

**ORACLE® HYPERION ENTERPRISE  
PERFORMANCE MANAGEMENT SYSTEM  
HYPERION® SHARED SERVICES**

*RELEASE 11.1.1.X*

**ACTIVE-PASSIVE FAILOVER  
CLUSTERS (UNIX ENVIRONMENTS)**



CONTENTS IN BRIEF

About Shared Services High Availability on UNIX . . . . .	2
Strategy for Deploying Shared Services in a Failover Cluster on UNIX . . . . .	2
Configuring Shared Services in an Oracle Clusterware Failover Cluster . . . . .	4
Oracle Clusterware Postinstallation Procedures . . . . .	9
Managing the Shared Services Cluster . . . . .	12
Oracle Clusterware Backup and Recovery . . . . .	13
Log Files . . . . .	14
Oracle Internet Directory Clustering . . . . .	14
Tips and Tricks . . . . .	15
Script Template for createvip.sh . . . . .	17
Script Template for hssregister.sh . . . . .	17
Action Script Template for hss11.pl . . . . .	18
Additional Information . . . . .	20

## About Shared Services High Availability on UNIX

To make Oracle's Hyperion® Shared Services highly available, you must use clustering solutions to ensure that none of these components is a single point of failure:

- Web application
- Native Directory and other user directories
- Database

Database clustering solutions depend on the relational database management system (RDBMS) that you use. EPM System products support Oracle Real Application Clusters (RAC) and third-party RDBMS software. See the documentation for your RDBMS.

These configurations are supported:

- Web application and OpenLDAP Native Directory in a failover cluster using Oracle Clusterware  
See “[Configuring Shared Services in an Oracle Clusterware Failover Cluster](#)” on page 4.
- Web application with Oracle Clusterware and Oracle Internet Directory (OID) in any active-passive failover cluster supported by Oracle Internet Directory  
See “[Oracle Internet Directory Clustering](#)” on page 14.

You cluster Shared Services and Native Directory for failover using Oracle Clusterware 11.1, which is available for free to protect Oracle Hyperion Enterprise Performance Management System components. See “[Installing and Configure Oracle Clusterware](#)” on page 7. .

You can download Oracle Clusterware from <http://www.oracle.com/technology/software/products/database/index.html>, under Database 11.1.0.6.0. Information about Oracle Clusterware is available from <http://www.oracle.com/technology/products/database/clusterware/index.html>. Licensing information is available at [http://download.oracle.com/docs/cd/B28359\\_01/license.111/b28287/editions.htm](http://download.oracle.com/docs/cd/B28359_01/license.111/b28287/editions.htm).

## Strategy for Deploying Shared Services in a Failover Cluster on UNIX

The Oracle Clusterware installation is described for a two-node topology where one node stays active and the other node is passive.

In a failover cluster, the Shared Services processes (Web Application and Native Directory) are accessible at a specific virtual IP (VIP) address referenced by a floating cluster host name (*hsscuster*) or DNS alias. If the primary node fails, the VIP and the Shared Services processes move automatically to the secondary node, as shown in [Figure 1 on page 4](#).

Both nodes mount the Shared Services files stack from a network file system (NFS) server that is separate from the Clusterware nodes. Both nodes and the NFS server use the same user name to manage the entire lifecycle of Shared Services, from installation to the operational behavior under Oracle Clusterware control.

In the Oracle Clusterware context, it is important to follow three rules regarding the Shared Services lifecycle:

- The installation and the configuration are performed on a separate NFS server, whose host name must be changed at system level to match the chosen cluster floating host name. The Shared Services stack learns the floating cluster host name in all its internal properties, files, and database settings. When the configuration is completed, the NFS host name is reset to the original name.
- The installation and the configuration on the NFS server must use the exact name of the mount point that the nodes will mount from the NFS server, so the NFS server mounts itself at Shared Services setup time. In other words the setup is performed through client mount settings, not the NFS sharing settings.
- The same user name (identical UID) and group name (identical GID) must be used for Shared Services servicing on the three hosts: installation and configuration on the NFS server, and maneuvering (start, stop, relocation) on the nodes. This document refers to the user name `oracle` (UID 102) and the group name `install` (GID 100).

Although the Oracle Clusterware installation is performed from one node, the Oracle Clusterware binaries are laid out locally on each node, outside the shared storage NFS server.

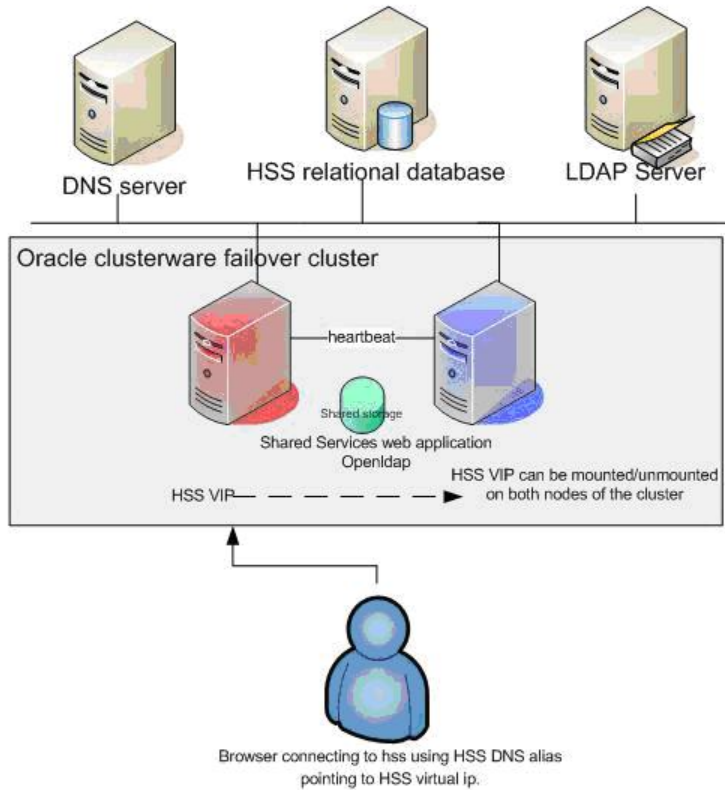
Together, the NFS mounted file systems hold the following elements:

- Shared Services configured binaries
- Oracle Clusterware data files—Oracle Cluster Registry (OCR) and the voting file
- Shell scripts to create VIP and Shared Services profiles and register them with the Oracle Clusterware registry

The Shared Services profile registers a perl action script that defines the start, stop, and health-check behavior of Shared Services processes.

There is no restriction on the underlying shared storage (NAS, local NFS server, and so on), which should be made highly available in production environments.

Figure 1 Shared Services Oracle Clusterware Failover Cluster



## Configuring Shared Services in an Oracle Clusterware Failover Cluster

Before clustering Shared Services Web application for failover, you must meet Oracle Clusterware prerequisites and then install and configure Oracle Clusterware. This document refers to the “Oracle Clusterware Preinstallation Tasks,” “Configuring Oracle Clusterware Storage,” and “Installing Oracle Clusterware” sections of the Oracle Clusterware installation guides for UNIX, which are available from <http://www.oracle.com/pls/db111/db111.homepage>:

- Linux—[http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28263.pdf](http://download.oracle.com/docs/cd/B28359_01/install.111/b28263.pdf)
- Solaris—[http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28262.pdf](http://download.oracle.com/docs/cd/B28359_01/install.111/b28262.pdf)
- HP-UX—[http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28259.pdf](http://download.oracle.com/docs/cd/B28359_01/install.111/b28259.pdf)
- AIX—[http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28258.pdf](http://download.oracle.com/docs/cd/B28359_01/install.111/b28258.pdf)

Review Chapter 1, “Summary List: Installing Oracle Clusterware,” in the *Oracle Clusterware Installation Guide* before proceeding with this section.

## Oracle Clusterware Prerequisites

Perform the steps described in Chapter 2, “Oracle Clusterware Preinstallation Tasks,” in the *Oracle Clusterware Installation Guide*, using these notes:

- Create the system group `oinstall`, and then create the user `oracle` in the `oinstall` group to start, stop, and check resources.
- Create an Oracle Clusterware home directory. As root on each cluster node, create a path:

```
mkdir -p /u01/app
chown -R oracle:oinstall /vol1/app
```

During installation, you can choose a location for `oraInventory` (owned by user `oracle`) and for Oracle Clusterware; for example, `/vol1/app/oraInventory` and `/vol1/app/11.1.0/crs`.

- Using NTP, ensure that the server clocks are synchronized.
- Ensure that each server has at least two network interfaces (more if the cards are teamed).
- The public and private networks must be created and must be physically distinct, on different subnets. VIPs are not created manually; the only manual task is to create the entries in `/etc/hosts`.
- Define IPs and VIPs, making sure to meet Oracle Clusterware requirements, as shown in [Figure 2](#). Plan a VIP (for example, `10.10.12.98/255.255.254.0`) for Shared Services Web application, with a corresponding DNS entry (for example, `hsscluster`). The subnet of the VIP created and managed by Oracle Clusterware must be the same as the subnet of the physical IP on the interface to which the VIP is assigned.

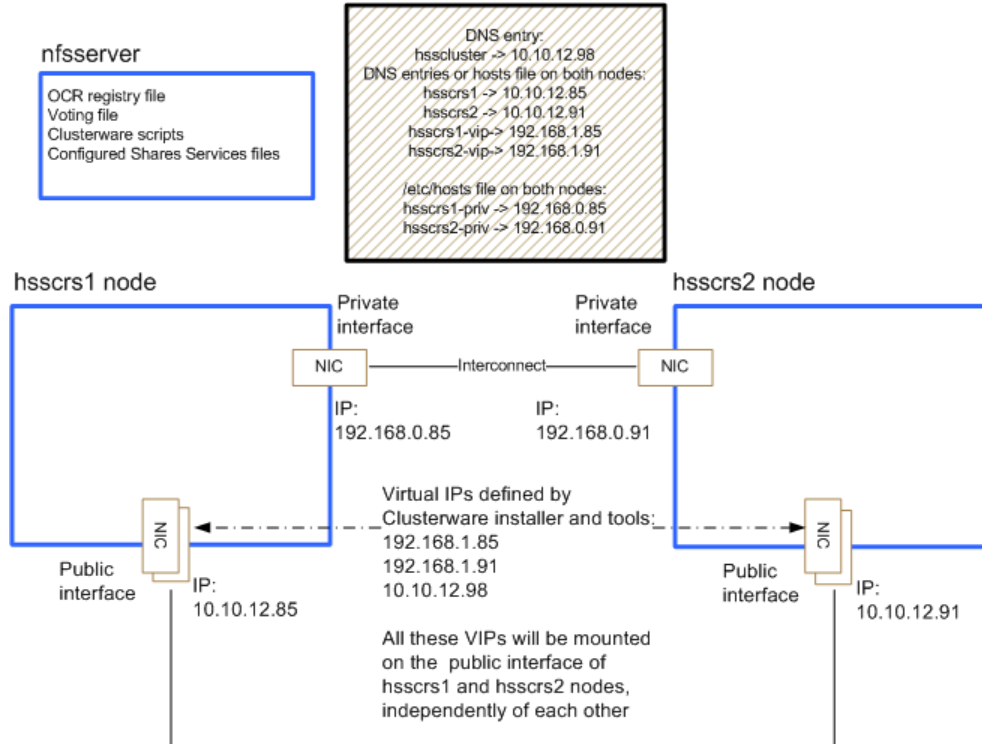
VIPs are not assigned to a fixed machine. When running, a VIP is physically bound to a specific interface on one machine in the cluster. However, Oracle Clusterware requires that each node have one specific VIP. A third VIP is used for Shared Services. Oracle Clusterware migrates the mount point of VIPs when failover occurs.

