



# HYPERION® SHARED SERVICES

RELEASE 9.3.1

## SHARED SERVICES AND REPORTING AND ANALYSIS HIGH AVAILABILITY (WINDOWS ENVIRONMENTS)

ORACLE | Hyperion

### CONTENTS IN BRIEF

About Shared Services and Reporting and Analysis High Availability on Windows . . . . .	2
Deploying Shared Services and Reporting and Analysis in a Failover Cluster . . . . .	2
Reporting and Analysis Installation . . . . .	13
Managing the Cluster . . . . .	20
Additional Information . . . . .	24
Appendix . . . . .	24

## About Shared Services and Reporting and Analysis High Availability on Windows

To make Oracle's Hyperion Oracle's Hyperion® Shared Services and Oracle's Hyperion® Reporting and Analysis – System 9 highly available, you must use clustering solutions to ensure that none of these components is a single point of failure:

- Database
- Native Directory and other user directories
- Web applications (Shared Services, Workspace, Oracle's Hyperion® Web Analysis – System 9, Oracle's Hyperion® Financial Reporting – System 9)
- Reporting and Analysis services (Core, IR, DAS, FR)

See the documentation for your RDBMS.

The supported configuration includes the Native user directory (OpenLDAP), Shared Services Web application, Reporting and Analysis Service, and Reporting and Analysis Web applications in a failover cluster using Oracle Clusterware.

You cluster these components for failover using Oracle Clusterware 11.1, which is available for free to protect Oracle Hyperion components. See “[Installing and Configuring Oracle Clusterware](#)” on page 7. You can download Oracle Clusterware from <http://www.oracle.com/technology/software/products/database/index.html>, and information about Oracle Clusterware is available at this Oracle Technology Network address: <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).

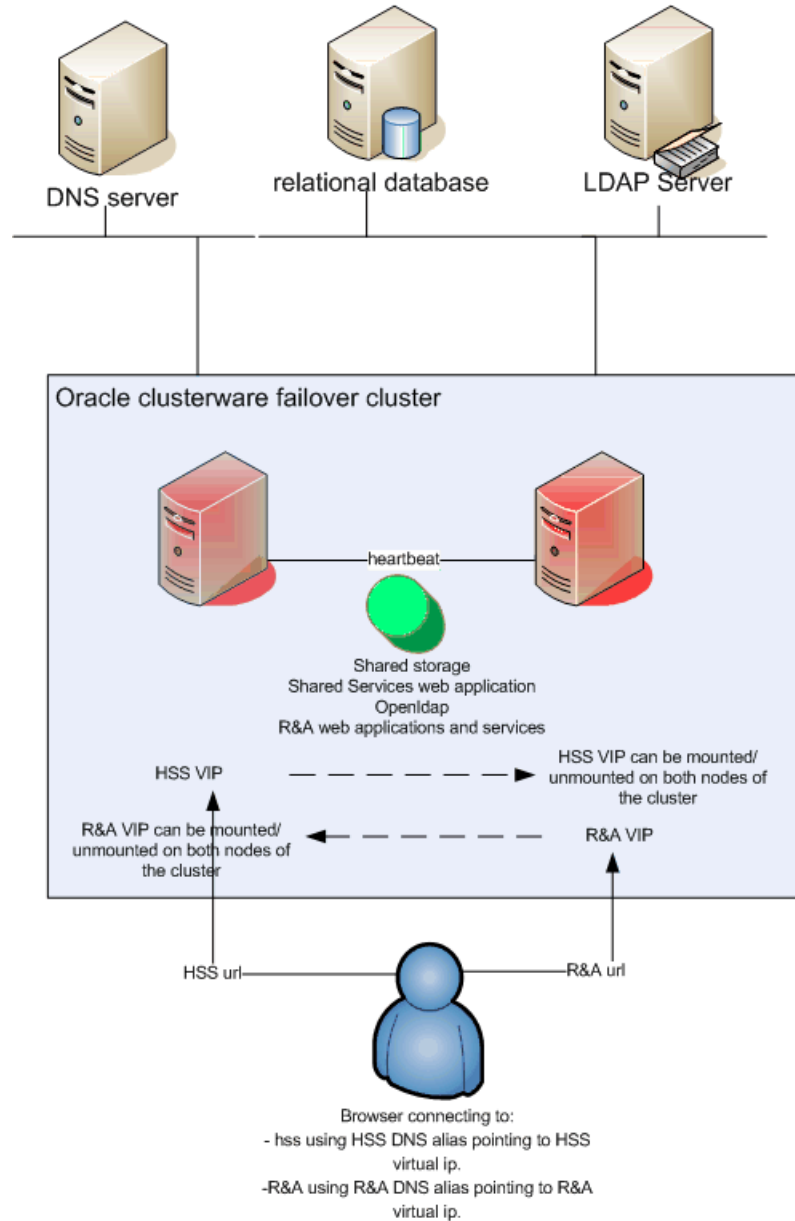
Database clustering solutions depend on the RDBMS that you use. Oracle's Hyperion products support Oracle Real Application Clusters and third-party RDBMS software.

