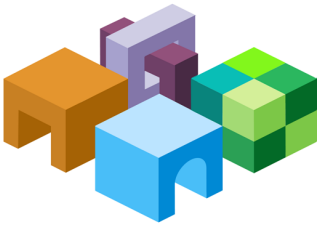


ORACLE® ESSBASE

HIGH AVAILABILITY DEPLOYMENT USING ORACLE CLUSTERWARE

ORACLE ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

Contents In Brief

Overview

This document provides instructions on setting up an Active/Passive (failover) configuration of Oracle Essbase using Oracle Clusterware (OCW). A clustered setup allows configuring two nodes running Essbase services so that they cooperate as a single system. In the event of a node failure, OCW automatically fails over and starts all Essbase services on the passive node . Virtual IP (VIP) failover allows a single floating IP to be bound to the active node in the cluster which then, fails over in case of a node crash.

Supported Platforms

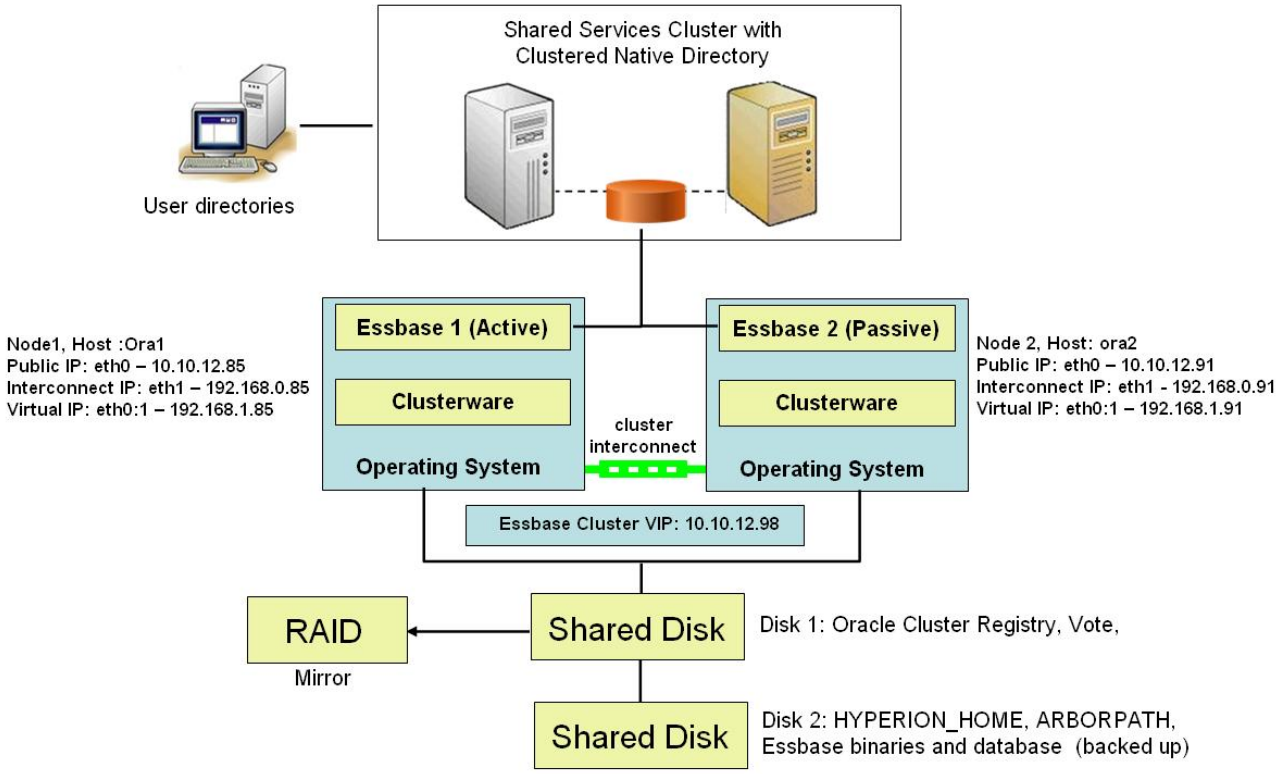
This white paper describes how to setup Essbase in an Active/Passive configuration on UNIX platforms. See Oracle Enterprise Performance Management Products - Supported Platforms Matrices.

