

Configuring udev and device mapper for Oracle RAC 10g Release 2 on RedHat RHEL4 and Oracle EL4

An Oracle White Paper
June 2007

Table of Contents

Introduction.....	3
Udev Overview	3
Udev Configuration	4
Udev Best Practices	7
Device Mapper.....	9
DM overview	9
Device-mapper devices.....	9
DM Configuration.....	11
Configuring DM for Automatic Storage Management.....	13
ASMLIB	13
Appendix A. Steps for Oracle Clusterware Installation on RHEL4 or OEL4 with DM	15
Appendix B. Priority Groups	17
Appendix C. HBA configuration for DM.....	18

Introduction

With the advent of Oracle Database 10g and Linux 2.6, customers have a true low cost computing solution. However, some customers that want high availability solutions have deployed additional products that have added to the overall price-point of their implementation, thus increasing overall total cost of ownership (TCO). One such example is the deployment of 3rd party I/O multi-pathing utilities from storage vendors. Customers now have a choice of low or no cost replacement for these utilities that provides an out of the box and stable solution.

This paper will focus on the configuration of two utilities that provide out-of-the-box I/O multipathing solutions and device naming persistence in the Linux 2.6 environment. Specifically we will illustrate how Automatic Storage Management (ASM) and Oracle Real Application Clusters (RAC) deployments can leverage these utilities in a RedHat (RHEL4) or Oracle Enterprise Linux 4 (OEL4) environments.

The following utilities will be reviewed:

- **udev** – the role of udev is to provide device persistency and naming consistency. This is especially important for the Oracle Cluster Registry (OCR) and Voting disks required by Oracle Clusterware.
- **device-mapper** – Device-mapper will provide I/O multi-path management for all Oracle devices, including ASM disk devices and OCR/Voting disks.
- **ASMLIB** – ASMLIB will provide device management specifically for ASM disk devices.

Udev Overview

In Linux 2.6, a new feature was introduced to simplify device management and hot plug capabilities. This feature is called *udev* and is a standard package in RHEL4 or Oracle Enterprise Linux 4 (OEL4) as well as Novell's SLES9 and SLES10. Udev is a user space utility for dynamic device node creation. A device node is an entry in the /dev directory; e.g., sda1 or sdh1. One main benefit of udev for Oracle environments, is that it provides persistent disk naming; i.e., prevents device renames upon SCSI reconfigurations, node reboots or even storage recabling.

By default, udev is used only to create the device nodes for hot plug devices. Many devices on the system are considered “cold-plug” and will not be handled by default. In order to have udev handle the device node creation for all devices, the udevstart script should be enabled. This will cause all devices on the system to be named by udev. It can also be started manually by calling /sbin/udevstart. Udev uses hot plug events sent by the kernel whenever a device is added or removed from the system. The details about the newly added devices are exported by the kernel to the sysfs filesystem. The sysfs

filesystem, which is a new filesystem in the 2.6 kernel, is managed by the kernel and exports basic information about the devices currently plugged into your system. Udev manages the /dev directory by monitoring the /sys directory.

By leveraging the information in sysfs, udev can determine which devices need to have a /dev entry created and the corresponding device names to be used. This infrastructure of udev and sysfs allows predictable behavior when devices are added or removed from the system.

Udev Configuration

Once udev is notified that a device is added, it uses the `scsi_id` call to retrieve and generate a unique SCSI identifier. The `scsi_id` call queries a SCSI device via the SCSI INQUIRY command and leverages the vital product data (VPD) page 0x80 or 0x83. The output from this query is used to generate a value that is unique across all SCSI devices that properly support pages 0x80 or 0x83¹.

udev uses three basic configuration files that control its functionality².

- /etc/udev/udev.conf - the main udev configuration file
- /etc/udev/permissions.d/50-udev.permissions - udev permission files
- /etc/udev/rules.d/50-udev.rules. – naming rule definitions.

See the udev man page for details on modifying these files.

udev.conf file

The main udev configuration file, /etc/udev/udev.conf, controls the directory locations for the udev permission and rules files, the udev database, and the default location where udev device nodes are created. The udev man page shows a sample udev.conf file.

udev.permissions

Once udev is started and the device nodes are created, udev applies the ownership and permissions of the device node to what is defined in the `*-udev.permissions` files in the `udev.permissions` directory. This is done every time udev is reloaded. This file is essentially used to keep file attributes persistent across reboots. Thus, this file should contain an entry for the disks that will be used for ASM, OCR, and Voting disks. Either specify individual disks or use wildcards.

