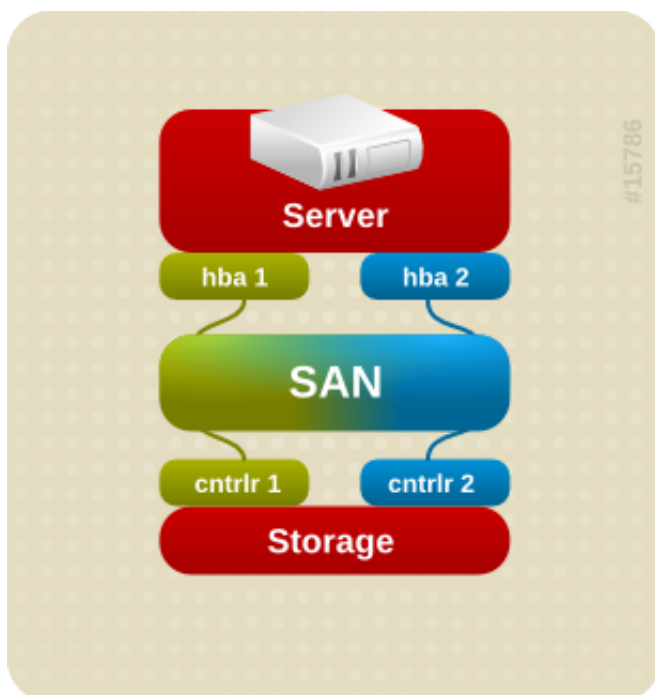


Figure 1.3. Active/Active Multipath Configuration with One RAID Device

1.2. 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.

1.3. DM-Multipath Components

[Table 1.1, “DM-Multipath Components”](#), describes the components of DM-Multipath.

Table 1.1. DM-Multipath Components

| Component | Description |
|---|--|
| <code>dm-multipath</code> kernel module | Reroutes I/O and supports failover for paths and path groups. |
| <code>multipath</code> command | Lists and configures multipath devices. Normally started up with <code>/etc/rc.sysinit</code> , it can also be started up by a <code>udev</code> program whenever a block device is added or it can be run by the <code>initramfs</code> file system. |
| <code>multipathd</code> 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 <code>/etc/multipath.conf</code> file. |
| <code>kpartx</code> 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 <code>kpartx</code> is provided in its own package, but the <code>device-mapper-multipath</code> package depends on it. |

1.4. 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.

Detailed setup instructions for several example multipath configurations are provided in see [Chapter 3, Setting Up DM-Multipath](#).

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 **mpathn**.

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 **mpathn**.

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/mpathn**, **/dev/mpath/mpathn**, 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. 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 in the cluster you perform one of the following procedures:

- Use the **alias** option in the **multipaths** section of the multipath configuration file to set the name of the multipath device. The **alias** for the multipath device is consistent across all the nodes in a cluster. For information on the **multipaths** section of the multipath configuration file, see [Section 4.4, “Multipaths Device Configuration Attributes”](#).

- If you want the system-defined user-friendly names to be consistent across all nodes in the cluster, set up all of the multipath devices on one machine. Then copy the **bindings** file from that machine to all the other machines in the cluster. The bindings file is located at `/var/lib/multipath/bindings` by default, but as of RHEL 4.6 and later you can set this value to a different location with the **bindings_file** parameter of the **defaults** section of the configuration file.

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.

```
pvccreate /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.

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** file 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 multipath configuration file (**lvm.conf**), include the following filter in the **devices** section of the file.

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

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.

```
devnode_blacklist {
    devnode "*"
}
```

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

```
# devnode_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 **mpathn**; 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, but only for the devices created by this command. If the command does yield any output, you can check your multipath devices as follows:

- Run the **multipath -ll** command. This lists all the multipath devices.
- If running the **multipath -ll** command does not show the device, verify that multipath is configured properly by checking the **/etc/multipath** file and making sure that the SCSI devices you want to be multipathed exist on the system.
- If the SCSI devices do not appear, ensure that all SAN connections are set up properly.

