Red Hat Reference Architecture Series

Deploying Oracle Database 12c on Red Hat Enterprise Linux 6

Best Practices

Roger Lopez, Principal Software Engineer

RHCE

Version 1.0

February 2014

redhat.

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

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

Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to provide their thoughts on potential improvements and topic expansion to the papers.

Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

Like us on Facebook:

https://www.facebook.com/rhrefarch

Follow us on Twitter:

https://twitter.com/RedHatRefArch

Plus us on Google+:

https://plus.google.com/u/0/b/114152126783830728030/

Accessing Reference Architectures

There are two ways to access Red Hat reference architectures:

- The Red Hat customer portal at http://access.redhat.com using a customer login
- The Red Hat resource library at http://www.redhat.com/resourcelibrary/reference-architectures/

Table of Contents

1 Executive Summary

IT organizations face challenges of optimizing Oracle database environments to keep up with the ever increasing workload demands and evolving security risks. This reference architecture provides a step-by-step deployment procedure with the latest best practices to install and configure an Oracle Database 12c Release 1 (12.1) with Oracle Automatic Storage Management (ASM). It is suited for system, storage, and database administrators deploying Oracle Database 12c Release 1 (12.1) on Red Hat Enterprise Linux 6. It is intended to provide a Red Hat | Oracle reference architecture that focuses on the following tasks:

- Deploying Oracle Grid Infrastructure 12c Release 1 (12.1.0.1.0)
- Deploying Oracle Database Software 12c Release 1 (12.1.0.1.0)
- Deploying an Oracle Database 12c Release 1 (12.1.0.1.1) with shared SAN disks
- Using Oracle ASM disks with *udev* rules
- Using Oracle ASM with Oracle *ASMLib* (with *SELinux* enabled, requires *SELinux* version 3.7.19-231 available by default in RHEL 6.5)
- Securing the Oracle Database 12c environment with *SELinux*

2 Reference Architecture Environment

This section focuses on the components used during the deployment of Oracle Database 12c with Oracle Automatic Storage Management (ASM) on Red Hat Enterprise Linux 6 x86_64 in this reference architecture.

2.1 Reference Architecture Overview

A pictorial representation of the environment used in this reference environment is shown in **Figure 1: Reference Architecture Overview**.

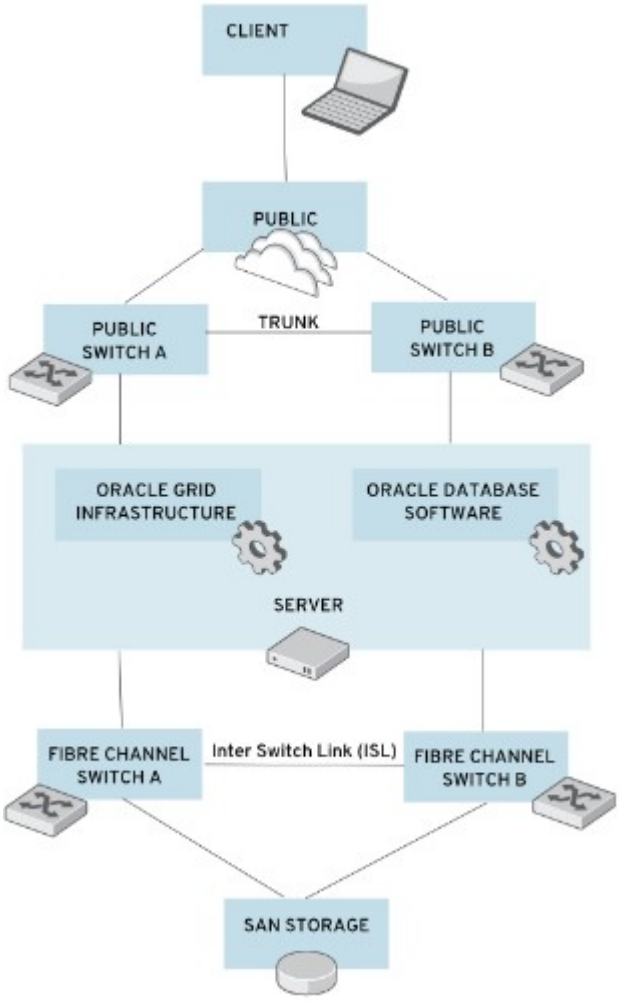

Figure 1: Reference Architecture Overview

2.2 Network Topology

The network topology used in this reference environment consists of two public switches with a link aggregation that connect the two switches together (*Public Switch A* and *Public Switch B*), creating a single logical switch. Ethernet device *em1* on the server connects to *Public Switch A*, while Ethernet device *em2* on the server connects to *Public Switch B*. Ethernet devices *em1* and *em2* are bonded together as a bond device, *bond0*, providing high availability for the network traffic. **Figure 2.2.1: Network Topology** shows the pictorial representation of the two public switches connecting to the server, while **Figure 2.2.2: Ethernet Bonding** shows the bonding of Ethernet device *em1* and *em2* as part of the *bond0* device.

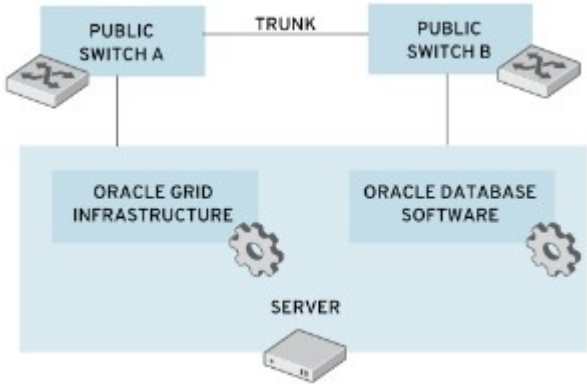

Figure 2.2.1: Network Topology

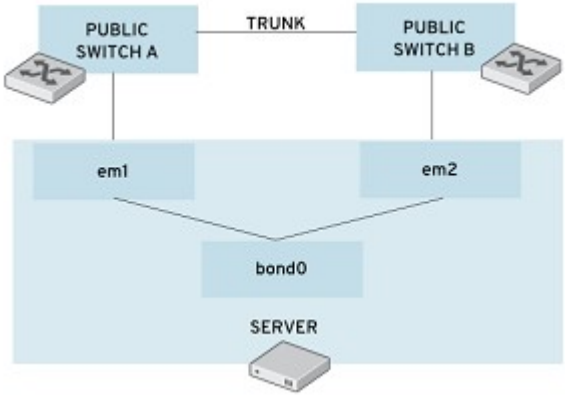

Figure 2.2.2: Ethernet Bonding

2.3 Hardware Details

The following are the minimum hardware requirements to properly install Oracle Database 12c on a x86_64 system:

- Minimum of 1 GB of RAM for the installation of both Oracle Grid Infrastructure and Oracle Database, however 2 GB of memory or more is recommended
- Minimum of 1 Network Interface Card (NIC), however 2 NICs are recommended for high availability (HA) as used in the reference environment
- Red Hat Enterprise Linux 6.x Server x86_64 with kernel 2.6.32-71 or higher
- Console access that supports 1024 x 768 resolution to ensure correct display of Oracle's Universal Installer (OUI).

Table 2.3.1: Server Details specifies the hardware for the server used within this reference environment. This hardware meets the minimum requirements for properly installing Oracle Database 12c Release 1 (12.1) on a x86_64 system.

Server Hardware	Specifications
Oracle 12c Release 1 Standalone Server (*db-oracle-node1*) [1 x HP ProLiant DL370 G6 Server]	Red Hat Enterprise Linux 6.5 kernel 2.6.32-431.el6.x86_64
	2 Socket, 8 Core (16 cores) Intel(R) Xeon(R) CPU W5580 @ 3.20 GHz
	48 GB of memory, DDR3 4096 MB @ 1333 MHz DIMMs
	2 x NetXen NX3031 1/10-Gigabit Network Interface Cards (NICs) for public network
	1 x Qlogic ISP2532 8GB Fibre Channel Dual Port HBA

Table 2.3.1: Server Details

Table 2.3.2: Switch Details specifies the Fibre Channel and Ethernet switches used within this reference environment.

Switch Hardware
2 x Brocade Silkworm Fibre Switches
2 x HP ProCurve Gigabit Switches

Table 2.3.2: Switch Details

Table 2.3.3: Storage Details specifies the storage used for storing Oracle data files within this reference environment.

Storage Hardware	Specification
HP StorageWorks MSA2324fc Dual Controller Array	24 x 146 GB 15K SAS Hard disks

Table 2.3.3: Storage Details

2.4 File System Layout & Disk Space Details

The following are the minimum disk space requirements for properly installing Oracle Database 12c Release 1 (12.1) software for this reference environment.

Software	Disk Space
Oracle Grid Infrastructure Home (includes software files)	9.7 GB
Oracle Database Home Enterprise Edition (includes software files and data files)	9.8 GB
/tmp	1 GB

Table 2.4.1:Disk Space Requirements

NOTE: The actual amount of disk space consumed for Oracle Grid Infrastructure Home and Oracle Database Home Enterprise Edition may vary.

Table 2.4.2: File System Layout specifies the file system layout for the server used in this reference environment. The layout ensures the disk space requirements to properly install the Oracle Grid Infrastructure and Oracle Database software for Oracle Database 12c Release 1 (12.1).

File System Layout	Disk Space Size
/	15 GB
/dev/shm	24 GB
/boot	248 MB
/home	8 GB
/tmp	4 GB
/u01	50 GB
/usr	5 GB
/var	8 GB

Table 2.4.2: File System Layout

While the size of the Oracle data files varies for each solution, the following are the Oracle data file sizes used for this reference environment.

Volume	Volume Size
Oracle Database Volume 1 (db1)	100 GB
Oracle Database Volume 2 (db2)	100 GB
Fast Recovery Area (fra)	200 GB
Oracle Redo Log Volume (redo)	10 GB

Table 2.4.3: Oracle Data File Sizes for Reference Architecture

2.5 Storage Layout

Table 2.5.1: Storage Disk Layout for Reference Architecture shows the storage disk layout for each volume.

Virtual Diskgroup Name	Volume Name	Volume Size	RAID Group Type	Hard Drive Count	Hot Spares Available	Size of Virtual Disk
vd01	db1	100 GB	Raid 10	8	1	586 GB
vd02	db2	100 GB	Raid 10	8	1	586 GB
vd03	fra	200 GB	Raid 5	5	0	586 GB
vd04	redo	10 GB	Raid 1	2	0	146 GB

Table 2.5.1: Storage Disk Layout for Reference Architecture

2.6 Swap Space

Swap space is determined by the amount of RAM found within the system. The following table displays the swap space recommendation. This reference environment allocates 16 GB of RAM for swap space.

RAM	Swap Space
2 GB up to 16 GB	Equal to the size of RAM
Greater than 16 GB	16 GB of RAM

Table 2.6.1: Recommended Swap Space

NOTE: When calculating swap space, ensure not to include RAM assigned for *hugepages*. More information on *hugepages* can be found in **Section 4.1.5 Enabling HugePages**

2.7 Security: Firewall Settings

This section focuses on providing the details required to run **`iptables`** successfully for an Oracle Database environment. **Table 2.7.1: Firewall Settings** lists the enabled ports in this reference environment.

Port	Protocol	Description
22	TCP	Secure Shell (SSH)
443	TCP	Hypertext Transfer Protocol over SSL/TLS (HTTPS)
1521	TCP	Oracle Transparent Network Substrate (TNS) Listener default port
5500	TCP	EM Express 12c default port

Table 2.7.1: Firewall Settings

2.8 Security: SELinux

Starting with Oracle 11g Release 2 version 11.2.0.3, *SELinux* is supported for Oracle database environments. The system in this reference environment runs with *SELinux* enabled and set to *ENFORCING* mode. **Table 2.8.1: SELinux Packages** lists the required *SELinux* packages. In order to take advantage of *SELinux* enablement with Oracle *ASMLib*, *SELinux* version 3.7.19-211 or higher is required. With the release of Red Hat Enterprise Linux 6.5, this requirement is met and no further action is required.

Package	Version
selinux-policy	3.7.19-231
selinux-policy-targeted	3.7.19-231

Table 2.8.1: SELinux Packages

3 Reference Architecture Configuration Details

This reference architecture focuses on the deployment of Oracle Database 12c Release 1 (12.1) with Oracle Automatic Storage Management (ASM) on Red Hat Enterprise Linux 6 x86_64. The configuration is intended to provide a comprehensive Red Hat | Oracle solution. The key solution components covered within this reference architecture consists of:

- Red Hat Enterprise Linux 6 Update 5
- Oracle Grid Infrastructure 12c Release 1 (12.1.0.1.0)
- Oracle Database 12c Release 1 Software Installation (12.1.0.1.0)
- Deploying an Oracle Database 12c Release 1 (12.1.0.1.0) with shared SAN disks
- Enabling Security-Enhanced Linux (*SELinux*)
- Configuring Device Mapper Multipathing
- Using *udev* Rules
- Using Oracle ASMLib (with *SELinux* enabled, requires *SELinux* version 3.7.19-231 available by default within RHEL 6.5)

3.1 Setting OS Hostname

A unique host name is required for the installation of Oracle Database 12c Release 1 (12.1) The host name within this reference environment is: *db-oracle-node1.* To set the host name, please follow the instructions below.

Set the host name via the use of the **`hostname`** command. An example of setting *db-oracle-node1* host name is shown below.

```
# hostname db-oracle-node1.cloud.lab.eng.bos.redhat.com
```

Edit the */etc/sysconfig/network* file's host name variable with the host name provided above. An example of the */etc/sysconfig/network* file on *db-oracle-node1* is shown below.

```
# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=db-oracle-node1.cloud.lab.eng.bos.redhat.com
NOZEROCONF=yes
```

3.2 Network Configuration

The network configuration focuses on the proper creation of a bonded network interface. It ensures the *NetworkManager* is disabled, and configures the */etc/resolv.conf* file. The bonded network interface provides an Oracle environment with high availability in case of a public network interface failure.

3.2.1 Configuring */etc/resolv.conf* file

The resolver is a set of routines in the C library that provides access to the Internet Domain Name System (DNS). The resolver configuration file contains information that is read by the resolver routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.[1] The */etc/resolv.conf* file for this reference environment consists of two configuration options: *nameserver* and *search*. The search option is used to search for a host name that is part of a particular domain. The *nameserver* option is the IP address of the name server the system (*db-oracle-node1*) must query. If more than one *nameserver* is listed, the resolver library queries them in order. An example of the */etc/resolv.conf* file used on the reference environment is shown below.

```
cat /etc/resolv.conf
search cloud.lab.eng.bos.redhat.com
nameserver 10.16.143.247
nameserver 10.16.143.248
nameserver 10.16.255.2
```

3.2.2 Public Network Configuration

The public network configuration consists of two network interfaces bonded together to provide high availability. The example below shows how to bond physical interfaces *em1* and *em2* with a bond device labeled *bond0*. If *NetworkManager* is installed, ensure it is *disabled*.

Check the status of *NetworkManager*:

```
# chkconfig –-list | grep NetworkManager
NetworkManager   0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

Disable *NetworkManager*:

```
# service NetworkManager stop
# chkconfig NetworkManager off
```

As the `root` user, execute the following command which creates a file named *bonding.conf* within the */etc/modprobe.d/* directory needed to create a bonded device for multiple network interfaces. The *bonding.conf* file is also part of **Appendix K Configuration Files**

```
# echo "alias bond0 bonding" > /etc/modprobe.d/bonding.conf
```

1 Linux man pages – man resolv.conf

As the `root` user, create a backup of the *ifcfg-em1 & ifcfg-em2* files, create the *ifcfg-bond0* file and edit the *ifcfg-em1 & ifcfg-em2* configuration files found within */etc/sysconfig/network-scripts.* An example can be seen below.

```
# cp /etc/sysconfig/network-scripts/ifcfg-em1 /etc/sysconfig/network-
scripts/ifcfg-em1.bkup
# cp /etc/sysconfig/network-scripts/ifcfg-em2 /etc/sysconfig/network-
scripts/ifcfg-em2.bkup
```

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0

DEVICE="bond0"
BONDING_OPTS="mode=1 miimon=100 primary=em1"
NM_CONTROLLED="no"
IPADDR="10.16.142.51"
NETMASK="255.255.248.0"
GATEWAY="10.16.143.254"
ONBOOT="yes"
```

```
# cat /etc/sysconfig/network-scripts/ifcfg-em1

DEVICE="em1"
BOOTPROTO="none"
HWADDR="00:25:B3:A8:6F:18"
IPV6INIT="no"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
UUID="3db45d28-e63c-401b-906a-ef095de4fc1e"
SLAVE="yes"
MASTER="bond0"
```

```
# cat /etc/sysconfig/network-scripts/ifcfg-em2

DEVICE="em2"
BOOTPROTO="none"
HWADDR="00:25:B3:A8:6F:19"
IPV6INIT="no”
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
UUID="7d29d87f-52bb-4dc6-88ca-d0857c7d7fd9"
SLAVE="yes"
MASTER="bond0"
```

After all the network scripts are configured, restart the network service via the command:

```
# service network restart
Shutting down interface bond0:                             [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface bond0:                               [  OK  ]
```

3.2.3 NTP Configuration

The **ntpd** program is an operating system daemon which sets and maintains the system time, synchronizing with Internet standard time servers[2]. The **ntpd** program operates by exchanging messages with one or more configured servers at designated poll intervals.

To configure the **ntpd** daemon, follow the instructions below.

1. Edit the */etc/ntp.conf* file with a text editor such as **vi**.

```
# vi /etc/ntp.conf
```

2. Locate the following public server pool section, and modify to include the appropriate NTP servers. For the purposes of this reference environment, only one NTP server is used, but three is recommended. The **iburst** option is added to speed up the time in which it takes to properly sync with the NTP servers.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 10.16.255.2 iburst
```

3. Save all the changes within the */etc/ntp.conf* file
4. Restart the ntpd daemon via the command:

```
# service ntpd restart
Shutting down ntpd:                                        [FAILED]
Starting ntpd:                                             [  OK  ]
```

NOTE: Shutting down **ntpd** daemon provides a status of 'FAILED' if the **ntpd** daemon is currently off.

5. Ensure that the ntpd daemon is started when the system is booted.

```
# chkconfig ntpd on
```

2 ntpd – Network Time Protocol (NTP) daemon man page – man ntpd (8)

3.3 OS Configuration

3.3.1 Using the Red Hat Subscription Manager (Option A)

The **subscription-manager** command is used to register systems to the Red Hat Network (RHN) and to manage the subscription entitlements for the systems. The **--help** option can be specified on the command line to query the command for the available options. If the **--help** option is issued along with a command directive, then options available for the specific command directive are listed.

To use Red Hat Subscription Management for providing packages to a system, the system must first register with the service. To register a system, use the **subscription-manager** command and pass the register command directive. If the **--username** and **--password** options are specified, then the command will not prompt for the RHN Network authentication credentials.

An example of registering a system using **subscription-manager** is shown below.

