Device Mapper Configuration Guide for HP Arrays

Part number: AA-RW8RA-TE
First edition: April 2006

Legal and notice information

Printed in the U.S.

Device Mapper Configuration Guide for HP Arrays

Contents

About this guide 5
Intended audience 5
Document conventions and symbols 5
HP technical support 6
Helpful web sites 6

1 Abbreviations 7

2 Introduction to Device Mapper Multipath 9
Overview 9
Features 9
Configurations currently supported 10
Software components 10
Device Mapper Multipath Operation 10

3 Managing Device Mapper Multipath for HP supported storage arrays 11
Preparing for installation 11
Device Mapper Multipath support matrix 12
Setting up Device Mapper Multipath for HP Arrays 13
Configuring HBA parameters 13
Configuring QLogic HBA parameters 13
Configuring Emulex HBA parameters 13
Setting up Device Mapper Multipath daemons 14
Configuring Device Mapper Multipath for HP arrays 14
Starting Device Mapper Multipath for HP arrays 18
Using Device Mapper Multipath for HP arrays 18

4 Known issues 21

Tables

1 Document conventions 5
2 Abbreviations and their definitions 7
3 Device Mapper Multipath supported hardware and software 12
4 HP supported array attributes 16
5 Basic operations of Device Mapper Multipath 20

About this guide

This guide describes the Device Mapper Multipath software and provides information to help you:

- Determine hardware and software prerequisites.
- Install Device Mapper Multipath software.
- Use and manage Device Mapper Multipath software.

Intended audience

This document is intended for users who are responsible for installing, configuring, and maintaining Device Mapper Multipath in their Linux server environment, assuming that you are familiar with Linux system administration, including hardware and software installation.

Document conventions and symbols

Table 1 Document conventions

Convention	Element
Medium blue, underlined text (http://www.hp.com)	Web site addresses
Bold font	• Key names • Text typed into a GUI element, such as into a box • GUI elements that are clicked or selected, such as menu and list items, buttons, and check boxes
Italic font	Text emphasis
`Monospace font`	• File and directory names • System output • Code • Text typed at the command line
`Monospace, italic font`	• Code variables • Command-line variables
`Monospace, bold font`	Emphasis of file and directory names, system output, code, and text typed at the command line

⚠ WARNING!

Indicates that failure to follow directions can result in bodily harm or death.

△ CAUTION:

Indicates that failure to follow directions can result in damage to equipment or data.

IMPORTANT:

Provides clarifying information or specific instructions.

NOTE:

Provides additional information.

HP technical support

Telephone numbers for worldwide technical support are listed on the HP support web site: http://www.hp.com/support/.

Collect the following information before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

For continuous quality improvement, calls may be recorded or monitored.

HP recommends that customers sign up online using the Subscriber's choice web site: http://www.hp.com/go/e-updates.

- Subscribing to this service provides you with e-mail updates on the latest product enhancements, newer versions of drivers, and firmware documentation updates as well as instant access to numerous other product resources.
- After signing up, you can locate your products by selecting **Business support** and then **Storage** under Product Category.

Helpful web sites

For other product information, see the following HP web sites:

- http://www.hp.com
- http://www.hp.com/country/us/eng/prodserv/storage.html
- http://www.hp.com/support/
- http://www.docs.hp.com
- http://christophe.varoqui.free.fr/
- http://christophe.varoqui.free.fr/multipath.html
- http://sourceware.org/dm/

1 Abbreviations

Table 2 lists the abbreviations and their definitions used in this document

Table 2 Abbreviations and their definitions

Abbreviations/Acronyms	Definition
DM	Device Mapper
GUI	Graphical User Interface
HBA	Host Bus Adapter
I/O	Input/Output
LUN	Logical Unit Number
LVM	Logical Volume Manager
CLI	Command Line Interpreter
OS	Operating System
RAID	Redundant Array of Independent (or Inexpensive) Disks
RHEL	Red Hat Enterprise Linux
SAN	Storage Area Network
SLES	SuSE LINUX Enterprise Server
UID	Unique Identifier
WWID	World-Wide Identifier
WWN	World-Wide Name
FC	Fibre Channel
SCSI	Small Computer System Interface

2 Introduction to Device Mapper Multipath

This chapter describes the following:

- Overview
- Features
- Configurations currently supported
- Software components
- Device Mapper Multipath Operation

Overview

The Device Mapper is a new infrastructure in Linux 2.6 kernel that provides a generic way to create virtual layers of block devices. The Device Mapper supports stripping, mirroring, snapshots, concatenation and multipathing. Device Mapper block devices are also supported by Logical Volume Manager (LVM). The Device Mapper Multipath support is provided in 2.6 kernel by the combination of both the Device Mapper Multipath kernel modules and the multipath-tools user-space package.

