Using Oracle Clusterware to Protect Single Instance Oracle Database 11.1

An Oracle White Paper

Version 1.14

Introduction

This paper updates the existing paper 'Using Oracle Clusterware to Protect a Single Instance Oracle Database'

This paper alters the way in which Oracle Clusterware must protect the single instance database. The database is no longer being treated as a single resource. It must be failed over together with any other relevant resources. To achieve this the concept of a 'Resource Group' for Oracle Clusterware is explained in this document. A resource group acts as a container for the managed resources. Oracle Clusterware starts all the 'contained' resources on the same node and they are all failed over as a consolidated group, dependencies exist between the various resources.

A number of dependencies are created when the individual resources are registered with Oracle Clusterware. This guarantees that the order in which Oracle Clusterware starts these processes is correct.

One key difference between the original scripts provided for Single Instance protection and these scripts is that they have been made generic. There is no longer any requirement to modify the scripts. Instead, as the resources are registered with Oracle Clusterware, extra parameters are provided as part of the crs_profile command line. These parameters are stored inside the Oracle Cluster Registry (OCR) and are specific to the individual resources. Oracle Clusterware then passes those parameters on to the action scripts when invoked.
The listener script requires two parameters:

- The location of the listener ORACLE_HOME
- The name of the listener.

The database script requires two parameters:

- The location of the database ORACLE_HOME – which can be the same home as the listener
- The name of the instance.

Example Scenario

Starting Case: No Oracle Software Installed

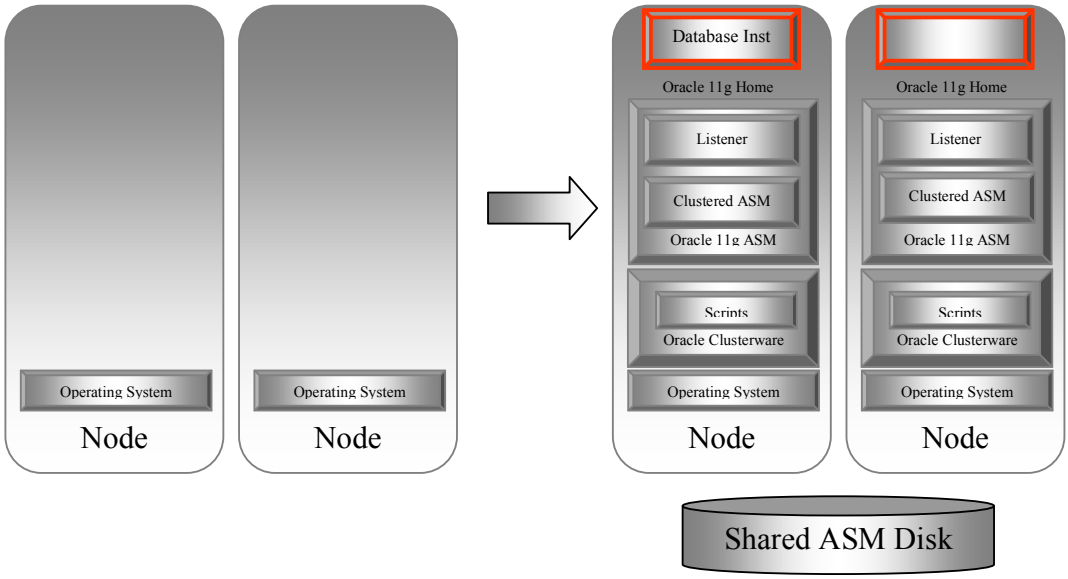


Figure 1

In this configuration the starting case is a clean cluster. The end case will be a 'cold failover' Oracle instance, The database files will be managed by a clustered ASM installation. In this case Oracle Clusterware is providing protection for the Single Instance database.

Pre Configuration Steps

Install Oracle Clusterware

Install Oracle Clusterware onto both nodes in the cluster. This paper assumes that this is /opt/oracle/product/11.1/crs and an environment variable CRS_HOME points to this.

Install a Database Home

Install a new home across both nodes in the cluster for the single instance database. These notes assume that this is /opt/oracle/product/11.1/si. If you would like to 'rolling patch' the database homes then it is suggested that you install single local copies of the database home rather than clustered local copies.

Install a Clustered ASM home and create an ASM instance [optional step]

If you choose to locate your database files inside ASM then you should install an ASM home across the nodes and create an ASM instance on each.

Create a new single Instance database

On Node1 create a new single instance database placing all the database files inside a clustered ASM database filesystem.

Allocate a new IP address

This IP address should be from the same subnet as the node public address. These notes assume that the address resolves as nodeappvip.

Multiple Active / Passive databases

If you have multiple databases you wish to protect on the same cluster you can either:

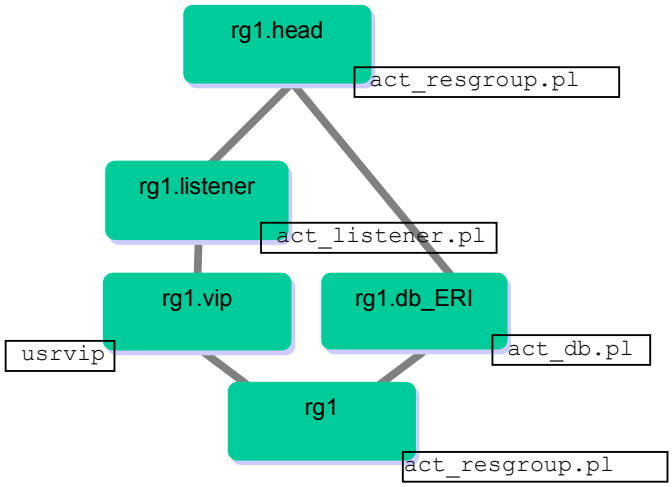
- Place the instance in the same resource group – This is simple to do but has the side effect that if one instance fails over to the other node then it will bring with it the other instance
- Create a new resource group and associated resources for each instance. Setup is more complicated but provides flexibility for resource group location.

Oracle Clusterware Resources

5 new Oracle Clusterware resources will be created

Resource Name	Resource Description
rg1	This resource acts as a container or a resource group for the other resources
rg1.vip	This resource is a new Application VIP. It allows clients to locate the instance
rg1.listener	This a new listener, it sits behind the Application VIP resource
rg1.db_$SID	This is the database instance resource
rg1.head	This is a top level container. It controls the startup order of resources of the same level (the agent, the listener & the database resource)

Schematic of resource dependencies, resources and action scripts.



rg1 resource

rg1 is a resource group, it acts as a container for all the other resources used in the Active/Passive database on both nodes as oracle

copy the supplied 'act_resgroup.pl script to $CRS_HOME/crs/public/
ensure that oracle has execute privileges on the scripts. Note if you find that the public directory does not exist check the path you are using carefully. If the Oracle Clusterware home is /opt/oracle/product/11.1/crs then the full path to the script directory will be /opt/oracle/product/11.1/crs/crs/public/

As oracle from node1

```
[oracle@node1 bin]$ crs_profile -create rg1 -t application \
                        -a $CRS_HOME/crs/public/act_resgroup.pl \
                        -o ci=600
[oracle@node1 bin]$ crs_register rg1
```

rg1.vip resource

rg1.vip is a new application VIP and will be used to connect to the new active passive database.

As oracle on node1

```
[oracle@node1 bin]$ crs_profile -create rg1.vip -t application -r rg1 \
                        -a $CRS_HOME/bin/usrvip \
                        -o oi=eth0,ov=144.25.214.49,on=255.255.252.0
[oracle@node1 bin]$ crs_register rg1.vip
```

In the above:
eh0 is the name of the public adapter
144.25.214.49 is the IP Address of the new Application VIP
255.255.252.0 is the subnet for the public network and is the value of the mask parameter from the /sbin/ifconfig eth0 command

To add a new IP address to an network adapter the Linux Operating system enforces the requirement for root privileges. You must modify the resource such that they run as the root user by Oracle Clusterware.

As root on node1

```
[root@node1 root]# crs_setperm rg1.vip -o root
[root@node1 root]# crs_setperm rg1.vip -u user:oracle:r-x
```

You can test that this has been set up correctly by issuing a crs_start command

As oracle on node1

```
[oracle@node1 bin]$ crs_start -c node1 rg1.vip
Attempting to start `rg1` on member `node1`
Start of `rg1` on member `node1` succeeded.
Attempting to start `rg1.vip` on member `node1`
Start of `rg1.vip` on member `node1` succeeded.
```

In the above command the –'c node1' forces Oracle Clusterware to start the resource on node1. The command asks Oracle Clusterware to start the Application VIP but there is a dependency on the rg1 resource so that resource is started first followed by the VIP resource. The dependency guarantees that:

- A resource will always be started after a resource it is dependent on has reported a correct start to Oracle Clusterware.
- The resources will be started on the same node.

The new VIP should now be 'pingable' from a client

As oracle on node1

```
[oracle@node1 bin]$ ping -c 1 144.25.214.49
PING 144.25.214.49 (144.25.214.49) 56(84) bytes of data.
64 bytes from 144.25.214.49: icmp_seq=0 ttl=64 time=0.020 ms
--- 144.25.214.49 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.020/0.020/0.020/0.000 ms, pipe 2
```

The IP address above is that of the new application VIP.

You should be able to see 2 new resources being managed by Oracle Clusterware. Use the crs_stat command to confirm this.

As oracle on node1

```
[oracle@node1 bin]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node1
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node1
```

You can test relocating the resource to node2

As oracle on node1

```
[oracle@node1 bin]$ crs_relocate -f rg1
Attempting to stop `rg1.vip` on member `node1`
Stop of `rg1.vip` on member `node1` succeeded.
Attempting to stop `rg1` on member `node1`
Stop of `rg1` on member `node1` succeeded.
Attempting to start `rg1` on member `node2`
Start of `rg1` on member `node2` succeeded.
Attempting to start `rg1.vip` on member `node2`
Start of `rg1.vip` on member `node2` succeeded.
```

You use the '–f' parameter to force Oracle Clusterware to relocate not only the resource you have chosen but also all resources that depend on that resource.

To confirm the resources have relocated use crs_stat again

As oracle on node1

```
[oracle@node1 bin]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

To continue you must relocate the resources back to node1

As oracle on node1

```
[oracle@node1 bin]$ crs_relocate -f rg1
Attempting to stop `rg1.vip` on member `node2`
Stop of `rg1.vip` on member `node2` succeeded.
Attempting to stop `rg1` on member `node2`
Stop of `rg1` on member `node2` succeeded.
Attempting to start `rg1` on member `node1`
Start of `rg1` on member `node1` succeeded.
Attempting to start `rg1.vip` on member `node1`
Start of `rg1.vip` on member `node1` succeeded.
```

Use the crs_stat command to confirm this.

As oracle on node1

```
[oracle@node1 bin]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node1
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node1
```

rg1.listener resource

rg1.listener is a new listener that listens on the Application VIP address for connection requests to the database. The failover database will register automatically with this listener.

Copy the supplied 'act_listener.pl' script to $CRS_HOME/crs/public/ on both nodes and ensure that oracle has execute privileges on the scripts.
Modify the supplied listener.ora and tnsnames.ora files to include the correct IP address for the Application VIP.
On both nodes as oracle copy the modified 'tnsnames.ora' and 'listener.ora' files to ORA_SI_HOME/network/admin.

Check that the listener starts on node1

As oracle on node1

```
[oracle@node1 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 bin]$ $ORACLE_HOME/bin/lsnrctl start LISTENER_RG1

LSNRCTL for Linux: Version 11.1.0.4.0 - Production on Apr 23 04:51:21 2007
.
.
.
The command completed successfully
```

Next check that the script can control the listener

As oracle on node1

```
[oracle@node1 bin]$ export CRS_HOME=/opt/oracle/product/11.1/crs
[oracle@node1 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 bin]$ export _USR_ORA_LANG=$ORACLE_HOME       < required to test the script
[oracle@node2 bin]$ export _USR_ORA_SRV=LISTENER_RG1        < from the command line
[oracle@node1 bin]$ $CRS_HOME/crs/public/act_listener.pl stop
LSNRCTL for Linux: Version 11.1.0.4.0 - Production on 15-APR-2006 05:43:39
Copyright (c) 1991, 2005, Oracle.  All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=NODEAPPVIP)(PORT=1521)(IP=FIRST)))
The command completed successfully
```

Next we need to make sure the script works on the other node
First fail the resource group, including the Application VIP over to node2

As oracle on node1

```
[oracle@node1 bin]$ crs_relocate -f rg1
Attempting to stop `rg1.vip` on member `node1`
Stop of `rg1.vip` on member `node1` succeeded.
Attempting to stop `rg1` on member `node1`
Stop of `rg1` on member `node1` succeeded.
Attempting to start `rg1` on member `node2`
Start of `rg1` on member `node2` succeeded.
Attempting to start `rg1.vip` on member `node2`
Start of `rg1.vip` on member `node2` succeeded.
```

Then we need to test the script on node2

As oracle on node2

```
[oracle@node2 bin]$ export CRS_HOME=/opt/oracle/product/11.1/crs
[oracle@node2 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node2 bin]$ export _USR_ORA_LANG=$ORACLE_HOME       < required to test the script
[oracle@node2 bin]$ export _USR_ORA_SRV=LISTENER_RG1        < from the command line
[oracle@node2 bin]$ $CRS_HOME/crs/public/act_listener.pl start
LSNRCTL for Linux: Version 11.1.0.4.0 - Production on 15-APR-2006 05:48:49
.
.
.
The command completed successfully
[oracle@node2 bin]$ $CRS_HOME/crs/public/act_listener.pl stop
LSNRCTL for Linux: Version 11.1.0.4.0 - Production on 15-APR-2006 05:43:59
.
.
.
The command completed successfully
```

Finally we add the listener as a resource to the resource group. In the following command replace the LISTENR_RG1 parameter with the name of your listener.

As oracle on node1

```
[oracle@node1 bin]$ export CRS_HOME=/opt/oracle/product/11.1/crs
[oracle@node1 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 bin]$ crs_profile -create rg1.listener \
                              -t application \
                              -r rg1.vip \
                              -a $CRS_HOME/crs/public/act_listener.pl \
                              -o ci=20,ra=5,osrv=LISTENER_RG1,ol=$ORACLE_HOME
[oracle@node1 bin]$ crs_register rg1.listener
```

Then ask Oracle Clusterware to start the resource

As oracle on node1

```
[oracle@node1 bin]$ crs_start rg1.listener
Attempting to start `rg1.listener` on member `node2`
Start of `rg1.listener` on member `node2` succeeded.
```

Test failover of the resource group using the crs_relocate command

As oracle on node1

```
[oracle@node1 bin]$ crs_relocate -f rg1
Attempting to stop `rg1.listener` on member `node1`
Stop of `rg1.listener` on member `node1` succeeded.
Attempting to stop `rg1.vip` on member `node1`
Stop of `rg1.vip` on member `node1` succeeded.
Attempting to stop `rg1` on member `node1`
Stop of `rg1` on member `node1` succeeded.
Attempting to start `rg1` on member `node2`
Start of `rg1` on member `node2` succeeded.
Attempting to start `rg1.vip` on member `node2`
Start of `rg1.vip` on member `node2` succeeded.
Attempting to start `rg1.listener` on member `node2`
Start of `rg1.listener` on member `node2` succeeded.
```

You can see that the resource group now contains 3 items

As oracle on node1

```
[oracle@node1 public]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.listener   application    0/5    0/0    ONLINE    ONLINE    node2
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

At this point our resource group consists of a RG container, which includes an Application VIP and a new listener LISTENER_RG1.

rg1.db_ERI resource

rg1.db_ERI is a new database resource that monitors and controls the Single Instance database who's SID is ERI. The database should have been running on node1 after the instance was created.

Copy the supplied 'act_db.pl' script to $CRS_HOME/crs/public/ and ensure that oracle has execute privileges on the scripts.

You need to tell the instance which listener to dynamically register with. This is indicated by the 'LOCAL_LISTENER=' database initialization parameter.

The alias LISTENER_rg1 is defines in the supplied TNSNAME.ORA file and points at the new listener that listens on the Application VIP. Assuming you are using an SPFILE:

As oracle on node1

```
[oracle@node1 oracle]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 oracle]$ export ORACLE_SID=ERI
[oracle@node1 oracle]$ $ORACLE_HOME/bin/sqlplus / as sysdba
SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 06:08:51 2006
Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
SQL> alter system set LOCAL_LISTENER=LISTENERS_rg1 scope=BOTH;
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
```

We need to make sure that the database will startup on the other node in the cluster
Shut the database down

As oracle on node1

```
[oracle@node1 oracle]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 oracle]$ export ORACLE_SID=ERI
[oracle@node1 oracle]$ $ORACLE_HOME/bin/sqlplus / as sysdba
SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 06:08:51 2006
Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
```

Next we copy the required files and directories from node1 to node2

As oracle on node1

```
[oracle@node1 oracle]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 oracle]$ scp $ORACLE_HOME/dbs/initERI.ora    node2:/$ORACLE_HOME/dbs/
[oracle@node1 oracle]$ ssh node2 mkdir -p /opt/oracle/admin
[oracle@node1 oracle]$ scp -r /opt/oracle/admin/* node2:/opt/oracle/admin/
```

You must also ensure that the diagnostic directory tree is copied to node2. In Oracle 11g this is located by the diagnostic_dest location, this may not be in the admin subdirectory.

Next we must make sure that the Single Instance will start on node2

As oracle on node2

```
[oracle@node2 oracle]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node2 oracle]$ export ORACLE_SID=ERI
[oracle@node2 oracle]$ $ORACLE_HOME/bin/sqlplus / as sysdba

SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 07:16:00 2006

Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1207959552 bytes
Fixed Size                  1260516 bytes
Variable Size             318768156 bytes
Database Buffers          872415232 bytes
Redo Buffers               15515648 bytes
Database mounted.
Database opened.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
```

Next we need to test the database scripts

As oracle on node1

```
[oracle@node1 bin]$ export CRS_HOME=/opt/oracle/product/11.1/crs
[oracle@node1 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 bin]$ export _USR_ORA_LANG=$ORACLE_HOME       < required to test the script
[oracle@node1 bin]$ export _USR_ORA_SRV=ERI                 < from the command line
[oracle@node1 bin]$ export _USR_ORA_FLAGS=1                 < set if db uses ASM
[oracle@node1 bin]$ $CRS_HOME/crs/public/act_db.pl start

SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 06:27:39 2006

Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area 1207959552 bytes
Fixed Size                  1260516 bytes
Variable Size             318768156 bytes
Database Buffers          872415232 bytes
Redo Buffers               15515648 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 -
Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
[oracle@node1 public]$ $CRS_HOME/crs/public/act_db.pl stop

SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 06:28:00 2006

Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 -
Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
```

As oracle on node2

```
[oracle@node2 bin]$ export CRS_HOME=/opt/oracle/product/11.1/crs
[oracle@node2 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node2 bin]$ export _USR_ORA_LANG=$ORACLE_HOME       < required to test the script
[oracle@node2 bin]$ export _USR_ORA_SRV=ERI                 < from the command line
[oracle@node1 bin]$ export _USR_ORA_FLAGS=1                 < set if db uses ASM
[oracle@node2 bin]$ $CRS_HOME/crs/public/act_db.pl start

SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 06:27:39 2006

Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area 1207959552 bytes
Fixed Size                  1260516 bytes
Variable Size             318768156 bytes
Database Buffers          872415232 bytes
Redo Buffers               15515648 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 -
Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
[oracle@node2 public]$ $CRS_HOME/crs/public/act_db.pl stop

SQL*Plus: Release 11.1.0.4.0 - Production on Sun Apr 15 06:28:00 2006

Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.4.0 -
Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
```

Finally we add the database instance as a resource to the resource group

Use only one of these two commands.

Setting the oflags=1 parameter modifes the action script called by Oracle Clusterware before it starts the database.
Before issuing the start database command it checks to see if the ASM instance is up on the node. If it is not up then the start action first starts the ASM instance and then starts the database instance.

If the database uses ASM then use this command

As oracle on node1

```
[oracle@node1 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 bin]$ crs_profile -create rg1.db_ERI -t application \
                        -r rg1 \
                        -a $CRS_HOME/crs/public/act_db.pl \
                        -o ci=20,ra=5,osrv=ERI,ol=$ORACLE_HOME,oflags=1,rt=600
[oracle@node1 bin]$ crs_register rg1.db_ERI
```

If the database does NOT uses ASM then use this command

As oracle on node1

```
[oracle@node1 bin]$ export ORACLE_HOME=/opt/oracle/product/11.1/si
[oracle@node1 bin]$ crs_profile -create rg1.db_ERI -t application \
                        -r rg1 \
                        -a $CRS_HOME/crs/public/act_db.pl \
                        -o ci=20,ra=5,osrv=ERI,ol=$ORACLE_HOME,oflags=0,rt=600
[oracle@node1 bin]$ crs_register rg1.db_ERI
```

As the startup of the instance may take more than 60 seconds, especially if an ASM instance must be started prior to the database instance start, the START script timeout is set to 600 seconds.

Then ask Oracle Clusterware to start the resource

As oracle on node1

```
[oracle@node1 bin]$ crs_start rg1.db_ERI
Attempting to start `rg1.db_ERI` on member `node2`
Start of `rg1.db_ERI` on member `node2` succeeded.
```

There are now 4 resources in the resource group

As oracle on node1

```
[oracle@node1 public]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node2
rg1.listener   application    0/5    0/0    ONLINE    ONLINE    node2
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

you should test relocating the resource group

As oracle on node1

```
[oracle@node1 public]$ crs_relocate -f rg1
Attempting to stop `rg1.listener` on member `node2`
Stop of `rg1.listener` on member `node2` succeeded.
Attempting to stop `rg1.vip` on member `node2`
Stop of `rg1.vip` on member `node2` succeeded.
Attempting to stop `rg1.db_ERI` on member `node2`
Stop of `rg1.db_ERI` on member `node2` succeeded.
Attempting to stop `rg1` on member `node2`
Stop of `rg1` on member `node2` succeeded.
Attempting to start `rg1` on member `node1`
Start of `rg1` on member `node1` succeeded.
Attempting to start `rg1.db_ERI` on member `node1`
Start of `rg1.db_ERI` on member `node1` succeeded.
Attempting to start `rg1.vip` on member `node1`
Start of `rg1.vip` on member `node1` succeeded.
Attempting to start `rg1.listener` on member `node1`
Start of `rg1.listener` on member `node1` succeeded.
```

rg1.head resource

rg1 is an additional resource group, it acts as a top level container for the 3 resources that are each at the top of their resource trees

You should already have the on resourcegroup.scr script in the correct location on both nodes in the cluster

This resource helps in two distinct ways:

- When relocating a resource group from one node to another the order in which Oracle Clusterware starts resources at the same level in the resource tree is indeterminate. Listing the required resources for this resource forces the correct startup order for the resources. It is advantageous to have the listener started before the database instance starts so that when the database instance starts it automatically registers with the listener immediately.
- It is now possible to start all resources using one command: crs_start rg1.head

As oracle from node1

```
[oracle@node1 bin]$ crs_profile -create rg1.head -t application \
                        -a $CRS_HOME/crs/public/act_resgroup.pl \
                        -r "rg1.listener rg1.db_ERI" \
                        -o ci=600
[oracle@node1 bin]$ crs_register rg1.head
```

Now we can test that the resource starts OK

As oracle on node1

```
[oracle@node1 bin]$ crs_start rg1.head
Attempting to start `rg1.head` on member `node1`
Start of `rg1.head` on member `node1` succeeded.
```

Now we have all 5 resources in the resource group.

As oracle on node1

```
[oracle@node1 bin]$ crs_stat -t -v | grep ^rg1
rg1             application    0/1    0/0    ONLINE    ONLINE    node1
rg1.db_ERI      application    0/5    0/0    ONLINE    ONLINE    node1
rg1.head        application    0/1    0/0    ONLINE    ONLINE    node1
rg1.listener    application    0/5    0/0    ONLINE    ONLINE    node1
rg1.vip         application    0/1    0/0    ONLINE    ONLINE    node1
```

Testing failover

We can test planned and unplanned failover

Planned Failover

To test planned failover we will use crs_stat to see which node the resource group is running on. Then we will use crs_relocate to move the entire resource group to a new node. Then we will use crs_stat to see the resources running on the new node

First lets see where the resources are currently running

As oracle on node1

```
[oracle@node1 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node1
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node1
rg1.head       application    0/1    0/0    ONLINE    ONLINE    node1
rg1.listener   application    0/5    0/0    ONLINE    ONLINE    node1
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node1
```

Here you can see the resources are running on node1

Now lets carry out a planned failover to the other node using crs_relocate

As oracle on node1

```
[oracle@node1 oracle]$ crs_relocate -f rg1
Attempting to stop `rg1.head` on member `node1`
Stop of `rg1.head` on member `node1` succeeded
Attempting to stop `rg1.listener` on member `node1`
Stop of `rg1.listener` on member `node1` succeeded.
Attempting to stop `rg1.vip` on member `node1`
Stop of `rg1.vip` on member `node1` succeeded.
Attempting to stop `rg1.db_ERI` on member `node1`
Stop of `rg1.db_ERI` on member `node1` succeeded.
Attempting to stop `rg1` on member `node1`
Stop of `rg1` on member `node1` succeeded.
Attempting to start `rg1` on member `node2`
Start of `rg1` on member `node2` succeeded.
Attempting to start `rg1.db_ERI` on member `node2`
Start of `rg1.db_ERI` on member `node2` succeeded.
Attempting to start `rg1.vip` on member `node2`
Start of `rg1.vip` on member `node2` succeeded.
Attempting to start `rg1.listener` on member `node2`
Start of `rg1.listener` on member `node2` succeeded.
Attempting to start `rg1.head` on member `node2`
Start of `rg1.head` on member `node2` succeeded.
```

If we now issue a crs_stat command again we can see that the resources are now running on node2

As oracle on node1

```
[oracle@node1 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node2
rg1.head       application    0/1    0/0    ONLINE    ONLINE    node2
rg1.listener   application    0/5    0/0    ONLINE    ONLINE    node2
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

Unplanned Failover

To force an unplanned failover we will force a failure of one of the resources until the restart count reaches the restart threshold. The third column in the table above lists the restarts. The rg1.listener resource is one of the easiest of the resources to fail.

As oracle on node2

```
[oracle@node2 oracle]$ ps -aef | grep LISTENER_rg1 | grep -v grep
oracle   25485     1  0 09:07 ?        00:00:00 /opt/oracle/product/11.1/si/bin/tnslsnr
LISTENER_rg1 -inherit
[oracle@node2 oracle]$ kill -9 25485
[oracle@node2 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node2
rg1.head       application    0/1    0/0    ONLINE    ONLINE    node2
rg1.listener   application    1/5    0/0    ONLINE    ONLINE    node2
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

As you can see Oracle Clusterware detected the failure in the listener process and restarted it. How quickly Oracle Clusterware reacts to a failure is a function of the "ci=" parameter used when the resource was profiled. When the rg1.listener was profiled the ci= parameter was set to 20 (seconds) which means that, on average, Oracle Clusterware will react with in ½ * 20 seconds.

Repeat the above 3 commands until the crs_stat show the following

As oracle on node2

```
[oracle@node2 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node2
rg1.head       application    0/1    0/0    ONLINE    ONLINE    node2
rg1.listener   application    5/5    0/0    ONLINE    ONLINE    node2
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

At this point we have reached the restart attempts limit for the rg1.listener resource. Another failure will cause Oracle Clusterware to relocate the resource to another node. Because the resource is a member of a resource group the other members of the group will also be relocated to the other node. Oracle Clusterware calls each of the action scripts in the correct sequence, based on the resource dependencies with the stop parameter. It then starts all the resources on the other node.

We need to kill the listener one last time to cause the failover.

As oracle on node2

```
[oracle@node2 oracle]$ ps -aef | grep LISTENER_rg1 | grep -v grep
oracle   31093     1  0 09:30 ?        00:00:00 /opt/oracle/product/11.1/si/bin/tnslsnr
LISTENER_rg1 -inherit
[oracle@node2 oracle]$ kill -9 31093
```

Repeat the crs_stat command

As oracle on node2

```
[oracle@node2 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node2
rg1.head       application    0/1    0/0    ONLINE    ONLINE    node2
rg1.listener   application    0/5    0/0    ONLINE    OFFLINE
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node2
```

At this point in time Oracle Clusterware has just detected the listener has gone offline
Repeat the crs_stat command

As oracle on node2

```
[oracle@node2 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node2
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node2
rg1.head       application    0/1    0/0    ONLINE    OFFLINE
rg1.listener   application    0/5    0/0    ONLINE    OFFLINE
rg1.vip        application    0/1    0/0    ONLINE    OFFLINE
```

At this point in time Oracle Clusterware has stopped almost all of the resources

Repeat the crs_stat command

As oracle on node2

```
[oracle@node2 oracle]$ crs_stat -t -v | grep ^rg1
rg1            application    0/1    0/0    ONLINE    ONLINE    node1
rg1.db_ERI     application    0/5    0/0    ONLINE    ONLINE    node1
rg1.head       application    0/1    0/0    ONLINE    ONLINE    node1
rg1.listener   application    0/5    0/0    ONLINE    ONLINE    node1
rg1.vip        application    0/1    0/0    ONLINE    ONLINE    node1
```

Oracle Clusterware has relocated all the resources in the resource group to the other node.

Appendix A: Action scripts

rg1 action script

Name: act_resgroup.pl
Location: $CRS_HOME/crs/public/ (on both nodes)
Modification required: none

```
#!/usr/bin/perl
#
# $Header: act_resgroup.pl 05-apr-2007.14:39:52 rvenkate Exp $
#
# act_resgroup.pl
#
# Copyright (c) 2007, Oracle. All rights reserved.
#
#    NAME
#      act resgroup.pl - action script for generic resource group
#
#    DESCRIPTION
#      This perl script is the action script for a generic resource group
#
#    NOTES
#      Edit the perl installation directory as appropriate.
#
#      Place this file in <CRS HOME>/crs/public/
#
#    MODIFIED   (MM/DD/YY)
#    rvenkate    04/05/07 - checkin into demo dir
#    pnewlan     04/05/07 - Creation
#

exit 0;
```

rg1.vip script

Name: usrvip
Location: $CRS_HOME/bin/

<This is a standard Oracle Clusterware provided script>

rg1.listener script

Name: act_listener.pl
Location: $CRS_HOME/crs/public/ (on both nodes)

```
#!/usr/bin/perl
#
# $Header: act_listener.pl 05-apr-2007.14:14:24 rvenkate Exp $
#
# act_listener.pl
#
# Copyright (c) 2007, Oracle.  All rights reserved.
#
#    NAME
#      act_listener.pl - action script for the listener resource
#
#    DESCRIPTION
#      This perl script is the action script for start / stop / check
#      the Oracle Listener in a cold failover configuration.
#
#    NOTES
#      Edit the perl installation directory as appropriate.
#
#      Place this file in <CRS_HOME>/crs/public/
#
#    MODIFIED   (MM/DD/YY)
#    rknapp      06/24/07 - fixed bug with multiple listener
#    rvenkate    04/05/07 - checkin as demo
#    pnewlan     01/17/07 - Use Environment variables rather than hard code
#                           HOME & LISTENER
#    pnewlan     11/23/06 - oracle OS user invoker and listener name
#    rknapp      05/22/06 - Creation
#

$ORACLE_HOME = "$ENV{_USR_ORA_LANG}";
$ORA_LISTENER_NAME = "$ENV{_USR_ORA_SRV}";

if ($#ARGV != 0 ) {
        print "usage: start stop check required \n";
exit;
}

$command = $ARGV[0];

# start listener
if ($command eq "start") {
        system ("
        export ORACLE_HOME=$ORACLE_HOME
        export ORA_LISTENER_NAME=$ORA_LISTENER_NAME
#       export TNS_ADMIN=$ORACLE_HOME/network/admin  # optionally set TNS_ADMIN here
        $ORACLE_HOME/bin/lsnrctl start $ORA_LISTENER_NAME");
        }
# stop listener
if ($command eq "stop") {
        system ("
        export ORACLE_HOME=$ORACLE_HOME
        export ORA_LISTENER_NAME=$ORA_LISTENER_NAME
#       export TNS_ADMIN=$ORACLE_HOME/network/admin  # optionally set TNS_ADMIN here
        $ORACLE_HOME/bin/lsnrctl stop $ORA_LISTENER_NAME");
        }
# check listener
if ($command eq "check") {
        check_listener();
        }

sub check_listener {
        my($check_proc_listener,$process_listener) = @_;
        $process_listener = "$ORACLE_HOME/bin/tnslsnr $ORA_LISTENER_NAME";
```

```
        $check_proc_listener = qx(ps -aef | grep -w "tnslsnr $ORA_LISTENER_NAME" | grep -v
grep | head -n 1 | awk '{print \$8,\$9}');
        chomp($check_proc_listener);
        if ($process_listener eq $check_proc_listener) {
                exit 0;
        } else {
                exit 1;
                }
        }
```

➢ Indicates the line has wrapped here

rg1.db_ERI script

Name: act_db.pl
Location: $CRS_HOME/crs/public/ (on both nodes)

```
#!/usr/bin/perl
#
# $Header: act_db.pl 05-apr-2007.14:21:24 rvenkate Exp $
#
# act_db.pl
#
# Copyright (c) 2007, Oracle. All rights reserved.
#
#    NAME
#      act_db.pl - <one-line expansion of the name>
#
#    DESCRIPTION
#      This perl script is the action script for start / stop / check
#      the Oracle Instance in a cold failover configuration.
#
#      Place this file in <CRS_HOME>/crs/public/
#
#    NOTES
#      Edit the perl installation directory as appropriate.
#
#    MODIFIED   (MM/DD/YY)
#    pnewlan     05/25/07 - use grep -w
#    rvenkate    04/05/07 - checkin into demo dir
#    pnewlan     01/17/07 - Use Environment variables rather than hard code
#                         - HOME & SID
#    pnewlan     11/23/06 - oracle OS user invoker
#    rknapp      05/22/06 - Creation
#

$ORACLE_HOME = "$ENV{_USR_ORA_LANG}";
$ORACLE_SID = "$ENV{_USR_ORA_SRV}";
$USES_ASM = "$ENV{_USR_ORA_FLAGS}";
if ($#ARGV != 0 ) {
        print "usage: start stop check required \n";
exit;
}
$command = $ARGV[0];
# Database start stop check
# Start database
if ($command eq "start" ) {
        if ($USES_ASM eq "1") {
            #make sure ASM is running now
            system ("
                export ORACLE_HOME=$ORACLE_HOME
                $ORACLE_HOME/bin/srvctl start asm -n `hostname -s`
                    ");
        }
        system ("
        export ORACLE_SID=$ORACLE_SID
        export ORACLE_HOME=$ORACLE_HOME
        export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
#       export TNS_ADMIN=$ORACLE_HOME/network/admin  # optionally set TNS_ADMIN here
        $ORACLE_HOME/bin/sqlplus /nolog <<EOF
        connect / as sysdba
        startup
        quit
        EOF" );
        $MYRET = check();
        exit $MYRET;
        }
if ($command eq "stop" ) {
        system ("
```

```
        export ORACLE_SID=$ORACLE_SID
        export ORACLE_HOME=$ORACLE_HOME
        export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
#       export TNS_ADMIN=$ORACLE_HOME/network/admin  # optionally set TNS ADMIN here
        $ORACLE_HOME/bin/sqlplus /nolog <<EOF
        connect / as sysdba
        shutdown immediate
        quit
        EOF" );
        $MYRET = check();
        if ($MYRET eq 1) {
            exit 0;
            }
        else {
            exit 1;
        }
        }
# Check database
if ($command eq "check" ) {
        $MYRET = check();
        exit $MYRET;
}
sub check {
        my($check_proc,$process) = @_;
       $process = "ora_pmon_$ORACLE_SID";
        $check_proc = qx(ps -aef | grep -w ora_pmon_$ORACLE_SID | grep -v grep | awk
'{print \$8}');
        chomp($check_proc);
        if ($process eq $check_proc) {
                $RET=0;
        } else {
                $RET=1;
        }
        return $RET;
}
```

➢

➢ Indicates that the line has wrapped here.

Appendix B: TNS Network configuration files

In the following examples the name 'node1' is the network name for the new Virtual IP address created as part of the rg1.vip resource.

TNSNAMES.ORA

name: tnsnames.ora
Location: $ORACLE_HOME/network/admin/ (on both nodes add to existing file)
Modification required: HOSTNAME for the new Application VIP

```
LISTENERS_RG1 =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = nodeappvip)(PORT = 1521))
  )
```

LISTENER.ORA

name: listener.ora
Location: $ORACLE_HOME/network/admin/ (on both nodes add to existing file)
Modification required: HOSTNAME for the new Application VIP

```
LISTENER_RG1 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = nodeappvip)(PORT = 1521)(IP = FIRST))
    )
  )
```

Background Reading

1 The concepts used in this paper are documented in the Oracle Database 11g Documentation set : http://www.oracle.com/pls/db102/portal.portal_db?selected=3 Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide

2 For a general understanding of Oracle Clusterware: http://www.oracle.com/technology/products/database/clustering/pdf/twp_oracleclusterware3rdparty[1].pdf



Using Oracle Clusterware to Protect
Single Instance Oracle Database 11.1
May 2007
Version 1.14
Author: Philip Newlan
Contributing Athors : Roland Knapp, Alan Tam

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

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

Copyright © 2007, Oracle. All rights reserved.
This document is provided for information purposes only and the contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.