```
# subscription-manager register --username [User] --password '[Password]'
The system has been registered with id: abcd1234-ab12-ab12-ab12-481ba8187f60
```

After a system is registered, it must be attached to an entitlement pool. For the purposes of this reference environment, the **Red Hat Enterprise Linux Server** is the pool chosen. To identify and subscribe to the **Red Hat Enterprise Linux Server** entitlement pool, the following command directives are required.

```
# subscription-manager list --available | grep -A8 "Red Hat Enterprise Linux
Server"
Subscription Name: Red Hat Enterprise Linux Server, Standard (8 sockets)
(Unlimited guests)
Provides:          Red Hat Beta
                   Red Hat Enterprise Linux 7 Public Beta
                   Red Hat Enterprise Linux Server
                   Red Hat Software Collections Beta (for RHEL Server)
SKU:               RH0186633
Pool ID:           8a85f984411b738d01415681d1ec393f
Available:         199
Suggested:         1
Service Level:     STANDARD
Service Type:      L1-L3
Multi-Entitlement: No
```

```
# subscription-manager attach --pool 8a85f984411b738d01415681d1ec393f
Successfully attached a subscription for: Red Hat Enterprise Linux Server,
Standard (8 sockets) (Unlimited guests)
```

The Red Hat Enterprise Linux supplementary repository is part of subscribing to the Red Hat Enterprise Linux Server entitlement pool, however, it is disabled by default. To enable the supplementary repository, use the **yum-config-manager** command.

NOTE: The following step is optional and only required when implementing the use of Oracle *ASMLib* as shown in **Section 3.4.3.2 Configuring Oracle ASMLib.** Implementing Oracle *ASMLib* is not required in order to use Oracle ASM disks. This reference architecture features two methods of implementation **Section 3.4.3.1 Oracle ASMLib Alternative: Configuring udev Rules** and **Section 3.4.3.2 Configuring Oracle ASMLib**

```
# yum-config-manager --enable rhel-6-server-supplementary-rpms
Loaded plugins: product-id
================================================================ repo: rhel-6-
server-supplementary-rpms
================================================================
[rhel-6-server-supplementary-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/dist/rhel/server/6/6Server/x86_64/supplementa
ry/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-6-server-supplementary-rpms
cost = 1000
enabled = 1

[ ... Output Abbreviated ... ]
```

For more information on the use of Red Hat Subscription Manager, please visit the Red Hat Enterprise Linux 6 documentation[3].

3.3.2 Accessing the RHN Repository (Option B)

Instructions on how to register a system and manage subscriptions on Red Hat Enterprise Linux 6 can be found within the Red Hat Linux 6 Deployment Guide documentation[4].

NOTE: If **Section 3.3.1 Using the Red Hat Subscription Manager (Option A)** is setup, please skip this section.

3 Red Hat Enterprise Linux Subscription Management - https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/registering-cmd.html

4 Red Hat Enterprise Linux 6 Deployment Guide, https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html-single/Deployment_Guide/index.html#entitlements

The following table shows the required channels via the Red Hat Network to support the installation of Oracle.

Channel	Source
rhel-x86_64-server-6	RHN Classic
rhel-x86_64-server-supplementary-6 (optional if not configuring ASMLib)	RHN Classic

Table 3.3.2.1: Required Channels

NOTE: The rhel-x86_64-server-supplementary-6 channel is a requirement for the implementation of Oracle *ASMLib* found in **Section 3.4.3.2 Configuring Oracle ASMLib**. However, Oracle *ASMLib* is not required for the use of Oracle ASM disks. This reference architecture features two methods of implementation **Section 3.4.3.1 Oracle ASMLib Alternative: Configuring udev Rules** and **Section 3.4.3.2 Configuring Oracle ASMLib**

3.3.3 Oracle Database 12c Release 1 (12.1) Package Requirements

A specific set of packages is required to properly deploy Oracle Database 12c Release 1 (12.1) on Red Hat Enterprise Linux 6 (x86_64). The number of installed packages required varies depending on whether a default or minimal installation of Red Hat Enterprise Linux 6 (x86_64) is performed. For the purposes of this reference environment, a minimal Red Hat Enterprise Linux 6 installation is performed to reduce the number of installed packages. A sample kickstart file as been provided within **Appendix K Configuration Files**. Red Hat Enterprise Linux 6 installation requires the following group packages:

Required Group Packages
@Base
@Core

Table 3.3.3.1: Group Packages

Oracle Grid Infrastructure 12.1 and Oracle Database 12.1 required x86_64 RPM packages[5]:

Required Packages	
cloog-ppl	libXxf86misc
compat-libcap1	libXxf86vm
compat-libstdc++-33	libaio-devel
cpp	libdmx
gcc	libstdc++-devel
gcc-c++	mpfr
glibc-devel	make
glibc-headers	ppl
kernel-headers	xorg-x11-utils
libXmu	xorg-x11-xauth
libXt	libXv
ksh	libXxf86dga

Table 3.3.3.2: Required Packages

After the installation of Red Hat Enterprise Linux 6 is completed, create a file, *req-rpm.txt*, that contains the name of each RPM package listed above on a separate line. For simplicity, this *req-rpm.txt* file is included in **Appendix F Oracle Database Package Requirements Text File.**

Use the **yum** package manager to install the packages and any of their dependencies with the following command:

```
# yum install `awk '{print $1}' ./req-rpm.txt`
```

A minimum installation of Red Hat Enterprise Linux 6 does not install the *X Window System* server package, but only the required *X11* client libraries. In order to run the Oracle Universal Installer (OUI), a system with the *X Window System* server package installed is required. Using a system with *X Window System* installed, **SSH** into the Oracle Database server with the *-Y* option to ensure trusted *X11* forwarding is set. The command is as follows:

```
# ssh -Y db-oracle-node1
```

Alternatively, if a system with the X Window System server package is unavailable, install the *X Window System* server package directly on the Oracle Database Server.

```
# yum groupinstall "X Window System"
```

5 Linux OS Installation with Reduced Set of Packages for Running Oracle Database Server [ID 728346.1] via http://support.oracle.com

3.3.4 Configuring Security-Enhanced Linux (SELinux)

SELinux is an implementation of a mandatory access control (MAC) mechanism developed by the National Security Agency (NSA). The purpose of *SELinux* is to apply rules on files and processes based on defined policies. When policies are appropriately defined, a system running *SELinux* enhances application security by determining if an action from a particular process should be granted thus protecting against vulnerabilities within a system. The implementation of Red Hat Enterprise Linux 6 enables *SELinux* by default and appropriately sets it to the default setting of *ENFORCING*. It is highly recommended that *SELinux* be kept in *ENFORCING* mode when running Oracle Database 12c Release 1 (12.1).

Verify that *SELinux* is running and set to *ENFORCING*:

As the `root` user,

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

If the system is running in *PERMISSIVE* or *DISABLED* mode, modify the */etc/selinux/config* file and set *SELinux* to enforcing as shown below.

```
SELINUX=enforcing
```

The modification of the */etc/selinux/config* file takes effect after a reboot. To change the setting of *SELinux* immediately without a reboot, run the following command:

```
# setenforce 1
```

For more information on Security-Enhanced Linux, please visit the Red Hat Enterprise Linux 6 Security-Enhanced Linux User Guide

3.3.5 Configuring Firewall Settings

Firewall access and restrictions play a critical role in securing your Oracle Database 12c Release 1 (12.1) environment. It is recommended that the firewall settings be configured to permit access to the Oracle Database network ports only from authorized database or database-management clients. For example, in order to allow access to a specific database client with an IP address of 10.16.142.54 and to make requests to the database server via SQL*Net using Oracle's TNS (Transparent Network Substrate) Listener (default port of 1521), the following firewall rule must be added to the **`iptables`** configuration file found at */etc/sysconfig/iptables*.

```
-A INPUT -m state --state NEW -m tcp -p tcp -s 10.16.142.54 --dport 1521 -j
ACCEPT
```

Likewise, if a particular database client with an IP address of 10.16.142.54 required access to the web-based EM Express that uses the default port of 5500, the following firewall rule must be added to the *iptables* configuration file found at */etc/sysconfig/iptables*.

```
-A INPUT -m state --state NEW -m tcp -p tcp -s 10.16.142.54 --dport 5500 -j
ACCEPT
```

Once the rules have been modified within the */etc/sysconfig/iptables*, run the following command to activate:

```
# service iptables restart
iptables: Setting chains to policy ACCEPT: nat mangle filte[  OK  ]
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Applying firewall rules:                         [  OK  ]
```

NOTE: A full listing of all the firewall settings within the */etc/sysconfig/iptables* file for this reference architecture can be found at **Appendix D Iptables Configuration File.**

3.3.6 Setting Virtual Memory

Tuning virtual memory requires the modification of five kernel parameters that affect the rate at which virtual memory is used within Oracle databases. **It is important to note the recommended values listed are to be used as a starting point when setting virtual memory.**

A brief description[6] and recommended settings for the virtual memory parameters, as well as, the definition of dirty data are described below.

SWAPPINESS[6] - A value from 0 to 100 which controls the degree to which the system swaps. A high value prioritizes system performance, aggressively swapping processes out of physical memory when they are not active. A low value prioritizes interactivity and avoids swapping processes out of physical memory for as long as possible, which decreases response latency. The default value is 60. The Oracle recommended value is 0.

DIRTY DATA – Dirty data is data that has been modified and held in the page cache for performance benefits. Once the data is flushed to disk, the data is clean.

DIRTY_RATIO[6] – Contains, as a percentage of total system memory, the number of pages at which a process which is generating disk writes will itself start writing out dirty data. The default value is 20. The recommended value is 80. The reasoning behind increasing the value from the standard Oracle 15 recommendation to 80 is because dirty ratio defines the maximum percentage of total memory that be can be filled with dirty pages before user processes are forced to write dirty buffers themselves during their time slice instead of being allowed to do more writes. All processes are blocked for writes when this occurs, not just the processes that filled the write buffers. This can cause what is perceived as unfair behavior where a single process can hog all the I/O on a system. As for all parameters in this reference architecture, there is no “one-size fits all” value and the recommendation should be only seen as a starting point.

DIRTY_BACKGROUND_RATIO[6] – Contains, as a percentage of total system memory, the number of pages at which the background write back daemon will start writing out dirty data. The Oracle recommended value is 3.

DIRTY_EXPIRE_CENTISECS[6] - Defines when dirty in-memory data is old enough to be eligible for writeout. The default value is 3000, expressed in hundredths of a second. The Oracle recommended value is 500.

DIRTY_WRITEBACK_CENTISECS[6] - Defines the interval of when writes of dirty in-memory data are written out to disk. The default value is 500, expressed in hundredths of a second. The Oracle recommended value is 100.

Prior to making any changes to the */etc/sysctl.conf* ensure to create a backup as follows:

```
# cp /etc/sysctl.conf /etc/sysctl.conf.bkup
```

6 RHEL6 Kernel Documentation - /usr/share/doc/kernel-doc-2.6.32/Documentation/sysctl/vm.txt

The following is a snippet from the */etc/sysctl.conf* file with the five virtual memory parameters set with the recommended settings:

```
vm.swappiness = 0
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
```

For the changes to take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.7 Setting Shared Memory (SHMMAX, SHMALL, SHMMNI)

Shared memory allows processes to communicate with each other by placing regions of memory into memory segments. In the case of Oracle, shared memory segments are used by the System Global Area (SGA) to store incoming data and control information. The size of Oracle's SGA impacts the amount of shared memory pages and shared memory segments to be set within a system. By default, Red Hat Enterprise Linux 6 provides a large amount of shared memory pages and segments. However, the appropriate allocation for a system depends on the size of the SGA within an Oracle database instance.

In order to allocate the appropriate amount of shared memory pages and shared memory segments for a system running an Oracle database, the kernel parameters *SHMMAX*, *SHMALL*, and *SHMMNI* must be set.

SHMMAX – is the maximum size in bytes of a single shared memory segment

SHMALL – is the maximum total amount of shared memory pages

SHMMNI – is the maximum total amount of shared memory segments

A default installation of Red Hat Enterprise Linux 6 x86_64 provides a maximum size of a single shared memory segment, *SHMMAX*, to 68719476736 bytes, equivalent to 64 GB. This value is important since it regulates the largest possible size of one single Oracle SGA shared memory segment. If the Oracle SGA is larger than the value specified by *SHMMAX* (default 64 GB), then Oracle will be required to create multiple smaller shared memory segments to completely fit Oracle's SGA. This can cause a significant performance penalty, especially in NUMA environments. In an optimal NUMA configuration, a single shared memory segment for Oracle's SGA is created on each NUMA node. If *SHMMAX* is not properly sized and creates multiple shared memory segments, *SHMMAX* limitations may keep the system from evenly distributing the shared memory segments across each NUMA node. To determine a proper *SHMMAX* value, the Oracle Database parameter ***SGA_MAX_SIZE*** needs to be analyzed. If intending to set ***SGA_MAX_SIZE*** to a value larger than 64 GB, set *SHMMAX* to a size in bytes larger than the ***SGA_MAX_SIZE***. If in doubt on how to properly set *SHMMAX*, a value of 4398046511104 (4 TB) can be used. This value is purposely set higher than the architectural memory limits to ensure that any Oracle SGA value set within an Oracle database instance

may fit in one single shared memory segment.

After calculating *SHMMAX*, the next step is to determine the maximum amount of shared memory pages (*SHMALL*) in a system by capturing system's page size in bytes. The following command can be used to obtain the system page size.

```
# getconf PAGE_SIZE
4096
```

A default installation of Red Hat Enterprise Linux 6 x86_64 provides a *SHMALL* value of 4294967296, the equivalent of 16 TB in system pages. This is determined by the following:

*SHMALL IN BYTES * PAGE_SIZE*

From the example above, 4294967296 bytes * 4096 bytes = 17592186044416 bytes = 16 TB in system pages

To ensure an adequate amount of memory pages are allocated to a single Oracle SGA, it is recommended that the value of *SHMALL* be set to the following:

SHMMAX IN BYTES / PAGE_SIZE

For example, on a system with the *SHMMAX* set to 4398046511104, the value of *SHMALL* is calculated as follows:

```
# echo "4398046511104 / 4096" | bc
1073741824
```

SHMMNI is the maximum total amount of shared memory segments. A default installation of Red Hat Enterprise Linux 6 x86_64 provides a *SHMMNI* default value of 4096. By optimizing the *SHMMAX* value with one shared memory segment per Oracle SGA, this parameter reflects the maximum number of Oracle and ASM instances that can be started on a system. Oracle recommends the value of *SHMMNI* to be left at the default value of 4096.

Snippet of the */etc/sysctl.conf* file:

```
kernel.shmmax = 4398046511104
kernel.shmall = 1073741824
kernel.shmmni = 4096
```

In order for the changes take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.8 Setting Semaphores (SEMMSL, SEMMNI, SEMMNS)

Red Hat Enterprise Linux 6 provides semaphores for synchronization of information between processes. The kernel parameter *sem* is composed of four parameters:

SEMMSL – is defined as the maximum number of semaphores per semaphore set

SEMMNI – is defined as the maximum number of semaphore sets for the entire system

SEMMNS – is defined as the total number of semaphores for the entire system

NOTE: *SEMMNS* is calculated by *SEMMSL* * *SEMMNI*

SEMOPM – is defined as the total number of semaphore operations performed per semop system call.

The following line is required within the */etc/sysctl.conf* file to provide sufficient semaphores for Oracle:

```
kernel.sem = 250 32000 100 128
```

In order for the changes take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.9 Ephemeral Network Ports

Oracle recommends that the ephemeral default port range be set starting at 9000 to 65500. This ensures that all well known ports used by Oracle and other applications are avoided. To set the ephemeral port range, modify the */etc/sysctl.conf* file and add the following line:

```
net.ipv4.ip_local_port_range = 9000 65500
```

In order for the changes take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.10 Optimizing Network Settings

Optimizing the network settings for the default and maximum buffers for the application sockets in Oracle is done by setting static sizes to *RMEM* and *WMEM*. The *RMEM* parameter represents the receive buffer size, while the *WMEM* represents the send buffer size. The recommended values by Oracle are configured within the */etc/sysct.conf* file.

```
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

In order to make the changes take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.11 Increasing synchronous I/O Requests

The kernel parameter *FS.AIO-MAX-NR* sets the maximum number of current asynchronous I/O requests. Oracle recommends setting the value to 1048576. In order to add *FS-AIO-MAX-NR* to 1048576, modify the */etc/sysctl.conf* file as follows:

```
fs.aio-max-nr = 1048576
```

In order for the changes take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.12 Increasing File Handles

The kernel parameter *FS.FILE-MAX* sets the maximum number of open file handles assigned to the Red Hat Enterprise Linux 6 operating system. Oracle recommends that for each Oracle database instance found within a system, allocate 512**PROCESSSES* in addition to the open file handles already assigned to the Red Hat Enterprise Linux 6 operating system. *PROCESSES* within a database instance refers to the maximum number of processes that can be concurrently connected to the Oracle database by the `oracle` user. The default value for *PROCESSES* is 300 for Oracle Database 12c Release 1 (12.1). To properly calculate the *FS.FILE-MAX* for a system, first identify the current *FS.FILE-MAX* allocated to the system via the following command:

```
# cat /proc/sys/fs/file-max
32768
```

Next, add all the *PROCESSES* together from each Oracle database instance found within the system and multiple by 512 as seen in the following command.

```
# echo “512 * 300” | bc
153600
```

NOTE: To determine the current *PROCESSES* value, log into each Oracle database instance and run the following command below. Since no Oracle database has yet been created within this reference environment, the default value of 300 *PROCESSES* is used.

```
SQL> show parameter processes;

NAME                         TYPE    VALUE
------------------------------------ -----------
processes                    integer      300
```

Finally, add the current *FS.FILE-MAX* value with the new value found from multiplying 512*PROCESSES to attain the new *FS.FILE-MAX* value.

```
# echo “32768 + 153600” | bc
186368
```

While the value of the *FS.FILE-MAX* parameter varies upon every environment, this reference environment sets the value at 6815744. Oracle recommends a value no smaller than 6815744. Due to the calculation in the above example equating to 186368, the minimum Oracle recommended value is used. In order to add *FS.FILE-MAX* to 6815744, modify the */etc/sysctl.conf* file as follows:

```
fs.file-max = 6815744
```

In order for the changes take effect immediately, run the following command:

```
# sysctl -p
```

NOTE: Oracle *ASMLib* does not open file descriptors for each device, but instead opens one file descriptor per Oracle process. This reference environment features both methods of implementation, **Section 3.4.3.1 Oracle ASMLib Alternative: Configuring udev Rules** and **Section 3.4.3.2 Configuring Oracle ASMLib.** However, during the installation of Oracle Database 12c Release 1 (12.1) the *FS.FILE-MAX* kernel parameter must be set to at least 6815744.

NOTE: It is recommended to revisit the *FS.FILE-MAX* value if the *PROCESSES* value is increased for any Oracle database instances created.