Example, with `/etc/hosts` on all nodes:

```
#
# Internet host table
#
10.10.12.98          hsscluster    #and dns name
10.10.12.84 nfsserver    #and dns name
#
10.10.12.91          hsscrs1      #and dns name
192.168.0.91         hsscrs1-priv
192.168.1.91         hsscrs1-vip
#
10.10.12.85          hsscrs2      #and dns name
192.168.0.85         hsscrs2-priv
192.168.1.85         hsscrs2-vip
```

**Note:** You need not create the `hsscrs1-vip` and `hsscrs2-vip` interfaces manually. When configuring network interfaces, set TCP/IP parameters only for the public and the private interface, without configuring VIPs.

Figure 2 IP and VIP Definitions



## Configuring Oracle Clusterware Shared Storage

Perform the steps documented under “Configuring Oracle Clusterware Storage” in the *Oracle Clusterware Installation Guide*. The storage for Shared Services and Oracle Clusterware voting disks and OCR uses a separate NFS server in this configuration.

Successful NFS server options found in `/etc/dfs/dfstab`:

```
share -F nfs -o root=hsscrs1:hsscrs2,anon=102 /vol1/sharedcrs
share -F nfs -o root=hsscrs1:hsscrs2,anon=102 /vol1/sharedck2
```

where `anon=102` has the UID of the user `oracle`

The MetaLink document 359515.1, *Mount Options for Oracle Files When Used with NAS Devices*, provides the mount options required. The shared mount point can be created on a network attached storage (NAS) device or on a disk plain partition. However, redundancy is strongly advised for high availability of the physical file system.

This metalink separates binaries, data files and CRS voting disk and OCR mount options:

- For the partition containing the Shared Services binaries and data, use Oracle data files mount options.
- For the partitions containing voting disks and OCR, use CRS voting disks and OCR mount options.

The following example shows how to create the NFS mount point for Oracle Clusterware files OCR and CRS voting disk and the action scripts. This must be performed on all cluster nodes.

```
# mkdir -p /mtk2crs
# chown -R oracle:oinstall /vol1/sharedcrs (to be performed on the nfs
server)
# mount -F nfs -o
rw,hard,bg,nointr,rsiz=32768,wsiz=32768,noac,proto=tcp,vers=3,xattr
nfsserver:/vol1/sharedcrs /mtk2crs
```

When installing Oracle Clusterware, you choose an NFS mounted location for the OCR file (/mtk2crs/ocr) and for the voting file (/mtk2crs/voting).

By contrast, the following example shows how to create the NFS mount point for Shared Services files. This must be performed on all cluster nodes:

```
# mkdir -p /mtk2ss
# chown -R oracle:oinstall /vol1/sharedk2 (to be performed on the nfs
server)
# mount -F nfs -o
rw,hard,bg,nointr,rsiz=32768,wsiz=32768,proto=tcp,vers=3,xattr
nfsserver:/vol1/sharedk2 /mtk2ss
```

**Note:** For performance reasons, it is essential that you skip `noac` mount option when mounting the Shared Services files. Be sure to countercheck the mount options on all cluster nodes, using the mount command as root.

## Installing and Configure Oracle Clusterware

► To install and configure Oracle Clusterware:

**1** Perform the steps documented under “Installing Oracle Clusterware” in the *Oracle Clusterware Installation Guide*.

**2** Ensure that the nodes are secure shell (SSH)-accessible:

a. Run

```
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
```

Start the `runInstaller` command that can launch a X11 GUI. There is no console mode.

```
$ export DISPLAY=x11ip:0.0
$ ./runInstaller &
```

b. There is no console mode.

**3** Launch the Oracle Clusterware installer on one node only, with the user name `oracle`.

Start the `runInstaller` command that can launch an X11 GUI:

```
$ export DISPLAY=x11ip:0.0
$ ./runInstaller &
```

**4** On the **Specify Cluster Configuration** screen, enter the public, private, and virtual host names for both nodes.

**Note:** You must click Add to enter information about `hsscscs2`.

**5** In **Specify Network Interface Usage**, specify the private and public interface.

**6** On the **Cluster Configuration Storage** screen, to ensure that the mount point `/mtk2crs` is mounted, specify these locations:

- OCR external redundancy and OCR location; for example, `/mtk2crs/ocr`
- voting external redundancy and voting location; for example, `/mtk2crs/voting`

**7** Click **Finish**.

**8** From a shell, the Clusterware installation as user `oracle`:

```
$ /vol1/app/11.1.0/crs/bin/cluvfy stage -post crsinst -n hssc1,hssc2
```

**Note:** Post-configuration validation of virtual IPs may fail with Oracle's Hyperion Enterprise Performance Management System Configurator or the `cluvfy` command. This is not a problem if you can see the virtual IP on the nodes using the `ipconfig /all` command.