Recommended Architecture

Essbase service deployment is supported on a two-node topology in active/passive mode where one node (Node 1) is active and the other node (Node 2) is passive. Essbase services are accessible from a specific floating VIP address referenced by a host name that is DNS resolvable. If the active node fails, the VIP and all shared processes automatically move to the passive node.

The supported two-node deployment topology is depicted in the following illustration.

Figure 1 Recommended Architecture for Essbase High Availability Deployment



Note: This architecture was tested on a two-node Linux cluster running on Intel x86 using Storage Area Network (SAN) disks (Disk1 and Disk2) running OCFS2. For CRS certification matrix on other hardware and platforms, see Certify - Oracle's Certification Matrices, Desupport Notices & Product Availability.

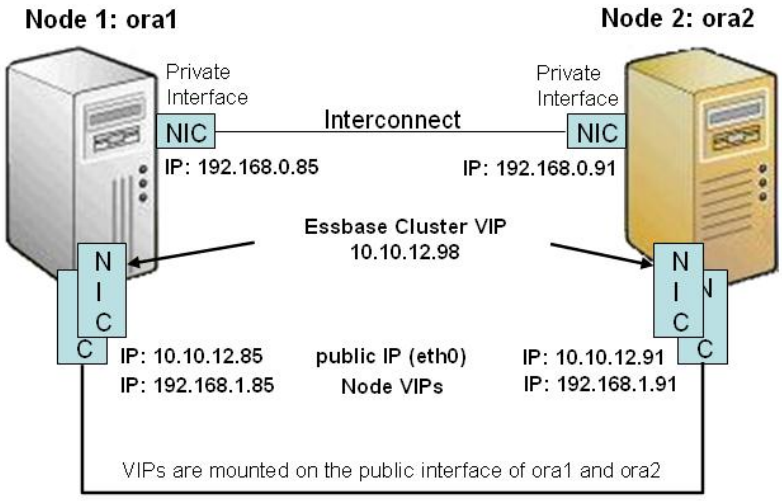
This architecture uses two hosts as clustered nodes. You can use physical hardware, or Virtual Machines running on redundant physical hardware as nodes. Node 1 is active and Node 2 is passive. Each node has a private interconnect, public IP, and Virtual IP. Cluster Ready Services is deployed on both nodes.

Disk 1, is a shared disk that hosts Oracle Cluster Registry (OCR). The OCR maintains cluster configuration information and manages information about processes that Oracle Clusterware controls. Disk 1 is also the voting disk, which manages cluster membership using a health check, and arbitrates cluster ownership among the instances in case of network failures. Because this architecture defines a single voting disk, an external RAID mirror is used to provide redundancy. Disk 2 hosts *HYPERION_HOME*, *ARBORPATH*, and Essbase binaries.

To ensure high availability, Oracle's Hyperion® Shared Services is clustered. In the sample architecture, Shared Services, including OpenLDAP, is clustered for high availability in a separate system.

The illustrated technical architecture:

Figure 2 High Availability Technical Architecture



Information Sources

- *Oracle Clusterware 11g* white paper http://www.oracle.com/technology/products/database/clusterware/pdf/TWP_Clusterware_11g.pdf
- *Using Oracle Clusterware to Protect 3rd Party Applications* white paper http://www.oracle.com/technology/products/database/clusterware/pdf/TWP-Oracle-Clusterware-3rd-party.pdf
- The "Making Applications Highly Available Using Oracle Clusterware" section of the *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* http://download.oracle.com/docs/cd/B19306_01/rac.102/b14197/crschp.htm
- The "Using CRSCTL to Diagnose Cluster Issues" section in the "Monitoring and Troubleshooting chapter of *Oracle Database 2 Day + Real Application Clusters Guide 10g Release 2 (10.2)*
- For CRS certification matrix, see Certify - Oracle's Certification Matrices, Desupport Notices & Product Availability.
  - In Certify Web site, click Select `View Certifications by Product`.
  - In Select one of the Product Groups Listed Below, select Real Application Cluster, and then select Submit.
- *Oracle Real Application Clusters in Oracle VM Environments* white paper http://www.oracle.com/technology/products/database/clusterware/pdf/oracle_rac_in_oracle_vm_environments.pdf.

Prerequisites

- "Oracle Clusterware" on page 5

- "Shared Services" on page 6
- "Essbase" on page 6

Note: Oracle recommends that you create operating system user `oracle` to perform all installation and deployment activities. Instructions in this document assume that you are using the `oracle` user account.

