



MULTIPATH
Setup Guide
for Linux

Version 0.85

MultiPath Task List

Before You Begin (below)

Task 1: Meeting Package Requirements (page 3)

Task 2: Preparing the Configuration File (page 5)

Task 3: Making Initial Host Settings (page 7)

Task 4: Create and Configure Devices (page 9)

Task 5: Create and Mount Partitions (page 10)

Sample multipath.conf File (page 12)

Setting-up ALUA (page 13)

Sample multipath.conf File for ALUA (page 16)

Promise has fully tested VTrak MultiPathing on RedHat RHEL5, and SuSE SLES10 and SLES10 (SP1). Coverage in this *Guide* is limited to those OSes.

MultiPathing is possible on other Linux OSes, in some cases with certain limitations, kernel updates, or different versions of the device mapper or multipath tool. Promise has not tested all of the possible combinations and therefore does not attempt to cover them in this document.

Before You Begin

Before you can set up multipathing on your Linux Host PC, you must:

- Install your Fibre Channel or SAS HBA card into the Host PC.
- Install the HBA card drivers onto the Host PC.
- Set up your VTrak, install your disk drives, and create your logical drives.
- Attach your Fibre Channel or SAS cables from the HBA card to the VTrak.

Refer to the Linux Administration Manual, your HBA documentation, and the *VTrak Quick Start Guide* or *Product Manual* as needed for more information.

Check Initial Setup

To check your initial setup, verify that you can view the logical drives on your VTrak from your Linux desktop or terminal window. Refer to the Linux Administration Manual for the procedure on your system.

- If you can see the logical drives, your system is properly configured. Go to “Task 1: Meeting Package Requirements” on page 3.
 - If you cannot see the logical drives, make the necessary adjustments and check again.
-

Task 1: Meeting Package Requirements

The latest device mapper and multipath packages must be loaded onto your Linux host before configuring Device Mapper Multipath (DM-MP). When this document was written, the current versions were:

- For RHEL5
 - device-mapper-1.02.13-1.el5
 - device-mapper-multipath-0.4.7-8.el5
- For SLES10
 - device-mapper-1.02.03-8.2.rpm
 - multipath-tools-0.4.6-25.8.rpm
- For SLES10 (SP1)
 - device-mapper-1.02.13-6.9
 - multipath-tools-0.4.7-34.18

Installing Packages

The easiest and most effective way to install the device mapper and multipath tool is during OS installation. The **device mapper** installs by default, regardless of the configuration you select. However, you must manually specify the **multipath tool**, as it does not install as a part of any of the configurations of either OS. The multipath tool is listed as an option under **Base System**.

For hosts with the OS already installed, you can add the device mapper and multipath tool, if they are missing.

Example To add the multipath tool for RHEL5, do the following actions:

1. Open a terminal window.
2. Type the following command and press Enter:

```
# rpm -ivh device-mapper-multipath-0.4.7-8.el5.i386.rpm
```

The system returns the following lines:

```
Preparing... ##### [100%]
1:device-mapper-multipath##### [100%]
#
```

(or a similar message)



Important

Where possible, obtain the device mapper and multipath tool from the original installation CDs to ensure full compatibility with your existing OS. Refer to your OS documentation for more information.

Verifying Packages – RedHat

To verify that the required packages are installed on a RHEL5 host, do the following actions:

1. Open a terminal window.
2. Type the following command and press Enter:

```
# rpm -qa | grep device-mapper
```

If the required packages are present, the system will return the following lines:

```
device-mapper-1.02.13-1.e15  
device-mapper-multipath-0.4.7-8.e15
```

Note that the actual version number might be different, depending on your configuration.

Verifying Packages – SuSE

To verify that the required packages are installed on a SLES10 host, do the following actions:

1. Open a terminal window.
2. Type the following command and press Enter:

```
# rpm -qa | grep device-mapper
```

If the required package is present, the system will return the following line:

```
device-mapper-1.02.03-8.2.rpm (SLES10)  
device-mapper-1.02.13-6.9 (SLES10 SP1)
```

Note that the actual version number might be different, depending on your configuration.

3. Type the following command and press Enter:

```
# rpm -qa | grep multipath-tools
```

If the required package is present, the system will return the following line:

```
multipath-tools-0.4.6-25.8.rpm (SLES10)  
multipath-tools-0.4.7-34.18 (SLES10 SP1)
```

Note that the actual version number might be different, depending on your configuration.

Task 2: Preparing the Configuration File

To setup multipathing with VTrak, or any other subsystem, you must provide the required device attributes in a configuration file. The multipath configuration file is named **multipath.conf**. The functional version of the file is saved in the **/etc** directory.

RedHat Systems

For RedHat systems, there is a default **/etc/multipath.conf** file. However, the default file does not have the required device attributes to work with VTrak.

There are also sample configuration files in the **/usr/share/doc/device-mapper-multipath-[version]** directory:

- **multipath.conf.annotated** – multipath device attributes listed and defined
- **multipath.conf.synthetic** – multipath device attributes listed only

SuSE Systems

For SuSE systems, there is no default **/etc/multipath.conf** file.

There are sample multipath configuration files in the **/usr/share/doc/packages/multipath-tools** directory:

- **multipath.conf.annotated** – multipath device attributes listed and defined
- **multipath.conf.synthetic** – multipath device attributes listed only

Editing a Configuration File

You must provide a configuration file with required device attributes to work with VTrak. See the sample configuration file on page 12.

Take the following actions to prepare a configuration file:

1. Choose an existing **multipath.conf** file and open the file in a text editor.
2. Save a working copy of the file under another name.
3. Edit the file to include the following line under **defaults**:

```
defaults {  
    user_friendly_names yes  
}
```

4. Edit the file to include the following lines under **devices**:

```
devices {
device {
    vendor                "Promise"
    product               "VTrak"
    path_grouping_policy multibus
    getuid_callout        "/sbin/scsi_id -g -u -s /block/%n"
    path_checker          readsector0
    path_selector         "round-robin 0"
    hardware_handler      "0"
    failback              immediate
    rr_weight             uniform
    rr_min_io             100
    no_path_retry         20
    features              "1 queue_if_no_path"
    product_blacklist     "VTrak V-LUN"
    }
}
```

5. Edit the file to include the following lines under **devnode_blacklist**:

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

6. Be sure all the relevant lines of your configuration file are uncommented. Remove the **#** character from the beginning of the line.
7. Save the file as **multipath.conf**.
8. Place a copy of the **multipath.conf** file into the Host's **/etc** directory.

Task 3: Making Initial Host Settings

After the packages and configuration file are installed, the Host is ready to accept multipath settings.

Setting the Daemon to Run

To set the MP daemon to run at boot time:

1. Open a terminal window.
2. Set the daemon to run at boot time.

```
# chkconfig multipathd on
```

The system does not return anything.

Verifying the Modules are Loaded

To verify that the DM-MP modules are loaded:

1. Open a terminal window.
2. Verify that the multipath module is loaded.

```
# lsmod | grep dm_multipath
```

If the module is loaded, the system returns:

```
dm_multipath 215770 (or a similar message)
```

3. Verify that the device mapper module is loaded.

```
# lsmod | grep dm_mod
```

If the module is loaded, the system returns:

```
dm_mod 56537 8 dm_snapshot... (or a similar message)
```

Verifying the Daemon is Running

To verify that the MP daemon is running:

1. Open a terminal window.
2. Check the daemon's status.

```
# /etc/init.d/multipathd status
```

3. Do one of the following actions:

- If the system returns:

```
multipathd is running (or a similar message)
```

Go to "Task 4: Create and Configure Devices" on page 9.

- If the system returns:

```
multipathd is stopped (or a similar message)
```

Start the MP daemon.

```
# /etc/init.d/multipathd start
```

Then go to “Task 4: Create and Configure Devices” on page 9.

Task 4: Create and Configure Devices

This step applies the settings from the **multipath.config** file to the Host.

1. Open a terminal window.
2. Type the following command and press Enter:

```
#multipath -v3
```

The system returns:

```
...
==== paths list ====
uuid          hctl    dev dev_t pri dm_st  chk_st  vend...
222490001555459b3b 2:0:0:0 sdb 8:16  1  [undef][undef] Prom...
222b40000155a75b49 2:0:0:2 sbc 8:32  1  [undef][undef] Prom...
20efcff5501000121a 3:0:0:0 sbd 8:48  1  [undef][undef] Prom...
222b44000155ebf0c  3:0:0:1 sde 8:64  1  [undef][undef] Prom...
params = 1 que_if_no_path 0 1 1 round-robin 0 1 1 8:64 100
status = 1 0 0 1 1 A 0 1 0 8:64 A 0
sde: mask = 0x4
sde: path checker = readsector0 (controller setting)
sde: state = 2
```

...

(or a similar message)

3. Restart the MP daemon.

```
#/etc/init.d/multipathd restart
```

The system returns:

```
Stopping multipathd daemon (or a similar message)
```

```
Starting multipathd daemon (or a similar message)
```

For more information about path monitoring functions, type **help** and press Enter.

Task 5: Create and Mount Partitions

This step creates partitions on the logical drives in the VTrak subsystem. Then it creates partitions and a file system to work with the device mapper. Finally, it mounts the partition on the Host.

1. Open a terminal window.
2. Run a multipath list to identify the SCSI devices.

```
# multipath -list
```

The system returns:

```
mpath2(22b4000155ebf01c) dm-2 Promise,VTrak E610s
[size-16.0G][features=1 queue_if_no_path][hwhandler=0]
  \_ round-robin 0 [prio=0][active]
     \_ 3:0:0:1 sde 8:64 [active][undef]
mpath1(222b4000155a75b49) dm-1 Promise,VTrak E610s
[size-12.0G][features=1 queue_if_no_path][hwhandler=0]
  \_ round-robin 0 [prio=0][active]
     \_ 2:0:0:2 sdc 8:32 [active][undef]
mpath0(22249000155459b3b) dm-0 Promise,VTrak E610s
[size-10.0G][features=1 queue_if_no_path][hwhandler=0]
  \_ round-robin 0 [prio=0][active]
     \_ 2:0:0:0 sdb 8:16 [active][undef]
```

Note the multipath and SCSI devices, circled in the example above.

- SCSI devices **sde**, **sdc**, and **sdb** are used in step 3.
 - Multipath devices **mpath0**, **mpath1**, and **mpath2** are used in step 4.
3. To partition each SCSI device using **fdisk**.

```
# fdisk /dev/sdb
```

Enter the appropriate commands in fdisk to build your partition.

Refer to the fdisk online help for more information.

4. Create corresponding partitions on each DM device (under /dev/mapper) using **kpartx**.

```
# kpartx -a /dev/mapper/mpath0
```

The system does not return anything.

Repeat the command for the other multipath devices.

5. List the DM device partitions.

```
# ls /dev/mapper
```

The system returns:

```
mpath0 mpath0p1 mpath1 pmath1p1 mpath2 mpath2p1
```

The number of multipath partitions (mpath0p1, mpath1p1, etc.) in the list depends on:

- The number of LUNs (SCSI devices) in your VTrak subsystem
- The number of partitions you created using fdisk

These partition numbers are used in the next step.

6. Create a file system on the DM device partition.

```
# mkfs /dev/mapper/mpath0p1
```

The system returns:

```
mke3fs 1.38 (30-Jun-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
130560 inodes, 261048 blocks
13052 blocks (5.00%) reserved for super user
First data block=0
8 block groups
32786 blocks per group, 32786 fragments per group
16320 inodes per group
Superblock backups stgored on blocks:
    32768, 98304, 163840, 229736

Writing inode tables: done
Writing superblocks and filesystem accounting informa...
This file system will be automatically checked every ...
180 days, whichever comes first. User tune2fs -c or -...
```

(or a similar message)

7. Mount the partition to a convenient location on the Host.

```
# mount /dev/mapper/mpath1p0 /mnt/temp
```

This command is only an example.

Sample multipath.conf File

Below is a complete **multipath.conf** file for VTrak when you are NOT using ALUA. If you are using ALUA, refer to the sample file on page 16.

If you have no other multipath devices on your Host, you can use this **multipath.conf** file as shown.

If you have other multipath devices, add these settings to your existing **multipath.conf** file.

```
##
## This is a template multipath-tools configuration file
## for the Promise VTrak subsystem
##
defaults {
    user_friendly_names    yes
}
blacklist {
    devnode "^sda$"
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z][[0-9]*]"
    devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}
devices {
    device {
        vendor            "Promise"
        product           "VTrak"
        path_grouping_policy    multibus
        getuid_callout    "/sbin/scsi_id -g -u -s /block/%n"
        path_checker      readsector0
        path_selector     "round-robin 0"
        hardware_handler  "0"
        failback          immediate
        rr_weight         uniform
        rr_min_io         100
        no_path_retry     20
        features          "1 queue_if_no_path"
        product_blacklist "VTrak V-LUN"
    }
}
```

Setting-up ALUA

The following procedures describe how to configure Asymmetric Logical Unit Access (ALUA) as part of your multipathing solution.

Before You Begin

Before you can set up ALUA on your Linux Host PC, you must:

- Perform the multipathing setup procedure beginning on page 2.
- Download the following RPM files from the [Promise Support Website](#).
 - `dm-alua-1.0-1.XXX.rpm`
 - `mpath_prio_promise-1.0-1.XXX.rpm`

XXX refers to platform version, such as `i386`, `i586`, `x86_64`, etc.

There are separate RPMs for 32-bit and 64-bit versions of RedHat and SuSE. Be sure you choose the right file for your system.

Configuring ALUA

To configure ALUA on the Host:

1. Open a terminal window.
2. Type the following command and press Enter:

```
# rpm -ivh dm-alua-1.0-1.i386.rpm
```

The system returns the following lines:

```
Preparing...                ##### [100%]
1:dm-alua                    ##### [100%]
#
```

(or a similar message)

3. Type the following command and press Enter:

```
# rpm -qa | grep dm-alua
```

If the package is present, the system will return the following line:

```
dm-alua-1.0-1
```

4. Type the following command and press Enter:

```
# modprobe dm-alua"
```

This action loads the ALUA device manager. The system does not return anything.

5. Type the following command and press Enter:

```
# lsmod |grep dm_alua
```

If the ALUA device manager has been loaded, the system will return the following line:

```
dm_alua 11776 (or a similar message)
```

6. Type the following command and press Enter:

```
# rpm -ivh mpath_prio_promise-1.0-1.i386.rpm
```

The system returns the following lines:

```
Preparing...                ##### [100%]  
1:mpath_prio_promise        ##### [100%]  
#
```

(or a similar message)

7. Type the following command and press Enter:

```
# ls /sbin/mpath_prio_promise
```

If the package is present, the system will return the following line:

```
/sbin/mpath_prio_promise
```

Editing the *multipath.conf* File

To edit the **multipath.conf** file on the Host:

1. Navigate to the **/etc** directory.
2. Open the existing **multipath.conf** file in a text editor.
3. Save a backup copy of the original file in a different directory.
4. Edit the file to include the following lines under **devices**:

```
devices {  
device {  
    vendor            "Intel"  
    product           "Multi-Flex"  
    path_grouping_policy group_by_prio  
    getuid_callout    "/sbin/scsi_id -g -u -s /block/%n"  
    path_checker       tur  
    path_selector      "round-robin 0"  
    hardware_handler   "1 alua"  
    failback           immediate  
    rr_weight          uniform  
    rr_min_io          100  
    no_path_retry      20  
    features           "1 queue_if_no_path"  
    }  
}
```

5. Save the **multipath.conf** file.
See the sample configuration file on page 16.

Configuring Paths

To configure your data paths on the Host:

1. Open a terminal window.
2. Type the following command and press Enter:

```
# multipath
```

The system does not return anything.

3. Type the following command and press Enter:

```
# multipath -ll
```

The system displays the current multipath topology.

Sample multipath.conf File for ALUA

Below is a complete **multipath.conf** file for VTrak when using ALUA. If you are not using ALUA, refer to the sample file on page 12.

If you have no other multipath devices on your Host, you can use this **multipath.conf** file as shown.

If you have other multipath devices, add these settings to your existing **multipath.conf** file.

```
##
## This is a template multipath-tools configuration file
## for the Promise VTrak subsystem when using ALUA
##
defaults {
    user_friendly_names    yes
}
blacklist {
    devnode "^sda$"
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z][[0-9]*]"
    devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}
devices {
    device {
        vendor                "Intel"
        product               "Multi-Flex"
        path_grouping_policy  group_by_prio
        getuid_callout        "/sbin/scsi_id -g -u -s /block/%n"
        path_checker          tur
        path_selector         "round-robin 0"
        hardware_handler      "1 alua"
        failback              immediate
        rr_weight             uniform
        rr_min_io             100
        no_path_retry         20
        features               "1 queue_if_no_path"
        product_blacklist    "VTrak V-LUN"
    }
}
```