For further information on the **multipath** command and its output, see [Section 5.1, “Multipath Command Output”](#), see [Section 5.2, “Multipath Queries with multipath Command”](#), and see [Section 5.3, “Multipath Command Options”](#).

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/mpathn**. 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 5.1, “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
```

- In order to prevent the device mapper from mapping `/dev/sda` in its multipath maps, edit the `devnode_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 -vs` 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.

```
devnode_blacklist {
    wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}
```

- 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”](#).

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-0.4.5/multipath.conf.defaults` file.



Note

The location and contents of the `multipath.conf.defaults` file may differ slightly from the example shown in this chapter depending on the release of RHEL 4 that you are running.

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-0.4.5/multipath.conf.annotated`.

4.1. Configuration File Overview

The multipath configuration file is divided into the following sections:

devnode_blacklist

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

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 **devnode_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.

```
devnode_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 default, a variety of device types are blacklisted, even after you comment out the initial **devnode_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 **devnode_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 are sd* devices.

```
devnode_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 entire blacklist do not generally support DM-Multipath.

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

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
#    rr_weight               priorities
#    failback                immediate
#    no_path_retry           fail
#    user_friendly_name      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 . |
| polling_interval | Specifies the interval between two path checks in seconds. The default value is 5. |
| 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 . |
| path_grouping_policy | Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include: failover = 1 path per priority group multibus = all valid paths in 1 priority group group_by_serial = 1 priority group per detected serial number group_by_prio = 1 priority group per path priority value group_by_node_name = 1 priority group per target node name 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. The default value is /sbin/scsi_id -g -u -s . |
| prio_callout | Specifies the the default program and arguments to call out to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable prio value for example. "none" is a valid value. The default value is no callout, indicating all paths are equal |
| features | Specifies the default extra features of multipath devices. The only existing feature is queue_if_no_path . The default value is (null). |
| path_checker | Specifies the default method used to determine the state of the paths. Possible values include: readsector0 , tur , emc_clariion , hp_sw , and directio . The default value is readsector0 . |
| failback | Specifies path group failback. 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. A numeric value greater than zero specifies deferred failback, expressed in seconds. A value of manual specifies that failback can happen only with operator intervention. The default value is manual . |

| Attribute | Description |
|----------------------------|--|
| 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 | (RHEL 4.7 and later) Sets the maximum number of open file descriptors for the multipathd process. A value of max sets the number of open file descriptors to the system maximum. |
| 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. If set to uniform , all path weights are equal. The default value is uniform . |
| 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. A value of fail indicates immediate failure, without queueing. A value of queue indicates that queueing should not stop until the path is fixed. The default value is (null). |
| flush_on_last_del | (RHEL 4.7 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 . |
| user_friendly_names | If set to yes , specifies that the system should use the bindings file /var/lib/multipath/bindings to assign a persistent and unique alias to the multipath, in the form of mpathn . 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. The default value is no . |
| bindings_file | (RHEL 4.6 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 | (RHEL 4.7 and later) The mode to use for the multipath device nodes, in octal. The default value is determined by the process. |
| uid | (RHEL 4.7 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 | (RHEL 4.7 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. |

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: failover = 1 path per priority group multibus = all valid paths in 1 priority group group_by_serial = 1 priority group per detected serial number group_by_prio = 1 priority group per path priority value group_by_node_name = 1 priority group per target node name |
| path_selector | Specifies the default algorithm to use in determining what path to use for the next I/O operation. |
| rr_min_io | (RHEL 4.8 and later) Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. |
| failback | Specifies path group failback. 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. A numeric value greater than zero specifies deferred failback, expressed in seconds. 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. 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. A value of fail indicates immediate failure, without queueing. A value of queue indicates that queueing should not stop until the path is fixed. |

| Attribute | Description |
|--------------------------|---|
| flush_on_last_del | (RHEL 4.7 and later) If set to yes , the multipathd daemon will disable queuing when the last path to a device has been deleted. The default value is no . |
| mode | (RHEL 4.7 and later) The mode to use for the multipath device nodes, in octal. The default value is determined by the process. |
| uid | (RHEL 4.7 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 | (RHEL 4.7 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_checker        readsector0
        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 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:

```
[root@cypher-06 ~]# cat /sys/block/sda/device/vendor
WINSYS
[root@cypher-06 ~]# 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 IO/ 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 |
|-----------------------------------|---|
| <code>vendor</code> | Specifies the vendor name of the storage device to which the <code>device</code> attributes apply, for example <code>COMPAQ</code> . |
| <code>product</code> | Specifies the product name of the storage device to which the <code>device</code> attributes apply, for example <code>HSV110 (C)COMPAQ</code> . |
| <code>bl_product</code> | Specifies a regular expression used to blacklist devices by vendor/ product. Note that for a device to get blacklisted, the <code>vendor</code> , <code>product</code> , and <code>bl_product</code> strings must all match. |
| <code>path_grouping_policy</code> | Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include: <code>failover</code> = 1 path per priority group <code>multibus</code> = all valid paths in 1 priority group <code>group_by_serial</code> = 1 priority group per detected serial number <code>group_by_prio</code> = 1 priority group per path priority value <code>group_by_node_name</code> = 1 priority group per target node name |
| <code>getuid_callout</code> | Specifies the default program and arguments to call out to obtain a unique path identifier. An absolute path is required. |
| <code>prio_callout</code> | Specifies the 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 failue. "none" is a valid value. |
| <code>path_checker</code> | Specifies the default method used to determine the state of the paths. Possible values include <code>readsector0</code> and <code>tur</code> , <code>emc_clariion</code> , <code>hp_sw</code> , and <code>directio</code> . |
| <code>path_selector</code> | 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. 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. A numeric value greater than zero specifies deferred failback, expressed in seconds. A value of manual specifies that failback can happen only with operator intervention. |
| features | The extra features of multipath devices. The only existing feature is queue_if_no_path , which is the same as setting no_path_retry to queue . |
| 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_min_io | (RHEL 4.8 and later) 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. |
| 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. If set to uniform , all path weights are equal. The default value is uniform . |
| 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. A value of fail indicates immediate failure, without queueing. A value of queue indicates that queueing should not stop until the path is fixed. The default value is (null). |
| flush_on_last_del | (RHEL 4.7 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 . |

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
# }
#}
```


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:

- Multipath Command Output
- Multipath Queries with multipath Command
- Multipath Command Options
- Multipath Queries with dmsetup Command
- Troubleshooting with the multipathd Interactive Console
- Resizing an Online Multipathed Device

5.1. 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"][hwandler="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.

5.2. 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"][hwandler="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]
```

5.3. 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 multipath devices. |

5.4. 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)
```

5.5. 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
```

5.6. Resizing an Online Multipath Device (RHEL 4.8 and later)

In systems running RHEL 4.8 and later, it is possible to resize a multipath device while it is online. This allows you to resize the device when it is open, as when a file system is currently mounted.

Use the following procedure to resize an online multipath device.

1. Resize your physical device.
2. Resize your paths. For SCSI devices, writing a 1 to the **rescan** file for the device causes the SCSI driver to rescan. You can use the following command:

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

3. Resize your multipath device by running the **multipath** command:

```
# multipath
```

Your hardware setup may require that you temporarily take the actual storage offline in order to resize your physical device. If you take your storage offline and your multipath device is not set to queue when all paths are down, any I/O activity while your storage is offline will fail. You can work around this by executing the following command before taking your storage offline:

```
# dmsetup suspend --noflush device_name
```

After you resize your storage and take it back online, you must run the following command before resizing your paths:

```
# dmsetup resume device_name
```