NOTE: A full listing of all the kernel parameters modified within the */etc/sysctl.conf* file can be found at **Appendix G Kernel Parameters.**

3.3.13 User Accounts & Groups

Prior to the installation of Oracle Database 12c Release 1 (12.1), Oracle recommends the creation of a `grid` user for the Oracle Grid Infrastructure and an `oracle` user for the Oracle Database software installed on the system. For the purposes of this reference environment, the Oracle Grid Infrastructure for a standalone server software owner is the user `grid` and the Oracle Database software owner is the user `oracle`. Each user is designated different groups to handle specific roles based on the software installed. However, the creation of separate users requires that both the `oracle` user and the `grid` user have a common primary group, the Oracle central inventory group (*OINSTALL*).

The following are the recommended system groups created for the installation of the Oracle Database and part of the `oracle` user.

OSDBA group (*DBA*) – determines OS user accounts with DBA privileges

OSOPER group (*OPER*) – an optional group created to assign limited DBA privileges (*SYSOPER* privilege) to particular OS user accounts

With the release of Oracle Database 12c (12.1), Oracle introduces three additional database groups that are optional, but recommended. The recommended following groups provide the ability to separate administrative privileges for common database operations and limit privileged access of the OSDBA group (*DBA*).

OSBACKUPDBA group (*BACKUPDBA*) – an optional group created to assign limited administrative privileges (*SYSBACKUP* privilege) to a user for database backup and recovery

OSDGDBA group (*DGDBA*) – an optional group created to assign limited administrative privileges (*SYSDG* privilege) to a user for administering and monitoring Oracle Data Guard

OSKMDBA group (*KMDBA*) – an optional group created to assign limited administrative privileges (*SYSKM* privilege) to a user for encryption key management when using Oracle Wallet Manager

The following are the recommended system groups created for the installation of the Oracle Grid Infrastructure and part of the `grid` user

OSDBA group (*ASMDBA*) – provides administrative access to Oracle ASM instances

OSASM group (*ASMADMIN*) – provides administrative access for storage files via the *SYSASM* privilege

OSOPER group (*ASMOPER*) – an optional group created to assign limited DBA privileges with regards to ASM to particular OS user accounts

As the `root` user, create the following user accounts, groups, and group assignments using a consistent UID and GID assignments across your organization:

```
# groupadd --gid 54321 oinstall
# groupadd --gid 54322 dba
# groupadd --gid 54323 asmdba
# groupadd --gid 54324 asmoper
# groupadd --gid 54325 asmadmin
# groupadd --gid 54326 oper
# groupadd --gid 54327 backupdba
# groupadd --gid 54328 dgdba
# groupadd --gid 54329 kmdba

# useradd --uid 54321 --gid oinstall --groups dba,oper,asmdba,asmoper,\
>  backupdba,dgdba,kmdba oracle
# passwd oracle

# useradd --uid 54322 --gid oinstall --groups dba,asmadmin,asmdba,asmoper grid
# passwd grid
```

Verify the `oracle` and `grid` user correctly display the appropriate primary and supplementary groups via the commands:

```
# id oracle
uid=54321(oracle) gid=54321(oinstall)
groups=54321(oinstall),54322(dba),54323(asmdba),54324(asmoper),54326(oper),
54327(backupdba),54328(dgdba),54329(kmdba)
# id grid
uid=54322(grid) gid=54321(oinstall)
groups=54321(oinstall),54322(dba),54323(asmdba),54324(asmoper),54325(asmadmin)
```

3.3.14 Setting Shell Limits for the `Grid` and `Oracle` User

Oracle recommends the following settings for the soft and hard limits for the number of open file descriptors (nofile), number of processes (nproc), and size of the stack segment (stack) allowed by each user respectively. The purpose of setting these limits is to prevent a system wide crash that could be caused if an application, such as Oracle, were allowed to exhaust all of the OS resources under an extremely heavy workload.

Create a file labeled *99-grid-oracle-limits.conf* within */etc/security/limits.d/* as follows:

```
# touch /etc/security/limits.d/99-grid-oracle-limits.conf
```

NOTE: The reason that the */etc/security/limits.conf* file is not directly modified is due to the order in which limit files are read in the system. After reading the */etc/security/limits.conf* file, files within the */etc/security/limits.d/* directory are read. If two files contain the same entry, the entry read last takes precedence. For more information visit Red Hat Article: “What order are

the limit files in the limits.d directory read in?"[7]

Within the */etc/security/limits.d/99-grid-oracle-limits.conf* file, add the following soft and hard limits for the *oracle* and *grid* user:

```
oracle soft nproc 16384
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
grid soft nproc 16384
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
```

Due to Bug 15971421[8], the soft limit of nproc is not adjusted at runtime by the Oracle database. Due to this, if the nproc limit is reached, the Oracle database may become unstable and not be able to fork additional processes. A high enough value for the maximum number of concurrent threads for the given workload must be set, or use the hard limit value of 16384 as done above if in doubt.

NOTE: Modifications made to the *99-grid-oracle-limits.conf* file take effect immediately. However, please ensure that any previously logged in `oracle` or `grid` user sessions (if any) are logged out and logged back in for the changes to take effect**.**

As the `root` user, create a shell script labeled *oracle-grid.sh* within */etc/profile.d/* to create the *ulimits* for the `oracle` and `grid` user. The contents of the *oracle-grid.sh* script:

```
#Setting the appropriate ulimits for oracle and grid user
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
       ulimit -u 16384
       ulimit -n 65536
    else
       ulimit -u 16384 -n 65536
    fi
fi

if [ $USER = "grid" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
       ulimit -u 16384
       ulimit -n 65536
    else
       ulimit -u 16384 -n 65536
    fi
fi
```

7 What order are the limits files in the limits.d directory read in? - https://access.redhat.com/site/solutions/199993

8 Oracle Documentation 12.1 - http://docs.oracle.com/cd/E16655_01/readmes.121/e17908/toc.htm#BABICCBB

NOTE: While the *ulimit* values can be set directly within the */etc/profile* file, it is recommended to create a custom shell script within */etc/profile.d* instead. The *oracle-grid.sh* script can be downloaded from the **Appendix K Configuration Files**

As `oracle` and `grid` user, verify the *ULIMIT* values by running the following command:

```
# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 385878
max locked memory       (kbytes, -l) 14854144
max memory size         (kbytes, -m) unlimited
open files                      (-n) 65536
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 10240
cpu time               (seconds, -t) unlimited
max user processes              (-u) 16384
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

3.4 Storage Configuration

The following storage configuration section describes the best practices for setting up device mapper multipath, the use of *udev* rules or Oracle ASMLib for ASM disk management, and the use of the *tuned* package for optimal performance.

3.4.1 Setting up DM-Multipath

Device mapper multipath provides the ability to aggregate multiple I/O paths to a newly created device mapper mapping to achieve high availability, I/O load balancing, and persistent naming. The following procedures provide the best practices to installing and configuring *device mapper multipath* devices.

NOTE: Ensure Oracle database volumes are accessible via the operating system prior to continuing with the section below.

1. As the `root` user, install the *device-mapper-multipath* package using the **`yum`** package manager.

```
# yum install device-mapper-multipath
```

2. Copy the *multipath.conf* file found within */usr/share/doc/device-mapper-multipath-0.4.9/* to */etc/*

```
# cp /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf /etc/
```

3. Capture the scsi id of the local disk(s) on the system.

```
# scsi_id --whitelisted --replace-whitespace –-device=/dev/sda
3600508b1001030353434363646301200
```

4. Uncomment and modify the blacklist section within the */etc/multipath.conf* file to include the *scsi id* of the local disk on the system. Once complete, save the changes made to the *multipath.conf* file.

```
blacklist {
       wwid 3600508b1001030353434363646301200
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
}
```

5. Start the *multipath* daemon.

```
# service multipathd start
Starting multipathd daemon:                                [  OK  ]
```

6. Enable the *multipath* daemon to ensure it is started upon boot time.

```
# chkconfig multipathd on
```

7. Identify the *dm- device, size, and WWID* of each *device mapper* volume for Oracle data disks and recovery disks. In this example, volume *mpathb* is identified via the following command:

```
# multipath -ll
```

Figure 3.4.1.1: Multipath Device (mpathb)

Figure 3.4.1.1: Multipath Device (mpathb) properly identifies the current multipath alias name, size, *WWID,* and *dm* device. This information is required for the application of a custom alias to each volume as shown in step 9.

8. Uncomment the defaults section found within the */etc/multipath.conf* file.

```
defaults {
        udev_dir                /dev
        polling_interval        10
        path_selector           "round-robin 0"
        path_grouping_policy    multibus
        getuid_callout          "/lib/udev/scsi_id --whitelisted
--device=/dev/%n"
        prio                    alua
        path_checker            readsector0
        rr_min_io               100
        max_fds                 8192
        rr_weight               priorities
        failback                immediate
        no_path_retry           fail
        user_friendly_names     yes
}
```

NOTE: The standard options can be customized to better fit your storage array capabilities. Check with your storage vendor for details.

9. Uncomment the multipath section found within the */etc/multipath.conf* file and create an alias for each *device mapper* volume in order to enable persistent naming of those volumes. Once complete, save the changes made to the *multipath.conf* file. The output should resemble the example below. For reference, refer to the Oracle data volumes created for this reference environment displayed in **Table 2.4.3: Oracle Data File Sizes for Reference Architecture.**

```
multipaths {
        multipath {
                wwid                    3600c0ff000d7e7a899d8515101000000
                alias                   db1
        }
        multipath {
                wwid                    3600c0ff000dabfe5a7d8515101000000
                alias                   db2
        }
        multipath {
                wwid                    3600c0ff000d7e7a8dbd8515101000000
                alias                   fra
        }
        multipath {
                wwid                    3600c0ff000dabfe5f4d8515101000000
                alias                   redo
        }
}
```

10. Restart the *device mapper multipath* daemon.

```
# service multipathd restart
ok
Stopping multipathd daemon:                                [  OK  ]
Starting multipathd daemon:                                [  OK  ]
```

11. Verify the *device mapper* paths and aliases are displayed properly. Below is an example of one *device mapper* device labeled *fra*.

```
# multipath -ll
fra (3600c0ff000d7e7a89e85ac5101000000) dm-10 HP,MSA2324fc
size=186G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=130 status=active
| |- 3:0:0:3 sdd  8:48   active ready running
| |- 3:0:1:3 sdh  8:112  active ready running
| |- 4:0:0:3 sdt  65:48  active ready running
| `- 4:0:1:3 sdx  65:112 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 3:0:2:3 sdl  8:176  active ready running
  |- 3:0:3:3 sdp  8:240  active ready running
  |- 4:0:2:3 sdab 65:176 active ready running
  `- 4:0:3:3 sdaf 65:240 active ready running
```

3.4.2 Partitioning Device Mapper Shared Disks

Partitioning of the *device mapper* shared disks is only required when using Oracle *ASMLib*. This reference environment provides instructions to configure either Oracle *ASMLib* or *udev* rules. Partitions for each device mapper volume are created to comply with either option.

Create a partition for each device mapper volume (db1,db2,fra,redo) using parted as displayed below for device db1.

```
# parted /dev/mapper/db1 mklabel gpt mkpart primary "1 -1"
Information: You may need to update /etc/fstab.
```

Once the partition is created, a newly created device mapper device is created as *db1p1*.

```
# ls -l /dev/mapper/db1p1
lrwxrwxrwx. 1 root root 8 Apr 16 15:15 /dev/mapper/db1p1 -> ../dm-11
```

NOTE: A newly created partition requires the alias name followed by *p1* such as d*b1p1* seen above. If *p1* is missing, please run the following **kpartx** command to add the partition mappings to the device mapper disks.

```
# kpartx -a /dev/mapper/db1
```

If the following **kpartx** command does not add the *p1* suffix to each partition, reboot the system.

3.4.3 Configuring Oracle ASM Disks

The configuration of Oracle ASM requires the use of either *udev* rules or Oracle's *ASMLib*. Oracle's *ASMLib* is an optional utility used to manage and assist users with Oracle ASM devices and is not required for proper operation of Oracle ASM disks. Moreover, Oracle *ASMLib* does not have any impact on Oracle Database performance and requires a kernel module labeled **kmod-oracleasm** and proprietary user space utilities to properly function. *udev* rules represent an alternative to Oracle's *ASMLib* and does not require additional kernel modules, thus keeping an overall smaller footprint on the Linux system. While this reference architecture documents both methods of implementation, **only one method** can be applied for a given solution. This section covers best practices of using Red Hat's native *udev* rules to setup the appropriate permissions for each device mapper disk, as well as, the best practices for Oracle's *ASMLib*.

3.4.3.1 Oracle ASMLib Alternative: Configuring udev Rules

The configuration of Oracle ASM requires the use of either *udev* rules or Oracle's ASMLib. This section, focuses on the best practices of using Red Hat's native *udev* rules to setup the appropriate permissions for each device mapper disk.

NOTE: If following the steps in this section, please ignore **Section 3.4.3.2 Configuring Oracle ASMLib**

1. As the `root` user, identify the *Device Mapper Universally Unique IDentifier* (*DM_UUID*) for each *device mapper* volume. The example below shows the *DM_UID* for the partitions of the volumes labeled db1,db2,fra, and redo.

```
# for i in db1p1 db2p1 frap1 redop1; do printf "%s %s\n" "$i" "$(udevadm
info --query=all --name=/dev/mapper/$i | grep -i dm_uuid)"; done
db1p1 E: DM_UUID=part1-mpath-3600c0ff000d7e7a899d8515101000000
db2p1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5a7d8515101000000
frap1 E: DM_UUID=part1-mpath-3600c0ff000d7e7a8dbd8515101000000
redop1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5f4d8515101000000
```

2. Create a file labeled *99-oracle-asmdevices.rules* within */etc/udev/rules.d/*
3. Within *99-oracle-asmdevices.rules* file, create rules for each device similar to the example below:

```
KERNEL=="dm-*",ENV{DM_UUID}=="part1-mpath-
3600c0ff000dabfe5f4d8515101000000",OWNER="grid",GROUP="asmadmin",MODE="06
60"
```

   To understand the rule above, it can be read as follows:

   If any *dm- device (dm-*)* matches the *DM_UUID* of *part1-mpath-3600c0ff000dabfe5f4d8515101000000*, assign that *dm- device* to be owned by the `grid` user and part of the *ASMADMIN* group with the permission mode set to 0660.

4. Save the file labeled *99-oracle-asmdevices.rules*
5. Locate the *dm-* device for each Oracle related partition. An example of how to find the *dm-* device for each partition is to run the following command:

```
# for i in db1p1 db2p1 frap1 redop1; do printf “%s %s\n” “$i” “$(ls
-ll /dev/mapper/$i)”; done
db1p1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/db1p1 -> ../dm-11
db2p1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/db1p1 -> ../dm-12
frap1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/db1p1 -> ../dm-13
redop1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/db1p1 -> ../dm-14
```

6. Apply and test the rules created within the *99-oracle-asmdevices.rules* by running a **`udevadm test`** on each device.

```
 # udevadm test /sys/block/dm-11
[ ... Output Abbreviated ... ]
udevadm_test: DM_NAME=db1p1
udevadm_test: DM_UUID=part1-mpath-3600c0ff000d7e7a86485ac5101000000
udevadm_test: DM_SUSPENDED=0
udevadm_test: DEVLINKS=/dev/mapper/db1p1 /dev/disk/by-id/dm-name-db1p1
/dev/disk/by-id/dm-uuid-part1-mpath-3600c0ff000d7e7a86485ac5101000000
/dev/block/253:11
udevadm_test: ID_FS_TYPE=oracleasm
```

7. Confirm the device has the desired permissions

```
# ls -lh /dev/dm-11
brw-rw----. 1 grid asmadmin 253, 11 Jun  6 20:59 /dev/dm-11
```

NOTE: For simplicity, this *99-oracle-asmdevices.rules* file is included in **Appendix I 99-oracle-asmdevices.rules**

3.4.3.2 Configuring Oracle ASMLib

In order to configure Oracle *ASMLib* the following components are required:
`kmod-oracleasm, oracleasm-support, and oracleasmlib`

NOTE: If **Section 3.4.3.1 Oracle ASMLib Alternative: Configuring udev Rules** has been configured, ignore this section and continue to **Section 3.4.4 Optimizing Database Storage using Automatic System Tuning**

The *ASMLib* kernel module package (**`kmod-oracleasm`**) is provided for Red Hat customers via the Supplementary Channel on Red Hat Network (RHN). In order to properly install and configure *ASMLib* the following procedures must be followed.

As the `root` user,

1. Enable the Red Hat Enterprise Linux 6 Supplementary repository as shown in Section **3.3.1 Using the Red Hat Subscription Manager (Option A)** or via the RHN Classic knowledge base article[9].
2. Download the *ASMLib* library package (**`oracleasmlib`**)

```
# wget http://download.oracle.com/otn_software/asmlib/oracleasmlib-2.0.4-1.el6.x86_64.rpm
```

3. Download the *ASMLib* utilites package (**`oracleasm-support)`**

```
# wget http://public-yum.oracle.com/repo/OracleLinux/OL6/latest/x86_64/getPackage/oracleasm-support-2.1.8-1.el6.x86_64.rpm
```

4. Install the *ASMLib* kernel module package (**`kmod-oracleasm`**), *ASMLib* library package (**`oracleasmlib`**), *ASMLib* utilities package (**`oracleasm-support`**) using the following command:

```
# yum install kmod-oracleasm oracleasmlib-2.0.4-1.el6.x86_64.rpm
oracleasm-support-2.1.8-1.el6.x86_64.rpm
```

9 Enabling the Supplementary Repository Knowledge Base Article, https://access.redhat.com/knowledge/articles/58637

5. Configure *ASMLib* using the following command:

```
# /usr/sbin/oracleasm configure -i
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM library
driver.  The following questions will determine whether the driver is
loaded on boot and what permissions it will have.  The current values
will be shown in brackets ('[]').  Hitting <ENTER> without typing an
answer will keep that current value.  Ctrl-C will abort.

Default user to own the driver interface []: grid
Default group to own the driver interface []: asmadmin
Start Oracle ASM library driver on boot (y/n) [ n ]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done
```

6. Within */etc/sysconfig/oracleasm*, set the **ORACLEASM_SCANORDER** and **ORACLEASM_SCANEXCLUDE** parameters as follows:

```
# ORACLEASM_SCANORDER: Matching patterns to order disk scanning
ORACLEASM_SCANORDER="dm"

# ORACLEASM_SCANEXCLUDE: Matching patterns to exclude disks from scan
ORACLEASM_SCANEXCLUDE="sda"
```

NOTE: The **ORACLEASM_SCANORDER** set to *dm* ensures that when **oracleasm** scans the disks, it is appropriately scanning devices known by the kernel. The **ORACLEASM_SCANEXCLUDE** is set to *sda* ensuring that local disk *sda* is to be ignored by **oracleasm**.

7. Run the following **oracleasm** command to ensure that **oracleasm** is enabled.

```
# /etc/init.d/oracleasm enable
Writing Oracle ASM library driver configuration: done
Initializing the Oracle ASMLib driver:                     [  OK  ]
Scanning the system for Oracle ASMLib disks:               [  OK  ]
```

8. Run the following **oracleasm** command to create and label all Oracle related volumes as an ASM disk. The example below creates an ASM disk labeled *DATA1* for the following */dev/mapper/db1p1* partition.