**9** Add `/vol1/app/11.1.0/crs/bin/` to the `$PATH` statement for the user `oracle`.

**Note:** Postconfiguration validation of virtual IPs may fail with EPM System Configurator or the `cluvfy` command. This is not a problem if you can see the virtual IP on the nodes using the `ifconfig -a` command.

## Creating an Oracle Clusterware Application VIP

Create an application VIP for Shared Services that is started and stopped on the Public interface by Oracle Clusterware. The VIP resource is owned and started by root. See [http://download.oracle.com/docs/cd/B28359\\_01/rac.111/b28255/crschp.htm#sthref369](http://download.oracle.com/docs/cd/B28359_01/rac.111/b28255/crschp.htm#sthref369).

You can use the `$HYPERION_HOME/common/utilities/CRS/hss_clusterware_scripts/createvip.sh` template to create a Virtual IP creation script. Put your script into the NFS mount point directory script path:

```
/mtk2crs/crs_actions/hss/createvip.sh
```

Edit the variables as required:

- VIPIP default—`10.10.12.98`
- VIPSUBNET default—`255.255.254.0`
- CRS\_HOME default—`/vol1/app/11.1.0/crs`
- ADAPTER—The physical interface on which the virtual IP will be mounted

The default is `eri0`.

You can check the adapter using `ifconfig -a`.

Run the script as root `# /mtk2crs/crs_actions/hss/createvip.sh`.



# Oracle Clusterware Postinstallation Procedures

## Shared Services Installation

Use Oracle Hyperion Enterprise Performance Management System Installer, Fusion Edition to install Shared Services on the NFS server, following the instructions in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* and these guidelines:

- Change the NFS server host name to *hsscuster*. On Solaris, for example, edit *etc/hostname.eri0* and */etc/nodename*.
  - Populate */etc/hosts* and ensure that you can ping *hsscuster*: Reboot the system.  
`10.10.12.84 hsscuster` (where `10.10.12.84` is the IP address of the NFS server)
  - Reboot the system (init 6)
- Clean up *\$HOME* directory for user *oracle*. There should exist no entries such as these  
`.hyperion.*`  
`.oracle.*.products`  
`set_hyphome*_1.sh`  
`InstallShield/`

- As user *root*, As user *root*, mount the NFS server on itself:

```
-->mkdir -p /mtk2ss
-->chown -R orclhss:oinstall /vol1/sharedk2
-->mount -F nfs -o
rw,hard,bg,nointr,rsize=32768,wsiz=32768,proto=tcp,vers=3,xattr
nfserver:/vol1/sharedk2 /mtk2ss
```

Make sure the user *oracle* has the correct rights to the */mtk2ss* mount point after the mount has been done. After every reboot, make sure that you mount the NFS mount points if you did not add entries in */etc/dfs/dfstab*.

- As user *oracle*, install Shared Services on the NFS server using the shared NFS client mount point: .

```
$ ./installTool.sh -console
```

For example, if the mount point is */mtk2ss*, you can use */mtk2ss/hyperion* as *HYPERION\_HOME*.

**Note:** Installation and deployment can take significant time if the mount option does not cache files attributes (`skip noac`).

## Configuring Shared Services

Use EPM System Configurator to configure Shared Services on one node of the cluster only. See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

**Prerequisite**—Ensure that no Shared Services processes are running.

► To configure Shared Services:

**1** As user `oracle`, launch the Oracle's Hyperion Enterprise Performance Management System Configurator:

```
$ cd /mtk2ss/hyperion/common/config/9.5.0.0/  
$ ./configtool.sh -console
```

**2** Select **Configure database, Common Settings, and Deploy to Application Server**.

**3** With a new database running, select **Perform 1st time configuration of Shared Services database**.

**4** On the application server deployment screen, click **Advanced Setup**, and enter `hsscluster:28080`, where `hsscluster` is a DNS entry pointing to the Shared Services virtual IP, and ensure that you can ping `hsscluster`.

This step defines the logical name for Shared Services Web application.

**5** Finish the configuration.

**6** Right-click **CSSConfig**, and select **Import**.

See the mount options for Shared Services files described in “[Configuring Oracle Clusterware Shared Storage](#)” on page 6.

**Note:** For performance reasons it is essential to skip `noac` mount option when mounting the Shared Services files.

**7** Start Native Directory and Shared Services Web application, using the start script `$HYPERION_HOME/products/Foundation/bin/start.sh`.

**8** Use Lifecycle Management from Shared Services to edit the `CSSConfig` file in the Oracle's Hyperion Shared Services Registry.

- a. Log on to Shared Services: <http://hsscluster:28080/interop/index.jsp>.
- b. Select **Projects, Foundation**, and **Deployment Metadata**.
- c. Expand **EPM System registry, Foundation Services Product**, and **Shared Services**.
- d. Right-click **CSSConfig**, select **Export for Edit**, and save the file on a local drive.
- e. Change the host for hub location as follows:

```
hub location=http://hsscluster:28080
```

where `hsscluster` is a DNS entry pointing to the Shared Services VIP.

The NFS server host name can be reset. From this point, all the operations regarding Shared Services (registration scripts; start, stop, check in perl action script) are performed on the Clusterware nodes mounting the NFS files:

```
# mkdir -p /mtk2ss  
# chown -R oracle:oinstall /vol1/sharedk2 (to be performed on the nfs  
server)  
# mount -F nfs -o  
rw,hard,bg,nointr,rsize=32768,wsiz=32768,proto=tcp,vers=3,xattr  
nfserver:/vol1/sharedk2 /mtk2ss
```

## Registering Shared Services in the Cluster

You can use the `$HYPERION_HOME/common/utilities/CRS/hss_clusterware_scripts/hss11.pl` action script template to start, stop, check for health the Shared Services processes. Put your scripts in the NFS mount point directory script path: `/mtk2crs/crs_actions/hss/`

You can use the `$HYPERION_HOME/common/utilities/CRS/hss_clusterware_scripts/registerhss.sh` template to create a Shared Services application profile and to register the profile in the Oracle Clusterware registry. The profile points to the perl action script.

Put your scripts in the NFS mount point directory script path:

```
/mtk2crs/crs_actions/hss/
```

Make sure that `perl` is in the `$PATH` statement.

► To register Shared Services in the cluster:

### 1 Check the HSS perl action script:

a. Start `hssvip` on one node:

```
# crs_start hssvip
# crs_stat -t -v
```

Check on which node `hssvip` runs.

b. Edit the `hss11.pl` action script provided in the Appendix to adapt it for your configuration. Check it works before registering in a cluster:

i. On the node where the VIP is mounted, as user `oracle`, launch from a shell:

```
$ perl hss11.pl start
$ echo $?
0 #denotes success
```

ii. Check the other commands:

```
$ hss11.pl check
$ echo $?
0 #denotes success of the Shared Services processes
1 #denotes failure of the Shared Services processes
$ hss11.pl stop
```

### 2 Create and register an application profile for Shared Services.

● Edit the variables in `registerhss.sh` as required:

```
ACTION_SCRIPT=/mtk2crs/crs_actions/hss/hss11.pl
CRS_HOME=/vol1/app/11.1.0/crs
```

● Ensure that Shared Services is started under the correct user. The process that is started on the behalf of the resource runs with the user specified through the `crs_setperm` command. The default script uses the user name `oracle`:

```
$CRS_HOME/bin/crs_setperm $APPNAME -o oracle
$CRS_HOME/bin/crs_setperm $APPNAME -g oinstall
```

● Run the script as root # `/mtk2crs/crs_actions/hss/registerhss.sh`.

## Managing the Shared Services Cluster

You use Oracle Clusterware to manage the Shared Services cluster. You can control Oracle Clusterware manually with the Command Line tool `crsctl`. You use `crsctl` commands to start and stop Oracle Clusterware. Using `crsctl` options, you can perform tasks such as enabling online debugging and dynamically adding, removing, and backing up voting disks.

Put `/vol1/app/11.1.0/crs/bin/` in user's `$PATH` statement.

### Checking the Cluster

Use the command `crs_stat -t -v` to check the cluster status. Running `crs_stat -t -v` provides the status of the cluster resource, including Shared Services and VIPs.

Perform login checks from a Web browser, using <http://hsscluster:28080/interop/index.jsp>, to ensure that Shared Services runs correctly.

### Monitoring the Cluster

The logical application name registered in the cluster with `registerhss.sh` is `hsswebapp`, and resources monitored include Native Directory and Web application. The following commands are available to for monitoring your Shared Services cluster, where `hsswebapp` is the application name:

- `# crsctl start crs #`—Starts Oracle Clusterware and related resources on the current node.
- `# crsctl stop crs #`—Stops Oracle Clusterware and related resources on the current node.

If the other node is running, all applications that are managed by Oracle Clusterware are relocated from the current node to the other node. If the command is launched on all nodes, all applications and services are stopped.

- `$ crsctl check crs #`—Verifies the health of `crs` and related daemons on the node from which you are checking.
- `$ crs_start hsswebapp #`—Starts the process related to the specified application. This causes Oracle Clusterware to call the action programs with the start parameter.

---

**Caution!** Start the application only with this command.

---

You can start the Oracle Clusterware registered application using the option `-f` (force) when there is relationship between applications (for example, Shared Services and Shared Services VIP):

```
$ crs_start -all -f
```

- `$ crs_stat hsswebapp #`—Calls the action programs (such as `hss11.pl`) with the check parameter. Using the option `-t` or `-v`, you can query the state of all Oracle Clusterware registered applications to view additional information about their status.
- `$ crs_stop hsswebapp #`—Stops the application. This causes Oracle Clusterware to call the action programs with the stop parameter.

---

**Caution!** Use only this command to stop the application. When you use this command for a specific application, its target status is changed to Offline, and the application stays offline after an Oracle Clusterware startup.

---

You can stop the Oracle Clusterware registered application using the option `-f` (force) when there is relationship between applications (for example, Shared Services and Shared Services VIP).

**Note:** Oracle Clusterware does not restart the application in case of a failure.

When you stopping the application using this command, the action program is called with the stop parameter, and the related application services are shut down on the server where you are working.

- `$ crs_relocate hsswebapp -f -c hsscrs2 #`—Stops the specified application and starts it on another node in the cluster.

The `crs_start`, `crs_stop`, and `crs_stat` commands become unavailable when the `crsd` daemon is stopped.

## Oracle Clusterware Backup and Recovery

These Oracle Clusterware components should be backed up so you can recover them in case of failure:

- Voting disk—Manages information about node membership.
- OCR—Manages cluster and configuration information.

Oracle recommends that you back up the voting disk and OCR immediately after the initial installation.

See the “Administering Oracle Clusterware” section in the *Oracle Clusterware Administration and Deployment Guide*, which you can download from [http://download.oracle.com/docs/cd/B28359\\_01/rac.111/b28255/votocr.htm](http://download.oracle.com/docs/cd/B28359_01/rac.111/b28255/votocr.htm).