This file was obsoleted in later versions of the 2.6 kernels such as SLES10, RHEL5 and Oracle Enterprise Linux 5 (OEL5), that functionality was merged into the rules files, which is explained later in this paper.

¹ The devices must support either page x80/x83 to enable this.

² Exact file placement will vary between Linux distribution.

Below is a sample *40-udev.permissions* file:

Figure 1.

```
#name:user:group:mode
# oracle data files
  dm*:oracle:oinstall:660
#OCR disks
  raw1:root:oinstall:640
  raw2:root:oinstall:640
#Vote disks
  raw3:oracle:oinstall:660
  raw4:oracle:oinstall:660
  raw5:oracle:oinstall:660
```

udev.rules

The udev rules file is used to drive the naming scheme for device nodes. Udev passes in a device node and applies the appropriate rule. The output is a meaningful and consistent device name. Every line in the rules files defines how a specific device attribute is mapped to a device file. If all keys that are specified in a rule match the device that was found, the specified device file is created. Below is an example of udev rule mapped to a SCSI device.³

Figure 2.

```
KERNEL="dm-[0-9]*", PROGRAM="/sbin/dmsetup ls --target multipath -
-exec basename -j %M -m %m", RESULT="?* ", NAME="%k",
SYMLINK="mpath/%c"

KERNEL="dm-[0-9]*", PROGRAM="dmsetup info -c --noheadings -j %M -m
%m | grep -q .*:.*:.*:.*:.*:.*:.*:.*:part[0-9]*-mpath-",
PROGRAM="/sbin/dmsetup ls --target line ar --exec basename -j %M -
m %m", RESULT="?* ", NAME="%k", SYMLINK="mpath/%c"
```

This rule indicates that for each SCSI device, we pass the device name to the dmsetup program. The “%k” represents the kernel assigned device name, such as sdc. The “%c” stands for the output string to be generated from the specified program. The program will output the persistent name for that device and create a symbolic link, as indicated in the SYMLINK keyword, that points to the device mapper device. Two device nodes will be created, one referenced in /dev/mpath/ and another in /dev/mapper.

The RHEL4 command /usr/bin/udevtest can be used to simulate the udev execution:

Figure 3.

```
# udevtest /sys/block/dm-8
```

³ This is an example rule, any valid rule can be used in this rules file.

```
creating device node '/dev/dm-8', major = '253', minor = '8'  
mode=060640 uid='0', gid='0'
```

If the disk contains multiple partitions, each partition will have a symbolic link with the partition number added to the name. Persistent naming is achieved by referring to the device by the persistent named symbolic link.

The `/usr/bin/udevinfo -d` command can be used to display all devices defined in the udev database. In addition to displaying the contents of the udev database, the `udevinfo` command can be used to display information about the devices udev has configured. For example the following `udevinfo` command will show all the udevinfo stored in the udev database about a multipathed device called `/dev/dm-8`:

```
udevinfo -q all -n /dev/sdc1
```

See the `udevinfo` man page for more details.

Adding new storage device to a host

There will be cases when new storage disks/LUNs will be added to a host. Therefore, when adding new storage, the HBA driver must be instructed to rescan the SCSI bus to discover the new storage area (LUN)⁴. This is required since the Linux 2.6 kernel is unable to dynamically scan for new devices. However, the exact commands or method may depend on the driver. If the driver supports it, the following command can be used to request a SCSI scan for each HBA port. .

```
echo "-- --" >/sys/class/scsi_host/host{N}/scan
```

where {N} is the HBA port number under scsi_host.

⁴ This is required if you are not rebooting to pickup the new storage.