Oracle Clusterware

Install and configure Oracle Database 11*g* Clusterware on the two node cluster (CRS binaries are deployed on the local disk which is not included in the architecture diagram, Figure 1). Dedicated network interconnect is used to maintain the integrity of the nodes in the cluster.

See the *Using Oracle Clusterware to Protect 3rd Party Applications* white paper at http://www.oracle.com/technology/products/database/clusterware/pdf/TWP-Oracle-Clusterware-3rd-party.pdf for information.

If you are using virtual machines as cluster nodes, oracle recommends that you perform the following tasks:

1. Set `diagwait` to 13 seconds. See "Setting Diagwait for Clusterware Process Monitor Daemon" on page 5.
2. Address clock drift in the domain. See "Address Clock Drift" on page 6.

Setting Diagwait for Clusterware Process Monitor Daemon

The Oracle Clusterware Process Monitor Daemon (`oprocd`) process is started automatically to detect and restart hung nodes. Because of wide variations in scheduling latencies across operating systems, `oprocd` default value may cause unnecessary `oprocd`-initiated restarts (false restarts).

To overcome these scheduling latencies, Oracle recommends that you set the value of the Oracle Clusterware parameter `diagwait` to 13. This setting increases the time for failed nodes to flush final trace files, which helps to debug the cause of a node failure. You must shut down the cluster to change the `diagwait` setting.

➤ To change the `diagwait` setting:

1 **Log in as root, and stop CRS by running the following command:**

ORA_CRS_HOME`/bin/crsctl stop crs`

2 **On each node, run the following command to stop** `oprocd`**:**

ORA_CRS_HOME`/bin/oprocd stop`

3 **Log into a cluster node as** `root` **and change the value of** `diagwait` **to 13 seconds by executing the following command:**

ORA_CRS_HOME`/bin/crsctl set css diagwait 13 -force`

4 **Restart the Oracle Clusterware by executing the following command on all nodes:**

ORA_CRS_HOME/bin/crsctl start crs

5 **Execute the following command to verify that Oracle Clusterware is functioning properly:**

ORA_CRS_HOME/bin/crsctl check crs

Address Clock Drift

By default, VM time clocks are synchronized to the clock in the control domain (dom-0). There may be a clock drift between VM guests (DomU) and the VM Manager (dom-0). To avoid this, the VM guests should derive their time directly from dom-0. For more information, see Metalink Note 580296.1, *Clock drift issue between Dom-0 and Dom-U on OVM Server*.

➤ To address clock drift in VMs:

1 **Log into the VM host as `root`.**

2 **Configure `xen.independent_wallclock=1` in `/etc/sysctl.conf` to allow the VM host to manage its own time clock.**

3 **Configure Network Time Protocol Daemon (NTPD) in dom-0.**

Shared Services

Oracle's Hyperion® Shared Services and Native Directory must be deployed for high availability and failover. See these white papers:

- Shared Services and Reporting and Analysis High Availability (UNIX Environments)
- Active-Passive Failover Clusters (UNIX Environments)

Use Oracle Hyperion Enterprise Performance Management System Lifecycle Management to migrate existing users from a previous version to this version. See the *EPM System Lifecycle Management Guide*.

Essbase

Install and configure Essbase on Node 1 and Node 2. See EPM System Installation and Configuration Guide. Use Disk 2, the data disk (see Figure 1 for suggested architecture), as the location of *ARBORPATH*.

Perform these steps after installing Essbase.

- Configure Essbase services using Oracle's Hyperion Enterprise Performance Management System Configurator.
- Externalize Essbase to use Oracle Hyperion Enterprise Performance Management System security. You can use MaxL or the Oracle Essbase Administration Services Console to externalize Essbase. See the Oracle Essbase Database Administrator's Guide.
- Migrate existing Essbase users, if necessary.

Note: High availability support is not available for Administration Services. You can use Oracle Essbase Administration Services Console to administer the Essbase server on Node 1 or Node 2 by modifying connection settings.

Assumptions

- *ORA_CRS_HOME* indicates the location of Oracle Clusterware home; for example, `/installs/oracle/crshome`.
- All installation and configuration tasks were completed by operating system user `oracle`.

Placing Essbase Under Oracle Clusterware Protection

Creating a VIP For Essbase

You must complete a four-step process to place Essbase under Oracle Clusterware protection.

Note: Oracle recommends that you use a dedicated IP that is DNS-resolvable as the Essbase VIP. This VIP will float between the nodes during failover or switchover.

Clients use the VIP to locate Essbase irrespective of the node it is running on. Oracle provides a script to create application VIPs, which exist as protected resources.