## Deploying Shared Services and Reporting and Analysis in a Failover Cluster

Before clustering Shared Services or Reporting and Analysis for failover, you must meet Oracle Clusterware prerequisites and then install and configure Oracle Clusterware. See “[Meeting Oracle Clusterware Prerequisites](#)” on page 3 and “[Installing and Configuring Oracle Clusterware](#)” on page 7.

In a failover cluster, the Shared Services processes (Web application and Native Directory) and Reporting and Analysis processes (Core, IR, FR, WA services) are accessible at a specific IP address referenced by a DNS name alias. If the primary node fails, the virtual IP (VIP) and Shared Services or Reporting and Analysis processes move automatically to the secondary node, as shown in Figure 1.

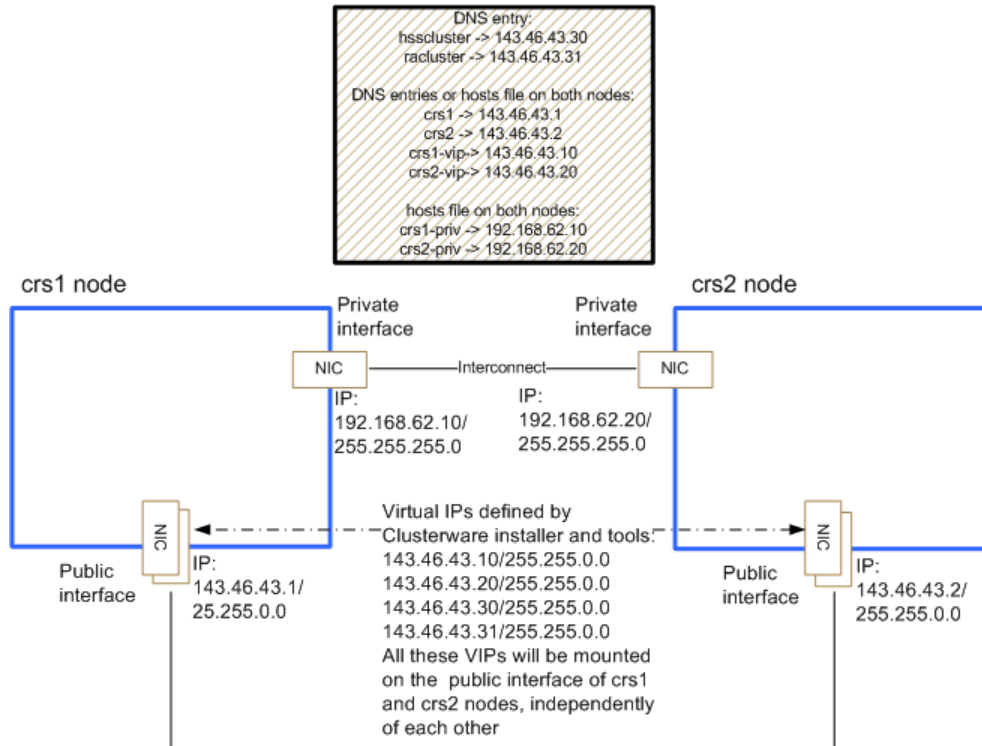
Figure 1 Shared Services and Reporting and Analysis Oracle Clusterware Failover Cluster



