



Red Hat Enterprise Linux 7

DM Multipath

Configuring and managing Device Mapper Multipath

Red Hat Enterprise Linux 7 DM Multipath

Configuring and managing Device Mapper Multipath

Steven Levine

Red Hat Customer Content Services

slevine@redhat.com

법적 공지

Copyright © 2018 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

This book provides information on using the Device Mapper Multipath feature on Red Hat Enterprise Linux 7.

차례

1장. DM-MULTIPATH (DEVICE MAPPER MULTIPATHING)	3
1.1. NEW AND CHANGED FEATURES	3
1.2. OVERVIEW OF DM MULTIPATH	5
1.3. 저장 장치 어레이 지원	7
1.4. DM MULTIPATH COMPONENTS	7
1.5. DM MULTIPATH SETUP OVERVIEW	8
2장. 멀티패스 장치	9
2.1. 멀티패스 장치 식별자	9
2.2. 클러스터에서 일관된 멀티패스 장치 이름	9
2.3. 멀티패스 장치 속성	10
2.4. 논리 볼륨에 있는 멀티패스 장치	10
3장. SETTING UP DM MULTIPATH	12
3.1. SETTING UP DM MULTIPATH	12
3.2. 멀티패스 장치 생성 시 로컬 디스크 무시	13
3.3. CONFIGURING STORAGE DEVICES	15
3.4. SETTING UP MULTIPATHING IN THE INITRAMFS FILE SYSTEM	15
4장. THE DM MULTIPATH CONFIGURATION FILE	16
4.1. 설정 파일 개요	16
4.2. 설정 파일 블랙리스트	17
4.3. 설정 파일 기본값	20
4.4. 멀티패스 장치 설정 속성	30
4.5. 설정 파일 장치	36
4.6. ISCSI AND DM MULTIPATH OVERRIDES	43
5장. DM MULTIPATH ADMINISTRATION AND TROUBLESHOOTING	44
5.1. AUTOMATIC CONFIGURATION FILE GENERATION WITH MULTIPATH HELPER	44
5.2. RESIZING AN ONLINE MULTIPATH DEVICE	44
5.3. MOVING ROOT FILE SYSTEMS FROM A SINGLE PATH DEVICE TO A MULTIPATH DEVICE	44
5.4. MOVING SWAP FILE SYSTEMS FROM A SINGLE PATH DEVICE TO A MULTIPATH DEVICE	46
5.5. THE MULTIPATH DAEMON	46
5.6. ISSUES WITH QUEUE_IF_NO_PATH FEATURE	46
5.7. MULTIPATH COMMAND OUTPUT	47
5.8. MULTIPATH QUERIES WITH MULTIPATH COMMAND	48
5.9. MULTIPATH COMMAND OPTIONS	48
5.10. DMSETUP 명령을 사용하여 장치 매퍼 항목 지정	49
5.11. THE MULTIPATHD COMMANDS	49
5.12. TROUBLESHOOTING WITH THE MULTIPATHD INTERACTIVE CONSOLE	50
5.13. CLEANING UP MULTIPATH FILES ON PACKAGE REMOVAL	51
부록 A. 개정 내역	52
색인	53

1장. DM-MULTIPATH (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.

This chapter provides a summary of the features of DM-Multipath that were added subsequent to the initial release of Red Hat Enterprise Linux 7. Following that, this chapter provides a high level overview of DM Multipath and its components, as well as an overview of DM-Multipath setup.

1.1. NEW AND CHANGED FEATURES

This section lists features of the DM Multipath that are new since the initial release of Red Hat Enterprise Linux 7.

1.1.1. New and Changed Features for Red Hat Enterprise Linux 7.1

Red Hat Enterprise Linux 7.1 includes the following documentation and feature updates and changes.

- [표 5.1. "유용한 multipath 명령 옵션"](#). now includes entries for the `-w` and `-W` options of the **multipath** command, which allow you to better manage the `wwids` file.
- Additional options for the **values** argument of the **features** parameter in the **multipath.conf** file are documented in [4장. The DM Multipath Configuration File](#).
- [표 4.1. "멀티패스 설정 기본값"](#). includes an entry for the **force_sync** parameter, which prevents path checkers from running in `async` mode when set to "yes".

In addition, small technical corrections and clarifications have been made throughout the document.

1.1.2. New and Changed Features for Red Hat Enterprise Linux 7.2

Red Hat Enterprise Linux 7.2 includes the following documentation and feature updates and changes.

- This document includes a new section, [5.1절. "Automatic Configuration File Generation with Multipath Helper"](#). The Multipath Helper application gives you options to create multipath configurations with custom aliases, device blacklists, and settings for the characteristics of individual multipath devices.
- The **defaults** section of the **multipath.conf** configuration file supports the new **config_dir**, **new_bindings_in_boot**, **ignore_new_boot_devs**, **retrigger_tries**, and **retrigger_delays** parameters. The **defaults** section of the **multipath.conf** file is documented in [표 4.1. "멀티패스 설정 기본값"](#).
- The **defaults**, **devices**, and **multipaths** sections of the **multipath.conf** configuration file now support the **delay_watch_checks** and **delay_wait_checks** configuration parameters. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).

In addition, small technical corrections and clarifications have been made throughout the document.

1.1.3. New and Changed Features for Red Hat Enterprise Linux 7.3

Red Hat Enterprise Linux 7.3 includes the following documentation and feature updates and changes.

- The **multipathd** command supports new format commands that show the status of multipath devices and paths in "raw" format versions. In raw format, no headers are printed and the fields are not padded to align the columns with the headers. Instead, the fields print exactly as specified in the format string. For information on the **multipathd** commands, see [5.11절. "The multipathd Commands"](#).
- As of Red Hat Enterprise Linux 7.3, if you specify **prio "alua exclusive_pref_bit"** in your device configuration, multipath will create a path group that contains only the path with the **pref** bit set and will give that path group the highest priority. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).
- The **defaults**, **devices**, and **multipaths** sections of the **multipath.conf** configuration file now support the **skip_kpartx** configuration parameter. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).

In addition, small technical corrections and clarifications have been made throughout the document.

1.1.4. New and Changed Features for Red Hat Enterprise Linux 7.4

Red Hat Enterprise Linux 7.4 includes the following documentation and feature updates and changes.

- The **defaults**, **devices**, and **multipaths** sections of the **multipath.conf** configuration file support the **max_sectors_kb** configuration parameter. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).
- The **defaults** and **devices** sections of the **multipath.conf** configuration file support the **detect_path_checker** configuration parameter. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).
- The **defaults** section of the **multipath.conf** configuration file supports the **remove_retries** and **detect_path_checker** parameters. The **defaults** section of the **multipath.conf** file is documented in [표 4.1. "멀티패스 설정 기본값"](#).

1.1.5. New and Changed Features for Red Hat Enterprise Linux 7.5

Red Hat Enterprise Linux 7.5 includes the following documentation and feature updates and changes.

- As of Red Hat Enterprise Linux 7.5, the **blacklist** and **blacklist_exceptions** section of the **multipath.conf** configuration file support the **property** parameter. For information on the **property** parameter, see [4.2.6절. "블랙리스트 예외 설정"](#).
- the **defaults** and **multipaths** sections of the **multipath.conf** file now support a value of **file** for the **reservation_key** parameter. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).
- The **defaults** section of the **multipath.conf** configuration file supports the **prkeys_file** parameter. The **defaults** section of the **multipath.conf** file is documented in [표 4.1. "멀티패스 설정 기본값"](#).

1.1.6. New and Changed Features for Red Hat Enterprise Linux 7.6

Red Hat Enterprise Linux 7.6 includes the following documentation and feature updates and changes.

- As of Red Hat Enterprise Linux 7.6, you can specify the protocol for a device to be excluded from multipathing in the **blacklist** section of the configuration file with a **protocol** section. For information on blacklisting by device protocol, see [4.2.5절. "Blacklisting By Device Protocol"](#)

(Red Hat Enterprise Linux 7.6 and Later)”.

- The **defaults** and **devices** sections of the **multipath.conf** configuration file support the **all_tg_pt** parameter. For information on the configuration parameters, see [4장. The DM Multipath Configuration File](#).

1.2. OVERVIEW OF DM MULTIPATH

DM Multipath can be used to provide:

- 이중화

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.

- 성능 향상

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 rebalance the load.

그림 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.

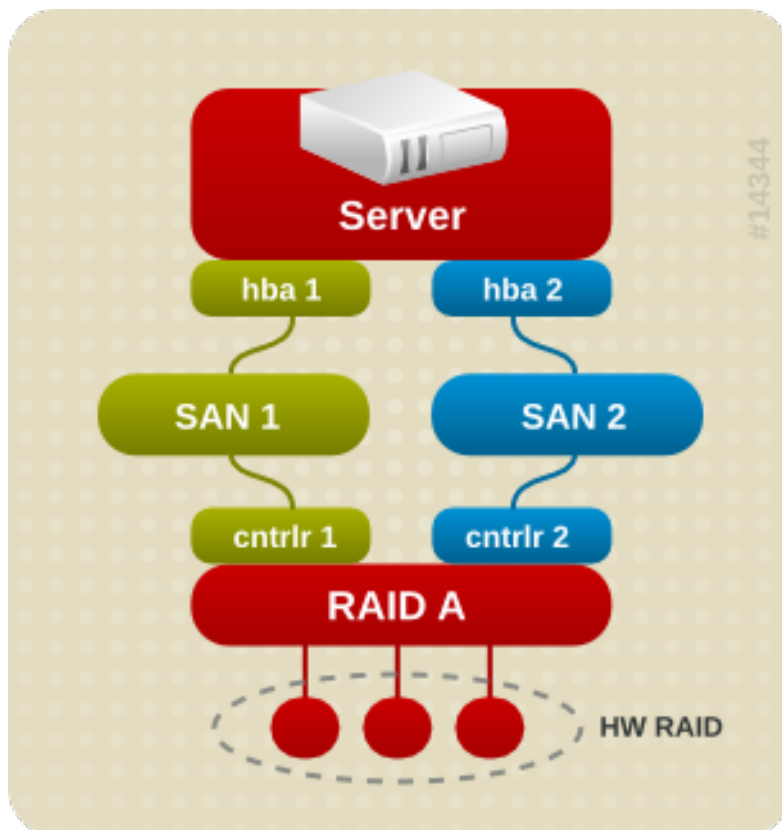


그림 1.1. Active/Passive Multipath Configuration with One RAID Device

이 설정에서, hba1, SAN1, 제어기 1을 통해 지나가는 첫 번째 I/O 경로와 hba2, SAN2, 제어기 2를 통해 지나가는 두 번째 I/O 경로가 있습니다. 이러한 설정에는 장애가 발생할 수 있는 지점이 여러개 있습니다.

- HBA 장애

- FC 케이블 장애
- SAN 스위치 장애
- 어레이 제어기 포트 장애

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

그림 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.

