



## Viskosity Technologies

### Leaders in Oracle ViSion and Knowledge

### Real Application Clusters (RAC) Pocket Reference Guide

ViSKosity professionals are experts in the areas of:

- Real Application Clusters (RAC) - 9i, 10g, 10gR2 and 11g
- EBusiness Suite Administration (11i and 12i)
- Data Warehousing
- Business Intelligence
- Database Security
- Enterprise Systems Management (ITIL and ITSM)

Email: [info@viskosity.com](mailto:info@viskosity.com)

Phone: 214.502.1231

URL: <http://viskosity.com>

#### Cluster Verify (cluvfy)

##### Post-checks for hardware and operating system setup

```
runcluvfy.sh stage -post hwas -n node1,node2
```

##### Pre-checks for cluster services setup

```
runcluvfy.sh stage -pre crsinst -n rac1,rac2
```

##### Check Node Reachability

```
runcluvfy.sh comp nodereach -n rac1,rac2 -srcnode rac1
```

##### Shared Storage Check

```
sh runcluvfy.sh comp ssa -n rac1,rac2
```

##### Post-checks for cluster services setup

```
cluvfy stage -post crsinst -n rac1,rac2
```

##### Verify Cluster Integrity

```
cluvfy comp clu -n rac1,rac2 -verbose
```

##### Verify Cluster Manager Integrity

```
cluvfy comp clumgr -n rac1,rac2 -verbose
```

##### Verify Node Applications

```
cluvfy comp nodeapp -n rac1,rac2 -verbose
```

##### Verify CRS Integrity

```
cluvfy comp crs -n rac1,rac2 -verbose
```

##### Verify OCR Integrity

```
cluvfy comp ocr -n rac1,rac2 -verbose
```

##### Precheck for Database Configuration

```
cluvfy stage -pre dbcfg -n rac1,rac2 -d /u01/app/oracle/product/10.2.0/DB -verbose
```

##### Precheck for Database Installation

```
cluvfy stage -pre dbinst -n rac1,rac2 -verbose
```

##### Precheck for Database Installation

```
cluvfy stage -pre dbinst -n rac1,rac2 -verbose
```

##### Verify OCR Integrity

```
cluvfy comp ocr -n all
```

#### CRS Tracing

```
export ORA_CRSDEBUG_ALL= 1 to 5  
Check logs @ $ORA_CRS_HOME/log/node  
See Metalink Note:357808.1 for details
```

#### Enable the tracing for the resources running:

```
# crsctl debug log res "ora.VISK.VISK1.inst:5"  
# crsctl debug log res "ora.VISK.VISK2.inst:5"
```

#### Increase the tracing for crsd threads:

```
# crsctl debug log crs CRSRES:5  
# crsctl debug log crs CRSAPP:5  
# crsctl debug log crs CRSEVT:5
```

#### CRS commands

##### Stop CRS (as root)

```
crsctl start crs - 10gR2  
/etc/init.d/init.crs start crs - 10gR1
```

##### Stop CRS (as root)

```
crsctl stop crs - 10gR2  
/etc/init.d/init.crs stop crs - 10gR1
```

##### Disable CRS

```
/etc/init.d/init.crs disable
```

##### Check CRS

```
crsctl check crs
```

- Check CSSD - crsctl check cssd
- Check CRSD - crsctl check crsd
- Check EVM - crsctl check evmd

##### Query Voting Disks

```
crsctl query css votedisk
```

##### Query Software Version

```
crsctl query crs  
softwareversion
```

##### Query Active Version

```
crsctl query crs activeversion
```

##### List Modules

- crsctl lsmodules crs
- crsctl lsmodules evm

##### Start / Stop Resources

```
crs_start resource-name -all (all resource)  
crs_stop resource-name -all (all resource)
```

##### Diagnostic Collection -

##### \$ORA\_CRS\_HOME/diagcollection.pl

Level 1: Configuration info (OCR and init scripts)  
Level 2: Logs and traces  
Level 3: Core files



### Additional CRS command:

crs\_relocate – change where resource is active

### srvctl commands

#### ADD

```
srvctl add database -d VISK -o
/apps/oracle/product/10.2.0/RACDB
srvctl add instance -d VISK -i VISK1 -n rac1
```

#### CONFIG (information)

```
srvctl config
srvctl config -v
srvctl config database -d VISK
srvctl config database -d VISK -a
srvctl config nodeapps -a -g -s -l -n rac1
srvctl config listener -n rac1
srvctl config service -d VISK
srvctl config service -d VISK -s "VISK_RAC" -a
```