## Meeting Oracle Clusterware Prerequisites

- To meet Oracle Clusterware prerequisites:
  - 1 Review Chapter 2, "Server and Network Preinstallation Tasks," in the *Oracle Clusterware Installation Guide* ([http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28250.pdf](http://download.oracle.com/docs/cd/B28359_01/install.111/b28250.pdf)).
  - 2 Ensure that the clocks of the machines in the cluster are synchronized.
  - 3 Define IPs and virtual IPs, making sure to meet Oracle Clusterware requirements. For example:

Figure 2 IP and VIP Definitions



#### Node 1

CRS1  
Type: Public  
IP Address: 143.46.43.1  
Subnet: 255.255.0.0  
Location: DNS

CRS1-vip  
Type: Virtual  
IP Address: 143.46.43.10  
Subnet: 255.255.0.0  
Location: DNS

CRS1-priv  
Type: Private  
IP Address: 192.168.62.10  
Subnet: 255.255.255.0  
Location: Host files on both nodes

#### Node 2

CRS2  
Type: Public  
IP Address: 143.46.43.2  
Subnet: 255.255.0.0  
Location: DNS

CRS2-vip  
Type: Virtual

IP Address: 143.46.43.20  
Subnet: 255.255.0.0  
Location: DNS

CRS2-priv  
Type: Private  
IP Address: 192.168.62.20  
Subnet: 255.255.255.0  
Location: Host files on both nodes

**Note:** You need not create CRS1-vip and CRS2-vip physically. When configuring network interfaces, set TCP/IP parameters only for the public and the private interface, without configuring virtual IPs.

- 4 Plan VIP/subnets for Shared Services components (for example, 143.46.43.30 / 255.255.0.0) and Reporting and Analysis (for example, 143.46.43.31 / 255.255.0.0), with corresponding DNS entries (for example, hsscluster and racluster).**

- 5 Ensure that a shared disk is connected to both nodes.**

The shared disks are not formatted and will be configured for RAW device access and Oracle Cluster File System access by the Oracle Clusterware installer.

- 6 Ensure that each server has at least two network interfaces (more if the cards are teamed):**

- a. Interfaces must be renamed “Public” and “Private” identically on both nodes:
  - Public provides connectivity from client to server.
  - Private provides server interconnection.
  - Public and Private interfaces have different subnet masks.

To rename a network interface, click Start, then Control Panel, then Network Connections, and right-click the network interface.

- In the network adapter binding order, the Public interface must be listed first. To open the screen where you set this order, select Network Connections from the Windows Control Panel, and then select the Advanced menu and the Advanced Settings command.

- 7 Disable write caching on all disks that will be used to share data between nodes in the cluster:**

- a. Click Start, then select Settings, then Control Panel, then Administrative Tools, then Computer Management, and then Device Manager.
- b. In the device list, expand Disk drives, and double-click the first drive that is listed.
- c. On the Policies tab, deselect the option that enables the write cache.

You can also modify Windows registry keys to enable or disable the write cache.

To disable the write cache:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\SCSI\“Device-Name”\“Resource”\Device Parameters\Disk]
"CacheIsPowerProtected"=dword:00000000
"UserWriteCacheSetting"=dword:00000000
```

To enable the write cache:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\SCSI\“Device-Name”\“Resource”\Device Parameters\Disk]
"CacheIsPowerProtected"=dword:00000001
"UserWriteCacheSetting"=dword:00000001
```

The Microsoft tool `Dskcache.exe` can also be used to enable or disable the write cache.