Features

Device Mapper Multipath provides the following features:

- Allows the multivendor Storage RAID systems and host servers equipped with multivendor Host Bus Adaptors (HBAs) redundant physical connectivity along the independent Fibre Channel fabric paths available
- Monitors each path and automatically reroutes (failover) I/O to an available functioning alternate path if an existing connection fails
- Provides an option to perform fail-back of the LUN to the repaired paths
- Implements failover or failback actions transparently without disrupting applications
- Monitors each path and notifies if there is a change in the path status
- Facilitates the load balancing among the multiple paths
- Provides CLI with display options to configure and manage Multipath features
- Provides all Device Mapper Multipath features support for any LUN newly added to the host
- Provides an option to have customized names for the Device Mapper Multipath devices
- Provides persistency to the Device Mapper Multipath devices across reboots if there are any change in the Storage Area Network
- Provides policy based path grouping for the user to customize the I/O flow through specific set of paths

NOTE:

This first release of Device Mapper is intended for use in limited environments. It is recommended to use with test environments and non-mission critical applications, to become familiar with the Device Mapper functionality. It is not recommended to use Device Mapper Multipath for your mission critical applications at this time, because caution should be exercised when implementing any new software into a production environment. Only features described in this document are supported with this release. Future releases will have increased functionality and usability.

Configurations currently supported

Device Mapper Multipath supports the following configurations:

- XP support on RHEL4 U3 and SLES9 SP3 and EVA3000/5000/4000/6000/8000 on RHEL4 U3 only
- Single Host with 2 HBAs connected to a single storage device
- The HBAs must be from the same vendor
- Maximum number of LUNs supported is 50
- Only base partition is supported
- Non coexisting multipath products
- No Array Active-Passive support (HSV 100, HSV 110, MSA 1000, MSA 1500, HSG)
- Only x86 and x86_64 architectures are supported
- Wild card or regular expression support for the product and vendor string (in the `/etc/multipath.conf` file) is only supported in RHEL4 U3
- Manually enter the device properties for HP arrays in the `/etc/multipath.conf` file (for both RHEL4 and SLES9 systems)
- Time based I/O request queuing (no_path_retry in the `/etc/multipath.conf file`) is only supported in RHEL4 U3
- Support for SAN boot is not available
- Supports only single path access to root device
- Provides only CLI support and not GUI support
- Supports only round-robin load balancing policy

Software components

This section describes the software components for Device Mapper Multipath support:

- **dm-multipath kernel module** -- Routes I/O and does failover to paths and path groups
- **multipath configuration tool** -- Provides commands to configure, list, and flush Multipath devices. The command is run in `rc.sysinit` during startup, and by `udev`, whenever a block device is added
- **multipathd daemon** -- Monitors paths to check if faulty paths have been fixed. When paths revert, `multipathd` may also initiate path group switches to ensure that the optimal path group is being used. Also, it is possible to interactively modify a multipath device
- **kpartx utility** -- Creates Device Mapper Multipath devices for partitions on a device. You must use this command for MS DOS based partitions with Device Mapper Multipath

Device Mapper Multipath Operation

Device Mapper Multipath allows hosts to route I/O over the multiple paths available to an end storage unit (LUN). A path refers to the connection from an HBA port to a storage controller port. When an active path through which I/O happens fails, Device Mapper Multipath reroutes the I/O over other available paths. In a Linux host, when there are multiple paths to a storage controller, each path appears as a separate block device and hence results in multiple block devices for single LUN. Device Mapper Multipath creates a new Multipath block device for those devices having the same LUN WWN. For example, a host with two HBAs attached to a storage controller with two ports via a single unzoned FC switch sees four block devices: `/dev/sda`, `/dev/sdb`, `/dev/sdc`, and `/dev/sdd`. Device Mapper Multipath creates a single block device, `/dev/mpath/mpath1` that reroutes I/O through those four underlying block devices.