The following procedure uses these variables, which must be modified for your environment:

- *ORA_CRS_HOME* is the location of Oracle Clusterware home
- `eth0` is the name of the public network adapter
- `ov=10.10.12.98` is the virtual IP that floats between the cluster nodes. This IP should resolve through DNS to `essbasevip.example.com`.
- `on=255.255.254.0` is the subnet mask of the IP

➤ To place Essbase under Oracle Clusterware protection:

1 **Log on to Node 1 as `oracle` (user account that was used to install Oracle Clusterware).**

2 **Create a CRS profile for Essbase VIP by executing this command:**

```
ORA_CRS_HOME/bin/crs_profile –create myvip -t application -a
ORA_CRS_HOME/bin\usrvip -o oi=eth0,ov=10.10.12.98,on=255.255.254.0
```

3 **Register the Essbase VIP profile with Oracle Clusterware by executing this command:**

ORA_CRS_HOME/bin/crs_register myvip.

4 **Grant ownership of the profile to `root` to allow the VIP script to run as `root`.**

a. Log in as `root`.

b. Execute this command:

```
ORA_CRS_HOME/bin/crs_setperm myvip –o root
```

5 **As `root`, grant permissions to `oracle` (user account that was used to install Oracle Clusterware) to execute the VIP script.**

```
ORA_CRS_HOME/bin/crs_setperm myvip –u user:oracle:r-x
```

6 **Start the VIP.**

a. Log in as `oracle`, (user account that was used to install Oracle Clusterware).

b. Execute this command:

```
ORA_CRS_HOME/bin/crs_start myvip
```

Verify that you get a message such as the following indicating that the VIP has been started on Node 1.

```
Attempting to start `myvip` on member `ora1`
Start of `myvip` on member `ora1` succeeded.
```

Creating an Action Program

Oracle Clusterware uses an action program to interact with Essbase. The action program provides a method for starting, stopping, and checking Essbase status.

You may write the action program; for example, `essbaseappcheck`, in any scripting language. The action program should accept an option: start, stop, or check. Successful execution of the start/stop option should return 0; else, `essbaseappcheck` should return 1. Similarly, the action program should return 0 if the check option discovers that the process is running; else it should return 1. Compile and save the action program in a convenient directory; that is accessible to all cluster nodes; for example, `/Disk2/scripts`.

See "Sample Action Script" on page 10 for a sample action program.

See Appendix B in the Using Oracle Clusterware to Protect 3rd Party Applications white paper for the source code of the action program used to check Oracle Time Date Service.

Creating Essbase Application Profile

The application profile is a simple text file with some name-value keypairs. To create a syntactically correct Essbase application profile file, run the `crs_profile` utility. The following procedure creates `essbase.cap` in *ORA_CRS_HOME*`/crs/public` directory.

➤ To create Essbase application profile:

1 **Log on to Node 1 as `oracle` (user account that was used to install Oracle Clusterware).**

2 **Execute this command:**

```
ORA_CRS_HOME/bin/crs_profile
-create myEssbase -t application -d "Essbase_Server " -r myvip
-a /Disk2/scripts/essbaseappcheck
-o ci=5,ra=3
```

In this command:

- `myEssbase` indicates the application name as stored in OCR.
- `Essbase_Server` is the long name of the application.
- `myvip` is the Essbase VIP (resource that must be online for the application to start). This forces resource `myvip` to be running before resource `myEssbase` (`Essbase_Server`) can start.
- `/Disk2/scripts/essbaseappcheck` is the name and location of the action program.
- `ci=5` is the check interval.
- `ra=3` is the number of restart attempts.

Registering the Essbase Application Profile with Oracle Clusterware

Use the `crs_register` command to register the Essbase application profile. This process registers the Essbase application with Oracle Clusterware and creates OCR entries for Essbase allowing Oracle Clusterware to control and manage Essbase.

When run by a non root user, `crs_register` checks *`ORA_CRS_HOME`*`/crs/public` for the application profile; for example, `essbase.cap`, which you created earlier. See "Creating Essbase Application Profile" on page 8.

➤ To register the Essbase application profile:

1 **Log on to Node 1 as `oracle` (user account that was used to install Oracle Clusterware).**

2 **Execute this command:**

```
ORA_CRS_HOME/bin/crs_register myEssbase
```

In this command, `myEssbase` is the application name as stored in OCR.