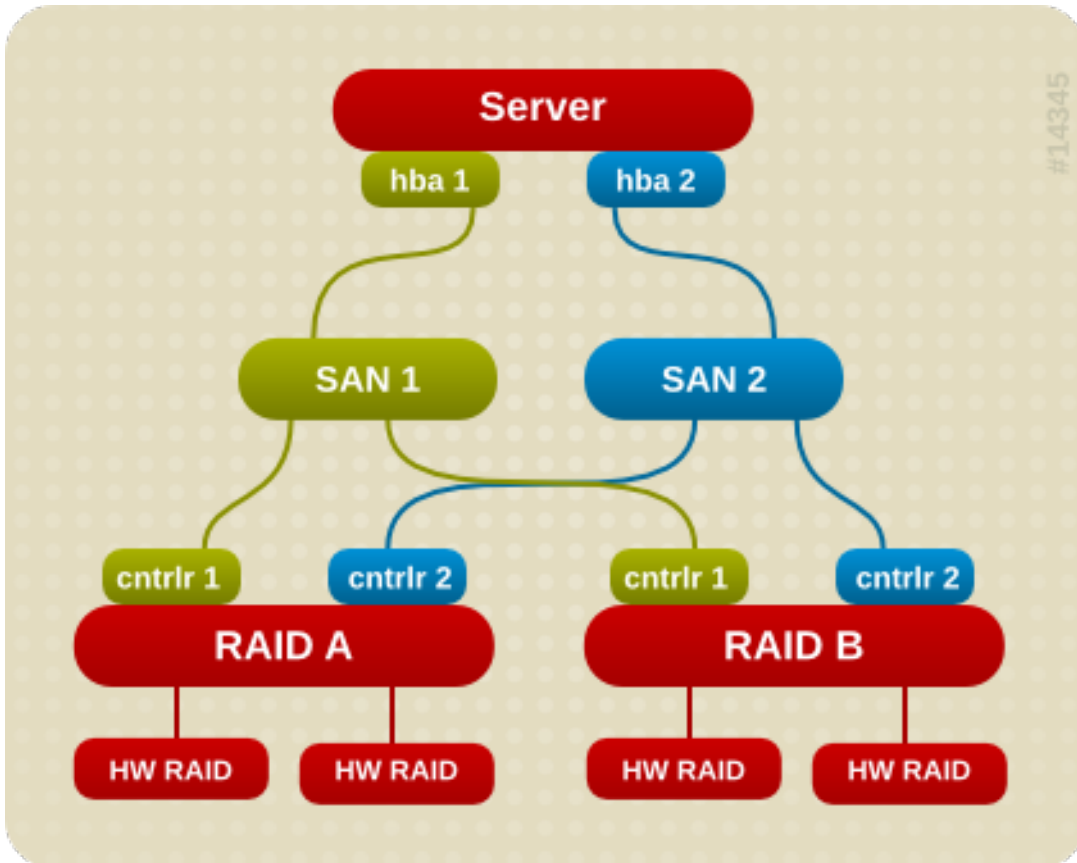


그림 1.2. Active/Passive Multipath Configuration with Two RAID Devices

In the example shown in 그림 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 그림 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.

그림 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:

- hba1에서 제어기1 까지
- hba1에서 제어기 2 까지
- hba2에서 제어기1 까지
- hba2에서 제어기2 까지

설정에서 I/O는 이러한 네 개의 경로 중 배치될 수 있습니다.



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

1.3. 저장 장치 어레이 지원

By default, DM Multipath includes support for the most common storage arrays that themselves support DM Multipath. For information on the default configuration values, including supported devices, run either of the following commands.

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

If your storage array supports DM Multipath and is not configured by default, you may need to add it to the DM Multipath configuration file, **multipath.conf**. For information on the DM Multipath configuration file, see [4장. The DM Multipath Configuration File](#).

일부 저장 장치 어레이는 특정한 I/O 오류 처리 및 경로를 변경해야 합니다. 이에는 분리된 하드웨어 처리 커널 모듈이 필요합니다.

1.4. DM MULTIPATH COMPONENTS

[표 1.1. "DM Multipath Components"](#). describes the components of DM Multipath.

표 1.1. DM Multipath Components

구성 요소	설명
dm_multipath kernel module	I/O를 재라우팅하고 경로 및 경로 그룹에 대해 장애 조치를 합니다.

구성 요소	설명
mpathconf utility	Configures and enables device mapper multipathing.
multipath command	Lists and configures multipath devices. Normally started with /etc/rc.sysinit , it can also be started by audev program whenever a block device is added.
multipathd daemon	Monitors paths; as paths fail and come back, it may initiate path group switches. Allows interactive changes to multipath devices. The daemon must be restarted following any changes to the /etc/multipath.conf file.
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 Multipath. The kpartx command is provided in its own package, but the device-mapper-multipath package depends on it.

1.5. DM MULTIPATH SETUP OVERVIEW

DM Multipath includes compiled-in default settings that are suitable for common multipath configurations. The basic procedure for configuring your system with DM Multipath is as follows:

1. Install the **device-mapper-multipath** rpm.
2. Create the configuration file and enable multipathing with the **mpathconf** command. You can also start the multipath daemon with this command if you do not need to edit the configuration file.
3. If necessary, edit the **multipath.conf** configuration file to modify default values and save the updated file.
4. Start the multipath daemon.

For detailed setup instructions for multipath configuration see [3장. Setting Up DM Multipath](#).

2장. 멀티패스 장치

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. 멀티패스 장치 식별자

각각의 멀티패스 장치에는 WWID (World Wide Identifier)가 있어, 고유하게 구별되고 변경되지 않게 합니다. 기본적으로 멀티패스 장치 이름은 해당 WWID로 설정됩니다. 다른 방법으로, 멀티패스 설정 파일에서 **mpathn** 형식의 고유한 노드명으로 별칭을 설정하는 **user_friendly_names** 옵션을 설정할 수 있습니다.

For example, a node with two HBAs attached to a storage controller with two ports by means of 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 two different places under the **/dev** directory: **/dev/mapper/mpathn** and **/dev/dm-n**.

- **/dev/mapper**에 있는 장치는 이전의 부팅 프로세스에서 생성된 것입니다. 논리 볼륨을 생성할 때와 같이 멀티패스된 장치를 액세스하기 위해 이 장치를 사용합니다.
- Any devices of the form **/dev/dm-n** are for internal use only should never be used by the administrator directly.

For information on the multipath configuration defaults, including the **user_friendly_names** configuration option, see 4.3절. "설정 파일 기본값".

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 4.4절. "멀티패스 장치 설정 속성".

2.2. 클러스터에서 일관된 멀티패스 장치 이름

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 multipath devices on other machines by running the following commands:

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

3. Copy the **/etc/multipath/bindings** 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:

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

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 **multipath.conf** file on one machine.
2. Disable all multipath devices on other machines by running the following commands:

```
# systemctl stop multipathd.service
# 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:

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

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

2.3. 멀티패스 장치 속성

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 [4.4절. "멀티패스 장치 설정 속성"](#).

2.4. 논리 볼륨에 있는 멀티패스 장치

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/mpatha** is the name of a multipath device, the following command will mark **/dev/mapper/mpatha** as a physical volume.

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

기타 다른 LVM 물리 장치를 사용하듯이 LVM 볼륨 그룹을 생성할 때 LVM 물리 장치를 사용할 수 있습니다.



참고

If you attempt to create an LVM physical volume on a whole device on which you have configured partitions, the **pvcreeate** 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 than 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 **/etc/lvm/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 LVM configuration file (**lvm.conf**), include the following filter in the **devices** section of the file.

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

3장. 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
- 로컬 디스크 무시
- 설정 파일에 장치 추가
- Starting multipath in the **initramfs** file system

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.

You set up multipath with the **mpathconf** utility, which creates the multipath configuration file **/etc/multipath.conf**.

- If the **/etc/multipath.conf** file already exists, the **mpathconf** utility will edit it.
- If the **/etc/multipath.conf** file does not exist, the **mpathconf** utility will use the **/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf** file as the starting file.
- If the **/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf** file does not exist the **mpathconf** utility will create the **/etc/multipath.conf** file from scratch.

For more information on the **mpathconf** utility, see the **mpathconf(8)** man page.

If you do not need to edit the **/etc/multipath.conf** file, you can set up DM Multipath for a basic failover configuration by running the following command. This command enables the multipath configuration file and starts the **multipathd** daemon.

```
# mpathconf --enable --with__multipathd y
```

If you need to edit the **/etc/multipath.conf** file before starting the **multipathd** daemon, use the following procedure to set up DM Multipath for a basic failover configuration.

1. Enter the **mpathconf** command with the **--enable** option specified:

```
# mpathconf --enable
```

For information on additional options to the **mpathconf** command you may require, see the **mpathconf** man page or enter the **mpathconf** command with the **--help** option specified.

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

Commands:
Enable: --enable
Disable: --disable
Set user_friendly_names (Default y): --user_friendly_names <y|n>
```



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

2. Edit the `/etc/multipath.conf` file if necessary. 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 edit the `/etc/multipath.conf` file. For information on changing the values in the configuration file to something other than the defaults, see [4장. The DM Multipath Configuration File](#).

The initial defaults section of the configuration file configures your system so 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, if necessary.
4. Execute the following command:

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

Since the value of `user_friendly_names` 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 [4장. The DM Multipath Configuration File](#).

If you do not want to use user friendly names, you can enter the following command:

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



참고

If you find that you need to edit the multipath configuration file after you have started the multipath daemon, you must execute the `systemctl reload multipathd.service` command for the changes to take effect.

3.2. 멀티패스 장치 생성 시 로컬 디스크 무시

Some machines have local SCSI cards for their internal disks. DM Multipath is not recommended for these devices. If you set the `find_multipaths` configuration parameter to `yes`, you should not have to blacklist these devices. For information on the `find_multipaths` configuration parameter, see [4.3절. "설정 파일 기본값"](#).

If you do not set the `find_multipaths` configuration parameter to `yes`, can use the following procedure to modify the multipath configuration file to ignore the local disks when configuring multipath.

1. 어떤 디스크가 내부 디스크인지를 지정하고 블랙리스트에 이를 표시합니다.

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

This examples specifies the **-d** option of the **multipath** command to indicate that this is a dry run that will not create the multipath devices. For further information on the **multipath** command output, see 5.7절. "Multipath Command Output".

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

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

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

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

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

- 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 a 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.

multipath -v2 명령의 출력 결과에서 **/dev/sda** 장치의 WWID는 SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1입니다. 이러한 장치를 블랙리스트하려면, **/etc/multipath.conf** 파일에 다음을 포함시킵니다.

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

- /etc/multipath.conf** 파일을 업데이트한 후, 파일을 다시 불러오기 위해 수동으로 **multipathd** 데몬을 작동시켜야 합니다. 다음의 명령은 업데이트된 **/etc/multipath.conf** 파일을 다시 불러오기합니다.