Appendix A. Revision History

Revision 1.0 Wed Apr 01 2009

Index

A

- active/active configuration
 - definition, 1
 - illustration, 2
- active/passive configuration
 - definition, 1
 - illustration, 1
- alias parameter , 16
 - configuration file, 5

B

- bindings_file parameter, 14
- blacklist
 - configuration file, 12
 - default devices, 13
 - device name, 12
 - in configuration file, 7
 - WWID, 12
- bl_product parameter, 18

C

- chkconfig command, 7
- configuration file
 - alias parameter, 16
 - bindings_file parameter, 14
 - blacklist, 7, 12
 - bl_product parameter, 18
 - devnode_blacklist, 12
 - failback parameter, 14, 16, 18
 - features parameter, 14, 18
 - flush_on_last_del parameter, 14
 - getuid_callout parameter, 14, 18
 - gid parameter, 14
 - hardware_handler parameter, 18
 - max_fds parameter, 18
 - mode parameter, 14
 - no_path_retry parameter, 14, 16, 18
 - overview, 11
 - path_checker parameter, 14, 18
 - path_grouping_policy parameter, 14, 16, 18
 - path_selector parameter, 16, 18
 - polling-interval parameter, 14
 - prio_callout parameter, 14, 18
 - product parameter, 18
 - rr_min_io parameter, 14, 16
 - rr_weight parameter, 14, 16, 18
 - selector parameter, 14
 - udev_dir parameter, 14
 - uid parameter, 14
 - user_friendly_names parameter, 13, 14
 - vendor parameter, 18

- wwid parameter, 16
- configuring
 - DM-Multipath, 7

D

- defaults section
 - multipath.conf file, 13
- dev/mapper directory, 5
- dev/mpath directory, 5
- device name, 5
- device-mapper-multipath package, 7
- devices
 - adding, 10, 17
- devices section
 - multipath.conf file, 17
- devnode_blacklist
 - configuration file, 12
- DM-Multipath
 - and LVM, 6, 6
 - components, 3
 - configuration file, 11
 - configuring, 7
 - definition, 1
 - device name, 5
 - devices, 5
 - failover, 1
 - overview, 1
 - redundancy, 1
 - setup, 7
 - setup, overview, 4
- dm-multipath kernel module , 3
- dm-n devices, 5
- dmsetup command, determining device mapper entries, 22

E

- etc/multipath.conf package, 7

F

- failback parameter, 14, 16, 18
- failover, 1
- features parameter, 14, 18
- feedback, vi, vi
- flush_on_last_del parameter, 14

G

- getuid_callout parameter, 14, 18
- gid parameter, 14

H

- hardware_handler parameter, 18

K

kpartx command , 3

L

local disks, ignoring, 8

LVM physical volumes

 multipath devices, 6

lvm.conf file , 6

M

max_fds parameter, 18

mode parameter, 14

modprobe command, 7

multipath command , 3, 7

 options, 22

 output, 21

 queries, 22

multipath devices, 5

 logical volumes, 6

 LVM physical volumes, 6

multipath.conf file, 3, 11

 defaults section, 13

 devices section, 17

 multipaths section, 16

multipath.conf.annotated file, 11

multipath.conf.defaults file, 3, 11

multipathd

 command, 23

 interactive console, 23

multipathd daemon , 3

multipathd start command, 7

multipaths section

 multipath.conf file, 16

N

no_path_retry parameter, 14, 16, 18

P

path_checker parameter, 14, 18

path_grouping_policy parameter, 14, 16, 18

path_selector parameter, 16, 18

polling_interval parameter, 14

prio_callout parameter, 14, 18

product parameter, 18

R

rr_min_io parameter, 14, 16

rr_weight parameter, 14, 16, 18

S

selector parameter, 14

setup

DM-Multipath, 7

storage array support, 3

storage arrays

 adding, 10, 17

U

udev_dir parameter, 14

uid parameter, 14

user_friendly_names parameter , 5, 13, 14

V

vendor parameter, 18

W

World Wide Identifier (WWID), 5

wwid parameter, 16