## Log Files

For log files location, see the “Troubleshooting” section of the *Oracle Clusterware Administration and Deployment Guide*, which you can download from [http://download.oracle.com/docs/cd/B28359\\_01/rac.111/b28255/troubleshoot.htm](http://download.oracle.com/docs/cd/B28359_01/rac.111/b28255/troubleshoot.htm).

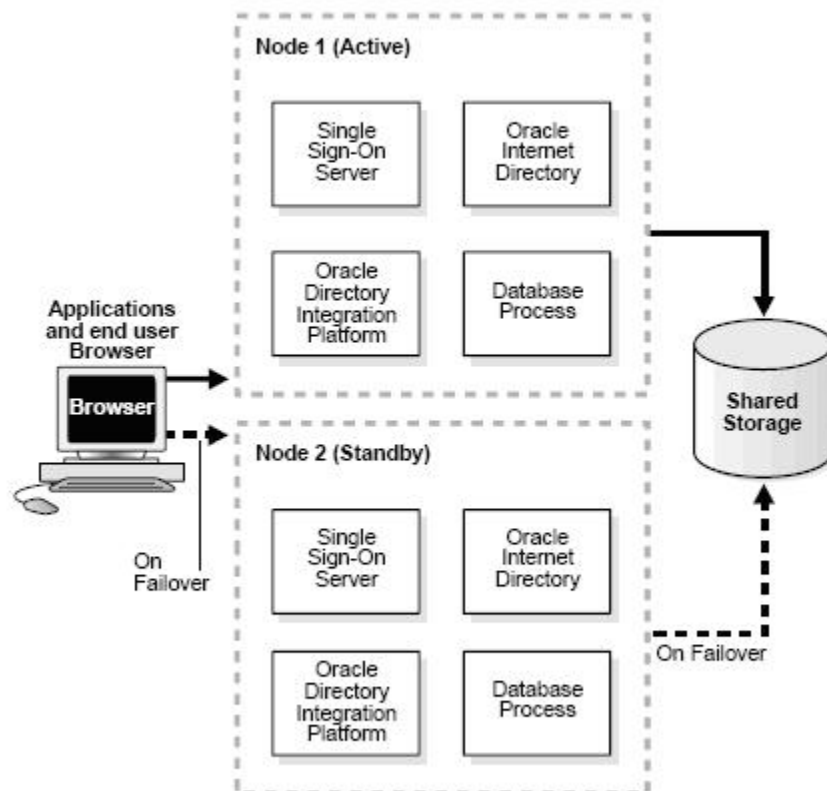
## Oracle Internet Directory Clustering

You can cluster Shared Services OID for high availability. OID is supported as native provider in any active-passive mode that OID supports.

You can use several techniques for making Oracle Internet Directory highly available. References:

- <http://www.oracle.com/technology/products/oid/pdf/oid-largescaledirectory-ha-performance.pdf>
- [http://www.oracle.com/technology/products/oid/pdf/oid\\_tuning\\_configuration\\_quickreference\\_01.pdf](http://www.oracle.com/technology/products/oid/pdf/oid_tuning_configuration_quickreference_01.pdf)

Figure 3 Oracle Internet Directory Deployment Using Cold Failover



## Tips and Tricks

- Please see the Oracle Clusterware documentation for alternate shared storage technologies. Other techniques such as setting up the nodes to act as NFS servers, or using physical shared storage, may work. The OCFS file system is not available on all UNIX flavors. This document describes a detailed NFS setup that is supported reliably and that one can port easily across UNIX flavors. Regardless of which file storage implementation you use, you must observe the three rules described in “[Strategy for Deploying Shared Services in a Failover Cluster on UNIX](#)” on page 2:
  - Floating host name and VIP (*hsscluster*)
  - Identical HSS paths during HSS installation/configuration and HSS runtime operation (mount point)
  - Nonprivileged user (*oracle*)
- The commands syntax in this document derive from Solaris 10 experience (no zones so far). Visit ROSETTA STONE (<http://bhami.com/rosetta.html>) for the other UNIX dialects.
- The installation and configuration of Shared Services is performed as user *oracle*/ group *oinstall*. User *oracle* \$HOME on both Oracle Clusterware nodes should contain this setting:

```
hyperion.hsscluster , .hyperion.`hostname` #identical, copy the
missing file(s) set_hyphome_hsscluster_1.sh ,
set_hyphome_`hostname`_1.sh #identical, copy the missing file(s)
```
- The NFS server and the Oracle Clusterware nodes must use identical UID/GID for *oracle/oinstall*.
- The NFS server and the Oracle Clusterware nodes must show the same time, time zone, and date.
- Keep the *oracle* user's `ulimit -n 1024`, especially during Shared Services configuration.
- The procedure directs changing the NFS server host name to *hsscluster* during Shared Services installation and configuration. The reliable approach (`vi /etc/nodename`) requires a system reboot.

An entry *hsscluster* must be resolved in */etc/hosts* by itself, preferably on top:

```
10.10.12.98 hsscluster
```

An alternate approach using `# uname -S hsscluster` dynamically may fail during the installation phase of OpenLDAP. (Installation freezes at `slapadd`).

With the alternate approach, *hsscluster* should be added at the head of the line entry in */etc/hosts*:

```
10.10.12.84 hsscluster nfsserver nfsserver.com loghost
(10.10.12.84 is the nfsserver IP).
```

- Keep system-level practice of using symbolic links such as */etc/hosts* -> */etc/inet/hosts*.
- To clear (invalidate) the system caches use `# /usr/sbin/nscd -i hosts ipnodes`.