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

3.3. CONFIGURING STORAGE DEVICES

By default, DM Multipath includes support for the most common storage arrays that themselves support DM Multipath. For information on the default configuration value, including supported devices, run either of the following commands.

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

알려진 멀티패스 장치로서 기본값으로 지원되지 않는 저장 장치를 추가하고자 할 경우, **/etc/multipath.conf** 파일을 편집하여 해당하는 장치 정보를 삽입합니다.

For example, to add information about the HP Open-V series the entry looks like this. This example sets the device to queue for a minute (or 12 retries and 5 seconds per retry) after all paths have failed.

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

For more information on the **devices** section of the configuration file, see [4.5절. "설정 파일 장치"](#).

3.4. SETTING UP MULTIPATHING IN THE INITRAMFS FILE SYSTEM

You can set up multipathing in the **initramfs** file system. After configuring multipath, you can rebuild the **initramfs** file system with the multipath configuration files by executing the **dracut** command with the following options:

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

If you run multipath from the **initramfs** file system and you make any changes to the multipath configuration files, you must rebuild the **initramfs** file system for the changes to take effect.

4장. 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 themselves support DM Multipath. For information on the default configuration values, including supported devices, run either of the following commands.

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

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.



참고

You can run set up multipathing in the **initramfs** file system. If you run multipath from the **initramfs** file system and you make any changes to the multipath configuration files, you must rebuild the **initramfs** file system for the changes to take effect. For information on rebuilding the **initramfs** file system with multipath, see [3.4절. "Setting Up Multipathing in the initramfs File System"](#).

This chapter provides information on parsing and modifying the **multipath.conf** file. It contains sections on the following topics:

- 설정 파일 개요
- 설정 파일 블랙리스트
- 설정 파일 기본값
- 설정 파일 멀티패스
- 설정 파일 장치

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. 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.

설정 파일에서는 정규 표현식 기술 구문을 허용합니다.

Further information about the configuration file can be found on the **multipath.conf(5)** man page.

4.1. 설정 파일 개요

멀티패스 설정 파일은 다음과 같은 부분으로 나뉘어져 있습니다:

blacklist

Listing of specific devices that will not be considered for multipath.

blacklist_exceptions

블랙리스트 부분의 매개 변수에 따라 블랙리스트될 수 있는 멀티패스 후보 목록.

defaults

General default settings for DM Multipath.

multipaths

개별적 멀티패스 장치의 특성 설정. 이러한 값은 설정 파일의 **defaults** 및 **devices**에서 지정된 값을 덮어쓰기 합니다.

devices

개별적 저장 장치 제어기 설정. 이러한 값은 설정 파일의 **defaults** 및 **devices**에서 지정된 값을 덮어쓰기 합니다. 기본값으로 지원되지 않는 저장 장치 배열을 사용하고 있을 경우, 배열에 해당하는 **devices** 하부 섹션을 생성해야 할 수도 있습니다.

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

4.2. 설정 파일 블랙리스트

멀티패스 설정 파일의 **blacklist** 부분에서는 시스템이 멀티패스 장치를 설정할 때 사용되지 않는 장치를 지정합니다. 블랙리스트된 장치는 멀티패스 장치 그룹에 포함되지 않게 됩니다.

In older releases of Red Hat Enterprise Linux, multipath always tried to create a multipath device for every path that was not explicitly blacklisted. As of Red Hat Enterprise Linux 6, however, if the **find_multipaths** configuration parameter is set to **yes**, then multipath will create a device only if one of three conditions are met:

- There are at least two paths that are not blacklisted with the same WWID.
- The user manually forces the creation of the device by specifying a device with the **multipath** command.
- A path has the same WWID as a multipath device that was previously created (even if that multipath device does not currently exist). Whenever a multipath device is created, multipath remembers the WWID of the device so that it will automatically create the device again as soon as it sees a path with that WWID. This allows you to have multipath automatically choose the correct paths to make into multipath devices, without have to edit the multipath blacklist.

If you have previously created a multipath device without using the **find_multipaths** parameter and then you later set the parameter to **yes**, you may need to remove the WWIDs of any device you do not want created as a multipath device from the **/etc/multipath/wwids** file. The following shows a sample **/etc/multipath/wwids** file. The WWIDs are enclosed by slashes (/):

```
# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and multipathd.
# You should not need to edit this file in normal circumstances.
#
# Valid WWIDs:
/3600d0230000000000e13955cc3757802/
/3600d0230000000000e13955cc3757801/
/3600d0230000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372 0E13955CC3757802/
/3600d0230000000000e13955cc3757803/
```

With the **find_multipaths** parameter set to **yes**, you need to blacklist only the devices with multiple paths that you do not want to be multipathed. Because of this, it will generally not be necessary to blacklist devices.

If you do need to blacklist devices, you can do so according to the following criteria:

- By WWID, as described in [4.2.1절. "WWID에 따라 블랙리스트하기 "](#)
- By device name, as described in [4.2.2절. "장치 이름에 따라 블랙리스트하기 "](#)
- By device type, as described in [4.2.3절. "장치 유형에 따라 블랙리스트하기 "](#)
- By **udev** property, as described in [4.2.4절. "Blacklisting By udev Property \(Red Hat Enterprise Linux 7.5 and Later\)"](#)
- By device protocol, as described in [4.2.5절. "Blacklisting By Device Protocol \(Red Hat Enterprise Linux 7.6 and Later\)"](#)

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 [4.2.2절. "장치 이름에 따라 블랙리스트하기 "](#).

4.2.1. WWID에 따라 블랙리스트하기

설정 파일의 **blacklist** 부분에 있는 **wwid** 항목으로 WWID (World-Wide IDentification)에 따라 블랙리스트하기 위해 개별 장치를 지정할 수 있습니다.

다음의 예에서는 WWID 26353900f02796769 로 장치를 블랙리스트하는 설정 파일 행을 보여주고 있습니다.

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

4.2.2. 장치 이름에 따라 블랙리스트하기

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 a specific type. This is not recommended, however, since 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 [4.2.6절. "블랙리스트 예외 설정 "](#).

```

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

```

4.2.3. 장치 유형에 따라 블랙리스트하기

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. Blacklisting By udev Property (Red Hat Enterprise Linux 7.5 and Later)

The **blacklist** and **blacklist_exceptions** sections of the **multipath.conf** configuration file support the **property** parameter. This parameter allows users to blacklist certain types of devices. The **property** parameter takes a regular expression string that is matched against the **udev** environment variable name for the device.

The following example blacklists all devices with the **udev** property **ID_ATA**.

```

blacklist {
    property "ID_ATA"
}

```

4.2.5. Blacklisting By Device Protocol (Red Hat Enterprise Linux 7.6 and Later)

You can specify the protocol for a device to be excluded from multipathing in the **blacklist** section of the configuration file with a **protocol** section. The protocol strings that multipath recognizes are `scsi:fc`, `scsi:spi`, `scsi:ssa`, `scsi:sbp`, `scsi:srp`, `scsi:iscsi`, `scsi:sas`, `scsi:adt`, `scsi:ata`, `scsi:unspec`, `ccw`, `cciss`, `nvme`, and `undef`. The protocol that a path is using can be viewed by running the command **multipathd show paths format "%d %P"**.

The following example blacklists all devices with an undefined protocol or an unknown SCSI transport type.

```

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

```

4.2.6. 블랙리스트 예외 설정

You can use the **blacklist_exceptions** section of the configuration file to enable multipathing on devices that have been blacklisted by default.

For example, if you have a large number of devices and want to multipath only one of them (with the WWID of 3600d0230000000000e13955cc3757803), instead of individually blacklisting each of the devices except the one you want, you could instead blacklist all of them, and then allow only the one you want by adding the following lines to the **/etc/multipath.conf** file.

```
blacklist {
    wwid "*"
}

blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}
```

설정 파일의 **blacklist_exceptions** 부분에서 장치를 지정할 때, 블랙리스트에 지정한 것과 동일한 방법으로 예외 설정을 해야 합니다. 예를 들어, 블랙리스트된 장치가 WWID와 관련되어 있을 지라도, WWID 예외 설정은 **devnode** 블랙리스트 항목으로 지정된 장치에는 적용되지 않게 됩니다. 이와 유사하게, **devnode** 예외 설정은 **devnode** 항목에만 적용되며 **device** 예외 설정은 장치 항목에만 적용됩니다.

The **property** parameter works differently than the other **blacklist_exception** parameters. If the parameter is set, the device must have a **udev** variable that matches. Otherwise, the device is blacklisted. This parameter allows users to blacklist SCSI devices that multipath should ignore, such as USB sticks and local hard drives. To allow only SCSI devices that could reasonably be multipathed, set this parameter to **SCSI_IDENT_ID_WWN** as in the following example.

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

4.3. 설정 파일 기본값

/etc/multipath.conf 설정 파일에는 다음과 같이 **user_friendly_names** 매개 변수가 **yes**로 설정된 **defaults** 부분이 포함되어 있습니다.

```
defaults {
    user_friendly_names yes
}
```

이는 **user_friendly_names** 매개 변수의 기본값을 덮어쓰기합니다.

설정 파일에는 설정 기본값 템플릿이 들어 있습니다. 다음과 같이 이 부분에 주석을 담니다.

```
#defaults {
#   polling_interval    10
#   path_selector       "round-robin 0"
#   path_grouping_policy multibus
#   uid_attribute       ID_SERIAL
#   prio                alua
#   path_checker        readsector0
#   rr_min_io           100
#   max_fds             8192
#   rr_weight           priorities
```



```
# failback          immediate
# no_path_retry     fail
# user_friendly_names  yes
#}
```

설정 매개 변수에 해당하는 기본값을 덮어쓰기하려면, 해당하는 행을 템플릿에서 **defaults** 부분으로 복사하여 주석 해제 처리를 합니다. 예를 들어, **path_grouping_policy** 매개 변수를 덮어쓰기하려면 이는 **failover** 기본값이 아닌 **multibus**이어야 하며, 해당 행을 템플릿에서 설정 파일의 초기 **defaults** 부분으로 복사한 후, 다음과 같이 주석 해제 처리합니다.

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

표 4.1. “멀티패스 설정 기본값” 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.

표 4.1. 멀티패스 설정 기본값

속성	설명
polling_interval	Specifies the interval between two path checks in seconds. For properly functioning paths, the interval between checks will gradually increase to (4 * polling_interval). The default value is 5.
multipath_dir	The directory where the dynamic shared objects are stored. The default value is system dependent, commonly /lib/multipath .