3 Managing Device Mapper Multipath for HP supported storage arrays

This chapter describes the following:

- Preparing for installation
- Device Mapper Multipath support matrix
- Setting up Device Mapper Multipath for HP Arrays
- Configuring Device Mapper Multipath for HP Arrays
- Starting Device Mapper Multipath for HP arrays
- Using Device Mapper Multipath for HP arrays

Preparing for installation

Before installing the Device Mapper Multipath, verify that your system components support Device Mapper Multipath and if you need to upgrade hardware or software.

To prepare your system for the installation of Device Mapper Multipath:

- Ensure that the Device Mapper package is installed in the system and is available with OS distribution CDs.
  - For RHEL4:
    `device-mapper-1.02.02-3.0.rpm`
    `device-mapper-multipath-0.4.5-12.0.RHEL4.rpm`
  - For SLES9:
    `device-mapper-1.01.01-1.6.i586.rpm`
    `multipath-tools-0.4.5-0.11.i586.rpm`
- Ensure that the latest supported HBA drivers are installed in the system

See Table 3 for information on supported servers, HBAs, drivers, and storage arrays.

Device Mapper Multipath support matrix

Table 3 lists the hardware and software supported by Device Mapper Multipath.

Table 3 Device Mapper Multipath supported hardware and software

System feature	Supported hardware and software
Operating system versions	Red Hat Enterprise Linux 4 Update 3 SuSE LINUX Enterprise Server 9, SP3
Host Bus Adapters (HBA)	**Qlogic:** • FCA2214, FCA2214DC (PCI-X v1.0 2Gb) • A7538A Qlogic 1p2g for Integrity Linux • A6826A Qlogic 2p2g for Integrity Linux • FC1142SR and FC1242SR PCI Express 4Gb **Emulex:** • EZ & E2 2Gb Mezzanine for p class Blade Servers • A8002A/LPe1150, A8003A/LPe11002 (PCI express 4Gb) • AD167A, AD168A (PCI-X v2.0 4Gb)
Servers	ProLiant Blade Servers Integrity, ProLiant x86, ProLiant AMD64, ProLiant EM64T
Supported arrays	EVA3000 (HSV101) fw rev 4001 EVA5000 (HSV111) fw rev 4001 EVA4000/6000/8000 fw rev 5031 XP128* fw rev 21-14-18-00/00 XP1024* fw rev 21-14-18-00/00 XP10000 fw rev 50-05-06-00/00 XP12000 fw rev 50-05-06-00/00 *If you use XP128/1024, the system modes 140 and 293 need to be switched on, otherwise the XP will not respond to a SCSI inquiry to code page 0x83 with a unique serial number (scsi_id tool). This would prevent the Device Mapper multipathing from detecting the redundant paths.
HBA drivers	Emulex: version 8.0.16.21, available at http://h18006.www1.hp.com/products/storageworks/4gbpciehba/index.html Qlogic: version 8.01.03-14, available at http://h18006.www1.hp.com/products/storageworks/fca2214/index.html

Setting up Device Mapper Multipath for HP Arrays

This section describes the following:

- Configuring HBA parameters
- Setting up Device Mapper Multipath daemons

Configuring HBA parameters

You must configure HBA parameters for the Device Mapper Multipath. The HBA time outs are typically setup for non Device Mapper Multipath environments, where longer time outs are necessary. This is because the only alternative is to send an error to the application. However, with Multipath, errors like cable failures must be intimated at the earliest so that the Multipath layer can quickly take action and redirect the I/O to another path.

This section describes the following:

- Configuring QLogic HBA parameters
- Configuring Emulex HBA parameters

Configuring QLogic HBA parameters

To configure QLogic HBA parameter, complete the following steps:

1. For QLogic `2xxx` family of HBAs, edit the `/etc/modprobe.conf` file in RHEL4 hosts and `/etc/modprobe.conf.local` file in SLES9 hosts with the following values:

   ```
   options qla2xxx qlport_down_retry=1 ql2xfailover=0 ql2xretrycount=5
   ```
2. Rebuild the initrd by executing the following script:

   ```
   /opt/hp/src/hp_qla2x00src/make_initrd
   ```
3. Reboot the host

> **NOTE:**
> For additional information, refer the Installation and Reference Guide of the respective HBA Drivers.

Configuring Emulex HBA parameters

To configure Emulex HBA parameter, complete the following steps:

1. For Emulex `lpfc` family of HBAs, edit the `/etc/modprobe.conf` file in RHEL4 hosts and `/etc/modprobe.conf.local` file in SLES9 hosts with the following values:

   ```
   options lpfc  lpfc_nodev_tmo=14 lpfc_lun_queue_depth=30 lpfc_discovery_threads=1
   ```
2. Rebuild the `initrd` by executing the following script:

   ```
   /opt/hp/hp-lpfc/make_initrd
   ```
3. Reboot the host

> **NOTE:**
> Retain the SCSI command timeout value for all the LUNs as 30, which is the default value. For additional information, refer the Installation and Reference Guide of the respective HBA Drivers.

Setting up Device Mapper Multipath daemons

You must set the Device Mapper Multipath daemons to start at boot time. The `multipathd` daemon does not automatically start while booting the system.

For RHEL4 hosts:

Complete the following steps on your RHEL4 host to start the `multipathd` daemon at boot time:

1. Run the following command to check if the daemon is configured to start in any run level:

```
# chkconfig --list multipathd
```

2. Run the following commands to start the `multipathd` daemon during boot time in the desired run level:

```
# chkconfig [--level levels] multipathd on
# chkconfig multipathd
```

For SLES9 hosts:

Complete the following steps on your SLES9 host to start the `multipathd` daemon at boot time:

1. Run the following commands to check if the daemon is configured to start at boot time

```
# chkconfig --list boot.device-mapper
# chkconfig --list boot.multipath
# chkconfig --list multipathd
```

2. Run the following command to start the Device Mapper Multipath daemons if they are not configured to start at boot time:

```
# chkconfig  boot.device-mapper [levels]
# chkconfig  boot.multipath [levels]
# chkconfig  multipathd [levels]
```

NOTE:

The levels values vary between 2 and 5 based on the desired run level.

After completing the above steps, you must reboot the host. To verify whether the modules required by Device Mapper Multipath are loaded and the `multipathd` daemon has started automatically at boot time, run the following command and verify if `dm_multipath` and `dm_mod` modules are listed:

```
lsmod | grep dm_multipath
lsmod | grep dm_mod
```

Configuring Device Mapper Multipath for HP arrays

Configuring Device Mapper Multipath for HP arrays involves editing the multipathing configuration file that is distributed as part of the Device Mapper Multipath tools user space package. It also allows you to customize Device Mapper Multipath for a variety of storage subsystems present in the Storage Area Network (SAN).

For RHEL4 systems:

Following is the sample configuration file:

```
/usr/share/doc/device-mapper-multipath-0.4.5/multipath.conf.annotated
/usr/share/doc/device-mapper-multipath-0.4.5/multipath.conf.synthetic
```

For SLES9 systems:

Following is the sample configuration file:

```
/usr/share/doc/packages/multipath-tools/multipath.conf.annotated
```

/usr/share/doc/packages/multipath-tools/multipath.conf.synthetic

The multipath.conf.annotated file contains a description of each of the parameters mentioned. The multipath.conf.synthetic file does not contain any description of the parameters.

Copy the multipath.conf.annotated file and rename it as /etc/multipath.conf and make the necessary changes to the device parameters for HP supported arrays.

In general, multipath.conf file consists of four sections that provides a variety of options to configure the attributes of a multipath device, such as I/O spreading policy, polling interval for subsequent path status verification, path selector algorithm and the program used by multipath to obtain a unique path identifier.

The /etc/multipath.conf file is divided into the following sections:

- System defaults (defaults)
- Black-listed devices (devnode_blacklist)
- Per storage array model settings (devices)
- Per multipath device settings (multipaths)

The per multipath device settings are used for the multipath device with a matching WWID value. The per storage array model settings are used for all multipath devices with matching vendor and product values. To determine the attributes of a multipath device, the per multipath settings are checked first, the per controller settings are checked second, and system defaults are checked at the last. The blacklisted device section provides various options to black list the devices and hence refrain Device Mapper Multipath from claiming those devices, based on the your requirements. The system defaults settings are applied for any multipath device, which does not fit into the other three sections.