Udev Best Practices

There may be occasions when Oracle Clusterware fails to start after a reboot due to permissions on raw devices. This is because the permissions on the raw devices for the OCR and Voting Disks e.g. `/dev/raw/raw{x}` revert to their default values (root:disk) and become inaccessible to the Oracle Clusterware. This change of behavior started with the 2.6 kernel. To work around this issue, the following two bullets need to be followed based on the Linux distribution:

For RHEL5, OEL5, SLES10

- Create a file `/etc/udev/rules.d/99-raw.rules` with the following content with the following content

```
KERNEL=="raw[1-2]*", GROUP="oinstall", MODE="640"  
KERNEL=="raw[3-5]*", OWNER="oracle", GROUP="oinstall", MODE="660"
```

For RHEL4, OEL4, SLES9

- Copy the `/etc/udev/permission.d/50-udev.permissions` file into a new permissions file. Choose a number that's lower than 50. After copying `/etc/udev/permission.d/50-udev.permissions` into the new permissions file, remove the lines that are not needed⁵.

The format of the udev permissions file consists of `device:userid:group:permission`. The userid and group must exist, and the permissions must be valid. With Oracle Clusterware 10g and 11g, the requirements are that the OCR is owned by root, and the group is the Oracle user group. The following is a sample entry of a permissions file for the OCR and Voting Disks:

```
#ocr file  
raw/raw[1-2]:root:oinstall:0640  
#votedisk file  
raw/raw[3-5]:oracle:oinstall:0660  
#If using Device Mapper for ASM, then use the dm devices  
#ASM disks  
dm-[1-9]:oracle:oinstall:0640
```

- The device specified in the first column should not include `/dev` prefix; e.g.; they should be `dm-1` or `raw/raw` and not `/dev/dm-1` or `/dev/raw/raw`
- Be cautious of wildcard definitions. For example, If the device names are 15,16,17, then the following delimiters can be used: `raw/raw1[5-7]`. Notice the 1 outside the '['.

⁵ Do not edit the `50-udev.permissions`, as it gets replaced with system upgrades. Additionally, a typo in the `50-udev.permissions` can render the system non-usable.

Note that the intuitive but incorrect use of the wildcard *raw/raw[15-17]* will not work and will not report any errors; i.e., it will silently fail.

- Device permissions are listed as 660 for Oracle 11g. Please consult your documentation for the platform specific permission settings. Also it is recommended that the OCR, the OCRMIRROR and the Voting devices are not on the same set of disks. Oracle Clusterware could be adversely impacted if access to these devices is slow. Place the OCR and OCRMIRROR on different disks and the Voting Disks separately as well.
- Ensure that the extended partition itself is not used to place datafiles or ASM diskgroups. Logical partitions inside the extended partition are OK for file or diskgroup placement. A typical method to avoid such accidental placement is to set the ownership of the extended partition to be root.

Device Mapper

DM overview

An I/O path generally consists of an initiator port, fabric port, target port, and LUN. Each permutation of this I/O path is considered an independent path. Dynamic Multipathing/failover tools aggregate these independent paths into a single logical path. This path virtualization provides I/O load-balancing across the host bus adapters (HBAs), as well as non-disruptive failover on HBA failures.

Multipathing is an important aspect of high availability configurations. Linux 2.6 offers a Path Manger utility called “device mapper” (DM). DM is dependent on the “device-mapper”, “udev”, and “multipath-tools” packages⁶. There are four components of DM, this include:

- **dm-multipath kernel module** – Provides I/O routing and failover to paths and path groups
- **multipath configuration tool** – Includes commands to configure, list, and flush the multipath devices.
- **multipathd daemon** - Monitors paths availability and polls for faulty paths resolution. When paths become enabled, multipathd may also initiate path group switches to ensure that the optimal path group is being used
- **kpartx utility** – This utility plays an integral part of the creation of partitions on DM multipath devices.

Device-mapper devices

DM uses pseudo devices with the “dm-” prefix in the “/dev” directory; e.g; /dev/dm-1. It automatically maps paths to drives, using the drive’s unique world wide id (WWID).

DM generally handles whole drives, if a drive has multiple partitions, the device mapping of each partition is handled by “kpartx”. kpartx is a user-space tool in the “multipath-tool” package. It is automatically invoked whenever device mapper device reconfigurations occur.

For example, suppose a system has one drive with two paths: “sdc” and “sdj”.

The drive has two partitions, “sdc1”/“sdj1” and “sdc2”/“sdj2”. device mapper and kpartx will create the following dm devices:

```
/dev/dm-0: sdc, sdj  
/dev/dm-1: sdc1, sdj1
```

⁶ DM packages are part of device-mapper-1.02.02-3.0.rpm and device-mapper-multipath-0.4.5-12.0.RHEL4.rpm

`/dev/dm-2: sdc2, sdj2`

The “`dmsetup info`” command can be used to display general information about configured DM devices.

DM should be integrated with `udev`. A `udev` rule can be written so that persistent names are obtained from `udev`. Figure 2 shows a sample `udev` rule that can be applied to DM devices.

If the `/etc/multipath.conf` `alias` attribute for a device is specified as the input parameter, then the `/sbin/dmsetup` command will return that value. If the `alias` attribute is not specified, then the script will return the WWID of the device. This has the affect of creating a persistent device name for a particular device. If the device is partitioned, the partitions will appear as `/dev/mapper/<WWID>pX` or `/dev/mapper/<alias>pX` where X is the partition number.

Therefore Oracle Clusterware can leverage `udev` and `device-mapper` for its OCR and Voting Disks by using the `dmsetup` generated paths. It is recommended that the OCR and Voting Disks use `/dev/dm-x` or `/dev/mapper/alias` path format. However, the Oracle Clusterware 10g Oracle Univesall Installer does not support block devices; such as `/dev/mapper/alias` devices, thus a raw device mapping is required to point to the OCR or Voting Disks multipath device. Thus, the Oracle Clusterware will leverage and benefit from the DM multipathing capabilities, even though raw devices are being accessed.

DM Configuration

Use the following commands to initialize and start DM for the first time:

```
# modprobe dm-multipath
# modprobe dm-round-robin
# service multipathd start
# multipath -v 2
```

It is recommended that DM be integrated into the boot sequence using the following commands:

```
# chkconfig --add multipathd
# chkconfig multipathd on
```

The `/sbin/multipath` command invokes DM using the settings defined in the “`/etc/multipath.conf`” file⁷. The `multipath.conf` can be edited to using the following stanza format⁸:

Figure 4.

```
multipaths {
    multipath {
        wwid                36006016064f51000d2e7d1feb052d911
        alias                ocr
        path_grouping_policy failover
        path_checker         readsector0
        path_selector        "round-robin 0"
        failback             immediate
    }
}
```

The “`wwid`” attribute can be determined using the `/sbin/multipath -v2 -d`. To make the devices more easily addressable, it is recommended that the `alias` attribute be added for the device. This is especially important for OCR and Voting Disks devices, so that these devices can be easily identified. The `/etc/multipath.conf` file should contain the entries for all required multipathed devices. Once the `/etc/multipath.conf` file is edited, run command `/sbin/multipath -v2` to allow the multipathed process to pickup the new configuration file, and enable the new multipath device nodes to be created with the following names `/dev/dm-x` and `/dev/mapper/wwn` or `/dev/mapper/alias` (if `alias` is used).

⁷ If the file does not exist, DM uses the default settings.

⁸ Note, that the stanza applies to a device level; i.e., all partitions of a device share the same attributes.

Certain devices should be excluded from the list of devices that DM scans for. This can be done using the blacklist stanza, which needs to be included in the `/etc/multipath.conf` file Figure 5.

```
devnode_blacklist {
    wwid 26353900f02796769
    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]*]"
}
```

DM's `multipath` command can be used to dynamically change policies or policy parameters as well as display multipathed devices. For example, the current multipath device status, including the state, serial number, priority, and node name can be obtained with the `/sbin/multipath -v2 -ll` command.

Configuring DM for Automatic Storage Management

Automatic Storage Management (ASM) is a storage management tool specifically built to simplify the job of the DBA. It provides a simple storage management interface across all servers and storage platforms. ASM provides the DBA flexibility to manage a dynamic database environment with increased efficiency. This feature is a key component of Enterprise Grid Computing and Database Storage Consolidation. A detailed overview of ASM is available on the Oracle Technology Network (OTN) ASM web site:

<http://www.oracle.com/technology/products/database/asm/index.html>

When ASM scans for disks, it will use the `asm_diskstring` parameter from the ASM instance `init.ora`. ASM will discover any devices that it has permissions to open. Upon successful discovery, the `V$ASM_DISK` view on the ASM instance, will reflect which disks were discovered. When using DM based paths with ASM, the `asm_diskstring` must be specify the following:

```
asm_diskstring = '/dev/dm-*
```

If ASMLIB is to be used in conjunction with ASM, then see the next section for disk device paths for ASM

If new disks are to be added to ASM diskgroups, see the section, [Adding new storage device to a host](#), first.