속성	설명
find_multipaths	<p>Defines the mode for setting up multipath devices. If this parameter is set to yes, then multipath will not try to create a device for every path that is not blacklisted. Instead multipath will create a device only if one of three conditions are met:</p> <ul style="list-style-type: none"> - There are at least two paths that are not blacklisted with the same WWID. - The user manually forces the creation of the device by specifying a device with the multipath command. - A path has the same WWID as a multipath device that was previously created. Whenever a multipath device is created with find_multipaths set, multipath remembers the WWID of the device so that it will automatically create the device again as soon as it sees a path with that WWID. This allows you to have multipath automatically choose the correct paths to make into multipath devices, without having to edit the multipath blacklist. For instructions on the procedure to follow if you have previously created multipath devices when the find_multipaths parameter was not set, see 4.2절. "설정 파일 블랙리스트". <p>The default value is no. The default multipath.conf file created by mpathconf, however, will enable find_multipaths as of Red Hat Enterprise Linux 7.</p>
reassign_maps	<p>Enable reassigning of device-mapper maps. With this option, The multipathd daemon will remap existing device-mapper maps to always point to the multipath device, not the underlying block devices. Possible values are yes and no. The default value is yes.</p>
verbosity	<p>The default verbosity. Higher values increase the verbosity level. Valid levels are between 0 and 6. The default value is 2.</p>
path_selector	<p>Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include:</p> <ul style="list-style-type: none"> round-robin 0: Loop through every path in the path group, sending the same amount of I/O to each. queue-length 0: Send the next bunch of I/O down the path with the least number of outstanding I/O requests. service-time 0: Send the next bunch of I/O down the path with the shortest estimated service time, which is determined by dividing the total size of the outstanding I/O to each path by its relative throughput. <p>The default value is service-time 0.</p>

속성	설명
<p>path_grouping_policy</p>	<p>미지정된 멀티패스에 적용할 기본값 경로 그룹 정책을 지정합니다. 가능한 값은 다음과 같습니다:</p> <p>failover: 1 path per priority group.</p> <p>multibus: all valid paths in 1 priority group.</p> <p>group_by_serial: 1 priority group per detected serial number.</p> <p>group_by_prio: 1 priority group per path priority value. Priorities are determined by callout programs specified as global, per-controller, or per-multipath options.</p> <p>group_by_node_name: 1 priority group per target node name. Target node names are fetched in <code>/sys/class/fc_transport/target*/node_name</code>.</p> <p>The default value is failover.</p>
<p>prio</p>	<p>Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable prio value. Possible values include:</p> <p>const: Set a priority of 1 to all paths.</p> <p>emc: Generate the path priority for EMC arrays.</p> <p>alua: Generate the path priority based on the SCSI-3 ALUA settings. As of Red Hat Enterprise Linux 7.3, if you specify prio "alua exclusive_pref_bit" in your device configuration, multipath will create a path group that contains only the path with the pref bit set and will give that path group the highest priority.</p> <p>ontap: Generate the path priority for NetApp arrays.</p> <p>rdac: Generate the path priority for LSI/Engenio RDAC controller.</p> <p>hp_sw: Generate the path priority for Compaq/HP controller in active/standby mode.</p> <p>hds: Generate the path priority for Hitachi HDS Modular storage arrays.</p> <p>The default value is const.</p>

속성	설명
<p>features</p>	<p>The default extra features of multipath devices, using the format: "<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>Possible values for features include:</p> <p>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 5.6절. "Issues with queue_if_no_path feature".</p> <p>retain_attached_hw_handler: If this parameter is set to yes and the SCSI layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the multipath.conf file. If the SCSI layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual. The default value is no.</p> <p>pg_init_retries <i>n</i>: Retry path group initialization up to <i>n</i> times before failing where $1 \leq n \leq 50$.</p> <p>pg_init_delay_msecs <i>n</i>: Wait <i>n</i> milliseconds between path group initialization retries where $0 \leq n \leq 60000$.</p>
<p>path_checker</p>	<p>Specifies the default method used to determine the state of the paths. Possible values include:</p> <p>readsector0: Read the first sector of the device.</p> <p>tur: Issue a TEST UNIT READY command to the device.</p> <p>emc_clariion: Query the EMC Clariion specific EVPD page 0xC0 to determine the path.</p> <p>hp_sw: Check the path state for HP storage arrays with Active/Standby firmware.</p> <p>rdac: Check the path state for LSI/Engenio RDAC storage controller.</p> <p>directio: Read the first sector with direct I/O.</p> <p>The default value is directio.</p>

속성	설명
failback	<p>Manages path group failback.</p> <p>A value of immediate specifies immediate failback to the highest priority path group that contains active paths.</p> <p>A value of manual specifies that there should not be immediate failback but that failback can happen only with operator intervention.</p> <p>A value of followover specifies that automatic failback should be performed when the first path of a path group becomes active. This keeps a node from automatically failing back when another node requested the failover.</p> <p>0 보다 큰 숫자 값을 지정하여 초 단위로 지연 장애 복구가 되도록 합니다.</p> <p>기본값은 manual입니다.</p>
rr_min_io	<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. This setting is only for systems running kernels older than 2.6.31. Newer systems should use rr_min_io_rq. The default value is 1000.</p>
rr_min_io_rq	<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath. This setting should be used on systems running current kernels. On systems running kernels older than 2.6.31, use rr_min_io. The default value is 1.</p>
rr_weight	<p>If set to priorities, then instead of sending rr_min_io requests to a path before calling 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 function. If set to uniform, all path weights are equal. The default value is uniform.</p>
no_path_retry	<p>A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queuing.</p> <p>fail 값은 대기 상태 없이 즉시 장애가 발생함을 나타냅니다.</p> <p>queue 값은 경로가 수정될 때 까지 대기 상태가 계속 됨을 나타냅니다.</p> <p>The default value is 0.</p>

속성	설명
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 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 .
queue_without_daemon	If set to no , the multipathd daemon will disable queuing for all devices when it is shut down. The default value is no .
flush_on_last_del	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 .
max_fds	Sets the maximum number of open file descriptors that can be opened by multipath and the multipathd daemon. This is equivalent to the ulimit -n command. As of the Red Hat Enterprise Linux 6.3 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.
checker_timeout	The timeout to use for prioritizers and path checkers that issue SCSI commands with an explicit timeout, in seconds. The default value is taken from sys/block/sdx/device/timeout .
fast_io_fail_tmo	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. The fast_io_fail_tmo option overrides the values of the recovery_tmo and replacement_timeout options. For details, see 4.6절. "iSCSI and DM Multipath overrides" .
dev_loss_tmo	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. Setting this to infinity will set this to 2147483647 seconds, or 68 years. The default value is determined by the OS.

속성	설명
hw_string_match	<p>Each device configuration in the devices section of the multipath.conf file will either create its own device configuration or it will modify one of the built-in device configurations. If hw_string_match is set to yes, then if the vendor, product, and revision strings in a user's device configuration exactly match those strings in a built-in device configuration, the built-in configuration is modified by the options in the user's configuration. Otherwise, the user's device configuration is treated as a new configuration. If hw_string_match is set to no, a regular expression match is used instead of a string match.</p> <p>The hw_string_match parameter is set to no by default.</p>
retain_attached_hw_handler	<p>If this parameter is set to yes and the SCSI layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the multipath.conf file. If the SCSI layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual. The default value is no.</p>
detect_prio	<p>If this is set to yes, multipath will first check if the device supports ALUA, and if so it will automatically assign the device the alua prioritizer. If the device does not support ALUA, it will determine the prioritizer as it always does. The default value is no.</p>
uid_attribute	<p>Provides a unique path identifier. The default value is ID_SERIAL.</p>
force_sync	<p>(Red Hat Enterprise Linux Release 7.1 and later) If this is set to "yes", it prevents path checkers from running in async mode. This means that only one checker will run at a time. This is useful in the case where many multipathd checkers running in parallel causes significant CPU pressure. The default value is no.</p>
delay_watch_checks	<p>(Red Hat Enterprise Linux Release 7.2 and later) If set to a value greater than 0, the multipathd daemon will watch paths that have recently become valid for the specified number of checks. If they fail again while they are being watched, when they next become valid they will not be used until they have stayed up for the number of consecutive checks specified with delay_wait_checks. This allows you to keep paths that may be unreliable from immediately being put back into use as soon as they come back online. The default value is no.</p>

속성	설명
delay_wait_checks	(Red Hat Enterprise Linux Release 7.2 and later) If set to a value greater than 0, when a device that has recently come back online fails again within the number of checks specified with delay_watch_checks , the next time it comes back online it will be marked and delayed and it will not be used until it has passed the number of checks specified in delay_wait_checks . The default value is no .
ignore_new_boot_devs	(Red Hat Enterprise Linux Release 7.2 and later) If set to yes , when the node is still in the initramfs file system during early boot, multipath will not create any devices whose WWIDs do not already exist in the initramfs copy of the /etc/multipath/wwids . This feature can be used for booting up during installation, when multipath would otherwise attempt to set itself up on devices that it did not claim when they first appeared by means of the udev rules. This parameter can be set to yes or no . If unset, it defaults to no .
retrigger_tries, retrigger_delay	(Red Hat Enterprise Linux Release 7.2 and later) The retrigger_tries and retrigger_delay parameters are used in conjunction to make multipathd retrigger uevents if udev failed to completely process the original ones, leaving the device unusable by multipath. The retrigger_tries parameter sets the number of times that multipath will try to retrigger a uevent if a device has not been completely set up. The retrigger_delay parameter sets the number of seconds between retries. Both of these options accept numbers greater than or equal to zero. Setting the retrigger_tries parameter to zero disables retries. Setting the retrigger_delay parameter to zero causes the uevent to be reissued on the next loop of the path checker. If the retrigger_tries parameter is unset, it defaults to 3. If the retrigger_delay parameter is unset, it defaults to 10.
new_bindings_in_boot	(Red Hat Enterprise Linux Release 7.2 and later) The new_bindings_in_boot parameter is used to keep multipath from giving out a user_friendly_name in the initramfs file system that was already given out by the bindings file in the regular file system, an issue that can arise since the user_friendly_names bindings in the initramfs file system get synced with the bindings in the regular file system only when the initramfs file system is remade. When this parameter is set to no multipath will not create any new bindings in the initramfs file system. If a device does not already have a binding in the initramfs copy of /etc/multipath/bindings , multipath will use its WWID as an alias instead of giving it a user_friendly_name . Later in boot, after the node has mounted the regular filesystem, multipath will give out a user_friendly_name to the device. This parameter can be set to yes or no . If unset, it defaults to no .