## 8 Disable media sensing:

---

**Caution!** Because you must modify the Windows registry to disable media sensing, first back up the registry and confirm that you can restore it, as described in Windows documentation.

---

a. To disable Windows media sensing for TCP/IP, set the value of the `DisableDHCPMediaSense` parameter to 1 on each node.

b. On each node of the cluster:

i. Back up the Windows registry.

ii. Use `Regedt32.exe` to view this key in the registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip/Parameters
```

iii. Add these registry values:

- Value Name—`DisableDHCPMediaSense`
- Data Type—`REG_DWORD`
- Value—1

iv. Restart the computer.

## 9 On each node, stop the Distributed Transaction Coordinator service, and set it as manual.

## 10 Partition the shared drive-hard drive:

a. On each node of the cluster, enable automount:

i. On a Windows command line, enter `diskpart`.

ii. On the command line, enter `AUTOMOUNT ENABLE`.

iii. Restart the computer.

b. On one cluster node only:

i. In the Windows **Control Panel**, double-click **Administrative Tools**.

ii. Double-click **Computer Management**.

iii. Double-click **Storage**.

iv. Double-click **Disk Management**.

v. Initialize the shared disk.

Do not convert the disk to a dynamic disk.

- vi. Right-click the disk, select **New Partition**, and then select **Create an extended partition** and use all available disk space.
- vii. Right-click the disk, and create logical drives:
  - Three partitions of 512 MB each for voting disks
  - Two partitions of 512 MB for OCR and mirror OCR disks
  - One partition for Shared Services software
  - One partition for Reporting and Analysis software

**Note:** Do not assign drive letters while creating the partitions, and do not format the partitions. Restart the second node after configuring the shared disk on the first node only. After restarting the second node, return to Computer Management and delete the drive letters of all the partitions of the shared storage. All the letters for volumes on the shared disks must be deleted on each node.

## 11 Check prerequisites:

- Ensure Administrator user name and password are identical on both cluster nodes.
- Ensure you can ping public interface and private interface using CRS1, CRS2, CRS1-priv, and CRS2-priv.

From Oracle Clusterware, check prerequisites in a command shell with this command:

```
runcluvfy stage -post hws -n CRS1,CRS2 -verbose|
```

The `runcluvfy.bat` file is in the Oracle Clusterware installation folder

**Note:** In some cases, these checks fail: User Equivalency, Node Reachability, and Administrative Privileges. See section “5.1.1 Troubleshooting Clusterware Setup for Windows” at [http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28250/crswin.htm](http://download.oracle.com/docs/cd/B28359_01/install.111/b28250/crswin.htm). The error message `Network interface information could not be obtain from any of the nodes is not blocking` if you can ping all nodes on the Public and Private interfaces, and if the `net use \\CRS2\c$` command succeeds (when you use `runcluvfy.bat` from CRS1).

## Installing and Configuring Oracle Clusterware

➤ > To install and configure Oracle Clusterware:

- 1 On one node only, launch `setup.exe` from the Oracle Clusterware installation directory to install Oracle Clusterware on the C: drive.

**Note:** You can keep the cluster name default.

- 2 On the **Specify cluster configuration** screen, enter the public, private, and virtual host names for each node.

**Note:** You must click Add to enter information about node 2.

**3 In Specify Network Interface usage, specify the private and public interface.**

**Note:** No change is required unless you want to distinguish between public or private for additional network interfaces.

**4 On the Cluster Configuration Storage screen:**

- Specify these locations:
  - OCR (cluster registry)—Two devices, OCR primary and mirror
  - Voting disk—Three devices  
Specify place voting disk on this partition 3 times.
- Partition for software:
  - Format with CFS check.
  - Assign drive letter Y for Shared Services software and driver letter Z for Reporting and Analysis.

**Tip:** If you do not select the partition for software, you can format the software partition later with the `ocfsformat` tool in the `CRS_HOME/cfs/` folder. See “Formatting Drives to Use Oracle Cluster File System after Installation” in the *Oracle Clusterware Installation Guide* ([http://download.oracle.com/docs/cd/B28359\\_01/install.111/b28250/crswin.htm](http://download.oracle.com/docs/cd/B28359_01/install.111/b28250/crswin.htm)).

**5 If Oracle Clusterware fails on virtual IP creation, use `product/11.1.0/crs/bin/VIPCA.bat` to create the virtual IPs manually.**

**6 Click Finish.**

**7 From a Windows Command Line prompt, run this command: `\product\11.1.0\crs\BIN\cluvfy stage -post crsinst -n CRS1,CRS2`**

## Creating Application VIPs

Create application virtual IPs (VIPs) for Shared Services and Reporting and Analysis that are mounted and unmounted on the Public interface by Oracle Clusterware. The VIPs are referenced by appropriate DNS aliases `hsscluster` and `raccluster`. The `hsscluster` and `raccluster` entries must be defined in your DNS.

**Tip:** From a command line, you can run the script `createvip_hss.bat` or `createvip_ra.bat` (which are provided in the Appendix of this document) to create a VIP, but first you must open the files and set the unique `VIPIP` values (for example, `143.46.43.30`) and specify `VIPSUBNET` (for example, `255.255.0.0`). Also ensure that the `CRS_HOME` value is set to `c:\product\11.1.0\crs\bin`, and that the `ADAPTER` value reflects the name of your public network interface, such as `Public`. This batch also defines default timeout parameters (300 seconds) for the VIP at runtime. More information on these timeouts is available at [http://download.oracle.com/docs/cd/B28359\\_01/rac.111/](http://download.oracle.com/docs/cd/B28359_01/rac.111/)



[b28255/crschp.htm#i1016484](http://b28255/crschp.htm#i1016484). These batches are run only once from one node of the cluster.

After each VIP is created, you can enter this command in a command shell in the `crs\bin` folder to view if it is correctly mounted on one node of the cluster: `crs_stat -t -v`

You should be able to ping `hsscluster` and `racluster` in a shell:

```
ping hsscluster
```

```
ping racluster
```

**Tip:** To troubleshoot an incorrect creation of the VIPs, run the script manually from a shell, line-by-line.

## Shared Services Installation

Use the Shared Services installer to install Shared Services following the instructions in the *Hyperion Shared Services Release 9.3.1.0.0x Installation Guide for Windows* (available from [http://download.oracle.com/docs/cd/E10530\\_01/doc/nav/portal\\_1.htm](http://download.oracle.com/docs/cd/E10530_01/doc/nav/portal_1.htm)) and these guidelines:

- Before beginning installation on either node, shut down the other node.

**Note:** If you cannot shut down one node, you can use the `crsctl stop crs` command to stop the OracleCSService, OracleEVMService, and OracleCRSService Windows services on that node instead.

- For each node, install Shared Services on the shared disk `Y:\Hyperion`. If you plan to install other Oracle Hyperion products in the cluster, install the common components (that is, the Hyperion home directory) on a local drive, and the product specific files such as Shared Services on the `Y` drive. This allows for greater independence of products during failover.

**Note:** If node 1 and node 2 were running, because of an OCFS limitation on Windows, after installing Shared Services on node 1 and shutting down the node, you must wait 5 minutes for files on the `Y` drive to become accessible on node 2.

- Click Finish (without launching the Hyperion Configuration Utility) on the last Shared Services installer screen.
- After each installation, set the Hyperion Foundation OpenLDAP service as manual and stop the service.
- Perform the same steps on the second node. When prompted for overwrite of existing file, click Yes.

## Configuring Shared Services

**Note:** Before configuring Shared Services, ensure that the `hsscuster` alias points to the virtual IP for Shared Services, through DNS or host files entries on both cluster nodes. You configure Shared Services on the node where the Shared Services virtual IP is mounted. You can use the `crs_stat -t -v` command to check where the virtual IP is mounted. This document assumes that the virtual IP is mounted on node 2.

Use the Hyperion Configuration Utility to configure Shared Services on both nodes of the cluster, starting with node 2. See the *Hyperion Shared Services Release 9.3.1.0.0x Installation Guide for Windows*.

► To configure Shared Services:

### 1 On node 2:

- a. From the Windows Start menu, select **Programs**, then **Hyperion System 9 Foundation**, then **Configuration Utility**.
- b. Select **Configure database and Deploy to Application Server**.
- c. Select a database type enter the database configuration parameters.
- d. Select an application server type.
- e. On the application server deployment screen, ensure that **Deploy as a service** is selected.
- f. After the configuration utility runs, change the `CSS.xml` file in `Y:\Hyperion\SharedServices\9.3.1\AppServer\InstallableApps\common` and `HYPERION_HOME\deployments\application server\version\config`  