ASMLIB

ASMLIB, which is an add-on to ASM, is a device management utility that is used to exploit the capabilities and strengths of vendor storage arrays. The purpose of ASMLIB, is to provide an alternative interface for ASM to identify and access block devices. This is particularly important in an Oracle RAC 10g environment. For a more detailed overview of ASMLIB please refer the following URL:

<http://www.oracle.com/technology/tech/linux/asmlib/index.html>

Disks to be managed by ASMLIB must have a partition table on them. Thus in the case of DM devices, the partition table must be created on one of the subpaths. For example, if `/dev/sdg` is to be used as an ASMLIB disk, then `fdisk` must be used to create a partition table, producing a `/dev/sdg1` device. Determine the DM device associated with `/dev/sdg1`. Now create the ASMLib disk using the DM device. For example:
/etc/init.d/oracleasm createdisk VOL1 /dev/dm-8

ASMLIB upon startup scans the devices available on the system and discovers ASMLIB labeled disks. ASMLIB needs to be limited to scan only the dm devices instead of any underlying subpaths. This can be done by specifying the following in the ASMLIB configuration file `/etc/sysconfig/oracleasm`:

```
ORACLEASM_SCANORDER="dm"  
ORACLEASM_SCANEXCLUDE="sd"
```

This must be done on each node of a RAC cluster. As stated in the ASM section, ASMLIB will scan for disks, based on the include and exclude rules above and find any devices that it has permissions to open. Upon successful discovery, the `V$ASM_DISK` view on the ASM instance, will reflect which disks were discovered. When ASMLIB and device-mapper based paths are used, the `asm_diskstring` must be specified as the following:

```
asm_diskstring = 'ORCL:*
```

Conclusion

As was demonstrated, the combination of udev and Device Mapper (DM) can provide a free alternative to multi-pathing and persistent device naming in a 2.6 Linux kernel. They fit nicely and can be used in a RAC environment with ASM or ASMLIB.

Appendix A. Steps for Oracle Clusterware Installation on RHEL4 OEL4 with DM

1. Install Red Hat Enterprise Linux 4 (RHEL4) or Oracle Enterprise Linux 4 (OEL4), along with update 3 or later and include the multipath-tools package during the installation.
2. The RHEL4 or OEL4 multipathing tools will not automatically detect newly created partitions when restarting the multipath tools. It is recommended that you create a partition table on all the disks that will be used for Oracle Clusterware and ASM. This can be done using `/sbin/fdisk` on the standard devices before starting the multipath tools. If you create a partition after starting the multipath tools, you will need to restart the multipath tools for the partition to appear with the other multipath devices.
3. Start the multipathing tools (if not started already) by running the following commands as the root user:

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

4. The shared storage devices should now be available as `/dev/mapper/<WWID>` and the partitions as `/dev/mapper/<WWID>pX` where X is the partition number.
5. Identify all the disk paths that will be used by Oracle Clusterware (OCR/Voting) and ASM
6. If you are using raw devices and your raw device layout is the following:

```
OCR          - /dev/raw/raw1
OCR mirror  - /dev/raw/raw2

Voting       - /dev/raw/raw3
              - /dev/raw/raw4
              - /dev/raw/raw5
```

Then the 40-permissions file will be like the following:

```
# Permissions
# OCR
raw/raw[1-2]:root:oinstall:640
# Voting
raw/raw[3-5]:oracle:oinstall:660
```

7. Edit the `/etc/multipath.conf` file to include all multipathed devices. Use the format shown on page 9 and the “`multipath -v2 -d`” command to assist in generating this file.

Add an alias attribute for each device. For example, use `ocrx`, for OCR; `votex` for Voting disks, and `asmx` for ASM disks.

8. Run command `/sbin/multipath -v2`. This will invoke `multipathd` daemon to re-read the `multipath.conf` file and generate the `/dev/mapper/<WWN>` (`/dev/mapper/alias`) and `/dev/dm-x` device nodes.