```
# /usr/sbin/oracleasm createdisk DATA1 /dev/mapper/db1p1
Writing disk header: done
Instantiating disk: done
```

NOTE: It is highly recommended to have all Oracle related disks to be included within Oracle ASM.

9. Verify all the Oracle ASM disks created are listed.

```
# /usr/sbin/oracleasm listdisks
DATA1
DATA2
FRA1
REDO1
```

10. If no disks are listed or if any disks are missing, run the following command to rescan the ASM disks.

```
# /usr/sbin/oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
```

NOTE: If the issue persists after a rescan of the Oracle ASM disks, a reboot of the system might be required via the **`reboot`**, **`shutdown`**, or **`init 6`** commands:

```
# shutdown -r now
```

3.4.4 Optimizing Database Storage using Automatic System Tuning

The **tuned** package in Red Hat Enterprise Linux 6 is recommended for automatically tuning the system for common workloads via the use of profiles. Each profile is tailored for different workload scenarios such as:

- enterprise-storage
- power savings
- high network throughput

It is recommended to create a custom profile (enterprise-storage-no-thp) with the same settings as the enterprise-storage profile but that disables Transparent HugePages (THP) for Oracle databases workload environments. For more information on why THP is disabled, see **Section 4.1.5 Enabling HugePages**. **Table 3.4.4.1: Profile Comparison** provides details between the default profile, enterprise-storage profile, and the custom profile enterprise-storage-no-thp.

Tuned Parameters	default	enterprise-storage	enterprise-storage-no-thp
I/O Elevator	CFQ	deadline	deadline
CPU governor	OnDemand	performance	performance
kernel.sched_min_granularity_ns	4ms	10ms	10ms
kernel.sched_wake_up_granularity_ns	4ms	15ms	15ms
Disk read-ahead	1x	4x	4x
vm.dirty_ratio	20%	40%[10]	40%[10]
File-system barrier	on	off	off
Transparent HugePages	on	on	off

Table 3.4.4.1: Profile Comparison

The following procedures provide the steps that are required to create, install, enable, and select the **enterprise-storage-no-thp** profile.

1. Install the **tuned** package via the yum package manager.

```
# yum install tuned
```

2. Enable **tuned** to ensure it is started upon boot time.

```
# chkconfig tuned on
```

10 The vm.dirty_ratio value explicitly set within the */etc/sysctl.conf* file has precedence over values set by tuned.

3. Start the **tuned** service

```
# service tuned start
Starting tuned:                                            [  OK  ]
```

4. Copy the existing enterprise-storage profile into a new directory called enterprise-storage-no-thp via the following commands:

```
# cd /etc/tune-profiles/
# cp -r enterprise-storage enterprise-storage-no-thp
```

5. Disable Transparent HugePages in the new profile via the following command:

```
# sed -ie 's,set_transparent_hugepages always,set_transparent_hugepages
never', /etc/tune-profiles/enterprise-storage-no-thp/ktune.sh
```

6. Verify the change was properly set via the following command:

```
# grep set_transparent_hugepages /etc/tune-profiles/enterprise-storage-
no-thp/ktune.sh
set_transparent_hugepages never
```

7. Activate the newly created enterprise-storage-no-thp profile:

```
# tuned-adm profile enterprise-storage-no-thp
Stopping tuned:                                            [  OK  ]
Switching to profile 'enterprise-storage-no-thp'
Applying ktune sysctl settings:
/etc/ktune.d/tunedadm.conf:                                [  OK  ]
Calling '/etc/ktune.d/tunedadm.sh start':                  [  OK  ]
Applying sysctl settings from /etc/sysctl.conf
Applying deadline elevator: dm-0 dm-1 dm-10 dm-11 dm-12 dm-13 dm-14 dm-15
dm-16 dm-17 dm-18 dm-19 dm-2 dm-20 dm-3 dm-4 dm-5 dm-6 dm-7 dm-8 dm-9 sda
sdaa sdab sdac sdb sdc sdd sde sdf sdg sdh sdi sdj sdk sdl sdm sdn sdo
sdp sdq sdr sds sdt sdu sdv sdw sdx sdy sdz
[  OK  ]
Starting tuned:                                            [  OK  ]
```

8. Verify that THP is now disabled via the following command:

```
# cat /sys/kernel/mm/redhat_transparent_hugepage/enabled
always madvise [never]
```

9. Ensure to also disable *transparent* huge pages persistently across reboots by adding to the kernel boot command line within the */etc/grub.conf* the following:

```
title Red Hat Enterprise Linux (2.6.32-431.el6.x86_64)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/mapper/myvg-root
rd_NO_LUKS LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=auto rd_NO_DM rd_LVM_LV=myvg/root  KEYBOARDTYPE=pc
KEYTABLE=us rd_LVM_LV=myvg/swap rhgb quiet transparent_hugepage=never
        initrd /initramfs-2.6.32-431.el6.x86_64.img
```

For simplicity, the *enterprise-storage-no-thp* profile is included in **Appendix K Configuration Files**

NOTE: If at any point in time a revert to the original settings are required with persistence across reboots, the following commands can be run:

```
# service tuned stop
# chkconfig tuned off
```

NOTE: Even if reverting to the original settings, it is recommended to keep *transparent* huge pages disabled within the */etc/grub.conf* file.

4 Oracle 12c R1 Configuration

4.1.1 Installing Oracle Grid Infrastructure (Required for ASM)

The installation of the Oracle Grid Infrastructure for Oracle Database 12c Release 1 (12.1) is required for the use of Oracle ASM. Prior to the installation of the Oracle Grid Infrastructure, ensure that the prerequisites from the following sections have been met:

- **Section 2 Reference Architecture Environment**
- **Section 3 Reference Architecture Configuration Details**

NOTE: The reference environment uses the */u01/app/grid* as the `grid` base. The owner is set to `grid` and the group is set to *OINSTALL*. Run the following commands to create the `grid` base directory and set the appropriate permissions:

As the `root` user,

```
# mkdir --parents /u01/app/grid
# chown --recursive grid.oinstall /u01/
```

1. Download the two Oracle Grid Infrastructure software files[11] from the Oracle Software Delivery Cloud site[11].
2. As the `grid` user, create a temporary directory within */u01/app/grid/grid-software* to store the Oracle Grid Software **`zip`** file, move the Oracle Grid Software **`zip`** file to the */u01/app/grid/grid-software* location, ensure the Oracle Grid Software **`zip`** has the proper permissions and unpack its contents.

   As the `grid` user,

   ```
   # mkdir /u01/app/grid/grid-software
   # mv V38501-01_1of2.zip /u01/app/grid/grid-software
   # mv V38501-01_2of2.zip /u01/app/grid/grid-software
   ```

   As the `root` user,

   ```
   # chown grid.oinstall V38501-01_1of2.zip
   # chown grid.oinstall V38501-01_2of2.zip
   ```

   As the `grid` user,

   ```
   # cd /u01/app/grid/grid-software
   # unzip V38501-01_1of2.zip
   # unzip V38501-01_2of2.zip
   ```

3. As the `grid` user, start the OUI via the command:

   ```
   # /u01/app/grid/grid-software/grid/runInstaller
   ```

11 Oracle Database 12c Release 1 Grid Infrastructure (12.1.0.1.0) - V38501-01_1of2.zip and V38501-01_2of2.zip from http://edelivery.oracle.com

NOTE: Ensure to **SSH** with the *-Y* option as the `grid` user from the client server. Otherwise, the following error will occur.

```
# /u01/app/grid/grid-software/grid/runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 120 MB.   Actual 3689 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 16415 MB
Passed
Checking monitor: must be configured to display at least 256 colors
>>> Could not execute auto check for display colors using command
/usr/bin/xdpyinfo. Check if the DISPLAY variable is set.    Failed <<<<

Some requirement checks failed. You must fulfill these requirements
before continuing with the installation,

Continue? (y/n) [n] n
```

4. Within the ***Download Software Updates*** window, select the option to either enter the My Oracle Support credentials to **Use My Oracle Support credentials for download** or select **Skip software updates**. This reference environment selected **Skip software updates** and click **Next**.
5. Within the ***Installation Option*** window, select **Install and Configure Oracle Grid Infrastructure for a Standalone Server** and click **Next**.
6. Within the ***Product Languages*** window, select the appropriate language, and click **Next**.
7. Within the ***Create ASM Disk Group*** window, provide the following:
   - a Disk Group Name, i.e. *DATA*
   - Redundancy Level
     - *External* – redundancy provided by the storage system RAID, and not by Oracle ASM.
     - *Normal* – provides two-way mirroring by Oracle ASM, thus provided two copies of every data extent.
     - *High* – provides three-way mirroring by Oracle ASM thus enduring the loss of two ASM disks within different failure groups.
   - Disks to be assigned to the Disk Group, i.e. */dev/mapper/db1p1, /dev/mapper/db2p1*

     NOTE: This reference environment uses *Normal Redundancy.*

- *Allocation Unit (AU) Size set to 4 MB*[12]
  - A 4 MB AU Size is used to decrease the amount of extents Oracle needs to manage. With less extents to manage, CPU utilization and memory consumption is reduced thus improving performance. The AU Size varies depending on the type of Oracle workload, I/O size per transaction, and overall diskgroup size. There is no “best size” for AU size, but a good starting point is a 4 MB. Please ensure to visit Oracle's documentation[12] for more information.

To display the appropriate candidate disks, click on the **Change Discovery Path** button and enter as the **Disk Discovery Path** one of the following as appropriate:

- For Device Mapper devices, type: **/dev/mapper/***
- For Oracle *ASMLib* marked disks, type: **/dev/oracleasm/disks/***

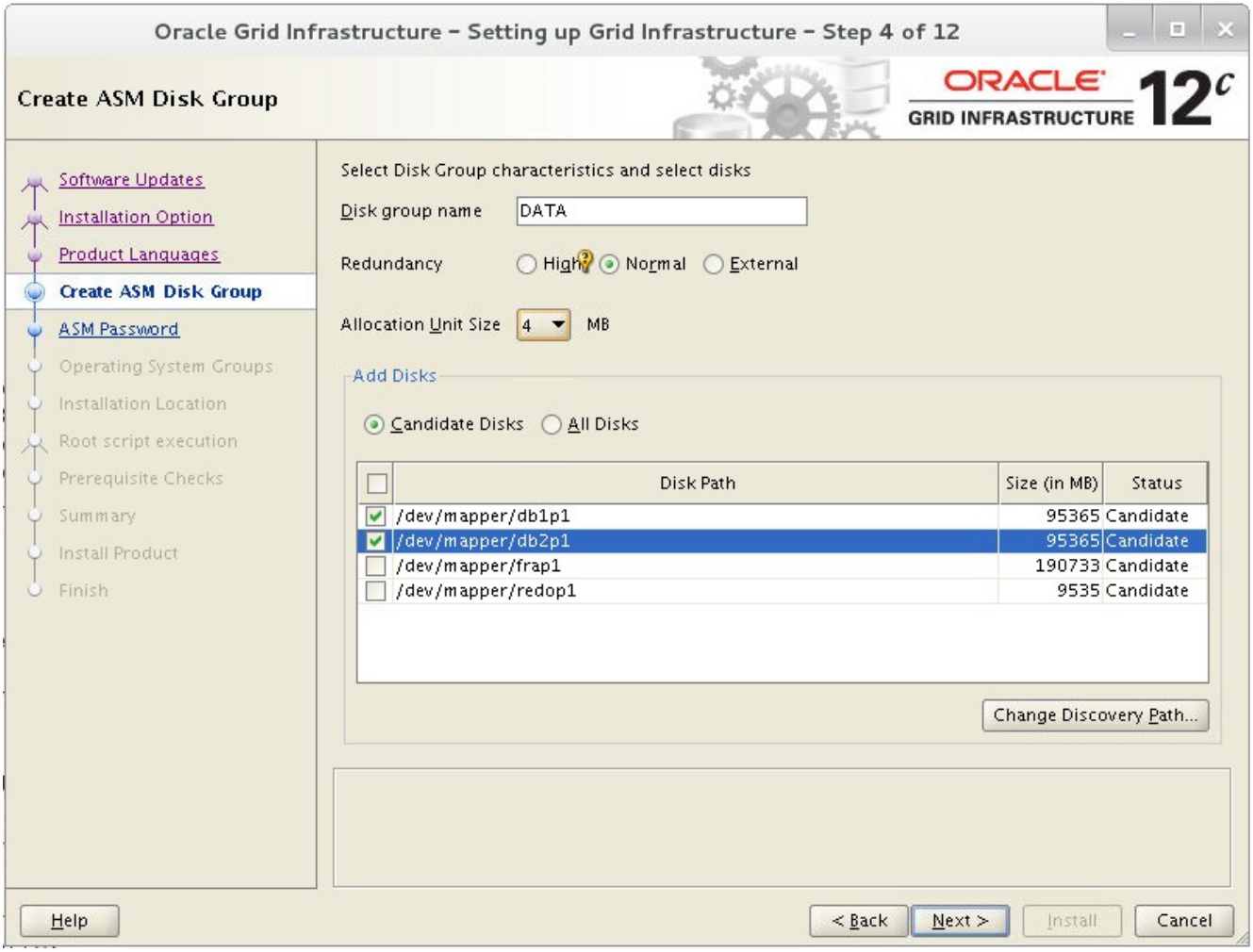

Figure 4.1.1.1: ASM Disk Group Window

12 Oracle ASM Extents - http://docs.oracle.com/cd/E16655_01/server.121/e17612/asmcon.htm#OSTMG94063

8. Click **Next** once complete within the ***Create ASM Disk Group*** window
9. Within the ***ASM Password*** window, specify the password for the *SYS* and *ASMSNMP* user accounts.
10. Within the ***Operating System Groups*** window, select the appropriate OS groups. The values as created and assigned within this reference environment are as follows:
    - Oracle ASM Administrator Group – *ASMADMIN*
    - Oracle ASM DBA Group – *ASMDBA*
    - Oracle ASM Operator Group – *ASMOPER*
11. Within the ***Installation Location*** window, specify the appropriate Oracle base and software locations. The values set by this reference environment are as follows:
    - *ORACLE BASE - /u01/app/grid*
    - *SOFTWARE LOCATION - /u01/app/grid/product/12.1.0/grid*
12. Within the ***Create Inventory*** window, specify the inventory directory. The values set by this reference environment are as follows:
    - Inventory Directory - */u01/app/oraInventory*
13. Within the ***Root script execution configuration*** window, select the check box labeled **Automatically run configuration scripts** and enter the `root` user credentials. The step specifying the `root` user credentials in order to run specific configuration scripts automatically at the end of the installation is optional. For the purposes of this reference environment, the `root` credentials are given in order to speed up the Oracle Grid Infrastructure installation process.
14. Within the ***Summary*** window, review all the information provided, and select **Install** to start the installation.
15. During the installation process, within the **Oracle Grid Infrastructure** pop up window, select **yes** to allow the installer to run as the `root` user to execute the configuration scripts.
16. Within the ***Finish*** window, verify the installation was successful and click **Close**.

4.1.2 Installing Oracle 12c R1 Database Software

Prior to the installation of the Oracle Database 12c Release 1 (12.1), ensure the following prerequisites from the following sections have been met:

- **Section 2 Reference Architecture Environment**
- **Section 3 Reference Architecture Configuration Details**

NOTE: The reference environment uses the */u01/app/oracle* as the Oracle base. The owner is set to `oracle` and the group is set to *OINSTALL*. Run the following commands to create the `oracle` base directory set the appropriate permissions:

As the `root` user,

```
# mkdir /u01/app/oracle
# chown --recursive oracle.oinstall /u01/app/oracle
```

1. Download the two Oracle Database software files[13] from the Oracle Software Delivery Cloud site.
2. As the `oracle` user, create a temporary directory within */u01/app/oracle/oracle-software* to store the Oracle Database Software **`zip`** files, move the Oracle Database software **`zip`** files to the */u01/app/oracle/oracle-software* location, ensure the Oracle Database software **`zip`** has the proper permissions and unpack its contents.

   As the `oracle` user,

   ```
   # mkdir /u01/app/oracle/oracle-software
   # mv V38500-01_1of2.zip V38500-01_2of2.zip  /u01/app/oracle/oracle-
   software/
   ```

   As the `root` user,

   ```
   # chown oracle.oinstall /u01/app/oracle/oracle-software/V38500-01_1of2.zip
   # chown oracle.oinstall /u01/app/oracle/oracle-software/V38500-01_2of2.zip
   ```

   As the `oracle` user,

   ```
   # cd /u01/app/oracle/oracle-software
   # unzip  V38500-01_1of2.zip
   # unzip   V38500-01_2of2.zip
   ```

3. As the `oracle` user, start the OUI via the command:

   ```
   # /u01/app/oracle/oracle-software/database/runInstaller
   ```

13 Oracle Database 12c Release 1 (12.1.0.1.0) - V38500-01_1of2.zip and V38500-01_2of2.zip from http://edelivery.oracle.com

NOTE: Ensure to issue **SSH** with the *-Y* option as the `oracle` user from the client server. Otherwise, the following error will occur.