Update Essbase Configuration File

Update *`ARBORPATH`*`/bin/essbase.cfg` to identify the cluster. You do this by including a new entry such as `NODENAME` *`Server_Name`*, for example,

```
NODENAME essbasevip.example.com
```

➤ To update Essbase configuration file:

1 **Using a text editor, open *`ARBORPATH`*`/bin/essbase.cfg`; for example, `/installs/Hyperion/products/Essbase/EssbaseServer/bin/essbase.cfg`**

2 **Append cluster information in** `NODENAME` *`Server_Name`* **format; where** *`Server_Name`* **is the host name of the NIC used by the CRS defined Virtual IP. For example,**

```
NODENAME essbasevip.example.com
```

See Figure 2.

3 **Save and close** `essbase.cfg`.

Managing Essbase Protected by Oracle Cluster

Several CRS commands are available to manage an Essbase cluster that is protected by Oracle Clusterware.

Table 1 CRS Commands to Manage Essbase Cluster

Command	Description
ORA_CRS_HOME`/bin/crs_stat myEssbase`*	Queries the status of the Essbase application.
ORA_CRS_HOME`/bin/crs_stat –t –v`	Queries the status of all registered applications
ORA_CRS_HOME`/bin/crs_stat –v myEssbase`	Queries the status of Essbase application. `crs_stat –v` returns additional information such as number of times the application failed over, and the number times the application failed
ORA_CRS_HOME`/bin/crs_start myEssbase`	Starts the Essbase application.
ORA_CRS_HOME`/bin/crs_relocate -f myEssbase`	Relocates the Essbase application to another node in the cluster
ORA_CRS_HOME`/bin/crs_stop myEssbase`	Stops the Essbase application
ORA_CRS_HOME`/bin/crs_unregister myEssbase`	Removes the Essbase application from Oracle Clusterware protection

*It is assumed that the name of the Essbase application in CRS is `myEssbase`

Sample Action Script

Modify this sample script to reflect the settings for your environment. You must modify the value of the action script properties indicated in Table 2:

Table 2 Essbase Settings in the Action Script

Property	Description
ESSSERVERNAME	Server where Essbase is running; for example, `essbasevip.example.com`.
ESSUSERNAME	User account that is to be used to use to perform Essbase operations; for example, `admin`.
ESSPASSWORD	Password of the user account that is to be used to perform Essbase operations; for example, `password`.
HYPERION_HOME	Directory where Oracle Hyperion Enterprise Performance Management System products are installed; for example, `/installs/Hyperion`.

Property	Description
ARBORPATH	Directory where Essbase server is installed; for example, `/installs/Hyperion/products/Essbase/EssbaseServer`.
ESSBASEPATH	Directory where Essbase server is installed; for example, `/installs/Hyperion/products/Essbase/EssbaseServer`
AGENTHANGDELAY	Time (in seconds) that the action script should wait before terminating an unresponsive Essbase instance to which a check command was issued by CRS.
AGENTSHUTDOWNDELAY	Time (in seconds) that the action script should wait to terminate an Essbase instance to which a Stop command was issued by CRS.

Caution! In the action script, AGENTHANGDELAY setting must be less than the SCRIPT_TIMEOUT value, which is set in CRS. If an Essbase operation; for example, application creation, data load, and database restructuring, is not completed within the time specified for AGENTHANGDELAY, the action script terminates all Essbase server process; including orphan processes. CRS only manages the Essbase Agent.

Note: SCRIPT_TIMEOUT is the time (in seconds) that CRS waits for a response from the `Check Status` action defined in the action script. If the action script does not return within this timeout, CRS terminates Oracle Essbase process and restarts it.