속성	설명
config_dir	(Red Hat Enterprise Linux Release 7.2 and later) If set to anything other than "", multipath will search this directory alphabetically for files ending in ".conf" and it will read configuration information from them, just as if the information were in the /etc/multipath.conf file. This allows you to have one main configuration that you share between machines in addition to a separate machine-specific configuration file or files. The config_dir parameter must either be "" or a fully qualified directory name. This parameter can be set only in the main /etc/multipath.conf file and not in one of the files specified in the config_dir file itself. The default value is /etc/multipath/conf.d .
deferred_remove	If set to yes , multipathd will do a deferred remove instead of a regular remove when the last path device has been deleted. This ensures that if a multipathed device is in use when a regular remove is performed and the remove fails, the device will automatically be removed when the last user closes the device. The default value is no .
log_checker_err	If set to once , multipathd logs the first path checker error at verbosity level 2. Any later errors are logged at verbosity level 3 until the device is restored. If it is set to always , multipathd always logs the path checker error at verbosity level 2. The default value is always .
skip_kpartx	(Red Hat Enterprise Linux Release 7.3 and later) If set to yes , kpartx will not automatically create partitions on the device. This allows users to create a multipath device without creating partitions, even if the device has a partition table. The default value of this option is no .
max_sectors_kb	(Red Hat Enterprise Linux Release 7.4 and later) Sets the max_sectors_kb device queue parameter to the specified value on all underlying paths of a multipath device before the multipath device is first activated. When a multipath device is created, the device inherits the max_sectors_kb value from the path devices. Manually raising this value for the multipath device or lowering this value for the path devices can cause multipath to create I/O operations larger than the path devices allow. Using the max_sectors_kb parameter is an easy way to set these values before a multipath device is created on top of the path devices and prevent invalid-sized I/O operations from being passed. If this parameter is not set by the user, the path devices have it set by their device driver, and the multipath device inherits it from the path devices.
remove_retries	(Red Hat Enterprise Linux Release 7.4 and later) Sets how many times multipath will retry removing a device that is in use. Between each attempt, multipath will sleep 1 second. The default value is 0, which means that multipath will not retry the remove.
disable_changed_wwids	(Red Hat Enterprise Linux Release 7.4 and later) If set to yes and the WWID of a path device changes while it is part of a multipath device, multipath will disable access to the path device until the WWID of the path is restored to the WWID of the multipath device. The default value is no , which does not check if a path's WWID has changed.

속성	설명
detect_path_checker	(Red Hat Enterprise Linux Release 7.4 and later) If set to yes , multipath will try to detect if the device supports ALUA. If so, the device will automatically use the tur path checker. If not, the path_checker will be selected as usual. The default value is no .
reservation_key	<p>This is the service action reservation key used by mpathpersist. It must be set for all multipath devices using persistent reservations, and it must be the same as the RESERVATION KEY field of the PERSISTENT RESERVE OUT parameter list which contains an 8-byte value provided by the application client to the device server to identify the I_T nexus. If the --param-aptpl option is used when registering the key with mpathpersist, :aptpl must be appended to the end of the reservation key.</p> <p>As of Red Hat Enterprise Linux Release 7.5, this parameter can be set to file, which will store the RESERVATION KEY registered by mpathpersist in the prkeys file. The multipathd daemon will then use this key to register additional paths as they appear. When the registration is removed, the RESERVATION KEY is removed from the prkeys file. It is unset by default.</p>
prkeys_file	(Red Hat Enterprise Linux Release 7.5 and later) The full path name of the prkeys file, which is used by the multipathd daemon to keep track of the reservation key used for a specific WWID when the reservation_key parameter is set to file . The default value is /etc/multipath/prkeys .
all_tg_pt	(Red Hat Enterprise Linux Release 7.6 and later) If this option is set to yes , when mpathpersist registers keys it will treat a key registered from one host to one target port as going from one host to all target ports. This must be set to yes to successfully use mpathpersist on arrays that automatically set and clear registration keys on all target ports from a host, instead of per target port per host. The default value is no .

4.4. 멀티패스 장치 설정 속성

표 4.2. "멀티패스 속성" 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.

표 4.2. 멀티패스 속성

속성	설명
----	----

속성	설명
wwid	Specifies the WWID of the multipath device to which the multipath attributes apply. This parameter is mandatory for this section of the multipath.conf file.
alias	Specifies the symbolic name for the multipath device to which the multipath attributes apply. If you are using user_friendly_names , do not set this value to mpath# ; this may conflict with an automatically assigned user friendly name and give you incorrect device node names.
path_grouping_policy	<p>미지정된 멀티패스에 적용할 기본값 경로 그룹 정책을 지정합니다. 가능한 값은 다음과 같습니다:</p> <p>failover = 우선 순위 그룹 당 1개의 경로</p> <p>multibus = 1개의 우선 순위 그룹에 있는 유효한 모든 경로</p> <p>group_by_serial = 검색된 시리얼 번호 당 1개의 우선 순위 그룹</p> <p>group_by_prio = 경로 우선 순위 값 당 1개의 우선 순위 그룹</p> <p>group_by_node_name = 대상 노드 이름 당 1개의 우선 순위 그룹</p>
path_selector	<p>Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include:</p> <p>round-robin 0: Loop through every path in the path group, sending the same amount of I/O to each.</p> <p>queue-length 0: Send the next bunch of I/O down the path with the least number of outstanding I/O requests.</p> <p>service-time 0: Send the next bunch of I/O down the path with the shortest estimated service time, which is determined by dividing the total size of the outstanding I/O to each path by its relative throughput.</p>

속성	설명
<p>failback</p>	<p>Manages path group failback.</p> <p>A value of immediate specifies immediate failback to the highest priority path group that contains active paths.</p> <p>A value of manual specifies that there should not be immediate failback but that failback can happen only with operator intervention.</p> <p>A value of followover specifies that automatic failback should be performed when the first path of a path group becomes active. This keeps a node from automatically failing back when another node requested the failover.</p> <p>0 보다 큰 숫자 값을 지정하여 초 단위로 지연 장애 복구가 되도록 합니다.</p>
<p>prio</p>	<p>Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable prio value. Possible values include:</p> <p>const: Set a priority of 1 to all paths.</p> <p>emc: Generate the path priority for EMC arrays.</p> <p>alua: Generate the path priority based on the SCSI-3 ALUA settings. As of Red Hat Enterprise Linux 7.3, if you specify prio "alua exclusive_pref_bit" in your device configuration, multipath will create a path group that contains only the path with the pref bit set and will give that path group the highest priority.</p> <p>ontap: Generate the path priority for NetApp arrays.</p> <p>rdac: Generate the path priority for LSI/Engenio RDAC controller.</p> <p>hp_sw: Generate the path priority for Compaq/HP controller in active/standby mode.</p> <p>hds: Generate the path priority for Hitachi HDS Modular storage arrays.</p>

속성	설명
features	<p>The default extra features of multipath devices, using the format: "<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>Possible values for features include:</p> <p>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 5.6절. "Issues with queue_if_no_path feature".</p> <p>retain_attached_hw_handler: If this parameter is set to yes and the SCSI layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the multipath.conf file. If the SCSI layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual. The default value is no.</p> <p>pg_init_retries <i>n</i>: Retry path group initialization up to <i>n</i> times before failing where $1 \leq n \leq 50$.</p> <p>pg_init_delay_msecs <i>n</i>: Wait <i>n</i> milliseconds between path group initialization retries where $0 \leq n \leq 60000$.</p>
no_path_retry	<p>A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queuing.</p> <p>fail 값은 대기 상태 없이 즉시 장애가 발생함을 나타냅니다.</p> <p>queue 값은 경로가 수정될 때 까지 대기 상태가 계속 됨을 나타냅니다.</p>
rr_min_io	<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. This setting is only for systems running kernels older than 2.6.31. Newer systems should use rr_min_io_rq. The default value is 1000.</p>
rr_min_io_rq	<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath. This setting should be used on systems running current kernels. On systems running kernels older than 2.6.31, use rr_min_io. The default value is 1.</p>

속성	설명
rr_weight	If set to priorities , then instead of sending rr_min_io requests to a path before calling 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 function. If set to uniform , all path weights are equal.
flush_on_last_del	If set to yes , then multipath will disable queuing 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 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.
delay_watch_checks	(Red Hat Enterprise Linux Release 7.2 and later) If set to a value greater than 0, the multipathd daemon will watch paths that have recently become valid for the specified number of checks. If they fail again while they are being watched, when they next become valid they will not be used until they have stayed up for the number of consecutive checks specified with delay_wait_checks . This allows you to keep paths that may be unreliable from immediately being put back into use as soon as they come back online.
delay_wait_checks	(Red Hat Enterprise Linux Release 7.2 and later) If set to a value greater than 0, when a device that has recently come back online fails again within the number of checks specified with delay_watch_checks , the next time it comes back online it will be marked and delayed and it will not be used until it has passed the number of checks specified in delay_wait_checks .
deferred_remove	If set to yes , multipathd will do a deferred remove instead of a regular remove when the last path device has been deleted. This ensures that if a multipathed device is in use when a regular remove is performed and the remove fails, the device will automatically be removed when the last user closes the device.
skip_kpartx	(Red Hat Enterprise Linux Release 7.3 and later) If set to yes , kpartx will not automatically create partitions on the device. This allows users to create a multipath device without creating partitions, even if the device has a partition table.

속성	설명
max_sectors_kb	(Red Hat Enterprise Linux Release 7.4 and later) Sets the max_sectors_kb device queue parameter to the specified value on all underlying paths of a multipath device before the multipath device is first activated. When a multipath device is created, the device inherits the max_sectors_kb value from the path devices. Manually raising this value for the multipath device or lowering this value for the path devices can cause multipath to create I/O operations larger than the path devices allow. Using the max_sectors_kb parameter is an easy way to set these values before a multipath device is created on top of the path devices and prevent invalid-sized I/O operations from being passed. If this parameter is not set by the user, the path devices have it set by their device driver, and the multipath device inherits it from the path devices.
reservation_key	<p>This is the service action reservation key used by mpathpersist. It must be set for all multipath devices using persistent reservations, and it must be the same as the RESERVATION KEY field of the PERSISTENT RESERVE OUT parameter list which contains an 8-byte value provided by the application client to the device server to identify the I_T nexus. If the --param-aptpl option is used when registering the key with mpathpersist, :aptpl must be appended to the end of the reservation key.</p> <p>As of Red Hat Enterprise Linux Release 7.5, this parameter can be set to file, which will store the RESERVATION KEY registered by mpathpersist in the prkeys_file file. The multipathd daemon will then use this key to register additional paths as they appear. When the registration is removed, the RESERVATION KEY is removed from the prkeys_file file.</p>

다음 예에서는 두 개의 특정한 멀티패스 장치에 해당하는 설정 파일에 지정된 멀티패스 속성을 보여주고 있습니다. 첫 번째 장치는 WWID **3600508b4000156d70001200000b0000**과 **yellow**라는 상징적 이름을 가지고 있습니다.

예에서 두 번째 멀티패스 장치는 WWID **1DEC_____321816758474**와 **red**라는 상징적 이름을 가지고 있습니다. 예에서, **rr_weight** 속성은 **priorities**로 설정되어 있습니다.

```

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. 설정 파일 장치

표 4.3. "장치 속성" 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. For information on the default configuration value, including supported devices, run either of the following commands.