```
# /u01/app/oracle/oracle-software/database/runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 120 MB.   Actual 3461 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 20479 MB
Passed
X11 connection rejected because of wrong authentication.
X11 connection rejected because of wrong authentication.
Checking monitor: must be configured to display at least 256 colors
>>> Could not execute auto check for display colors using command /
usr/bin/xdpyinfo. Check if the DISPLAY variable is set.    Failed <<<<

Some requirement checks failed. You must fulfill these requirements
before continuing with the installation,

Continue? (y/n) [n] n
```

4. Within the ***Configure Security Updates*** window, provide the My Oracle Support email address for the latest security issues information. Otherwise uncheck the **I wish to receive security updates via My Oracle Support** and click **Next**.
5. Within the ***Download Software Updates*** window, provide the My Oracle Support credentials to download the latest security updates. Otherwise select the **Skip software updates** radio button. This reference environment selected **Skip software updates**. Click **Next**.

6. Within the ***Installation Option*** window, select **Install database software only** and click **Next**.

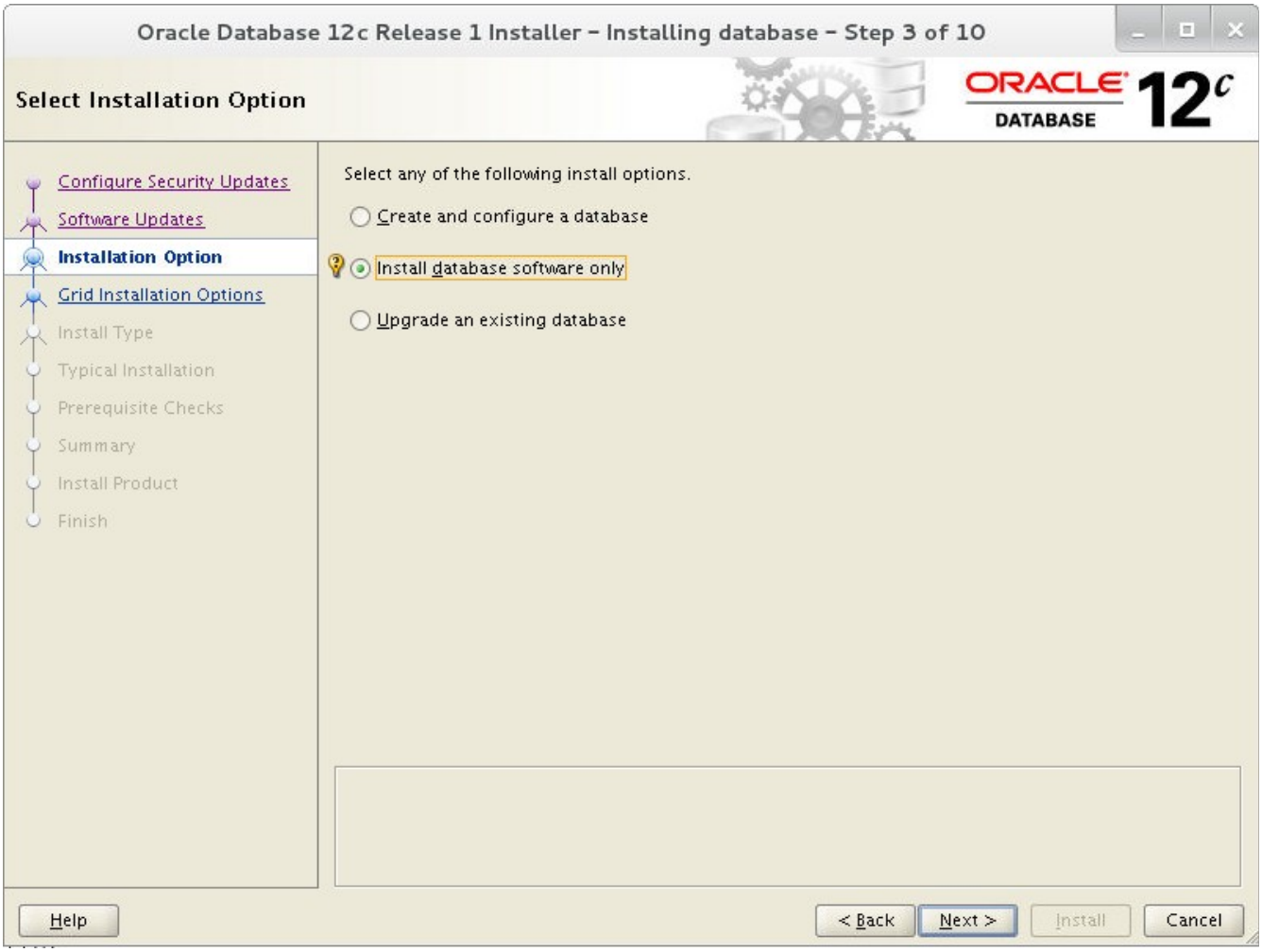

Figure 4.1.2.1: Installation Option Window

7. Within the ***Grid Installation Options*** window, select **Single Instance database installation** as the type of database installation being performed.

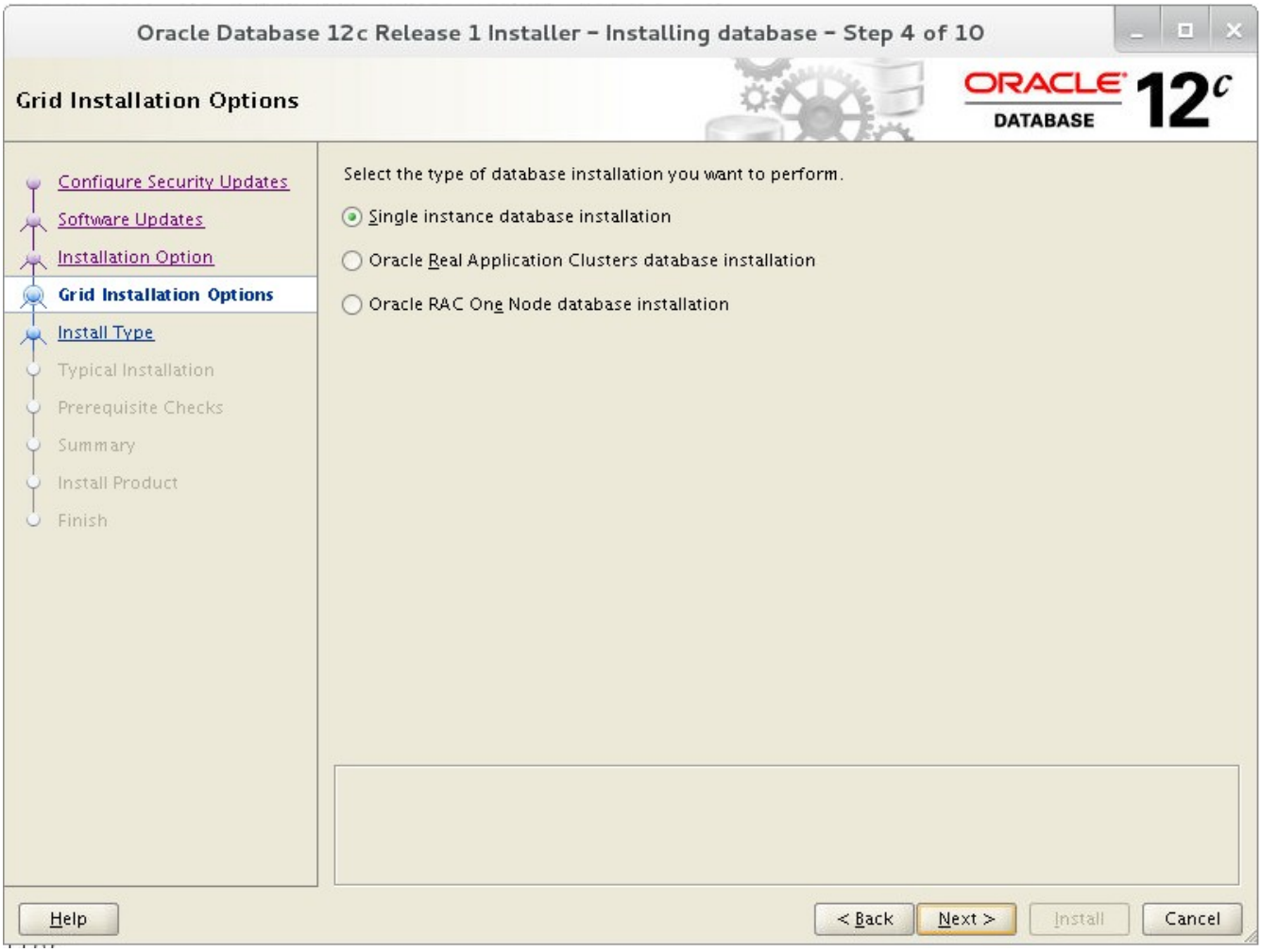

Figure 4.1.2.2: Grid Installation Options Window

8. Within the ***Product Languages*** window, select the appropriate language for the installation.
9. Within the ***Database Edition*** window, select the appropriate database edition and click **Next**. For the purposes of this reference environment, **Enterprise Edition** is the edition of choice.
10. Within the ***Installation Location*** window, select the appropriate Oracle base and software location and click **Next**. For the purposes of this reference environment, the following values are set:
    - *ORACLE BASE - /u01/app/oracle*
    - *SOFTWARE LOCATION - /u01/app/oracle/product/12.1.0/dbhome_1*

11. Within the ***Operating System Groups*** window, select the appropriate OS groups and click **Next**. For the purposes of this reference environment, the following values are set as:
    - Database Administrator Group – *DBA*
    - Database Operator Group – *OPER*
    - Database Backup and Recovery Group – *BACKUPDBA*
    - Data Guard Administrative group – *DGDBA*
    - Encryption Key Management Administrative group – *KMDBA*
12. Within the ***Summary*** window, review all the information provided, and select **Install** to start the installation.
13. Once the installation completes, execute the scripts within the **Execute Configuration scripts** window. As the `root` user, run the following:

```
# /u01/app/oracle/product/12.1.0/dbhome_1/root.sh
Performing root user operation for Oracle 12c

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/12.1.0/dbhome_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
The contents of "dbhome" have not changed. No need to overwrite.
The contents of "oraenv" have not changed. No need to overwrite.
The contents of "coraenv" have not changed. No need to overwrite.

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
```

NOTE: In the example above, */u01/app/oracle/product/12.1.0/dbhome_1* is the Oracle home directory.

14. Click **OK** within the ***Execute Configuration scripts*** window.
15. Within the ***Finish*** window, verify the installation was successful and click **Close**.

4.1.3 Creating ASM Diskgroups via the ASM Configuration Assistant (ASMCA)

Prior to the creation of an Oracle database, create the Fast Recovery Area (FRA) and Redo Logs Oracle ASM diskgroups via Oracle's ASM Configuration Assistant (ASMCA).

1. **ssh** with the -Y option as the `grid` user is required prior to running **asmca**.
2. As the `grid` user, start **asmca** via the following command:

```
# /u01/app/grid/product/12.1.0/grid/bin/asmca
```

NOTE: In the example above, */u01/app/grid/product/12.1.0/grid* is the `grid` home directory.

3. Via the **asmca** application, select the **Disk Groups** tab and click **Create**.

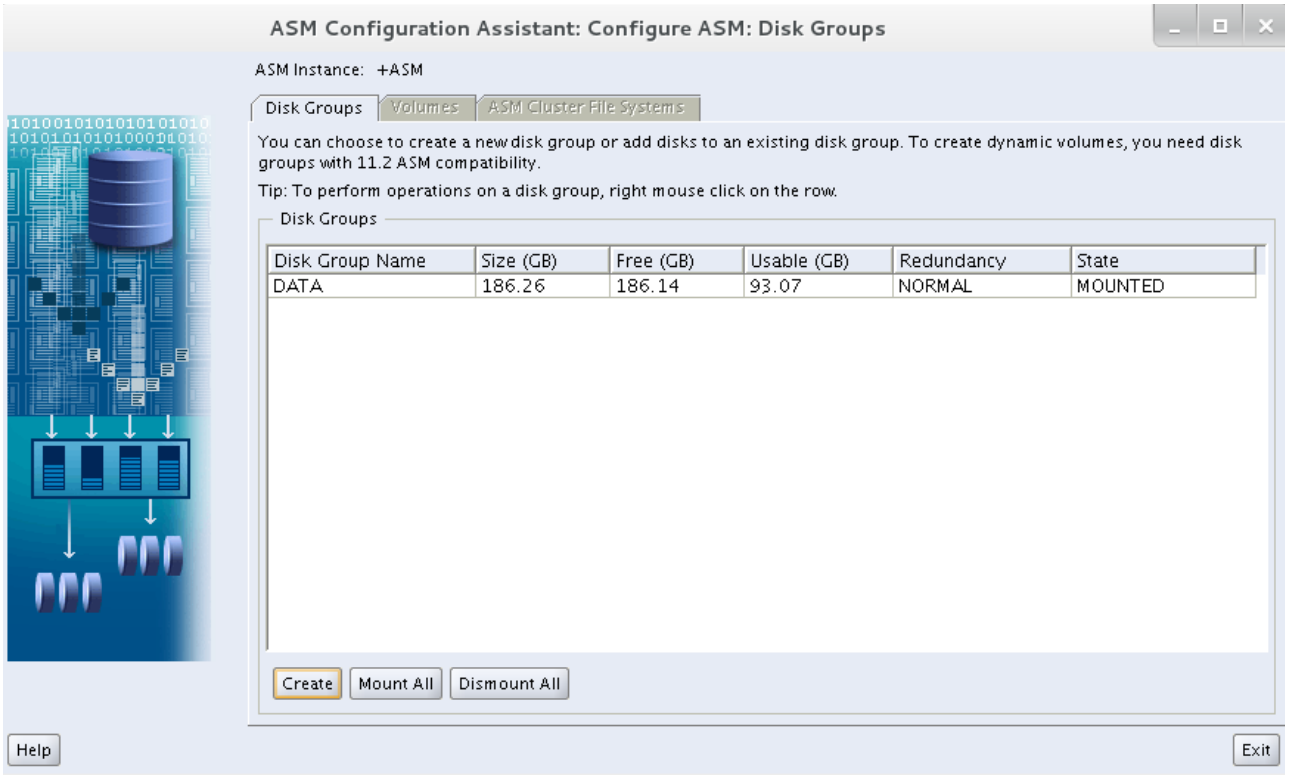

Figure 4.1.3.1: ASMCA Disk Groups Tab

4. Within the ***Create Disk Group*** window, provide the following:
   - A name for the disk group, i.e. *FRADG*
   - Redundancy level for the disk group, i.e. *External Redundancy*
   - Selection of the disks to be added to the disk group, i.e. */dev/mapper/frap1*
   - Click the **Show Advanced Options** radio button and provide the appropriate AU Size. This reference environment uses an AU Size of 4 MB for disk groups all disks groups: *DATA*, *FRADG*, and *REDODG*.

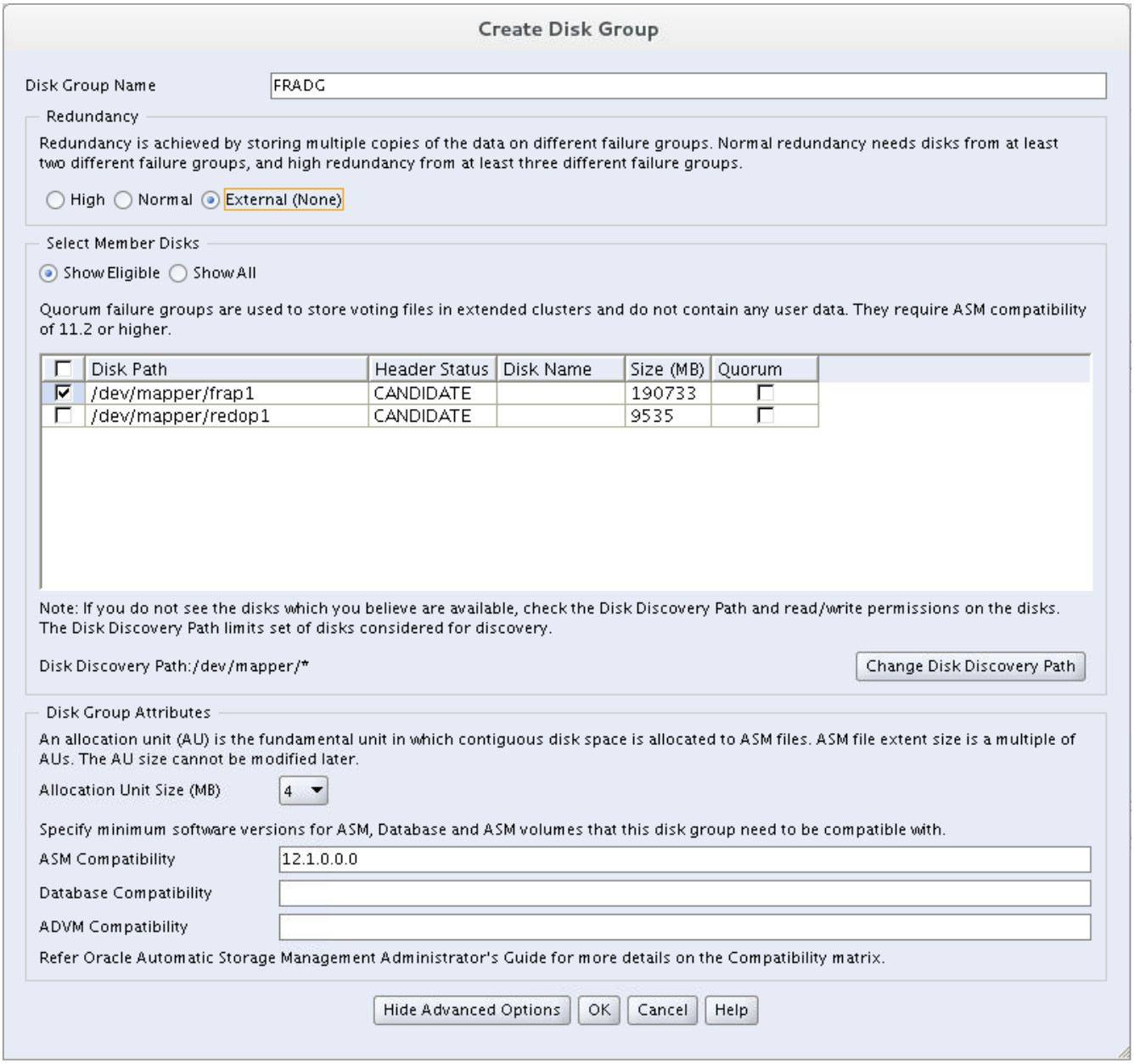

Figure 4.1.3.2:

NOTE: To display the appropriate eligible disks, click on the **Change Discovery Path** button and enter as the 'Disk Discovery Path' one of the following as appropriate:

- For Device Mapper devices, type: **/dev/mapper/***
- For Oracle ASMLib marked disks, type: **/dev/oracleasm/disks/***

Click the **OK** button once the steps above are complete.

5. Repeat steps 2 and 3 to configure a disk group for the redo logs.
   NOTE: Separation of redo logs into a separate Oracle ASM disk group is optional, but recommended.
6. Once all disk groups are created, click the **Exit** button from the main ***ASM Configuration Assistant*** window. Click **yes**, when asked to confirm quitting the application.

4.1.4 Creating Pluggable Databases using Database Configuration Assistant (DBCA)

With the introduction to Oracle Database 12c, Oracle introduced the Multitenant architecture. The Multitenant architecture provides the ability to consolidate multiple databases known as pluggable databases (PDBs) into a single container database (CDB). It provides advantages[14] which include easier management and monitoring of the physical database, fewer patches and upgrades, performance metrics consolidated into one CDB, and sizing one SGA instead of multiple SGAs. While using the Multitenant architecture is optional, this reference architecture focuses on describing the step-by-step procedure of taking advantage of it. When creating an Oracle database, the recommended method is the usage of the **dbca** utility.

Prior to getting into to the details of installing a container database (CDB) and deploying pluggable databases (PDB), an overview of the key concepts of the Multitenant Architecture is provided.

Container[14] – is a collection of schemas, objects, and related structures in a multitenant container database (CDB) that appears logically to an application as a separate database. Within a CDB, each container has a unique ID and name.

A CDB consists of two types of containers: the root container and all the pluggable databases that attach to a CDB.

Root container[14] – also called *the root*, is a collection of schemas, schema objects, and nonschema objects to which all PDBs belong. Every CDB has one and only one root container, which stores the system metadata required to manage PDBs (no user data is stored in the root container). All PDBs belong to the root. The name of the root container is *CDB$ROOT*

PDB[14] – is a user-created set of schemas, objects, and related structures that appears

14 Purpose of a Multitenant Environment - http://docs.oracle.com/cd/E16655_01/server.121/e17636/cdb_intro.htm#ADMIN13509

logically to an application as a separate database. Every PDB is owned by *SYS*, which is a common user in the CDB, regardless of which user created the CDB.

For more information on Oracle's Multitenant architecture, visit Oracle's documentation[15].

The following section describes the step-by-step procedure to create a container database (CDB) that holds two pluggable databases (PDB) thus taking advantage of Oracle's Multitenant architecture.

1. **ssh** with the -Y option as the `oracle` user prior to running **dbca**.
2. As the `oracle` user, run the **dbca** utility via the command:

```
# /u01/app/oracle/product/12.1.0/dbhome_1/bin/dbca
```

   NOTE: In the example above, */u01/app/oracle/product/12.1.0/dbhome_1* is the Oracle home directory.

3. Within the ***Database Operations*** window, select **Create a Database** radio button and click **Next**.
4. Within the ***Creation Mode*** window, select **Advanced Mode** radio button and click **Next**.
5. Within the ***Database Template*** window, select **Custom Database** radio button and click **Next**.

15 Oracle Documentation, Multitenant architecture - http://docs.oracle.com/cd/E16655_01/server.121/e17633/cdbovrvw.htm#CNCPT89234

6. Within the ***Database Identification*** window, set a global database name and Oracle System Identifier (SID), i.e. *cdb*. Check the check box that reads **Create as Container Database**. Select the number of PDBs to install and provide a PDB Name Prefix, i.e. *pdb* and click **Next**. This reference environment creates two PDBs.

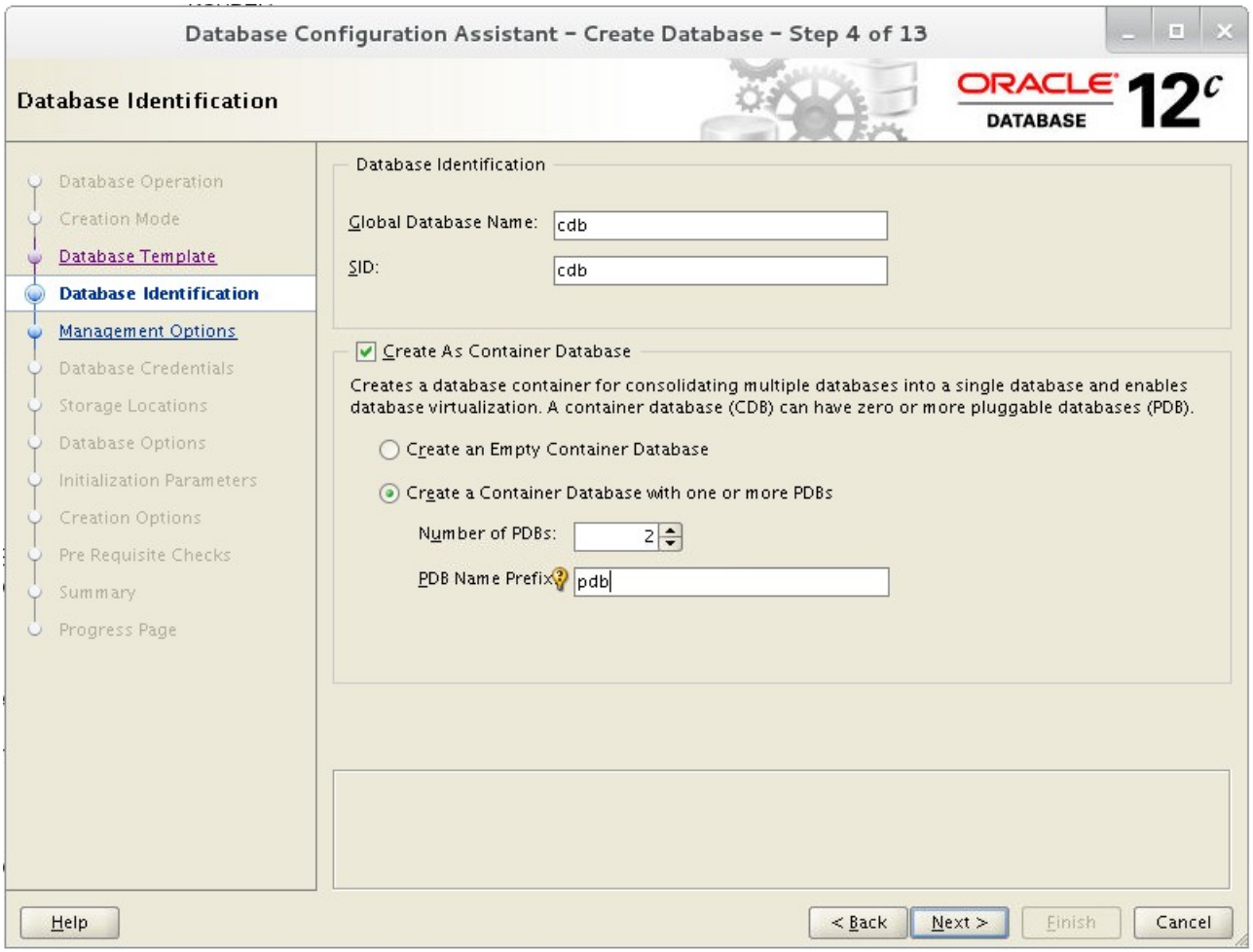

Figure 4.1.4.1: Database Identification Window

7. Within the ***Management Options*** window, ensure the check box **Configure Enterprise Manager (EM) Database Express** is checked (default) and click **Next**.
8. Within the ***Database Credentials*** window, provide the administrative passwords for *SYS*,*SYSTEM*, and *PDBADMIN*, click **Next**.
9. Within the ***Network Configuration*** window, ensure the Listener labeled **LISTENER** that resides in the Grid Infrastructure Home showing an **Up** status is checked (default) and click **Next**.

10. Within the ***Storage Locations*** window, select the appropriate **Storage Type** and storage locations. For the purposes of this reference environment, the following selections were made:
    - Database Files Storage Type – Automatic Storage Management (ASM)
        - Use Common Location for All Database Files selected
            - Database File Locations: *+DATA*
            - Use Oracle-Managed Files (checked)
    - Recovery Related Files Storage Type – Automatic Storage Management (ASM)
        - Storage Locations – Specify Fast Recovery Area
            - Fast Recovery Area: *+FRADG*
            - Fast Recovery Area Size: 142964 Megabytes[16]
            - Enable Archiving (checked)

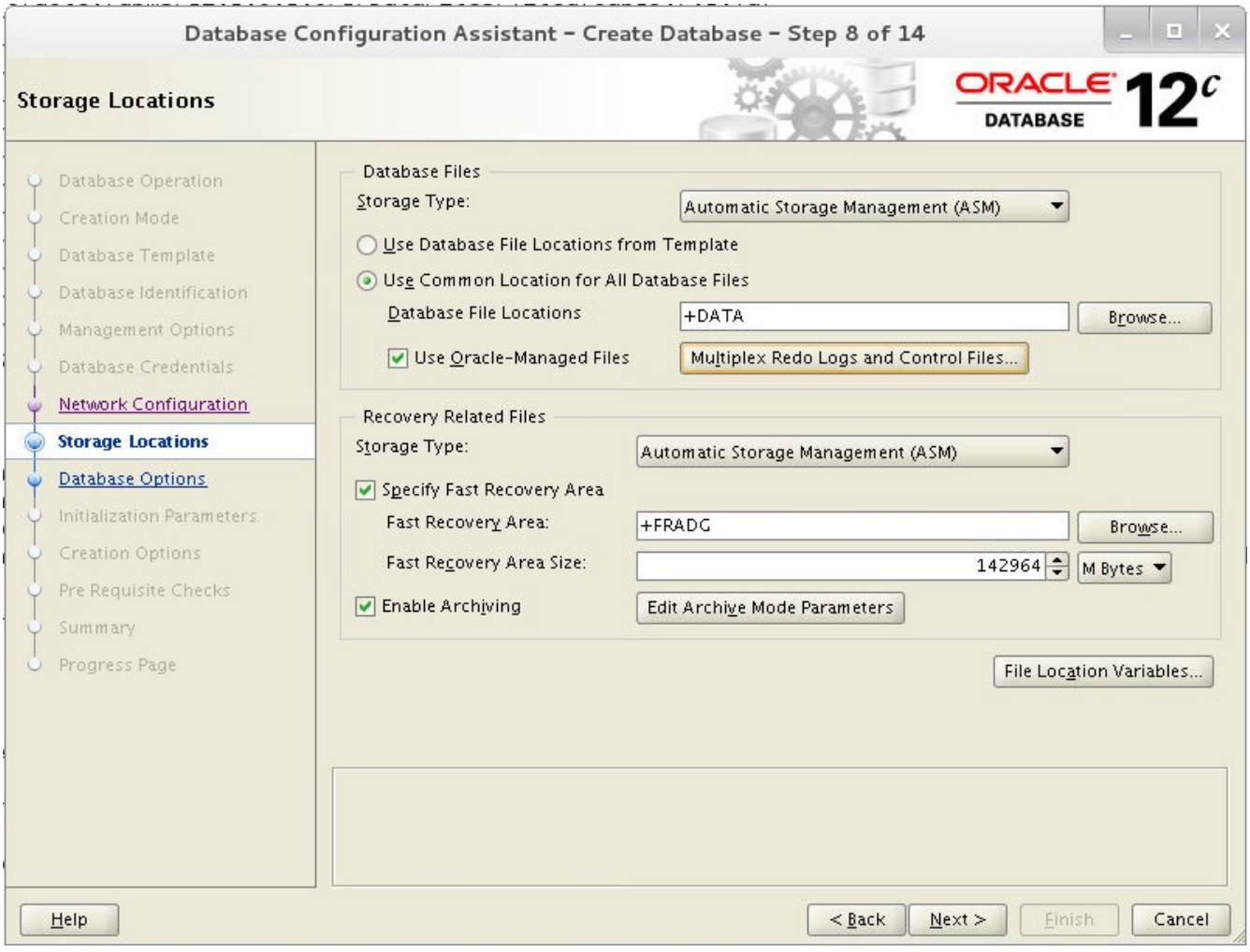

Figure 4.1.4.2: Storage Locations Window

16 Doc ID [305648.1] – "What is Flash Recovery Area and how to configure it?" at http://support.oracle.com

11. Within the same ***Storage Locations*** window, select the **Multiplex Redo Logs and Control Files** button and within the ***Multiplex Redo Logs and Control Files*** window, add the Redo Logs diskgroup, i.e. *+REDODG* and click **OK**, then click **Next** in the **Storage Locations** window.

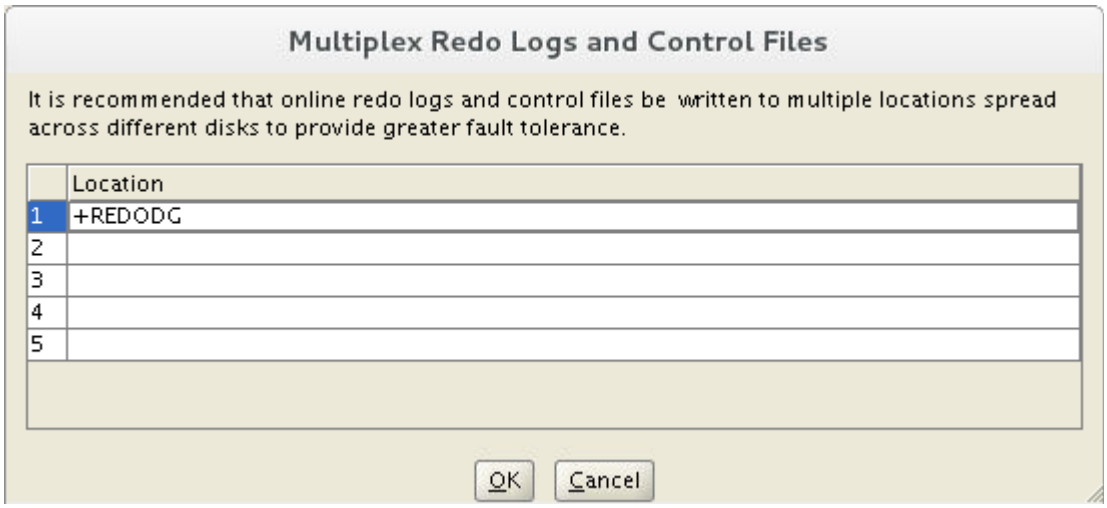

Figure 4.1.4.3: Multiplex Redo Logs and Control Files Window

12. Within the ***Database Options*** window, all the components should be greyed out. Click **Next**.

13. Within the ***Database Vault & Label Security*** window, this reference environment unchecks **Configure Database Vault** and **Configure Label Security**. These products provide a level of security with regards to access control decisions at the object level as done by Oracle Database Vault and access control decisions at the row level as done by Oracle Label Security. For more information on these two products, be sure to visit Oracle's Documentation about Oracle Database Vault[17] and Oracle Label Security[18].
14. Within the ***Initialization Parameters*** window under the **Memory** tab, select **Custom Settings** and enter the appropriate values for the *SGA* and *PGA* size. It is recommended that the Memory Management be set as **Automatic Shared Memory Management**. The values set for the reference environment with regards to *SGA* and *PGA* are the following, however, these values vary for every Oracle database environment.
    - SGA – 14460 Megabytes
    - PGA – 4820 Megabytes

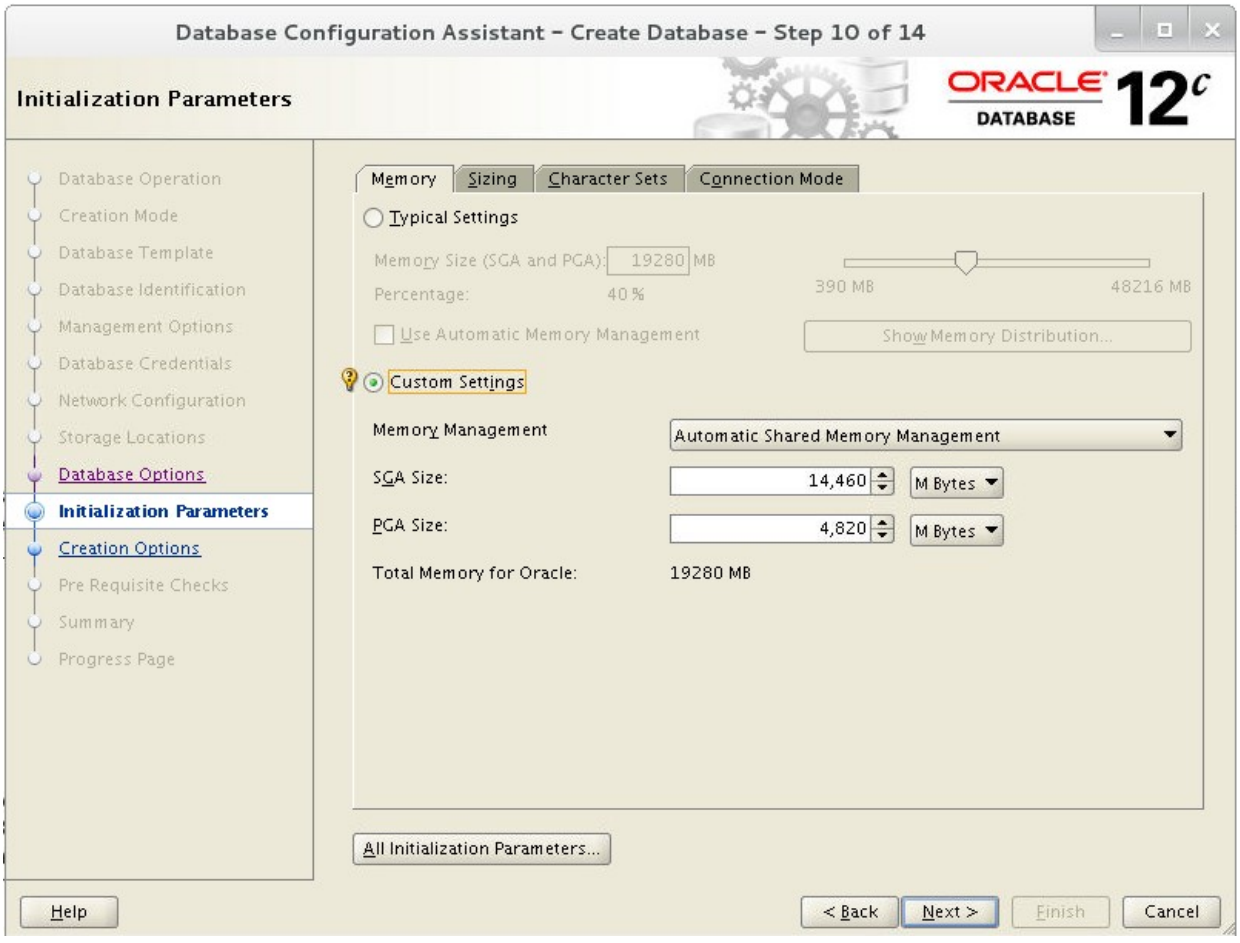

Figure 4.1.4.4: Initialization Parameters Window, Memory Tab

17 Oracle Documentation – Oracle Database Vault
http://docs.oracle.com/cd/E16655_01/server.121/e17608/dvintro.htm#DVADM001
18 Oracle Documentation – Oracle Label Security
http://docs.oracle.com/cd/E16655_01/network.121/e17730/intro.htm#OLSAG001

NOTE: It is recommended to modify the values of the SGA and PGA based on the Oracle database workload requirements.

15. Within the **Sizing** tab of the ***Initialization Parameters*** window, appropriately set the block size and maximum number of user processes that can be simultaneously connected to the database. For the purposes of this reference environment, the defaults are used.

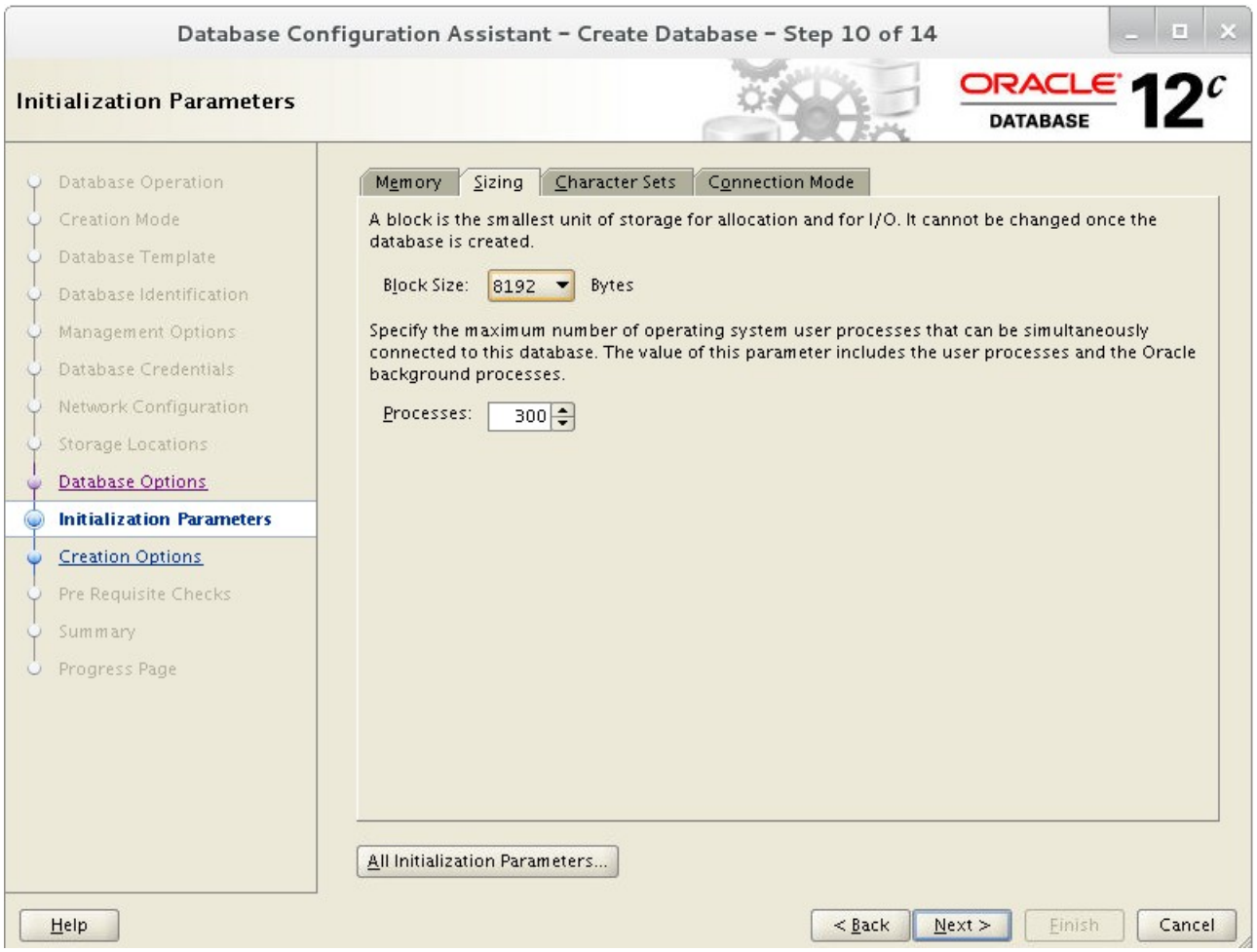

Figure 4.1.4.5: Initialization Parameters Window, Sizing Tab

NOTE: The **Character Sets** tab and **Connection Mode** tab within the ***Initialization Parameters*** window are not pictorially represented in this reference architecture, however, the default values are set.

16. Within the ***Creation Options*** window, ensure the **Create Database** box is checked and click **Next**.

17. Within the ***Summary*** window, review the **Create Database – Summary**, and click **Finish** to start the database creation.

4.1.5 Enabling HugePages

Transparent Huge Pages (THP) are implemented within Red Hat Enterprise Linux 6 to improve memory management by removing many of the difficulties of manually managing huge pages by dynamically allocating huge pages as needed. Red Hat Enterprise Linux 6, by default, uses *transparent* huge pages also known as *anonymous* huge pages. Unlike static huge pages, no additional configuration is needed to use them. Huge pages can boost application performance by increasing the chance a program will have quick access to a memory page. Unlike traditional huge pages, *transparent* huge pages can be swapped out (as smaller 4kB pages) when virtual memory clean up is required.

Unfortunately, Oracle Databases do not take advantage of *transparent* huge pages for interprocess communication. In fact, My Oracle Support [ID 1557478.1][19] states to disable THP due to unexpected performance issues or delays when THP is found to be enabled. To reap the benefit of huge pages for an Oracle database, it is required to allocate *static* huge pages and disable THP. Due to the complexity of properly configuring huge pages, it is recommended to copy the bash shell script found within **Appendix E Huge Pages Script** and run the script once the database is up and running. The reasoning behind allocating huge pages once the database is up and running is to provide a proper number of pages to handle the running shared memory segments. The steps are as follows:

1. Copy the bash script found within **Appendix E Huge Pages Script** and save it as *huge_pages_settings.sh*
2. As the `root` user, ensure the *huge_pages_settings.sh* is executable by running the following command:

```
# chmod +x huge_pages_settings.sh
```

3. As the `root` user, execute the *huge_pages_settings.sh* script as follows:

```
# ./huge_pages_settings.sh
Recommended setting within the kernel boot command line: hugepages = <value>
Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf:
oracle soft memlock  <value>
Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf:
oracle hard memlock <value>
```

4. Add the number of hugepages provided by the *huge_pages_settings.sh* script to the kernel boot command line within the */etc/grub.conf* and disable *transparent* huge pages persistently across reboots as follows:

```
title Red Hat Enterprise Linux (2.6.32-431.el6.x86_64)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/mapper/myvg-root
rd_NO_LUKS LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=auto rd_NO_DM rd_LVM_LV=myvg/root  KEYBOARDTYPE=pc
KEYTABLE=us rd_LVM_LV=myvg/swap rhgb quiet hugepages=<value-provided-by-
script> transparent_hugepage=never
        initrd /initramfs-2.6.32-431.el6.x86_64.img
```

19 ALERT: Disable Transparent HugePages on SLES11,RHEL6,OEL6 and UEK2 Kernels (DOC ID: 1557478.1)

NOTE: Allocating the number of huge pages within the kernel boot command line is the most reliable method due to memory not yet becoming fragmented.[20]

5. Oracle requires setting the soft and hard limits to *memlock.* Setting *memlock* allows the `oracle` user to lock a certain amount of memory from physical RAM that isn't swapped out. The value is expressed in kilobytes and is important from the Oracle perspective because it provides the `oracle` user permission to use huge pages. This value should be slightly larger than the largest SGA size of any of the Oracle Database instances installed in an Oracle environment. To set *memlock,* add within */etc/security/limits.d/99-grid-oracle-limits.conf* the following:

```
oracle soft memlock <value-provided-by-script>
oracle hard memlock <value-provided-by-script>
```

6. Reboot the system to ensure the huge pages setting takes effect properly.
7. Verify the value provided by the *huge_pages_settings.sh* matches the total number of huge pages available on the system with the following command:

```
# cat /proc/meminfo | grep -i hugepages_total
HugePages_Total:       <value-provided-by-script>
```

8. Verify the current status of the *transparent* huge pages is set to *NEVER* via the following command:

```
# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
```

NOTE: Starting with Oracle Database version 11.2.0.2, the initialization parameter "USE_LARGE_PAGES" was introduced to allocate huge pages on a per database use case. The default value for Oracle Database 12.0.1.0 is set to true. **Section 5 Logging into the Oracle Container Database 12c Release 1 (12.1)** shows how to set "USE_LARGE_PAGES" to the recommended value of **only** to ensure huge pages are always used upon Oracle database startup. For more information on the parameter and its value refer to My Oracle Support[21].

NOTE: Huge pages is not compatible with Automatic Memory Management (AMM).

20 https://www.kernel.org/doc/Documentation/vm/hugetlbpage.txt
21 USE_LARGE_PAGES To Enable HugePages [ID 1392497.1]

5 Logging into the Oracle Container Database 12c Release 1 (12.1)

This section focuses on ensuring once the Oracle Database 12c Release 1 (12.1) deployment is complete, the `oracle` user can successfully log into the Oracle container database (CDB), and ensure the Oracle database is using the allocated huge pages. The following steps provide the details.

As the `oracle` user,

1. Set the environment variable for *ORACLE_HOME* with the location of your Oracle Database 12c Release 1 (12.1) home. This reference environment sets ORACLE_HOME to */u01/app/oracle/product/12.1.0/dbhome_1*

```
# export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
# echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome_1
```

   NOTE: As a precaution, ensure not to include a trailing forward slash (*/*) when exporting the *ORACLE_HOME.*

2. Set the Oracle System ID (*ORACLE_SID*) used to identify the CDB database.

```
# export ORACLE_SID=cdb
# echo $ORACLE_SID
cdb
```

3. Invoke the **`sqlplus`** binary to log into the Oracle instance as a *sysdba*.

```
# $ORACLE_HOME/bin/sqlplus / as sysdba;

SQL*Plus: Release 12.1.0.1.0 Production on Thu Jan 16 16:17:08 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics and Real Application Testing options
```

4. Verify the current value of the Oracle parameter ***use_large_pages***

```
SQL> show parameter use_large_pages;

NAME                           TYPE        VALUE
------------------------------ ----------- -----------
use_large_pages                string      TRUE
```

CAUTION: Ensure there is enough physical RAM on the system to place the entire SGA in large pages, otherwise the Oracle database instance will not start. If there is not enough physical RAM on the system to place the entire SGA into large pages, leave the default setting of true and ignore the remaining steps in this section.

5. Set the value of the Oracle parameter ***use_large_pages*** to the value of ***only***.

```
SQL> alter system set use_large_pages=only scope=spfile sid='*';

System altered.
```

6. Shutdown the Oracle database instance and restart the Oracle database instance.

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup;
ORACLE instance started.

Total System Global Area 2.0176E+10 bytes
Fixed Size             3478432 bytes
Variable Size           2952793184 bytes
Database Buffers       1.7180E+10 bytes
Redo Buffers              40247296 bytes
Database mounted.
Database opened.
```

7. Verify the current value of the Oracle parameter ***use_large_pages*** is now set to ***only***.

```
SQL> show parameter use_large_pages;

NAME                           TYPE        VALUE
------------------------------ ----------- -----------
use_large_pages                string      ONLY
```

8. Open the container database's alert log, named alert_*<name-of-cdb>.log*, located under the *$ORACLE_BASE/diag/rdbms/<name-of-cdb>/<name-of-cdb>/trace/* using a text editor, such as **vi**, and search for the following snippet to ensure that the System Global Area (SGA) is 100% in large pages.

$ORACLE_BASE/diag/rdbms/<name-of-cdb>/<name-of-cdb>/trace/alert_<name-of-cdb>.log

```
********************** Large Pages Information ******************
Thu Jan 16 16:55:07 2014
Per process system memlock (soft) limit = UNLIMITED
Thu Jan 16 16:55:07 2014

Total System Global Area in large pages = 19 GB (100%)
Thu Jan 16 16:55:07 2014

Large pages used by this instance: 9665 (19 GB)
Thu Jan 16 16:55:07 2014
Large pages unused system wide = 3 (6144 KB)
Thu Jan 16 16:55:07 2014
Large pages configured system wide = 9668 (19 GB)
Thu Jan 16 16:55:07 2014
Large page size = 2048 KB
Thu Jan 16 16:55:07 2014
*****************************************************************
```

NOTE: This reference environment's SGA size is set to 19 GB, however, this value will vary depending on the value provided when creating an Oracle database using **dbca**.

6 Common Tasks when Managing Container Database (CDB) and Pluggable Databases (PDB)

This section describes tasks that are commonly used when dealing with a CDB and PDBs. The tasks covered within this section are as follows:

- Connect to a CDB
- Connect to a PDB
- Managing a CDB
- Managing a PDB
- Location of Data files in a CDB & PDB

6.1 Connect to a CDB

As the `oracle` user,

1. Set the environment variable for *ORACLE_HOME* with the location of your Oracle Database 12c Release 1 (12.1) home. This reference environment sets ORACLE_HOME to */u01/app/oracle/product/12.1.0/dbhome_1*

```
# export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
# echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome_1
```

   NOTE: As a precaution, ensure not to include a trailing forward slash (*/*) when exporting the ORACLE_HOME.

2. Set the Oracle System ID (*ORACLE_SID*) used to identify the CDB database.

```
# export ORACLE_SID=cdb
# echo $ORACLE_SID
cdb
```

3. Invoke the **`sqlplus`** binary to log into the Oracle instance as a *sysdba*.

```
# $ORACLE_HOME/bin/sqlplus / as sysdba;

SQL*Plus: Release 12.1.0.1.0 Production on Thu Jan 16 16:17:08 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
```

```
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics and Real Application Testing options
```

4. Once connected, verify that the instance is connected to the root container, *CDB$ROOT* with a *CON_ID* is 1.

```
SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT

SQL> show con_id

CON_ID
------------------------------
1
```

NOTE: The *CDB$ROOT* connection ID is always set to one.

5. To list all the available services and PDBs within the CDB:

```
SQL> select name, con_id from v$active_services;

NAME                                                  CON_ID
---------------------------------------------------- --------------
pdb2                                                   4
pdb1                                                   3
cdbXDB                                                 1
cdb                                                    1
SYS$BACKGROUND                                         1
SYS$USERS                                              1

6 rows selected.
```

6.2 Connect to a PDB

The syntax to connect to a PDB varies depending on whether or not there is an entry within the *tnsnames.ora* file for the PDB.

Without an entry to the tnsnames.ora file, the syntax to connect to a PDB named *pdb1* is as follows:

```
# $ORACLE_HOME/bin/sqlplus sys/<password>@localhost:1521/pdb1 as sysdba;
```

NOTE: The value 1521, represents the Oracle Listener port.

With an entry to the *tnsnames.ora* file, the syntax to connect to a PDB named *pdb1* is as follows:

```
# $ORACLE_HOME/bin/sqlplus sys/<password>@PDB1 as sysdba;
```

A snippet of the entry found within the *tnsnames.ora* file is displayed below:

/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsnames.ora

```
PDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = db-oracle-
12c.cloud.lab.eng.bos.redhat.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb1)
    )
  )
```

6.3 Managing a CDB

The process of starting and shutting down a CDB database is similar to the steps done in previous Oracle database versions for traditional databases. The key difference is to verify that the connection is to the root container prior to shutting down or starting up the Oracle database.

1. Connect to the CDB database as a *SYSDBA* using **`sqlplus`**. The steps are the same as shown in **Section 6.1 Connect to a CDB** steps one through three.
2. Once connected, verify the instance is the root container *CDB$ROOT* via the command:

```
SQL> show con_name;

CON_NAME
------------------------------
CDB$ROOT
```

3. To shutdown the Oracle CDB database use the following command:

```
SQL> shutdown immediate;
```

4. To start the Oracle CDB database use the following command:

```
SQL> startup;
```

The startup command starts the instance, mounts the control files, and then opens the root container.

6.4 Managing a PDB

This section focuses on verifying the *OPEN_MODE* of a PDB, how to open and close a specific PDB, and how to open and close all PDBs within a CDB.

1. To verify the open_mode status of all the PDBs, while logged in as a *SYSDBA* in the CDB, use the following command:

```
SQL> select name, open_mode from v$pdbs;

NAME                      OPEN_MODE
------------------------------ ----------
PDB$SEED                  READ ONLY
PDB1                      MOUNTED
PDB2                      MOUNTED
```

2. When a PDB is closed, the *OPEN_MODE* is set to *MOUNTED.* To open a PDB and verify the new *OPEN_MODE* of *READ WRITE*, run the following SQL syntax while logged in as a *SYSDBA* in the CDB:

```
SQL> alter pluggable database pdb1 open;

Pluggable database altered.

SQL> select name, open_mode from v$pdbs;

NAME                      OPEN_MODE
------------------------------ ----------
PDB$SEED                  READ ONLY
PDB1                      READ WRITE
PDB2                      MOUNTED
```

3. To open all the PDBs connected to a CDB and verify the new *OPEN_MODE* of *READ WRITE*, run the following SQL syntax while logged in as a *SYSDBA* in the CDB:

```
SQL> alter pluggable database all open;

Pluggable database altered.

SQL> select name, open_mode from v$pdbs;

NAME                      OPEN_MODE
------------------------------ ----------
PDB$SEED                  READ ONLY
PDB1                      READ WRITE
PDB2                      READ WRITE
```

4. To drop a particular PDB i.e. *pdb2*, and its data files, execute the following SQL syntax while logged in as a *SYSDBA* in the CDB:

```
SQL > alter pluggable database pdb2 close immediate;

Pluggable database altered.

SQL > drop pluggable database pdb2 including datafiles;

Pluggable database dropped.
```

5. To verify if the pluggable database with the name *pdb2* has been dropped:

```
SQL> select name, open_mode from v$pdbs;
NAME                    OPEN_MODE
------------------------------ ----------
PDB$SEED                READ ONLY
PDB1                    READ WRITE
```

6.5 Location of Data files of PDBs & CDB

The following section shows how to identify tablespace names, data files associated with the CDB and PDBs, including their temporary files.

1. Connect to the CDB database as a *SYSDBA* using **sqlplus**. The steps are the same as shown in **Section 6.1 Connect to a CDB** steps one through three.
2. To identify the tablespaces associated with the CDB or any of the PDBs installed, use the following syntax where the *con_id* varies upon the database chosen. The example below uses the *con_id* of 1 to show the CDB tablespaces.

```
SQL > select tablespace_name, con_id from cdb_tablespaces where con_id =
1;

TABLESPACE_NAME                 CON_ID
------------------------------ ----------
SYSTEM                          1
SYSAUX                          1
UNDOTBS1                        1
TEMP                            1
USERS                           1
```

3. To locate the data files from the CDB or PDBs installed, use the following syntax where the *con_id* varies upon the database chosen. The example below uses the *con_id* of 1 to show the CDB data file locations.

```
SQL> select file_name, con_id from cdb_data_files where con_id=1;

FILE_NAME                                                CON_ID
-------------------------------------------------- -----------
+DATA/CDB/DATAFILE/system.270.836232071                       1
+DATA/CDB/DATAFILE/sysaux.273.836232077                       1
+DATA/CDB/DATAFILE/undotbs1.262.836232081                     1
```

```
FILE_NAME                                                  CON_ID
---------------------------------------------------------- -----------
+DATA/CDB/DATAFILE/users.275.836232097                              1
```

4. To locate the temporary files from the CDB or PDBs installed, use the following syntax where the *con_id* varies upon the database chosen. The example below uses the *con_id* of 1 to show the CDB data file locations.

```
SQL> select file_name, con_id from cdb_temp_files where con_id =1 ;

FILE_NAME                                                  CON_ID
---------------------------------------------------------- -------------
+DATA/CDB/TEMPFILE/temp.278.836232081                               1
```

7 Conclusion

Red Hat Enterprise Linux 6 provides an excellent foundation for database deployments with demonstrated stability, scalability, and performance. With the support for Oracle Database 12c Release 1 (12.1) on Red Hat Enterprise Linux 6, customers can increasingly look to deploy Oracle Databases in advanced configurations.

The steps and procedures described in this reference architecture should provide system, database, and storage administrators the blueprint required to create a robust and performing solution based on Oracle Databases. Administrators can reference this document to simplify and optimize the deployment process and employ the latest best practices for configuring Red Hat technologies while implementing the following tasks:

- Deploying Oracle Grid Infrastructure 12c Release 1 (12.1.0.1.0)
- Deploying Oracle Database Software 12c Release 1 (12.1.0.1.0)
- Deploying an Oracle Database 12c Release 1 (12.1.0.1.1) with shared SAN disks
- Using Oracle ASM disks with *udev* rules or with Oracle ASMLib
- Securing the Oracle Database 12c environment with *SELinux*

For any questions or concerns, please email refarch-feedback@redhat.com and ensure to visit the Red Hat Reference Architecture page at http://www.redhat.com/resourcelibrary/reference-architectures/ to find out about all of our Red Hat solution offerings.