- Ensure that no prior OpenLDAP process runs during installation and configuration of Shared Services:

```
$ netstat -an | grep 280
```

Before configuration, the installed OpenLDAP data files consist of a limited set of `__db.00?`, a lock, and a few `.bdb` files. The configuration creates the remaining `.bdb` and the redo log file.

Check the `oracle/oinstall` ownership of the files `__db.00?`:

```
-rwxrwxr-x 1 oracle oinstall 8192 Sep 25 02:36 __db.001
```

The directory location is `/mtk2ss/hyperion/products/Foundation/openLDAP/usr/local/var/openldap-data/`.

- When configuring the Shared Services database, perform “1st time configuration of Shared Services database” on a new database. Do not perform “Connect to previously configured Shared Services database.”
- Ensure that the NFS server is available from the Oracle Clusterware nodes before mounting the client:

```
# dfshares nfsserver
```

The “hard” mount option makes it difficult to interrupt a failing mount operation (reboot).

- Successful NFS server options found in `/etc/dfs/dfstab`:

```
share -F nfs -o root=hsscscs1:hsscscs2,anon=102 /vol1/sharedcrs
share -F nfs -o root=hsscscs1:hsscscs2,anon=102 /vol1/sharedk2
```

where `anon=102` has the UID of the user `oracle`.

- The NFS client mount options may include `forcedirectio` as a performance flag.
- The NFS client mount options for Shared Services should not include `noac`. Including `noac` has an adverse effect on performance.
- Back up the fresh OCR and voting files on the NFS server when the Oracle Clusterware installation is completed.
- Moving the Clusterware OCR and voting files to another NFS server succeeds if the file rights and ownerships are preserved (root for ocr):

```
# ls -l /vol1/sharedcrs/
total 566088
drwxrwxrwx  3 root    root          512 Sep 26 08:32 ./
drwxr-xr-x  9 root    root          512 Sep 22 12:39 ../
drwxrwxrwx  2 root    root          512 Sep 25 06:44 crs_actions/
-rw-r--r--  1 root    root        8704 Sep 26 08:32
crs_actionsSolaris.tar
-rw-r--r--  1 root    oinstall 268644352 Sep 25 05:47 ocr
-rw-r--r--  1 oracle  oinstall 21004288 Sep 25 06:44 voting
```

- Moving the Shared Services files to another NFS server succeeds if the file rights and ownerships are preserved:

```
# ls -l /vol1/sharedk2/
total 6
drwxrwxrwx  3 root    root          512 Sep 26 08:29 ./
```



```
drwxr-xr-x  9 root    root          512 Sep 22 12:39 ../
drwxrwx---  9 oracle  oinstall    512 Sep 25 04:26 hyperion/
```

- Startup of Oracle's Hyperion® Shared Services processes takes a while. Wait until ports 28081 28082 come up.

```
$ netstat -an | grep 280
```

- In keeping with the UNIX tradition, the administrator or root prompt is #, and the oracle nonprivileged user prompt is \$.
- The only administrator-accessible start (stop) script is

```
# /etc/init.d/init.crs start (stop) (Solaris link /etc/rc3.d/S96init.crs
```

It starts other scripts and the entire stack of Oracle Clusterware processes.

The supported way of enabling (disabling) the Oracle Clusterware stack's automatic startup is

```
# /etc/init.d/init.crs enable (disable)
```

## Script Template for createvip.sh

```
#!/bin/sh
APPVIPNAME=hssvip
ACTIONSRIPT=/vol1/app/11.1.0/crs/bin/usrvip
ADAPTER=eri0
VIPIP=10.10.12.98
VIPSUBNET=255.255.254.0
START_TIMEOUT=90
STOP_TIMEOUT=90
SCRIPT_TIMEOUT=90
RESTART_ATTEMPTS=2
CRS_HOME=/vol1/app/11.1.0/crs
# profile application VIP
$CRS_HOME/bin/crs_profile -create $APPVIPNAME -t application -a
$ACTIONSRIPT -o rt=$START_TIMEOUT,pt=$STOP_TIMEOUT,st=$SCRIPT_TIMEOUT,ra=
$RESTART_ATTEMPTS,oi=$ADAPTER,ov=$VIPIP,on=$VIPSUBNET
# register application VIP
$CRS_HOME/bin/crs_register $APPVIPNAME
# application VIP address script must run as the root user
$CRS_HOME/bin/crs_setperm $APPVIPNAME -o root
# enable the oracle user to run application VIP
$CRS_HOME/bin/crs_setperm $APPVIPNAME -u user:oracle:r-x
# see permissions
$CRS_HOME/bin/crs_getperm $APPVIPNAME
# start the resource
# $CRS_HOME/bin/crs_start $APPVIPNAME
```

## Script Template for hssregister.sh

```
#!/bin/sh
APPNAME=hsswebapp
ACTION_SCRIPT=/mtk2crs/crs_actions/hss11.pl
DESCRIPTION=HyperionSharedServices
```

```

REQUIRED_RESSOURCES=hssvip
FAILOVER_DELAY=0 #60
FAILURE_THRESHOLD=0 #2 #5
FAILURE_INTERVAL=0 #600
START_TIMEOUT=90
STOP_TIMEOUT=90
SCRIPT_TIMEOUT=90
RESTART_ATTEMPTS=2
CRS_HOME=/vol1/app/11.1.0/crs
# profile application
$CRS_HOME/bin/crs_profile -create $APPNAME -t application -a $ACTION_SCRIPT
-d $DESCRIPTION -r $REQUIRED_RESSOURCES -o fd=$FAILOVER_DELAY,ft=
$FAILURE_THRESHOLD,fi=$FAILURE_INTERVAL,rt=$START_TIMEOUT,pt=
$STOP_TIMEOUT,st=$SCRIPT_TIMEOUT,ra=$RESTART_ATTEMPTS
# register application
$CRS_HOME/bin/crs_register $APPNAME
# run application as the oracle user
$CRS_HOME/bin/crs_setperm $APPNAME -o oracle
$CRS_HOME/bin/crs_setperm $APPNAME -g oinstall
# see permissions
$CRS_HOME/bin/crs_getperm $APPNAME
# start the resource
# $CRS_HOME/bin/crs_start $APPNAME