Create the executable script in a convenient directory; for example, as `/Disk2/scripts/essbaseappcheck`.

```
#!/bin/sh

SCRIPT_PATH=/oraohocfs/scripts
PATH=${SCRIPT_PATH}:${PATH}
mkdir -p ${SCRIPT_PATH}

ESSVER=1.0
LOGSCR=${SCRIPT_PATH}/clusteractionessbase.log

ESSSERVERNAME="essbasevip.example.com"
ESSUSERNAME="admin"
ESSPASSWORD="password"

HYPERION_HOME=/oradbocfs/hyperion
ARBORPATH=${HYPERION_HOME}/products/Essbase/EssbaseServer
ESSBASEPATH=${HYPERION_HOME}/products/Essbase/EssbaseServer

PATH=${ESSBASEPATH}/bin:${PATH}

# Delay time (seconds) for Essbase's agent to shutdown gracefully
AGENTSHUTDOWNDELAY=15

# Delay time (seconds) after which Essbase agent agent is considered to be unresponsive
AGENTHANGDELAY=120
```

```
# Essbase's agent state (0=running, 1=not running)
AGENTSTATE=0

#
# Declare functions
# ---------------------------------------

Logger()
{
    echo `date | awk '{ printf "[%s %s %s %s %s - ", $1, $2, $3, $4, $6 }'` $
{HOSTNAME}"] "$1 | tee -a ${LOGSCR}
}

Kill_Orphans()
{
    # Input delay time before killing all Essbase processes
    sleep $1

    # Find all Essbase's running processes if any
    ps -ef | grep -v grep | grep "${ESSBASEPATH}/bin/ESS" | awk '{printf("%s %s\n", $2,
$9)}' | sed 's/\*\*/ESSBASE/' > .pid

    # Kill all running processes since they appear to be hung
    while read line
    do
    Logger "Force killing process id ${line}..."
    kill -9 ${line% *}
    done < .pid
}

Check_Status()
{
    # Perform delay kill Essbase's orphan processes if the next check command
    # won't return when AGENTHANGDELAY expires
    Kill_Orphans ${AGENTHANGDELAY} &
    KID=$!

    echo "display system version;" | startMaxl.sh -l ${ESSUSERNAME} ${ESSPASSWORD} -i > /
dev/null 2>&1
    AGENTSTATE=$?

    # Abort Kill_Orphans() when script exited before AGENTHANGDELAY expires
    trap "kill -9 ${KID}" 0 1

    if [ ${AGENTSTATE} -eq 0 ]; then
        Logger "Essbase agent is running!"
    exit 0
    else
        Logger "Essbase agent is not running!"
    exit 1
    fi
}

Start_Essbase()
{
    # If's safe to kill all orphan processes here if any since Stop_Essbase()
    # has been called earlier by OCW and all well behaved processes assumed
```

```
    # being shutdown gracefully
    Kill_Orphans 0

    # Start Essbase agent
    startEssbase.sh ${ESSPASSWORD} > /dev/null 2>> ${LOGSCR}

    # Check status of new started agent
    echo "display system version;" | startMaxl.sh -l ${ESSUSERNAME} ${ESSPASSWORD} -i > /
dev/null 2>&1
    AGENTSTATE=$?

    if [ ${AGENTSTATE} -eq 0 ]; then
        Logger "Essbase agent has started successfully!"
    exit 0
    else
        Logger "Problem starting Essbase agent! Please check ${HYPERION_HOME}/logs/
essbase/Essbase.log for more details!"
    exit 1
    fi
}

Stop_Essbase()
{
    # Perform delay kill Essbase's orphan processes if the next shutdown
    # command won't return when AGENTSHUTDOWNDELAY expired
    Kill_Orphans ${AGENTSHUTDOWNDELAY} &
    KID=$!

    # Shutdown Essbase's agent using MaxL command
    echo "alter system shutdown;" | startMaxl.sh -l ${ESSUSERNAME} ${ESSPASSWORD} -i > /
dev/null 2>> ${LOGSCR}
    AGENTSTATE=$?

    # Abort Kill_Orphans() when script exited before AGENTSHUTDOWNDELAY expires
    trap "kill -9 ${KID}" 0

    # Perform soft or hard shutdown if neccessary
    Logger "Essbase agent was shutdown successfully!"
    exit 0
}

#
# Start script
# ---------------------------------------

if [ $# -eq 0 ]; then
    echo "Usage: ${0} [ start | stop | check ]"
    exit 1
fi

Logger "Using ESSBASEPATH=${ESSBASEPATH}"
Logger "Using ARBORPATH=${ARBORPATH}"

# Convert command to lower case
CMD=`echo $1 | tr [A-Z] [a-z]`
Logger "BEGIN: ${0} $1"
```

```
# Move to working dir
cd ${ARBORPATH}/bin

case ${CMD} in

start)
    Logger "Running command ${CMD}..."
    Start_Essbase
    ;;

stop)
    Logger "Running command ${CMD}..."
    Stop_Essbase
    ;;

check)
    Logger "Running command ${CMD}..."
    Check_Status
    ;;

*)
    Logger "Unknown command entered: ${CMD}"
    exit 1
    ;;

esac
```

COPYRIGHT NOTICE

Authors: EPM Information Development Team

ORACLE
ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM