Red Hat Enterprise Linux

Online Storage Reconfiguration Guide *(BETA)*

Red Hat Enterprise Linux Documentation

Don Domingo

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

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

PO Box 13588
Research Triangle Park, NC 27709
USA

Abstract

This guide outlines the different procedures involved in reconfiguring iSCSI, Fibre Channel and SAS storage devices.

Beta Document.

This document is currently in Beta. Red Hat welcomes any suggestions and corrections; for more information about submitting feedback, refer to *Section 1.2, “We Need Feedback!”*.

1. Introduction

This manual outlines the different procedures involved in reconfiguring online storage devices on Red Hat Enterprise Linux 5 host systems. Online storage devices typically use any of three protocols: fibre channel, iSCSI, and Serial Attached SCSI (SAS).

The scope of this manual is limited to adding, removing, monitoring and management of online storage devices. This manual does not discuss the fibre channel, iSCSI, or SAS protocols in detail. Refer to other documentation for more information about these protocols.

This manual assumes that you have advanced working knowledge of Red Hat Enterprise Linux

5, along with first-hand experience in managing storage devices in Linux.

Before consulting this book, verify if your host bus adapter vendor or hardware vendor have their own documentation. It is recommended that you consult such documents before reading this manual.

1.1. Document Conventions

Certain words in this manual are represented in different fonts, styles, and weights. This highlighting indicates that the word is part of a specific category. The categories include the following:

`Courier font`

Courier font represents `commands`, `file names and paths`, and `prompts` .

When shown as below, it indicates computer output:

```
Desktop       about.html       logs      paulwesterberg.png
Mail          backupfiles      mail      reports
```

`bold Courier font`

Bold Courier font represents text that you are to type, such as: **`service jonas start`**

If you have to run a command as root, the root prompt (`#`) precedes the command:

```
# gconftool-2
```

`italic Courier font`

Italic Courier font represents a variable, such as an installation directory: *`install_dir`*`/bin/`

bold font

Bold font represents **application programs** and **text found on a graphical interface**.

When shown like this: **OK** , it indicates a button on a graphical application interface.

Additionally, the manual uses different strategies to draw your attention to pieces of information. In order of how critical the information is to you, these items are marked as follows:

Note

A note is typically information that you need to understand the behavior of the system.

Tip

A tip is typically an alternative way of performing a task.

Important

Important information is necessary, but possibly unexpected, such as a configuration change that will not persist after a reboot.

Caution

A caution indicates an act that would violate your support agreement, such as recompiling the kernel.

Warning

A warning indicates potential data loss, as may happen when tuning hardware for maximum performance.

1.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: *http://bugzilla.redhat.com/bugzilla/* against the product **Red_Hat_Enterprise_Linux.**

When submitting a bug report, be sure to mention the manual's identifier: *Online_Storage_Reconfiguration_Guide*

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

2. Fibre Channel

This section discusses the fibre channel API, native Red Hat Enterprise Linux 5 fibre channel drivers, and the fibre channel capabilities of these drivers.

2.1. Fibre Channel API

Below is a list of `/sys/class/` directories that contain files used to provide the userspace API. In each item, host numbers are designated by <*H*>, bus numbers are <*B*>, targets are <*T*>, LUNs are <*L*>, and remote port numbers are <*R*>.

Transport: `/sys/class/fc_transport/target<H>:<B>:<T>/`

- `port_id` — 24-bit port ID/address
- `node_name` — 64-bit node name
- `port_name` — 64-bit port name

Remote Port: `/sys/class/fc_remote_ports/rport-<H>:<B>-<R>/`

- `port_id`
- `node_name`
- `port_name`
- `dev_loss_tmo` — number of seconds to wait before marking a link as "bad". Once a link is marked bad, IO running on its corresponding path (along with any new IO on that path) will be failed.

  The default `dev_loss_tmo` value varies, depending on which driver/device is used. If a Qlogic adapter is used, the default is 35 seconds, while if an Emulex adapter is used, it is 30 seconds. The `dev_loss_tmo` value can be changed via the `scsi_transport_fc` module parameter `dev_loss_tmo`, although the driver can override this timeout value.

  The maximum `dev_loss_tmo` value is 600 seconds. If `dev_loss_tmo` is set to zero or any value greater than 600, the driver's internal timeouts will be used instead.

- `fast_io_fail_tmo` — length of time to wait before failing IO executed when a link problem is detected. IO that reaches the driver will fail. If IO is in a blocked queue, it will not be failed until `dev_loss_tmo` expires and the queue is unblocked.

Host: `/sys/class/fc_host/host<H>/`

- `port_id`
- `issue_lip` — instructs the driver to rediscover remote ports.

2.2. Native Fibre Channel Drivers and Capabilities

Red Hat Enterprise Linux 5 ships with the following native fibre channel drivers:

- `lpfc`
- `qla2xxx`
- `zfcp`
- `mptfc`

Table 1, "Fibre-Channel API Capabilities" describes the different fibre-channel API capabilities of each native Red Hat Enterprise Linux 5 driver. X denotes support for the capability.

	`lpfc`	`qla2xxx`	`zfcp`	`mptfc`
Transport `port_id`	X	X	X	X
Transport `node_name`	X	X	X	X
Transport `port_name`	X	X	X	X
Remote Port `dev_loss_tmo`	X	X	X	X
Remote Port `fast_io_fail_tmo`	X			
Host `port_id`	X	X	X	X
Host `issue_lip`	X	X		

Table 1. Fibre-Channel API Capabilities

3. iSCSI

This section describes the iSCSI API and the `iscsiadm` utility.

3.1. iSCSI API

To get information about running sessions, run:

```
iscsiadm -m session -P 2
```

This command displays the session/device state, session ID (sid) and some negotiated parameters.

For shorter output (for example, to display only the sid-to-node mapping), run:

`iscsiadm -m session -P 0`

or

`iscsiadm -m session`

These commands print the list of running sessions with the format:

```
driver [sid] ip:port,target_portal_group_tag targetname
```

For example:

`iscsiadm -m session`

```
tcp [2] 10.15.84.19:3260,2 iqn.1992-08.com.netapp:sn.33615311
tcp [3] 10.15.85.19:3260,3 iqn.1992-08.com.netapp:sn.33615311
```

For more information about the iSCSI API, refer to `/usr/share/doc/iscsi-initiator-utils-`*`version`*`/`README.

4. Persistent Naming

The `/dev/disk/` directory contains symlinks to different symbolic names that point back to any attached raw device. These symbolic names are useful in determining what names for each device are persistent regardless of which ports or protocols are used.

The symlinks used by these persistent names are divided by type in the following `/dev/disk/` subdirectories:

`by-id/`

Names devices by SCSI VPD page 0x80 or 0x83 data

`by-uuid/`

Names devices based on file system Universally Unique Identifier (UUID)

`by-label/`

Names devices based on file system label

`by-path/`

Names devices by `sysfs` path. For fibre channel this may name the device using the PCI info and Host:BusTarget:LUN info.

For iSCSI devices, `by-path/` names use the target name and portal information. Note that `by-path` names are not reliable if you have two multiple paths to the same portal.

Note

In addition to this, you can also use `udev` rules to implement persistent naming. For more information about this, refer to `man udev`.

5. Scanning for New Devices

If you load a driver before adding the corresponding storage device, you will likely need to manually add the new storage to the operating system. As such, you will need the corresponding *logical unit number* (LUN) of the added storage device.

To scan all buses and targets for new LUNs, use:

```
echo - - - > /sys/class/scsi_host/<host>/scan
```

Here, *<host>* refers to the host number. This can be `host0`, `host1`, `host2`, and so on.

For iSCSI, if the targets sends an iSCSI async event indicating new storage is added, then the scan is done automatically. Cisco MDS™ and EMC Cellera™ support this feature.

For fibre channel, you may need to rediscover the target by executing the command `issue_lip`.

Note that the `proc` interface is deprecated; as such, do not use it.

6. Removing Devices

If remove a device on a target, the driver will normally not remove it automatically. To properly remove a device, perform the following steps:

Procedure 1. Ensuring a Clean Device Removal

1. Close all users of the device.

2. Unmount any file systems that mounted the device.

3. File systems that mounted the device should be unmounted using `umount`. In addition, you should also remove the device from any `dm`, `md`, LVM, multipath or RAID devices using it.

4. Remove the device from the SCSI layer using:

   ```
   echo 1 > /sys/block/<device name>/device/delete
   ```

As stated in the previous subsection, the `proc` interface is deprecated; as such, do not use it.

7. Modifying Link Loss Behavior

This section describes how to modify the link loss behavior of devices that use either fibre channel or iSCSI protocols.

7.1. Fibre Channel

If a driver implements the Transport `dev_loss_tmo` callback, access attempts to a device through a link will be blocked when a transport problem is detected. To verify if a device is blocked, run the following command:

```
cat /sys/block/<device>/device/state
```

This command will return `blocked` if the device is blocked. If the device is operating normally, this command will return `running`.

Procedure 2. Determining The State of a Remote Port

1. To determine the state of a remote port, run the following command:

   ```
   cat /sys/class/rport/rport-H:B:R/port_state
   ```

2. This command will return `Blocked` when the remote port (along with devices accessed through it) are blocked. If the remote port is operating normally, the command will return `Online`.

3. If the problem is not resolved within `dev_loss_tmo` seconds, the rport and devices will be unblocked and all IO running on that device (along with any new IO sent to that device) will be failed.

Procedure 3. Changing `dev_loss_tmo`

- To change the `dev_loss_tmo` value, `echo` in the desired value to the file. For example, to set `dev_loss_tmo` to 30 seconds, run:

  ```
  echo 30 > /sys/class/rport/rport-H:B:R/dev_loss_tmo
  ```

For more information about `dev_loss_tmo`, refer to *Section 2.1, “Fibre Channel API”*.

When a device is blocked, the fibre channel class will leave the device as is; i.e. `/dev/sdx` will remain `/dev/sdx`. This is because the `dev_loss_tmo` expired. If the link problem is fixed at a later time, the SCSI device will be used again.

Fibre Channel: remove_on_dev_loss.

If you prefer that devices are removed at the SCSI layer when links are marked bad (i.e. expired after dev_loss_tmo seconds), you can use the scsi_transport_fc module parameter remove_on_dev_loss. When a device is removed at the SCSI layer while remove_on_dev_loss is in effect, the device will be added back once all transport problems are corrected.

> **Warning**
>
> The use of remove_on_dev_loss is not recommended, as removing a device at the SCSI layer does not automatically unmount any file systems from that device. When file systems from a removed device are left mounted, the device may not be properly removed from multipath or RAID devices.
>
> Further problems may arise from this if the upper layers are not hotplug-aware. This is because the upper layers may still be holding references to the state of the device before it was originally removed. This can cause unexpected behavior when the device is added again.

7.2. iSCSI Settings With dm-multipath

If dm-multipath is implemented, it is advisable to set iSCSI timers to immediately defer commands to the multipath layer. To configure this, nest the following line under device { in /etc/multipath.conf:

```
features           "1 queue_if_no_path"
```

This ensures that I/O errors are retried and queued if all paths are failed in the dm-multipath layer.

You may need to adjust iSCSI timers further to better monitor your SAN for problems. Available iSCSI timers you can configure are *NOP-Out Interval/Timeouts* and replacement_timeout, which are discussed in the following sections.

7.2.1. NOP-Out Interval/Timeout

To help monitor problems the SAN, the iSCSI layer sends a NOP-Out request to each target. If a NOP-Out request times out, the iSCSI layer responds by failing any running commands and instructing the SCSI layer to requeue those commands when possible.

When dm-multipath is being used, the SCSI layer will fail those running commands and defer them to the multipath layer. The multipath layer then retries those commands on another path. If dm-multipath is *not* being used, those commands are retried five times before failing altogether.

Intervals between NOP-Out requests are 10 seconds by default. To adjust this, open `/etc/iscsi/iscsid.conf` and edit the following line:

```
node.conn[0].timeo.noop_out_interval = [interval value]
```

Once set, the iSCSI layer will send a NOP-Out request to each target every *`[interval value]`* seconds.

By default, NOP-Out requests time out in 15 seconds. To adjust this, open `/etc/iscsi/iscsid.conf` and edit the following line:

```
node.conn[0].timeo.noop_out_timeout = [timeout value]
```

This sets the iSCSI layer to timeout a NOP-Out request after *`[timeout value]`* seconds.

SCSI Error Handler.

If the SCSI Error Handler is running, running commands on a path will not be failed immediately when a NOP-Out request times out on that path. Instead, those commands will be failed *after*`replacement_timeout` seconds. For more information about `replacement_timeout`, refer to *Section 7.2.2, “`replacement_timeout`”*.

To verify if the SCSI Error Handler is running, run:

```
iscsiadm -m session -P 3
```

7.2.2. replacement_timeout

`replacement_timeout` controls how long the iSCSI layer should wait for a timed-out path/session to reestablish itself before failing any commands on it. The default `replacement_timeout` value is 120 seconds.

To adjust `replacement_timeout`, open `/etc/iscsi/iscsid.conf` and edit the following line:

```
node.session.timeo.replacement_timeout = [replacement_timeout]
```

The `1 queue_if_no_path` option in `/etc/multipath.conf` sets iSCSI timers to immediately defer commands to the multipath layer (refer to *Section 7.2, “iSCSI Settings With `dm-multipath`”*). This setting prevents I/O errors from propagating to the application; because of this, you can set `replacement_timeout` to 15-20 seconds.

By configuring a lower `replacement_timeout`, I/O is quickly sent to a new path and executed (in the event of a NOP-Out timeout) while the iSCSI layer attempts to re-establish the failed path/session. If all paths time out, then the multipath and device mapper layer will internally queue I/O based on the settings in `/etc/multipath.conf` instead of `/etc/iscsi/iscsid.conf`.

7.3. iSCSI Root

When accessing the root partition directly through a iSCSI disk, the iSCSI timers should be set so that iSCSI layer has several chances to try to reestablish a path/session. In addition, commands should not be quickly requeued to the SCSI layer. This is the opposite of what should be done when `dm-multipath` is implemented.

To start with, NOP-Outs should be disabled. You can do this by setting both NOP-Out interval and timeout to zero. To set this, open `/etc/iscsi/iscsid.conf` and edit as follows:

```
node.conn[0].timeo.noop_out_interval = 0
node.conn[0].timeo.noop_out_timeout = 0
```

In line with this, `replacement_timeout` should be set to a high number. This will instruct the system to wait a long time for a path/session to reestablish itself. To adjust `replacement_timeout`, open `/etc/iscsi/iscsid.conf` and edit the following line:

```
node.session.timeo.replacement_timeout = [replacement_timeout]
```

8. Controlling the SCSI Command Timer and Device Status

The Linux SCSI layer sets a timer on each command. When this timer expires, the SCSI layer will quiesce the *host bus adapter* (HBA) and wait for all outstanding commands to either time out or complete. Afterwards, the SCSI layer will activate the driver's error handler.

When the error handler is triggered, it attempts the following operations in order (until one successfully executes):

1. Abort the command.
2. Reset the device.
3. Reset the bus.
4. Reset the host.

If all of these operations fail, the device will be set to the `offline` state. When this occurs, all IO to that device will be failed, until the problem is corrected and the user sets the device to `running`.

The process is different, however, if a device uses the fibre channel protocol and the `rport` is blocked. In such cases, the drivers wait for several seconds for the `rport` to become online again before activating the error handler. This prevents devices from becoming offline due to temporary transport problems.

Device States.

To display the state of a device, use:

```
cat /sys/block/<device name>/device/state
```

To set a device to `running` state, use:

```
echo running > /sys/block/<device name>/device/state
```

Command Timer.

To control the command timer, you can write to `/sys/block/<device name>/device/timeout`. To do so, run:

```
echo <value> /sys/block/<device name>/device/timeout
```

Here, `<value>` is the timeout value (in seconds) you want to implement.

Alternatively, you can also modify the timeout `udev` rule. To do so, open `/etc/udev/rules.d/50-udev.rules`. You should find the following lines:

```
ACTION=="add", SUBSYSTEM=="scsi" , SYSFS{type}=="0|7|14", \
RUN+="/bin/sh -c 'echo 60 > /sys$$DEVPATH/timeout'"
```

`echo 60` refers to the timeout length, in seconds; in this case, timeout is set at 60 seconds. Replace this value with your desired timeout length.

Note that the default timeout for normal file system commands is 60 seconds when `udev` is being used. If `udev` is not in use, the default timeout is 30 seconds.

9. Troubleshooting

This section provides solution to common problems users experience during online storage reconfiguration.

LUN removal status is not reflected on the host.

> When a LUN is deleted on a configured filer, the change is not reflected on the host. In such cases, `lvm` commands will hang indefinitely when `dm-multipath` is used, as the LUN has now become *stale*.
>
> To work around this, perform the following procedure:

Procedure 4. Working Around Stale LUNs

1. Determine which `mpath` link entries in `/etc/lvm/.cache` are specific to the stale LUN. To do this, run the following command:

   ```
   ls -l /dev/mpath | grep <stale LUN>
   ```

2. For example, if `<stale LUN>` is 3600d0230003414f30000203a7bc41a00, the following results may appear:

```
lrwxrwxrwx 1 root root 7 Aug  2 10:33 /3600d0230003414f30000203a7bc41a00 ->
../dm-4
lrwxrwxrwx 1 root root 7 Aug  2 10:33 /3600d0230003414f30000203a7bc41a00p1
-> ../dm-5
```

   This means that 3600d0230003414f30000203a7bc41a00 is mapped to two `mpath` links: `dm-4` and `dm-5`.

3. Next, open `/etc/lvm/.cache`. Delete all lines containing `<stale LUN>` and the `mpath` links that `<stale LUN>` maps to.

   Using the same example in the previous step, the lines you need to delete are:

```
/dev/dm-4
/dev/dm-5
/dev/mapper/3600d0230003414f30000203a7bc41a00
/dev/mapper/3600d0230003414f30000203a7bc41a00p1
/dev/mpath/3600d0230003414f30000203a7bc41a00
/dev/mpath/3600d0230003414f30000203a7bc41a00p1
```

Index