9. Edit (SuSE: `/etc/raw` RedHat: `/etc/sysconfig/rawdevices`) file so that the proper raw devices are mapped to the proper multipath devices⁹. Note, the raw bindings for the ASM disks are not required, since ASM can use the block devices. This is only required for the OCR and Voting Disks. This should be done consistently across all nodes of the cluster.

```
# /etc/sysconfig/rawdevices
#
raw1:/dev/mapper/ocr1
raw2:/dev/mapper/ocr2
raw3:/dev/mapper/vote1
raw4:/dev/mapper/vote2
raw5:/dev/mapper/vote3
```

Restart the raw service as the root user, to pickup the new bindings:

```
# service rawdevices restart
```

10. Start the Oracle Clusterware installation. When prompted for the OCR and Voting disk names, enter the raw device names, when prompted for ASM devices, enter the device mapper paths.

⁹ Associate the raw mapping to `/dev/mapper` devices and not the `/dev/dm-x` devices. The `/dev/dm-x` device names may not be consistent across nodes.

Appendix B. Priority Groups

Priority groups are the central component of DM. Paths are grouped into an ordered list of Priority Groups. Each Priority Group has a Path Selector which chooses which of the Priority Group's paths is to be used for each I/O request. If an I/O request returns an error, then the path that it uses, gets disabled and an alternative path is tried. If all the paths in a Priority Group fail, then another Priority Group is selected. There are also management commands such as *fail_path* and *reinstate_path*, that manually disable or re-enable paths; respectively. A path tester module, which currently runs in userspace, is responsible for monitoring paths that have failed and reinstating them should they come back.

A drive can have one or more Priority Groups. At any given time; however, only one of the Priority Groups can be active. DM issues all I/O requests to the drive paths in the active priority group.

If a failed path in the active priority group is restored, DM will begin re-issuing I/O requests to it as well as. If a failed path in an inactive priority group is restored, DM will failback to the inactive priority group if and only if the group has a higher priority than the currently active group. Other management commands can be use to switch immediately to a specified Priority Group or to disable a particular Priority Group so it will only be tried after there are no more left.

The grouping policy controls how device mapper groups paths. It can be set as a multipath command option or from the “/etc/multipath.conf” file. DM offers five grouping policies:

- **failover** The failover policy places each path in a separate priority group. All priority groups have equal priority. This policy is good for low-bandwidth databases where multiple paths are for providing availability, not throughput. It is the default policy.
- **multibus** The multibus policy places all paths in one priority group. This policy provides both availability and throughput.
- **group_by_serial** In general, storage array controllers have unique serial numbers. The group_by_serial policy groups paths by serial number, placing paths through the same controller in the same priority group. It can be used to failover between active/passive controllers and to load-balance between paths through the active controller.
- **group_by_prio** The group_by_prio policy places paths with the same priority value into the same priority group. A program is specified in the “/etc/multipath.conf” file that takes a path name as its input and returns the priority for the given path. If no such program is specified, all paths have the same priority.
- **group_by_node_name** The group_by_node_name policy places all paths with the same target node name in the same priority group. The grouping policy is generally storage dependent. For most storage arrays the *multibus* policy is the recommended setting.

Appendix C. QLogic HBA configuration for DM

General Best Practices for deploying DM

The following are some best practices for deploying DM:

- The HBA driver may already be configured with its own failover mode enabled, thus preventing the paths except the first HBA from being discovered. To prevent this follow the procedures for your HBA.

○

For QLogic 2xxx family of HBAs, edit the `/etc/modprobe.conf` file with following:

```
options qla2xxx qlport_down_retry=1 ql2xfailover=0 ql2xretrycount=5
```

Similarly, for Emulex lpfc family of HBAs, edit the `/etc/modprobe.conf` file with following:

```
options lpfc lpfc_nodev_tmo=14 lpfc_lun_queue_depth=30  
lpfc_discovery_threads=1
```

*Further configuration may be needed for each driver change, such as `make_initrd`, please refer to the *Installation and Reference Guide of the respective HBA Drivers* for additional information.*

- Set “HOTPLUG_USE_SUBFS=no” in the “`/etc/sysconfig/hotplug`” file.
- Disable the DM scanning of non-multipathed devices; e.g., CDROM devices or internal/no-SCSI devices, by using the blacklist stanza in the `/etc/multipath.conf` file. This is discussed later.



Configuring udev and device mapper for Oracle RAC 10g Release 2 on RHEL4 or OEL4

June 2007

Author: Nitin Vengurlekar, Saar Maoz

Contributing Authors: Krishnadev Telikicherla, Barb Lundhild

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.