```

# multipathd show config
# multipath -t

```

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 for the device that the **multipathd show config** command displays and override the values that you want to change.

To add a device that is not configured automatically by default to this section of the configuration file, 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 표 4.3. "장치 속성".

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 available hardware handler is **emc**. If this is not sufficient for your device, you may not be able to configure the device for multipath.

표 4.3. 장치 속성

속성	설명
vendor	예를 들어 COMPAQ 과 같이 device 속성이 적용되는 저장 장치의 공급 업체 이름을 지정합니다.

속성	설명
product	예를 들어 HSV110 (C)COMPAQ 과 같이 device 속성이 적용되는 저장 장치의 제품 이름을 지정합니다.
revision	Specifies the product revision identifier of the storage device.
product_blacklist	제품에 따라 장치를 블랙리스트하는데 사용되는 정규 표현을 지정합니다.
alias_prefix	The user_friendly_names prefix to use for this device type, instead of the default "mpath".
hardware_handler	<p>Specifies a module that will be used to perform hardware specific actions when switching path groups or handling I/O errors. Possible values include:</p> <p>1 emc: hardware handler for EMC storage arrays.</p> <p>1 alua: hardware handler for SCSI-3 ALUA arrays.</p> <p>1 hp_sw: hardware handler for Compaq/HP controllers.</p> <p>1 rdac: hardware handler for the LSI/Engenio RDAC controllers.</p>
path_grouping_policy	<p>미지정된 멀티패스에 적용할 기본값 경로 그룹 정책을 지정합니다. 가능한 값은 다음과 같습니다:</p> <p>failover = 우선 순위 그룹 당 1개의 경로</p> <p>multibus = 1개의 우선 순위 그룹에 있는 유효한 모든 경로</p> <p>group_by_serial = 검색된 시리얼 번호 당 1개의 우선 순위 그룹</p> <p>group_by_prio = 경로 우선 순위 값 당 1개의 우선 순위 그룹</p> <p>group_by_node_name = 대상 노드 이름 당 1개의 우선 순위 그룹</p>

속성	설명
<p>path_selector</p>	<p>Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include:</p> <p>round-robin 0: Loop through every path in the path group, sending the same amount of I/O to each.</p> <p>queue-length 0: Send the next bunch of I/O down the path with the least number of outstanding I/O requests.</p> <p>service-time 0: Send the next bunch of I/O down the path with the shortest estimated service time, which is determined by dividing the total size of the outstanding I/O to each path by its relative throughput.</p>
<p>path_checker</p>	<p>Specifies the default method used to determine the state of the paths. Possible values include:</p> <p>readsector0: Read the first sector of the device.</p> <p>tur: Issue a TEST UNIT READY to the device.</p> <p>emc_clariion: Query the EMC Clariion specific EVPD page 0xC0 to determine the path.</p> <p>hp_sw: Check the path state for HP storage arrays with Active/Standby firmware.</p> <p>rdac: Check the path stat for LSI/Engenio RDAC storage controller.</p> <p>directio: Read the first sector with direct I/O.</p>

속성	설명
<p>features</p>	<p>The default extra features of multipath devices, using the format: "<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>Possible values for features include:</p> <p>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 5.6절. "Issues with queue_if_no_path feature".</p> <p>retain_attached_hw_handler: If this parameter is set to yes and the SCSI layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the multipath.conf file. If the SCSI layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual.</p> <p>pg_init_retries <i>n</i>: Retry path group initialization up to <i>n</i> times before failing where $1 \leq n \leq 50$.</p> <p>pg_init_delay_msecs <i>n</i>: Wait <i>n</i> milliseconds between path group initialization retries where $0 \leq n \leq 60000$.</p>
<p>prio</p>	<p>Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable prio value. Possible values include:</p> <p>const: Set a priority of 1 to all paths.</p> <p>emc: Generate the path priority for EMC arrays.</p> <p>alua: Generate the path priority based on the SCSI-3 ALUA settings. As of Red Hat Enterprise Linux 7.3, if you specify prio "alua exclusive_pref_bit" in your device configuration, multipath will create a path group that contains only the path with the pref bit set and will give that path group the highest priority.</p> <p>ontap: Generate the path priority for NetApp arrays.</p> <p>rdac: Generate the path priority for LSI/Engenio RDAC controller.</p> <p>hp_sw: Generate the path priority for Compaq/HP controller in active/standby mode.</p> <p>hds: Generate the path priority for Hitachi HDS Modular storage arrays.</p>

속성	설명
failback	<p>Manages path group failback.</p> <p>A value of immediate specifies immediate failback to the highest priority path group that contains active paths.</p> <p>A value of manual specifies that there should not be immediate failback but that failback can happen only with operator intervention.</p> <p>A value of followover specifies that automatic failback should be performed when the first path of a path group becomes active. This keeps a node from automatically failing back when another node requested the failover.</p> <p>0 보다 큰 숫자 값을 지정하여 초 단위로 지연 장애 복구가 되도록 합니다.</p>
rr_weight	<p>If set to priorities, then instead of sending rr_min_io requests to a path before calling 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 function. If set to uniform, all path weights are equal.</p>
no_path_retry	<p>A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queuing.</p> <p>fail 값은 대기 상태 없이 즉시 장애가 발생함을 나타냅니다.</p> <p>queue 값은 경로가 수정될 때 까지 대기 상태가 계속 됨을 나타냅니다.</p>
rr_min_io	<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. This setting is only for systems running kernels older than 2.6.31. Newer systems should use rr_min_io_rq. The default value is 1000.</p>
rr_min_io_rq	<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath. This setting should be used on systems running current kernels. On systems running kernels older than 2.6.31, use rr_min_io. The default value is 1.</p>
fast_io_fail_tmo	<p>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 fast_io_fail_tmo option overrides the values of the recovery_tmo and replacement_timeout options. For details, see 4.6절. "iSCSI and DM Multipath overrides".</p>

속성	설명
dev_loss_tmo	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. Setting this to infinity will set this to 2147483647 seconds, or 68 years.
flush_on_last_del	If set to yes , the multipathd daemon will disable queuing 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 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 .
retain_attached_hw_handler	If this parameter is set to yes and the SCSI layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the multipath.conf file. If the SCSI layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual. The default value is no .
detect_prio	If this is set to yes , multipath will first check if the device supports ALUA, and if so it will automatically assign the device the alua prioritizer. If the device does not support ALUA, it will determine the prioritizer as it always does.
uid_attribute	Provides a unique path identifier.
delay_watch_checks	(Red Hat Enterprise Linux Release 7.2 and later) If set to a value greater than 0, the multipathd daemon will watch paths that have recently become valid for the specified number of checks. If they fail again while they are being watched, when they next become valid they will not be used until they have stayed up for the number of consecutive checks specified with delay_wait_checks . This allows you to keep paths that may be unreliable from immediately being put back into use as soon as they come back online.
delay_wait_checks	(Red Hat Enterprise Linux Release 7.2 and later) If set to a value greater than 0, when a device that has recently come back online fails again within the number of checks specified with delay_watch_checks , the next time it comes back online it will be marked and delayed and it will not be used until it has passed the number of checks specified in delay_wait_checks .
deferred_remove	If set to yes , multipathd will do a deferred remove instead of a regular remove when the last path device has been deleted. This ensures that if a multipathed device is in use when a regular remove is performed and the remove fails, the device will automatically be removed when the last user closes the device.

속성	설명
skip_kpartx	(Red Hat Enterprise Linux Release 7.3 and later) If set to yes , kpartx will not automatically create partitions on the device. This allows users to create a multipath device without creating partitions, even if the device has a partition table.
max_sectors_kb	(Red Hat Enterprise Linux Release 7.4 and later) Sets the max_sectors_kb device queue parameter to the specified value on all underlying paths of a multipath device before the multipath device is first activated. When a multipath device is created, the device inherits the max_sectors_kb value from the path devices. Manually raising this value for the multipath device or lowering this value for the path devices can cause multipath to create I/O operations larger than the path devices allow. Using the max_sectors_kb parameter is an easy way to set these values before a multipath device is created on top of the path devices and prevent invalid-sized I/O operations from being passed. If this parameter is not set by the user, the path devices have it set by their device driver, and the multipath device inherits it from the path devices.
detect_path_checker	(Red Hat Enterprise Linux Release 7.4 and later) If set to yes , multipath will try to detect if the device supports ALUA. If so, the device will automatically use the tur path checker. If not, the path_checker will be selected as usual.
all_devs	When this parameter is set to yes , all of the options set in this device configuration will override the values for those options in all of the other device configurations, both the ones in the configuration file and the built-in defaults.
all_tg_pt	(Red Hat Enterprise Linux Release 7.6 and later) If this option is set to yes , when mppathpersist registers keys it will treat a key registered from one host to one target port as going from one host to all target ports. This must be set to yes to successfully use mppathpersist on arrays that automatically set and clear registration keys on all target ports from a host, instead of per target port per host.

다음의 예에서는 멀티패스 설정 파일에 있는 **device** 항목을 보여주고 있습니다.

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

The following configuration sets **no_path_retry** to **fail** for all of the built-in device configurations.

```
devices {
```

```
device {  
  all_devs yes  
  no_path_retry fail  
}  
}
```

4.6. ISCSI AND DM MULTIPATH OVERRIDES

The **recovery_tmo sysfs** option controls the timeout for a particular iSCSI device. The following options globally override **recovery_tmo** values:

- The **replacement_timeout** configuration option globally overrides the **recovery_tmo** value for all iSCSI devices.
- For all iSCSI devices that are managed by DM Multipath, the **fast_io_fail_tmo** option in DM Multipath globally overrides the **recovery_tmo** value. The **fast_io_fail_tmo** option in DM Multipath also overrides the **fast_io_fail_tmo** option in Fibre Channel devices.

The DM Multipath **fast_io_fail_tmo** option takes precedence over **replacement_timeout**. Red Hat does not recommend using **replacement_timeout** to override **recovery_tmo** in devices managed by DM Multipath because DM Multipath always resets **recovery_tmo** when the **multipathd** service reloads.

5장. DM MULTIPATH ADMINISTRATION AND TROUBLESHOOTING

This chapter provides information on administering DM Multipath on a running system.

5.1. AUTOMATIC CONFIGURATION FILE GENERATION WITH MULTIPATH HELPER

You can generate a basic configuration for multipath devices on Red Hat Enterprise Linux with the Multipath Helper application. The application gives you options to create multipath configurations with custom aliases, device blacklists, and settings for the characteristics of individual multipath devices. Upon completion, the application generates an installation script that includes the configuration parameters you selected and it provides a **multipath.conf** configuration file for review.

The Multipath Helper application can be found at <https://access.redhat.com/labsinfo/multipathhelper>.

5.2. RESIZING AN ONLINE MULTIPATH DEVICE

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

1. Resize your physical device.
2. Execute 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/path_device/device/rescan
```

Ensure that you run this command for each of the path devices. For example, if your path devices are **sda**, **sdb**, **sde**, and **sdf**, you would run the following commands:

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

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

```
# multipathd resize map multipath_device
```

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

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

5.3. MOVING ROOT FILE SYSTEMS FROM A SINGLE PATH DEVICE TO A MULTIPATH DEVICE

If you have installed your system on a single-path device and later add another path to the root file system, you will need to move your root file system to a multipathed device. This section documents the procedure for moving from a single-path to a multipathed device.

After ensuring that you have installed the **device-mapper-multipath** package, perform the following procedure:

1. Execute the following command to create the **/etc/multipath.conf** configuration file, load the multipath module, and set **chkconfig** for the **multipathd** to **on**:

```
# mpathconf --enable
```

For further information on using the **mpathconf** command to set up multipathing, see [3.1절. "Setting Up DM Multipath"](#).

2. If the **find_multipaths** configuration parameter is not set to **yes**, edit the **blacklist** and **blacklist_exceptions** sections of the **/etc/multipath.conf** file, as described in [4.2절. "설정 파일 블랙리스트"](#).
3. In order for multipath to build a multipath device on top of the root device as soon as it is discovered, enter the following command. This command also ensures that **find_multipaths** will allow the device, even if it only has one path.

```
# multipath -a root_devname
```

For example, if the root device is **/dev/sdb**, enter the following command.

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

4. To confirm that your configuration file is set up correctly, you can enter the **multipath** command and search the output for a line of the following format. This indicates that the command failed to create the multipath device.

```
date wwid: ignoring map
```

For example, if the WWID of the device is 3600d02300069c9ce09d41c4ac9c53200, you would see a line in the output such as the following:

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

5. To rebuild the **initramfs** file system with **multipath**, execute the **dracut** command with the following options:

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

6. Shut the machine down.
7. Configure the FC switch so that other paths are visible to the machine.
8. Boot the machine.
9. Check whether the root file system (**/**) is on the multipathed device.

5.4. MOVING SWAP FILE SYSTEMS FROM A SINGLE PATH DEVICE TO A MULTIPATH DEVICE

By default, swap devices are set up as logical volumes. This does not require any special procedure for configuring them as multipath devices as long as you set up multipathing on the physical volumes that constitute the logical volume group. If your swap device is not an LVM volume, however, and it is mounted by device name, you may need to edit the `/etc/fstab` file to switch to the appropriate multipath device name.

1. Determine the WWID number of the swap device by running the `/sbin/multipath` command with the `-v3` option. The output from the command should show the swap device in the paths list.

You should look in the command output for a line of the following format, showing the swap device:

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

For example, if your swap file system is set up on `sda` or one of its partitions, you would see a line in the output such as the following:

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

2. Set up an alias for the swap device in the `/etc/multipath.conf` file:

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

3. Edit the `/etc/fstab` file and replace the old device path to the root device with the multipath device.

For example, if you had the following entry in the `/etc/fstab` file:

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

You would change the entry to the following:

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

5.5. 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 [3장. Setting Up DM Multipath](#). The `multipathd` daemon must be running in order to use multipathed devices.

5.6. ISSUES WITH QUEUE_IF_NO_PATH FEATURE

If a multipath device is configured with **features "1 queue_if_no_path"**, 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).

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 **mpathc** from **"queue_if_no_path"** to **"fail_if_no_path"**, execute the following command.

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

경로 대신 **mpathn** 별칭을 지정하셔야 함에 유의합니다.

5.7. MULTIPATH COMMAND OUTPUT

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

각각의 멀티패스 장치의 경우:

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

각각의 경로 그룹의 경우:

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

각각의 경로의 경우:

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

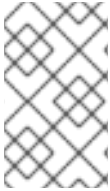
예를 들어, multipath 명령의 출력 결과는 다음과 같이 나타나게 됩니다:

```
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=active
|`- 6:0:0:0 sdb 8:16 active ready running
`+- policy='round-robin 0' prio=1 status=enabled
  `- 7:0:0:0 sdf 8:80 active ready running
```

If the path is up and ready for I/O, the status of the path is **ready** or **ghost**. If the path is down, the status is **faulty** or **shaky**. 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.

The possible values for **online_status** are **running** and **offline**. A status of **offline** means that this SCSI device has been disabled.



참고

When a multipath device is being created or modified, the path group status, the dm device name, the write permissions, and the dm status are not known. Also, the features are not always correct.

5.8. 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** option displays in addition to all other available components of the system.

멀티패스 설정 보기를 할 때, **multipath** 명령의 **-v** 옵션으로 지정할 수 있는 세 가지 레벨의 상세 보기가 있습니다. **-v0** 옵션을 지정하면 아무런 출력 결과를 얻지 못하게 됩니다. **-v1** 옵션으로는 생성되거나 업데이트된 멀티패스 이름만을 출력하게 되어, **kpartx**와 같은 다른 도구에 사용할 수 있습니다. **-v2** 옵션을 지정하면 검색된 모든 경로, 멀티패스, 장치 맵을 출력하게 됩니다.

다음의 예는 **multipath -l** 명령의 출력 결과를 보여주고 있습니다.

```
# multipath -l
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=active
|`- 6:0:0:0 sdb 8:16 active ready running
`+- policy='round-robin 0' prio=1 status=enabled
  `- 7:0:0:0 sdf 8:80 active ready running
```

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

```
# multipath -ll
3600d0230000000000e13955cc3757801 dm-10 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=enabled
|`- 19:0:0:1 sdc 8:32 active ready running
`+- policy='round-robin 0' prio=1 status=enabled
  `- 18:0:0:1 sdh 8:112 active ready running
3600d0230000000000e13955cc3757803 dm-2 WINSYS,SF2372
size=125G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=active
|`- 19:0:0:3 sde 8:64 active ready running
  `- 18:0:0:3 sdj 8:144 active ready running
```

5.9. MULTIPATH COMMAND OPTIONS

표 5.1. "유용한 **multipath** 명령 옵션" describes some options of the **multipath** command that you may find useful.

표 5.1. 유용한 **multipath** 명령 옵션

옵션	설명
-l	sysfs 및 장치 매핑에서 얻은 현재 멀티패스 설정을 보여줍니다.

옵션	설명
-ll	sysfs , 장치 매핑, 시스템에서 사용 가능한 기타 다른 구성 요소에서 얻은 현재 멀티패스 설정을 보여줍니다.
-f device	이름이 지정된 멀티패스 장치를 삭제합니다.
-F	Remove all unused multipath devices.
-w device	Remove the wwid of the specified device from the wwids file.
-W	Reset the wwids file to include only the current multipath devices.

5.10. DMSETUP 명령을 사용하여 장치 매핑 항목 지정

dmsetup 명령을 사용하여 어떤 장치 매핑 항목이 멀티패스 장치와 일치하는 지를 알 수 있습니다.

다음의 명령으로 모든 장치 매핑 장치 및 주 장치 번호와 부 장치 번호를 볼 수 있습니다. 부 장치 번호로는 dm 장치의 이름을 지정합니다. 예를 들어, 3에 대한 부 장치 번호는 멀티패스 장치 **/dev/dm-3**에 해당하게 됩니다.

```
# dmsetup ls
mpathd (253:4)
mpathep1 (253:12)
mpathfp1 (253:11)
mpathb (253:3)
mpathgp1 (253:14)
mpathhp1 (253:13)
mpatha (253:2)
mpathh (253:9)
mpathg (253:8)
VolGroup00-LogVol01 (253:1)
mpathf (253:7)
VolGroup00-LogVol00 (253:0)
mpathe (253:6)
mpathbp1 (253:10)
mpathd (253:5)
```

5.11. THE MULTIPATHD COMMANDS

The **multipathd** commands can be used to administer the **multipathd** daemon. For information on the available **multipathd** commands, see the **multipathd(8)** man page.

The following command shows the standard default format for the output of the **multipathd show maps** command.

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

Some **multipathd** commands include a **format** option followed by a wildcard. You can display a list of available wildcards with the following command.

```
# multipathd show wildcards
```

As of Red Hat Enterprise Linux release 7.3, the **multipathd** command supports new format commands that show the status of multipath devices and paths in "raw" format versions. In raw format, no headers are printed and the fields are not padded to align the columns with the headers. Instead, the fields print exactly as specified in the format string. This output can then be more easily used for scripting. You can display the wildcards used in the format string with the **multipathd show wildcards** command.

The following **multipathd** commands show the multipath devices that **multipathd** is monitoring, using a format string with multipath wildcards, in regular and raw format.

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

The following **multipathd** commands show the paths that **multipathd** is monitoring, using a format string with multipath wildcards, in regular and raw format.

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

The following commands show the difference between the non-raw and raw formats for the **multipathd show maps**. Note that in **raw** format there are no headers and only a single space between the columns.

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

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

5.12. 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 executing 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.

multipathd 대화식 콘솔은 시스템에서 발생할 수 있는 문제를 해결하는데 사용될 수 있습니다. 예를 들어, 다음과 같은 순서의 명령으로 콘솔이 나타나기 이전에 기본값이 포함된 멀티패스 설정 사항을 볼 수 있습니다.

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

다음과 같은 순서의 명령으로 멀티패스가 **multipath.conf**에 변경된 사항을 발견하는 지를 확인합니다.

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

다음과 같은 순서의 명령으로 경로 체크 기능이 올바르게 작동하는지를 확인합니다.

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

5.13. CLEANING UP MULTIPATH FILES ON PACKAGE REMOVAL

If you should have occasion to remove the **device-mapper-multipath rpm** file, note that this does not remove the **/etc/multipath.conf**, **/etc/multipath/bindings**, and **/etc/multipath/wwids** files. You may need to remove those files manually on subsequent installations of the **device-mapper-multipath** package.

부록 A. 개정 내역

고침 7.1-1 Document version for 7.7 GA publication.	Wed Aug 7 2019	Steven Levine
고침 6.1-1 Document version for 7.6 GA publication.	Thu Oct 4 2018	Steven Levine
고침 5.1-1 Document version for 7.5 GA publication.	Thu Mar 15 2018	Steven Levine
고침 5.1-0 Document version for 7.5 Beta publication.	Wed Dec 13 2017	Steven Levine
고침 4.0-6 Document version for 7.4 GA publication.	Wed Jul 19 2017	Steven Levine
고침 4.0-3 Preparing document for 7.4 Beta publication	Wed May 10 2017	Steven Levine
고침 0.4-5 Small update to 7.3 version	Wed Nov 16 2016	Steven Levine
고침 0.4-3 Version for 7.3 GA publication	Mon Oct 24 2016	Steven Levine
고침 0.4-1 Preparing document for 7.3 Beta publication	Mon Aug 15 2016	Steven Levine
고침 0.3-6 Preparing document for 7.2 GA publication	Mon Nov 9 2015	Steven Levine
고침 0.3-3 Preparing document for 7.2 Beta publication.	Wed Aug 19 2015	Steven Levine
고침 0.2-7 Version for 7.1 GA release	Mon Feb 16 2015	Steven Levine
고침 0.2-6 Version for 7.1 Beta release	Thu Dec 11 2014	Steven Levine
고침 0.1-22 Version for 7.0 GA release	Mon Jun 2 2014	Steven Levine
고침 0.1-10 7.0 Beta update	Wed Apr 9 2014	Steven Levine
고침 0.1-3 Version for 7.0 pre-Beta	Tue Nov 26 2013	Steven Levine
고침 0.1-1 Branched from the Red Hat Enterprise Linux 6 version of the document	Wed Jan 16 2013	Steven Levine

색인

Symbols

/etc/multipath.conf package, [Setting Up DM Multipath](#)

멀티패스 장치, [멀티패스 장치](#)

장애조치, [Overview of DM Multipath](#)

장치 이름, [멀티패스 장치 식별자](#)

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

configuration file, [멀티패스 장치 식별자](#)

alias 매개 변수, [멀티패스 장치 설정 속성](#)

alias_prefix parameter, [설정 파일 장치](#)

all_devs parameter, [설정 파일 장치](#)

all_tg_pt parameter, [설정 파일 기본값](#), [설정 파일 장치](#)

B

blacklist

configuration file, [설정 파일 블랙리스트](#)

default devices, [장치 이름에 따라 블랙리스트하기](#)

device name, [장치 이름에 따라 블랙리스트하기](#)

device protocol, [Blacklisting By Device Protocol \(Red Hat Enterprise Linux 7.6 and Later\)](#)

device type, [장치 유형에 따라 블랙리스트하기](#)

udev property, [Blacklisting By udev Property \(Red Hat Enterprise Linux 7.5 and Later\)](#)

WWID, [WWID에 따라 블랙리스트하기](#)

blacklist_exceptions section

multipath.conf file, [블랙리스트 예외 설정](#)

C

checker_timeout parameter, [설정 파일 기본값](#)

configuration file

alias parameter, [멀티패스 장치 설정 속성](#)

alias_prefix parameter, [설정 파일 장치](#)

all_devs parameter, 설정 파일 장치
 all_tg_pt parameter, 설정 파일 기본값, 설정 파일 장치
 blacklist, 설정 파일 블랙리스트
 checker_timeout parameter, 설정 파일 기본값
 config_dir parameter, 설정 파일 기본값
 deferred_remove parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 delay_wait_checks parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 delay_watch_checks parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 detect_path_checker parameter, 설정 파일 기본값, 설정 파일 장치
 detect_prio parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성
 dev_loss_tmo parameter, 설정 파일 기본값, 설정 파일 장치
 disable_changed_wwids parameter, 설정 파일 기본값
 fallback parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 fast_io_fail_tmo parameter, 설정 파일 기본값, 설정 파일 장치
 features parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 flush_on_last_del parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 force_sync parameter, 설정 파일 기본값
 hardware_handler parameter, 설정 파일 장치
 hw_string_match parameter, 설정 파일 기본값
 ignore_new_boot_devs parameter, 설정 파일 기본값
 log_checker_err parameter, 설정 파일 기본값
 max_fds parameter, 설정 파일 기본값
 max_sectors_kb parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 new_bindings_in_boot parameter, 설정 파일 기본값
 no_path_retry parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 overview, 설정 파일 개요
 path_checker parameter, 설정 파일 기본값, 설정 파일 장치
 path_grouping_policy parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 path_selector parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성, 설정 파일 장치
 polling-interval parameter, 설정 파일 기본값
 prio parameter, 설정 파일 기본값, 설정 파일 장치
 prkeys_file parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성
 product parameter, 설정 파일 장치
 product_blacklist parameter, 설정 파일 장치
 queue_without_daemon parameter, 설정 파일 기본값
 reassign_maps parameter, 설정 파일 기본값
 remove_retries parameter, 설정 파일 기본값
 retain_attached_hw_handler parameter, 설정 파일 기본값, 멀티패스 장치 설정 속성
 retrigger_delay parameter, 설정 파일 기본값
 retrigger_tries parameter, 설정 파일 기본값
 revision parameter, 설정 파일 장치

rr_min_io parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#)
 rr_weight parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)
 skip_kpartx parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)
 uid_attribute parameter, [설정 파일 기본값](#), [설정 파일 장치](#)
 user_friendly_names parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)
 vendor parameter, [설정 파일 장치](#)
 verbosity parameter, [설정 파일 기본값](#)
 wwid parameter, [멀티패스 장치 설정 속성](#)

configuring

DM Multipath, [Setting Up DM Multipath](#)

config_dir parameter, [설정 파일 기본값](#)

D

defaults section

multipath.conf file, [설정 파일 기본값](#)

deferred_remove parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)
 delay_wait_checks parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)
 delay_watch_checks parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)
 detect_path_checker parameter, [설정 파일 기본값](#), [설정 파일 장치](#)
 detect_prio parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#)
 dev/mapper directory, [멀티패스 장치 식별자](#)
 device-mapper-multipath package, [Setting Up DM Multipath](#)

devices

adding, [Configuring Storage Devices](#), [설정 파일 장치](#)

devices section

multipath.conf file, [설정 파일 장치](#)

dev_loss_tmo parameter, [설정 파일 기본값](#), [설정 파일 장치](#)

disable_changed_wwids parameter, [설정 파일 기본값](#)

DM Multipath

and LVM, [논리 볼륨에 있는 멀티패스 장치](#)
 components, [DM Multipath Components](#)
 configuration file, [The DM Multipath Configuration File](#)
 configuring, [Setting Up DM Multipath](#)
 definition, [DM-Multipath \(Device Mapper Multipathing\)](#)
 device name, [멀티패스 장치 식별자](#)
 devices, [멀티패스 장치](#)
 failover, [Overview of DM Multipath](#)
 overview, [Overview of DM Multipath](#)
 redundancy, [Overview of DM Multipath](#)

setup, [Setting Up DM Multipath](#)

setup, overview, [DM Multipath Setup Overview](#)

dm-n devices, [멀티패스 장치 식별자](#)

dmsetup command, determining device mapper entries, [dmsetup 명령을 사용하여 장치 매핑 항목 지정](#)

dm_multipath kernel module, [DM Multipath Components](#)

F

failback 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

fast_io_fail_tmo parameter, [설정 파일 기본값](#), [설정 파일 장치](#)

features 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

features, new and changed, [New and Changed Features](#)

flush_on_last_del 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

force_sync parameter, [설정 파일 기본값](#)

H

hardware_handler 매개 변수, [설정 파일 장치](#)

hw_string_match parameter, [설정 파일 기본값](#)

I

ignore_new_boot_devs parameter, [설정 파일 기본값](#)

initramfs

starting multipath, [Setting Up Multipathing in the initramfs File System](#)

K

kpartx command, [DM Multipath Components](#)

L

local disks, ignoring, [멀티패스 장치 생성 시 로컬 디스크 무시](#)

log_checker_err parameter, [설정 파일 기본값](#)

LVM physical volumes

multipath devices, [논리 볼륨에 있는 멀티패스 장치](#)

lvm.conf file, [논리 볼륨에 있는 멀티패스 장치](#)

M

max_fds 매개 변수, [설정 파일 기본값](#)

max_sectors_kb parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

mpathconf command, [DM Multipath Components](#)

multipath command, [DM Multipath Components](#)

options, [Multipath Command Options](#)

output, [Multipath Command Output](#)

queries, [Multipath Queries with multipath Command](#)

multipath daemon (multipathd), [The Multipath Daemon](#)

multipath devices

logical volumes, [논리 볼륨에 있는 멀티패스 장치](#)

LVM physical volumes, [논리 볼륨에 있는 멀티패스 장치](#)

Multipath Helper, [Automatic Configuration File Generation with Multipath Helper](#)

multipath.conf file, [저장 장치 어레이 지원](#)

blacklist_exceptions section, [블랙리스트 예외 설정](#)

defaults section, [설정 파일 기본값](#)

devices section, [설정 파일 장치](#)

multipaths section, [멀티패스 장치 설정 속성](#)

multipath.conf 파일, [The DM Multipath Configuration File](#)

multipathd

command, [Troubleshooting with the multipathd Interactive Console](#)

interactive console, [Troubleshooting with the multipathd Interactive Console](#)

multipathd daemon, [DM Multipath Components](#)

multipathd start command, [Setting Up DM Multipath](#)

multipathed root file system, [Moving root File Systems from a Single Path Device to a Multipath Device](#)

multipathed swap file system, [Moving swap File Systems from a Single Path Device to a Multipath Device](#)

multipaths section

multipath.conf file, [멀티패스 장치 설정 속성](#)

N

new_bindings_in_boot parameter, [설정 파일 기본값](#)

no_path_retry 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

O

overview

features, new and changed, [New and Changed Features](#)

P

path_checker 매개 변수, [설정 파일 기본값](#), [설정 파일 장치](#)

path_grouping_policy 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

path_selector 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

polling_interval parameter, [설정 파일 기본값](#)

prio parameter, [설정 파일 기본값](#), [설정 파일 장치](#)

prkeys_file parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#)

product 매개 변수, [설정 파일 장치](#)

product_blacklist 매개 변수, [설정 파일 장치](#)

Q

queue_without_daemon 매개 변수, [설정 파일 기본값](#)

R

reassign_maps parameter, [설정 파일 기본값](#)

remove_retries parameter, [설정 파일 기본값](#)

resizing a multipath device, [Resizing an Online Multipath Device](#)

retain_attached_hw_handler parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#)

retrigger_delay parameter, [설정 파일 기본값](#)

retrigger_tries parameter, [설정 파일 기본값](#)

revision parameter, [설정 파일 장치](#)

root file system, [Moving root File Systems from a Single Path Device to a Multipath Device](#)

rr_min_io 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#)

rr_weight 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

S

setup

DM Multipath, [Setting Up DM Multipath](#)

skip_kpartxr parameter, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

storage array support, [저장 장치 어레이 지원](#)

storage arrays

adding, [Configuring Storage Devices](#), [설정 파일 장치](#)

swap file system, [Moving swap File Systems from a Single Path Device to a Multipath Device](#)

U

uid_attribute parameter, [설정 파일 기본값](#), [설정 파일 장치](#)

user_friendly_names parameter, [멀티패스 장치 식별자](#)

user_friendly_names 매개 변수, [설정 파일 기본값](#), [멀티패스 장치 설정 속성](#), [설정 파일 장치](#)

V

vendor 매개 변수, [설정 파일 장치](#)

verbosity parameter, [설정 파일 기본값](#)

W

World Wide Identifier (WWID), [멀티패스 장치 식별자](#)

wwid 매개 변수, [멀티패스 장치 설정 속성](#)