For HP supported arrays, the multipath device attributes are configured using the section devices, which are per storage array model settings that are applied by matching vendor and product values. For each supported array model, a new sub section must be added in devices with the vendor and product values of the same array model. The attributes that are applied to the multipath devices belonging to the same array model are also added subsequently in the same sub section.

Table 4 lists the important attributes applied to HP supported arrays.

Table 4 HP supported array attributes

Attribute	Description	Supported values
path_grouping_policy	Used for path grouping policy to apply to multipath device hosted by this storage controler	• failover: One path per priority group • multibus: All valid paths in one priority group • group_by_serial: One priority group per detected controller serial number • group_by_prio: One priority group per path priority value
path_checker	Used for determining the state of the path	• tur
path_selector	Used to select the path selector algorithm to use for mpath. These algorithms are offered by the kernel mpath target	• round-robin 0
failback	Used to manage the path group failback	• manual • immediate • n > 0 : failback deferred for n seconds
prio_callout	Executable to obtain a path weight for a block device. Weights are summed for each path group to determine the next path group to use in case of path failure	• None • /sbin/mpath_prio_alua %d
no_path_retry (only with RHEL4 U3)	(n > 0) tells the number of retries until disable queuing (queues till n number of polling), or *fail* means immediate failure (no queuing), *queue* means never stop queuing (queue for ever till the path comes alive)	• n (>0) • fail • queue
features (only with SLES9 SP3)	1 queue_if_no_path feature makes the I/O requests to be queued if no path is available for I/O. I/O requests are queued until a path comes alive	• 1 queue_if_no_path • 0

The following example provides the recommended attributes and values for HP supported arrays:

NOTE:

The product string of storage controller contains 16 characters (including spaces) and the vendor string contains 8 characters (including spaces) before editing the /etc/multipath.conf file.

For EVA3000-(HSV101)

```
device {
       vendor               "HP      "
       product              "HSV101 \(C\)COMPAQ"   note: only for RHEL4 U3
       product              "HSV101 (C)COMPAQ"     note: only for SLES9 SP3
       path_grouping_policy  group_by_prio
       getuid_callout        "/sbin/scsi_id -g -u -s /block/%n"
       path_checker       tur
```

```
       path_selector          "round-robin 0"
      prio_callout           "/sbin/mpath_prio_alua %d"
       failback        immediate
       no_path_retry          60                    note: only for RHEL4 U3
}
```

For EVA5000-(HSV111)

```
device {
       vendor               "COMPAQ  "
       product              "HSV111 \(C\)COMPAQ"  note: only for RHEL4 U3
       product              "HSV111 (C)COMPAQ"    note: only for SLES9 SP3
       path_grouping_policy  group_by_prio
       getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
       path_checker       tur
       path_selector          "round-robin 0"
      prio_callout           "/sbin/mpath_prio_alua %d"
       failback        immediate
       no_path_retry          60                   note: only for RHEL4 U3
}
```

For EVA4000/6000

```
device {
       vendor                "HP      "
       product               "HSV200        "  note: only for RHEL4 U3
       path_grouping_policy  group_by_prio
       getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
       path_checker        tur
       path_selector          "round-robin 0"
      prio_callout           "/sbin/mpath_prio_alua %d"
       failback        immediate
       no_path_retry          60                    note: only for RHEL4 U3
}
```

For EVA8000

```
device {
      vendor                "HP      "
      product               "HSV210          "
      path_grouping_policy  group_by_prio
      getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
      path_checker       tur
      path_selector          "round-robin 0"
      prio_callout           "/sbin/mpath_prio_alua %d"
      failback          immediate
      no_path_retry          60                    note: only for RHEL4 U3
}
```

For XP arrays

```
device {
       vendor                 "HP      "
       product                "OPEN-x          " note: only for  SLES9 SP3
     product         "OPEN-*"            note: only for RHEL4 U3
       path_grouping_policy   multibus
       path_checker        tur
       failback         immediate
}
```

NOTE:

In XP arrays, there are different LUNs such as OPEN-<x>, 3390-3A, 3390-3B, OP-C:3390-3C, 3380KA, 3380-KB, OP-C:3380-KC where x = {3,8,9,K,T,E,V}. The product strings for XP LUNs are based on the above emulation types. Since each product string requires a new device sub section, a new device section must be added for each emulation type.