#### Export

Ocrconfig –export /tmp/dba/exp\_ocr.dmp –s online

#### START / STOP

```
srvctl stop database -d VISK
srvctl stop database -d VISK -o abort
srvctl stop instance -d VISK -i VISK1
```

```
srvctl stop listener -b rac1 -l VISK_RAC1
srvctl stop listener -b rac1 -l VISK_RAC2
```

```
srvctl start nodeapps -n RAC1
srvctl stop nodeapps -n RAC2
```

#### STATUS

```
srvctl status database -d VISK
srvctl status instance -d VISK -i "VISK1,VISK2" -v
srvctl status nodeapps -n rac1
```

#### Tracing

export SRVM\_TRACE=true

### OCR commands

```
ocrdump ocr_cluster_dump
ocrdump -stdout -keyname SYSTEM.css.misscount
ocrdump -stdout -xml
Change OCR Backup Location: ocrconfig -backuploc
<directory name>
Show OCR Backup: ocrconfig -showbackup
Check integrity of OCR: ocrcheck
Restore the OCR: ocrconfig -restore <filename>
```

### Determine OCR Location

```
cat /etc/oracle/ocr.loc
ocrconfig_loc=/dev/raw/raw1
ocrmirrorconfig_loc=/dev/raw/raw2
local_only=FALSE
```

#### Recommendation:

Mirror the OCR disk

- ocrconfig -replace ocr file\_or\_disk
- ocrconfig -replace ocrmirror file\_or\_disk

### oifcfg commands

```
oifcfg iflist
oifcfg iflist -p -n
oifcfg getif -type public
oifcfg getif
oifcfg getif -n rac2
oifcfg getif -if eth2
```



### CRS Statistics

```
crs_stat -t
crs_stat -t -v
http://dbaexpert.com/dba_crs.txt
```

### Additional commands

- List Nodes: olsnodes -n
- View Available Raw Devices: raw -qa
- Look for CRS processes: ps -ef | egrep -i "crs|css|evm|d.bin"

### Add RAC gv\$ views

@\$ORACLE\_HOME/rdbms/admin/catclust.sql

### RAC Specific Init.ora Parameters

```
*.cluster_database_instances= 2
*.cluster_database= true
VISK1.instance_number= 1
VISK2.instance_number= 2
VISK1.thread= 1
VISK2.thread= 2
VISK1.undo_tablespace= 'UNDO_RBS1'
VISK2.undo_tablespace= 'UNDO_RBS2'
VISK1.local_listener= VISK_L1
VISK2.local_listener= VISK_L2
VISK1.remote_listener= VISK_R1
VISK2.remote_listener= VISK_R2
VISK1.log_archive_format= VISK1_%t_%s_%r.arc
VISK2.log_archive_format= VISK2_%t_%s_%r.arc
```

### Voting Disks

Backup - dd if=/dev/raw/raw11  
of=/home/oracle/votingdisk.bkup

Restore - dd if=/home/oracle/votingdisk.bkup  
of=/dev/raw/raw11

### Recommendation:

Have a minimum three voting disks for RAC survivability

### Rolling Upgrade

./runInstaller -updateNodeList "CLUSTER\_NODES=rac1" -local  
ORACLE\_HOME=/apps/oracle/product/10.2.0/CRS

### Recommended kernel parameters - /etc/sysctl.conf

```
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.core.rmem_default = 262144
net.core.rmem_max = 262144
net.core.wmem_default = 262144
net.core.wmem_max = 262144
kernel.shmmax = 2147483648 #depends on total RAM
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
```

### Setup User Equivalence

1. ssh-keygen -t rsa #Take all defaults
2. cd \$HOME/.ssh
3. cat id\_rsa.pub #Copy the contents of the output
4. # ssh to remote host as oracle
5. cd .ssh
6. vi authorized\_keys #paste contents of id\_rsa.pub

Note:

- Repeat for all the nodes including local node
- Make sure that the group permissions are not writeable

### Oracle Clusterware 11g Upgrade Considerations

1. Make sure that you have enough space for the OCR
2. OCR space utilization doubles during the upgrade

### Relevant Logfiles

- OS - /var/log/messages
- CRS - \$ORA\_CRS\_HOME/log/node-name/crsd/node.log
- CSS - \$ORA\_CRS\_HOME/log/node-name/cssd/ocssd.log
- EVM - \$ORA\_CRS\_HOME/log/node-name/evmn/evm.log
- RACG - \$ORA\_CRS\_HOME/log/node-name/racg/ora.node1.ons.log
- VIP Tracing – edit \$ORA\_CRS\_HOME/bin/racgwrap  
\_USR\_ORA\_DEBUG=1 to 10 #Bounce Nodeapps