Appendix A: Revision History

Revision 1.0	Friday February 28, 2014	Roger Lopez
Initial Release		

Appendix B: Contributors

1. John Herr, content (Subscription Manager)
2. Balaji Jayavelu, content reviewer
3. Tom Coughlan, content reviewer
4. John Boero, content reviewer
5. Yan Fisher, content reviewer

Appendix C: DM Multipath Configuration File

```
# This is a basic configuration file with some examples, for device mapper
# multipath.
# For a complete list of the default configuration values, see
# /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.defaults
# For a list of configuration options with descriptions, see
# /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.annotated
#
# REMEMBER: After updating multipath.conf, you must run
#
# service multipathd reload
#
# for the changes to take effect in multipathd

## By default, devices with vendor = "IBM" and product = "S/390.*" are
## blacklisted. To enable mulitpathing on these devies, uncomment the
## following lines.
#blacklist_exceptions {
#       device {
#               vendor  "IBM"
#               product "S/390.*"
#       }
#}

## Use user friendly names, instead of using WWIDs as names.
defaults {
        user_friendly_names yes
}
##
## Here is an example of how to configure some standard options.
##
#
defaults {
        udev_dir                /dev
        polling_interval        10
        path_selector           "round-robin 0"
        path_grouping_policy    multibus
        getuid_callout          "/lib/udev/scsi_id --whitelisted
--device=/dev/%n"
        prio                    alua
        path_checker            readsector0
        rr_min_io               100
        max_fds                 8192
        rr_weight               priorities
        failback                immediate
        no_path_retry           fail
        user_friendly_names     yes
}
##
## The wwid line in the following blacklist section is shown as an example
```

```
## of how to blacklist devices by wwid.  The 2 devnode lines are the
## compiled in default blacklist. If you want to blacklist entire types
## of devices, such as all scsi devices, you should use a devnode line.
## However, if you want to blacklist specific devices, you should use
## a wwid line.  Since there is no guarantee that a specific device will
## not change names on reboot (from /dev/sda to /dev/sdb for example)
## devnode lines are not recommended for blacklisting specific devices.
##
blacklist {
       wwid 3600508b1001030353434363646301200
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
}
multipaths {
        multipath {
                wwid                    3600c0ff000d7e7a899d8515101000000
                alias                   db1
        }
        multipath {
                wwid                    3600c0ff000dabfe5a7d8515101000000
                alias                   db2
        }
        multipath {
                wwid                    3600c0ff000d7e7a8dbd8515101000000
                alias                   fra
        }
        multipath {
                wwid                    3600c0ff000dabfe5f4d8515101000000
                alias                   redo
        }
}
#devices {
#       device {
#               vendor                  "COMPAQ  "
#               product                 "HSV110 (C)COMPAQ"
#               path_grouping_policy    multibus
#               getuid_callout          "/lib/udev/scsi_id --whitelisted
--device=/dev/%n"
#               path_checker            readsector0
#               path_selector           "round-robin 0"
#               hardware_handler        "0"
#               failback                15
#               rr_weight               priorities
#               no_path_retry           queue
#       }
#       device {
#               vendor                  "COMPAQ  "
#               product                 "MSA1000         "
#               path_grouping_policy    multibus
#       }
#}
```

Appendix D: Iptables Configuration File

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp -s 10.16.142.54 --dport 1521 -j
ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp -s 10.16.142.54 --dport 5500 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Appendix E: Huge Pages Script

The following huge pages script is from Tuning Red Hat Enterprise Linux For Oracle & Oracle RAC by Scott Crot, Sr. Consultant, Red Hat[22] and modified to include the values Oracle's soft and hard memlock.

```
#!/bin/bash
KERN=`uname -r | awk -F. '{ printf("%d.%d\n",$1,$2); }'`
# Find out the HugePage size
HPG_SZ=`grep Hugepagesize /proc/meminfo | awk '{print $2}'`
# Start from 1 pages to be on the safe side and guarantee 1 free HugePage
NUM_PG=1
# Cumulative number of pages required to handle the running shared memory
segments
for SEG_BYTES in `ipcs -m | awk '{print $5}' | grep "[0-9][0-9]*"`
do
MIN_PG=`echo "$SEG_BYTES/($HPG_SZ*1024)" | bc -q`
if [ $MIN_PG -gt 0 ]; then
NUM_PG=`echo "$NUM_PG+$MIN_PG+1" | bc -q`
fi
done
# Finish with results
case $KERN in
'2.4') HUGETLB_POOL=`echo "$NUM_PG*$HPG_SZ/1024" | bc -q`;
echo "Recommended setting: vm.hugetlb_pool = $HUGETLB_POOL" ;;
'2.6') MEM_LOCK=`echo "$NUM_PG*$HPG_SZ" | bc -q`;
echo "Recommended setting within the kernel boot command line: hugepages =
$NUM_PG"
echo "Recommended setting within /etc/security/limits.d/99-grid-oracle-
limits.conf: oracle soft memlock $MEM_LOCK"
echo "Recommended setting within /etc/security/limits.d/99-grid-oracle-
limits.conf: oracle hard memlock $MEM_LOCK"  ;;
*) echo "Unrecognized kernel version $KERN. Exiting." ;;
esac
```

22 Tuning Red Hat Enterprise Linux For Oracle & Oracle RAC by Scott Crot, Sr. Consultant, Red Hat, http://www.redhat.com/promo/summit/2010/presentations/summit/decoding-the-code/fri/scott-945-tuning/summit_jbw_2010_presentation.pdf

Appendix F: Oracle Database Package Requirements Text File

```
cloog-ppl
compat-libcap1
compat-libstdc++-33
cpp
gcc
gcc-c++
glibc-devel
glibc-headers
kernel-headers
ksh
libXmu
libXt
libXv
libXxf86dga
libXxf86misc
libXxf86vm
libaio-devel
libdmx
libstdc++-devel
mpfr
make
ppl
xorg-x11-utils
xorg-x11-xauth
```

Appendix G: Kernel Parameters

```
vm.swappiness = 0
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
kernel.shmmax = 4398046511104
kernel.shmall = 1073741824
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
# fs.file-max needs to be set to at least 6815744 for Oracle Installation.
fs.file-max = 6815744
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

Appendix H: Limits Configuration File (99-grid-oracle-limits.conf)

```
oracle soft nproc 16384 #Oracle Bug 15971421
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
oracle soft memlock <value-provided-by-script>
oracle hard memlock <value-provided-by-script>

grid soft nproc 16384 #Oracle Bug 15971421
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
```

Appendix I: 99-oracle-asmdevices.rules

```
KERNEL=="dm-*",ENV{DM_UUID}=="<enter-value-according-to-your-
environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
KERNEL=="dm-*",ENV{DM_UUID}=="<enter-value-according-to-your-
environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
KERNEL=="dm-*",ENV{DM_UUID}=="<enter-value-according-to-your-
environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
KERNEL=="dm-*",ENV{DM_UUID}=="<enter-value-according-to-your-
environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
```

Appendix J: Sample Kickstart File

```
# Red Hat | Oracle Solutions Kickstart Script


install
url --url=<place-distro-url-here>
lang en_US.UTF-8
keyboard us
network --onboot yes --device em1 --mtu=1500 --bootproto dhcp
rootpw  <password-for-system>
# Reboot after installation
reboot
authconfig --enablemd5 --enableshadow
selinux --enforcing
timezone America/New_York
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb
quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all

volgroup myvg --pesize=32768 pv.008002
logvol /home --fstype=ext4 --name=home --vgname=myvg --size=8192
logvol / --fstype=ext4 --name=root --vgname=myvg --size=15360
logvol swap --name=swap --vgname=myvg --size=16400
logvol /tmp --fstype=ext4 --name=tmp --vgname=myvg --size=4096
logvol /u01 --fstype=ext4 --name=u01 --vgname=myvg --size=51200
logvol /usr --fstype=ext4 --name=usr --vgname=myvg --size=5120
logvol /var --fstype=ext4 --name=var --vgname=myvg --size=8192

part /boot --fstype=ext4 --size=256
part pv.008002 --grow –size=1000

%packages
@Base
@Core
```

Appendix K: Configuration Files

All configuration files can be downloaded from the Red Hat customer portal[23]. A listing of all the files and a brief description can be seen on the **Table 7.1: Configuration Files.** Some of the configuration files require input with the proper information pertaining to your environment.

Files	Description
req-rpm.txt	The required RPMs to deploy Oracle.
huge_page_settings.sh	Script that provides the proper hugepage values to set.
multipath.conf	Device Mapper Multipath configuration file.
sysctl.conf	Configuration file for the kernel parameters
99-grid-oracle-limits.conf	Configuration file to set limits for a user.
99-oracle-asmdevices.rules	*udev* configuration file for Oracle ASM disks
iptables	iptables configuration
bonding.conf	*/etc/modprobe.d/* bonding configuration file
oracle-grid.sh	Shell script used to set user limits
sample-ks.cfg	Sample Kickstart File
enterprise-storage-no-thp	Directory of the custom tuned profile enterprise-sorage-no-thp
CHANGELOG	Listing of the latest changes made to the .tar.gz file

Table 7.1: Configuration Files

23 https://access.redhat.com/site/node/725843/40/1

Appendix L: Troubleshooting ORA-* Errors

This section focuses on using the command line tool, Automatic Diagnostic Repository Command Interpreter (*ADRCI*), to troubleshoot Oracle database related errors. *ADRCI* was introduced in Oracle Database 11g in order to help users diagnose errors within their Oracle database environments and provide health reports if an issue should arise. The following example shows how one could troubleshoot an Oracle database instance error using the *ADRCI* tool.

NOTE: The following steps are intended to produce an ORA-07445 error that can be troubleshooted using the *ADRCI* tool. Do not attempt on a Oracle Database Production environment. The following is for demonstration purposes only and intended only to show how to troubleshoot ORA-* related errors using the *ADRCI* tool.

1. In order to create an ORA-07445 error, an essential Oracle process will be killed via the following commands:

```
# ps -A --format pid,args | grep ora_dbrm | grep -v grep
7811 ora_dbrm_cdb
# kill -SEGV 7811
```

2. Export the ORACLE_HOME via the command:

```
# export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
```

3. Start the *ADRCI* command tool via the command:

```
# $ORACLE_HOME/bin/adrci

ADRCI: Release 12.1.0.1.0 - Production on Mon Jan 20 15:19:24 2014

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights
reserved.

ADR base = "/u01/app/oracle"
adrci>
```

4. At the *ADRCI* prompt, show Oracle Home's available via the command:

```
adcri> show home
ADR Homes:
diag/rdbms/cdb/cdb
```

   NOTE: If more than one Oracle Home is available, one must specify a particular Oracle Database Home. An example on how to set to a particular Oracle Database Home is as follows:

```
adcri> set home diag/rdbms/cdb/cdb
```

5. At the *ADRCI* prompt, run the following command to see the last 50 entries in the alert log:

```
adrci> show alert -tail -f
[ ... Output Abbreviated ... ]
Exception [type: SIGSEGV, unknown code] [ADDR:0xD431000076FD]
[PC:0x3199AEB22A, semtimedop()+10] [exception issued by pid: 30461, uid:
54321] [flags: 0x0, count: 1]
Errors in file /u01/app/oracle/diag/rdbms/cdb/cdb/trace/cdb_dbrm_7811.trc
(incident=9673):
```

```
ORA-07445: exception encountered: core dump [semtimedop()+10] [SIGSEGV]
[ADDR:0xD431000076FD] [PC:0x3199AEB22A] [unknown code] []
Incident details in:
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_i96
73.trc
Use ADRCI or Support Workbench to package the incident.
See Note 411.1 at My Oracle Support for error and packaging details.
```

NOTE: In this particular case, we are looking for an ORA-07445 within the alert log as seen above. However, this step is just to confirm what is seen in the future *ADRCI* steps. To exit the alert log, execute CTRL+C.

6. Within the *ADRCI*, there are two key terms to be aware of, problem and incident. An incident is a particular time when a problem occurred. For example, it is possible for an Oracle process to crash at different times with the same ORA-07445. The multiple occurences of the crash are incidents, while the problem is still the ORA-07445 error. In order to view the problem, the following *ADRCI* command needs to be run.

```
adrci> show problem
ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb:
*************************************************************************
PROBLEM_ID       PROBLEM_KEY      LAST_INCIDENT          LASTINC_TIME
-------------------------------------------------------------------------
1    ORA 7445 [semtimedop]     9673      2014-01-20 15:20:50.273000
-05:00
1 row fetched
```

7. In order to view how many incidents, the following *ADRCI* command must be run. In this example, I only have one incident in which the ORA-07445 problem occurred.

```
adrci> show incident
ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb:
*************************************************************************
INCIDENT_ID      PROBLEM_KEY                       CREATE_TIME
-------------------------------------------------------------------------
9673   ORA 7445 [semtimedop]  2014-01-20 15:20:50.273000 -05:00
1 rows fetched
```

8. In order to view the incident in more detail, run the following command:

```
adrci> show incident -mode detail -p "incident_id=9673"
ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb:
*************************************************************************


*********************************************************
INCIDENT INFO RECORD 1
*********************************************************
  INCIDENT_ID                   9673
  STATUS                        ready
  CREATE_TIME                   2014-01-20 15:20:50.273000 -05:00
  PROBLEM_ID                    1
  CLOSE_TIME                    <NULL>
  FLOOD_CONTROLLED              none
```

```
   ERROR_FACILITY                 ORA
   ERROR_NUMBER                   7445
   ERROR_ARG1                     semtimedop
   ERROR_ARG2                     SIGSEGV
   ERROR_ARG3                     ADDR:0xD431000076FD
  [ ... Output Abbreviated ... ]
  OWNER_ID                       1
  INCIDENT_FILE
/u01/app/oracle/diag/rdbms/cdb/cdb/trace/cdb_dbrm_7811.trc
  OWNER_ID                       1
INCIDENT_FILE
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_i9673.trc
1 rows fetched
```

NOTE: The two parameters of importance here are the PROBLEM_ID and INCIDENT_FILE.

9. When looking at the incident in further detail, the following incident file can be examined further via the following command:

```
adrci> show trace
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_i96
73.trc
/
u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_i967
3.trc
------------------------------------------------------------
LEVEL PAYLOAD
--------------------------------------------------------------------------
--------------------------------------------------------------------------
---
Dump file
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_i96
73.trc
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit
Production With the Partitioning, Automatic Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1
System name:      Linux
Node name:        db-oracle-12c.cloud.lab.eng.bos.redhat.com
Release:  2.6.32-431.el6.x86_64
Version:  #1 SMP Sun Nov 10 22:19:54 EST 2013
Machine:  x86_64
Instance name: cdb
Redo thread mounted by this instance: 1
Oracle process number: 9
Unix process pid: 7811, image: ?


*** 2014-01-20 15:20:50.293
*** SESSION ID:(271.1) 2014-01-20 15:20:50.293
*** CLIENT ID:() 2014-01-20 15:20:50.293
*** SERVICE NAME:(SYS$BACKGROUND) 2014-01-20 15:20:50.293
*** MODULE NAME:() 2014-01-20 15:20:50.293
*** ACTION NAME:() 2014-01-20 15:20:50.293
*** CONTAINER ID:(1) 2014-01-20 15:20:50.293
```

```
Dump continued from file:
/u01/app/oracle/diag/rdbms/cdb/cdb/trace/cdb_dbrm_7811.trc
1>     ***** Error Stack *****
ORA-07445: exception encountered: core dump [semtimedop()+10] [SIGSEGV]
[ADDR:0xD431000076FD] [PC:0x3199AEB22A] [unknown code] []
1<     ***** Error Stack *****
1>     ***** Dump for incident 9673 (ORA 7445 [semtimedop]) *****
2>      ***** Beginning of Customized Incident Dump(s) *****
Dumping swap information
Memory (Avail / Total) = 42169.27M / 64420.08M
Swap (Avail / Total) = 16415.99M /  16415.99M
Exception [type: SIGSEGV, unknown code] [ADDR:0xD431000076FD]
[PC:0x3199AEB22A, semtimedop()+10] [exception issued by pid: 30461, uid:
54321]
[flags: 0x0, count: 1]
Registers:
%rax: 0xfffffffffffffffc %rbx: 0x00000000000a000b %rcx:
0xffffffffffffffff
  [ ... Output Abbreviated ... ]
```

10. While this concludes how to examine trace files that pertain to a particular ORA error using *ADRCI*; if the issue cannot be solved by the end user, the *ADRCI* provides the *Incident Packaging Service* (IPS) tool to ZIP the necessary trace files based on the problem. It can then be sent to support for further debugging. To create the appropriate ZIP file, use the following commands:

```
 adrci> ips create package problem 1 correlate all
Created package 1 based on problem id 1, correlation level all
```

NOTE: Problem 1 is the Problem_ID found in step 6.

```
 adrci> ips generate package 1 in “/home/oracle”
 Generated package 1 in file
/home/oracle/ORA7445se_20140120153727_COM_1.zip, mode complete
```

NOTE: Package 1 is the package ID captured from the ips create output command.

For more information about *ADRCI*, please visit the
http://docs.oracle.com/cd/E16655_01/server.121/e17639/adrci.htm#SUTIL700

Appendix M: References

TECH: Unix Semaphores and Shared Memory Explained [ID 15566.1]
http://docs.oracle.com/cd/E11882_01/install.112/e24321/pre_install.htm

Oracle Grid Infrastructure, Oracle Documentation
http://docs.oracle.com/cd/E16655_01/install.121/e17888/concepts.htm#CWLIN506

Tuning Red Hat Enterprise Linux For Oracle & Oracle RAC by Scott Crot, Sr. Consultant, Red Hat
http://www.redhat.com/promo/summit/2010/presentations/summit/decoding-the-code/fri/scott-945-tuning/summit_jbw_2010_presentation.pdf

Linux OS Installation with Reduced Set of Packages for Running Oracle Database Server [ID 728346.1]
https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=290805959329203&id=728346.1&_adf.ctrl-state=13886txzey_67

USE_LARGE_PAGES To Enable HugePages [ID 1392497.1]
https://support.oracle.com/epmos/faces/Dashboard?_adf.ctrl-state=nvtwimbst_252

Large Pages Information in the Alert Log [ID 1392543.1]
https://support.oracle.com/epmos/faces/ui/km/SearchDocDisplay.jspx?_adf.ctrl-state=nvtwimbst_226

Tuning Virtual Memory
https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/s-memory-tunables.html

Maximum SHMMAX values for Linux x86 and x86-64 [ID 567506.1]
https://support.oracle.com/epmos/faces/ui/km/SearchDocDisplay.jspx?_adf.ctrl-state=yp0o5bwk6_4

About the Oracle Database Fault Diagnosability Infrastructure
http://docs.oracle.com/cd/E11882_01/server.112/e25494/diag001.htm

Pro Oracle Database 11g RAC on Linux – Installation, Administration, Performance by Steve Shaw and Martin Bach
http://www.amazon.com/Pro-Oracle-Database-11g-Linux-ebook/dp/B004VJ472I/ref=sr_1_1?ie=UTF8&qid=1389977723&sr=8-1&keywords=pro+oracle+11g+rac