```

## Action Script Template for hss11.pl

```

#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
#
#      NAME
#      hss11.pl
#
#      DESCRIPTION
#      This perl script is the action script for start / stop / check
#      the Shared Services and Openldap in a cold failover configuration.
#
#      Usage: hss11.pl [start | stop | check]
#      ebelmon 06/26/08 - Creation
#      tdubois 08/29/08 - Update EPMS, pass $HOME to start/stop scripts

# Environment settings please adapt

my $SCRIPT_PATH      = "/mtk2crs/crs_actions";
my $VER              = "1.0";
my $HSS_URL          = "http://hsscluster:28080/interop/index.jsp";
my $RESP_TIMEOUT    = "40"; #If $HSS_URL does not respond in RESP_TIMEOUT
my $LDAP_HOST       = "hsscluster";
my $LDAP_PORT       = 28089;

my $home = glob ("~"); #save home directory of caller
my $HYPERION_HOME = "/mtk2ss/hyperion/";
my $HSS_STARTSCRIPT = "$HYPERION_HOME/products/Foundation/bin/start.sh";
my $HSS_STOPSCRIPT = "$HYPERION_HOME/products/Foundation/bin/stop.sh";
my $CRS_ACTION_LOG = "$SCRIPT_PATH/hss11.log";
# Please increase the sleep interval, especially if starting up the hss
processes take more than $SLEEP_INTERVAL

```

```

my $SLEEP_INTERVAL = 45;

my $LOG_ENABLED      = "true";

#my $DEBUG_ENABLED  = "true";
my $DEBUG_ENABLED    = "false";

if ($#ARGV != 0 ) {
    print "usage: hss11.pl [start | stop | check] \n";
    exit;
}

$command = $ARGV[0];

# hss web app and native directory start stop check

# Start Shared Services web application and openldap
if ($command eq "start" ) {
    ("Starting Shared Services");

    system (" HOME=$home ; export HOME ; $HSS_STARTSCRIPT>/dev/null
2>&1 ");
    sleep $SLEEP_INTERVAL;
    exit 0;
}

# Stop Shared Services
if ($command eq "stop" ) {
    ("Stopping Shared Services");

    system (" HOME=$home ; export HOME ; $HSS_STOPSCRIPT >/dev/null
2>&1 ");
    sleep $SLEEP_INTERVAL;
    exit 0;
}

# Check Shared Services web app and openldap
if ($command eq "check" ) {

    if ($DEBUG_ENABLED eq "true") { (" Check command for HSS Native
directory"); }
    my $ldap_check=ldap_check();
    if ($DEBUG_ENABLED eq "true") { ("Result openLDAP:$ldap_check"); }

    if ($DEBUG_ENABLED eq "true") { (" Check command for HSS web application
url $HSS_URL"); }
    my $http_check= http_check();
    if ($DEBUG_ENABLED eq "true") { ("Result HSS:$http_check"); }

    if ($http_check == 0 && $ldap_check ==0) {
        exit 0;
    } else {
        exit 1;
    }
}

#makes sense only on the node that runs slapd

```

```

sub ldap_check {
    my $process = "openLDAP/usr/local/libexec/slapd";
    my $check_proc = qx(ps -aef | grep slapd | grep -v grep | awk '{print
$8}');
    chomp($check_proc);
    if ($DEBUG_ENABLED eq "true") { ("check_proc: $check_proc"); }

    if ($check_proc=~ /$process/) {
        return 0;
    } else {
        return 1;
    }
}

sub http_check {
# /usr/sfw/bin/wget --delete-after --timeout=20 --tries=1 http://
hsscluster:28080/interop/index.jsp | grep -c "200 OK"

    my $res = qx(/usr/sfw/bin/wget --delete-after --timeout=$RESP_TIMEOUT --
tries=1 $HSS_URL 2>&1 | grep -c "200 OK");
    chomp($res);
    if ($DEBUG_ENABLED eq "true") { ("http number of 200 OK:$res"); }

    if ($res >= 1) {
        return 0;
    } else {
        return 1;
    }
}

sub log {
    if ($LOG_ENABLED eq "true") {
        open LOGFILE , ">>$CRS_ACTION_LOG";
        my $now=localtime(time);
        print LOGFILE "$now - $_[0]\n";
        close LOGFILE
    }
}

```

## Additional Information

*Oracle® Clusterware Administration and Deployment Guide*

11g Release 1 (11.1)

B28255-04

August 2008

[http://download.oracle.com/docs/cd/B28359\\_01/rac.111/b28255.pdf](http://download.oracle.com/docs/cd/B28359_01/rac.111/b28255.pdf)



## COPYRIGHT NOTICE

Shared Services Active-Passive Failover Clusters (UNIX Environments), 11.1.1.X

Copyright © 2008, 2009, Oracle and/or its affiliates. All rights reserved.

Authors: Eric Belmon and Thierry Dubois

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable: U.S. GOVERNMENT RIGHTS: Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.