In RHEL4 U3, the regular expression is supported in `/etc/multipath.conf` because product strings OPEN-* is sufficient for all the OPEN emulation.

After editing the `/etc/multipath.conf` file for HP arrays, you must restart Device Mapper Multipath. See Starting Device Mapper Multipath for HP arrays for details.

Starting Device Mapper Multipath for HP arrays

Whenever an attribute for the LUNs of any of the HP supported storage array changes by editing the `multipath.conf` file, the configuration has to be reloaded and the `multipathd` daemon has to be restarted by running the following commands:

```
# /sbin/multipath -v0
# /etc/init.d/multipathd restart
```

These steps are required only if the attributes are changed. You do not manually execute the steps on every reboot because these are also part of boot time scripts.

To view the status of the multipath devices, multipath utility is run with the following options:

```
# /sbin/multipath -ll
```

See Using Device Mapper Multipath for HP arrays for details on status maps in `multipath -ll` for Device Mapper multipath devices.

Using Device Mapper Multipath for HP arrays

The multipath devices are created under `/dev/mapper` directory in RHEL4 and SLES9 hosts. These devices are the same as any other block devices present in the host and they can be used for any block or File level I/O operations, such as creating the file system. For the device during reboots, you can add a new name for any multipath device by adding the multipath sub-sections in the `/etc/multipath.conf` file. This is carried out with the attributes alias and the WWID of the multipath device present in that multipath subsection.

For example, when the following sub-section is added in the `/etc/multipath.conf` file for the LUN with WWID `3600508b30090f5d0d2a9d64590490022`, a multipath device `mydatadisk1` is created under `/dev/mapper`.

```
multipaths {
multipath {
wwid 3600508b30090f5d0d2a9d64590490022
alias mydatadisk1
path_grouping_policy multibus
path_checker readsector0
path_selector "round-robin 0"
}
```

You can use `/dev/mapper/mydatadisk1` like any other block device.

The following section briefly explains the multipath map obtained by executing the command `multipath-ll` based on a sample `multipath.conf`:

The following is a sample `multipath.conf` (for RHEL4 U3) with parameters to support `EVA 4000` as a storage controller:

```
defaults {
multipath_tool             "/sbin/multipath -v0"
udev_dir                   /dev
polling_interval            10
default_selector             "round-robin 0"
default_path_grouping_policy   failover
default_getuid_callout       "/sbin/scsi_id -g -u -s /block/%n"
default_prio_callout         "/bin/true"
default_features             "0"
rr_wmin_io                   100
failback                     immediate
}
multipaths {
multipath {
wwid                         3600508b30090f5d0d2a9d64590490022
path_grouping_policy        multibus
path_checker                 readsector0
path_selector                "round-robin 0"
}
multipath {
…….
…...  for other target
}
multipath {
…….
…... for other target
}
}
devices {

device {
vendor                       "HP      "
product                      "HSV200          "
path_grouping_policy         group_by_prio
getuid_callout        "/sbin/scsi_id -g -u -s /block/%n"
path_checker                 tur
path_selector}                 "round-robin 0"
prio_callout                   "/sbin/mpath_prio_alua %d"
failback                       immediate
no_path_retry                  60
}

device {
…….
…... for targets from other storage controllers
}
}
```

A Linux host with a dual port HBA connected to EVA 4000 via two switches.

For any LUN presented to the host from the EVA 4000, the host will have 4 I/O paths to access the LUN if all the paths are available. The multipath.conf in the previous example will enable the host to see the multipath map as follows:

```
3600508b30090f5d0cf46bd1a4c420023
[size=1 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_round-robin 0 [active]
  \_0:0:0:1 sda 8:0 [ready][active]
  \_1:0:0:1 sdc 8:32[ready][active]
\_round-robin 0 [enabled]
  \_0:0:1:1 sdb 8:16 [ready][active]
  \_1:0:1:1 sdd 8:48 [ready][active]



3600508b30090f5d0d2a9d64590490022
[size=3 GB][features="0"][hwhandler="0"]
```

```
\_round-robin 0 [enabled]
  \_0:0:0:6 sde 8:64 [ready][active]
  \_1:0:1:6 sdf 8:80 [ready][active]
```

The information in the map is presented by grouping the paths for a LUN with unique Identifiers such as, UID/WWN.

The size, features and the corresponding hwhandlers are shown followed by the unique LUN identifier.

The grouping of paths is done and presented in a map based on the I/O spreading policy. In the above example, for the LUN 3600508b30090f5d0d2a9d64590490022, the grouping is done with the policy as multibus and the devices sde, sdf belong to same path group. The state of the group will be either active or enabled depending on whether I/O is happening or not. And for the LUN 3600508b30090f5d0cf46bd1a4c420023, the grouping is done with the policy as group_by_prio and the devices sda, sdc belong to one path group and sdb, sdd belong to different path group as they are from 2 controllers of different serial numbers from the EVA 4000 . The I/O always happens in the path group which is active. In the event of all paths failure in the active group, the fail-over occurs to the other path group which is enabled on changing that path group as active. When the paths are up again and if the failback parameter is set as immediate, the fail-back occurs to the earlier group and I/O will occur through the earlier group. For example if 'group a', has sda,sdc and 'group b', has sdb, sdd the fail-over occurs from 'group a' to 'group b' and the fail-back occurs from 'group b' to 'group a'.

The state of the path is given as [ready] [active] if the path is up and ready for I/O. If the path is down, this state will be shown as [faulty] [failed]. The path states will be updated periodically based on the polling interval as in `/etc/multipath.conf`, and this is taken care of by `multipathd` daemon.

Table 5 lists the basic operations supported by the `multipath` CLI utility provided with Device Mapper Multipath.

Table 5 Basic operations of Device Mapper Multipath

Command	Description
`# multipath -F`	Deletes all device mapper multipath devices.
`# multipath -d`	Displays potential paths, but do not create any device.
`# multipath`	Creates device mapper multipath devices.

`# multipath -l`
`# multipath -ll`
`# multipath -l|-ll <device>`
	Displays the device status.

`# multipath -v2 <device>`	Re-scans path information.
`# multipath -v3`	Displays all device information.

`# multipath -p failover [<device>]`
`# multipath -p multibus [<device>]`
`# multipath -p group_by_serial [<device>]`
`# multipath -p group_by_prio[<device>]`
`# multipath -p group_by_node_name [<device>]`
	Sets group policy.

4 Known issues

Following are the limitations and known issues of Device Mapper Multipath:

- ioctls support is not available on a multipath device created by Device Mapper. Hence scsi inquiry like operations are not possible with Device Mapper Multipath devices
- You have to manually enter the device properties for HP arrays in the `/etc/multipath.conf` file (for both RHEL4 U3 and SLES9 SP3 systems)
- Under all path failure condition, if the queue_if_no_path feature is enabled, the I/O requests are queued for till the system resources are available. This may end in system resource starvation
- Wild card or regular expression support for the product and vendor string (in the `/etc/multipath.conf` file) is only supported in RHEL4 U3
- Time based I/O request queuing (no_path_retry in the `/etc/multipath.conf` file) is only supported in RHEL4 U3
- You have to do the device aliasing manually for the devices presented. See Starting Device Mapper Multipath for HP arrays for details.
- Device Mapper Multipath may not create device nodes for partitions when the configuration has many LUNs
- For a host with many LUNs, "`multipath -ll`" may take longer time to display the status of all the Device Mapper Multipath devices, if there is a change in the path status
- Device Mapper Multipath does not support online LUN deletion. If the LUN is deleted from the host when its online, the Device Mapper Multipath device corresponding to that LUN in the map is displayed as #:#:#:# for h:t:b:l with `multipath-ll` output
- In SLES9 SP3, multipath -v3 exhibits odd behaviour like segmentation fault or irrelevant display information, if the parameters and values present in `/etc/multipath.conf` are not accurate
- Device Reset might occur, but will not impact device availability other than a possible I/O delay and it only occurs if a path failure occurs
- Online LUN addtition currently not supported
- Supports only single path access to root device