```
hub location=http://hsscuster:58080
```

where `hsscuster` is a DNS entry pointing to the Shared Services virtual IP.
- g. Stop the Hyperion Foundation Shared Services – Web application and Hyperion Foundation OpenLDAP services and set them as manual.

### 2 Stop node 2.

### 3 On node 1:

- a. From the Windows Start menu, select **Programs**, then **Hyperion System 9 Foundation**, then **Configuration Utility**.
- b. Select **Configure database and Deploy to Application Server**.
- c. Select a database type enter the database configuration parameters, using the same database as for node 2, and when prompted, select **Reuse the existing database**.
- d. Select an application server type.
- e. On the application server deployment screen, ensure that **Deploy as a service** is selected.
- f. After the configuration utility runs, change the `CSS.xml` file in `HYPERION_HOME\deployments\application server\version\config`

```
hub location=http://hsscuster:58080.
```

where *hsscuster* is a DNS entry pointing to the Shared Services virtual IP

- g. Stop the Hyperion Foundation Shared Services – Web application and Hyperion Foundation OpenLDAP services and set them as manual.
- h. Restart the Hyperion Foundation Shared Services – Web application and Hyperion Foundation OpenLDAP services, then log on to `http://hsscuster:58080/interop/index.jsp` for a test.
- i. Stop the Hyperion Foundation Shared Services – Web application and Hyperion Foundation OpenLDAP services and set them as manual.

If you stopped the OracleCSService, OracleEVMSERVICE, and OracleCRSService Windows services, use the commands `crsctl check crs` and `crsctl start crs` on each node to ensure that those services are all running now. The commands are in `c:\product\11.1.0\crs\bin`.

## Registering Shared Services in the Cluster

- To register Shared Services in the cluster:

### 1 Create an action program for Shared Services.

Oracle Clusterware uses this script to start, stop and check the status of Shared Services.

An appendix to this document provides the `hss931.pl` file as a sample action script to monitor Shared Services and OpenLDAP.

### 2 Modify `hss931.pl` to update the values of `SCRIPT_PATH`, `HSSURL`, `ORA_CRS_HOME`, `HYPERION_SHARED`, or other variables:

- a. Create a `y:\crs_action\hss` folder on the shared drive.
- b. Create `hss931-pl.bat` and `hss931.pl` files in the `y:\crs_action\hss` folder. You can copy the contents of these files from the Appendix of this document.
- c. Open `y:\crs_action\hss\hss931.pl` in a text editor, and in the following lines change the `SCRIPT_PATH`, `HSS_URL`, `ORA_CRS_HOME`, and `HYPERION_SHARED` values to match your configuration:

```
my $SCRIPT_PATH = "y:/CRS_ACTION/HSS"  
my $HSS_URL = "http://hsscuster:58080/interop/index.jsp";  
my $ORA_CRS_HOME = "c:\product\11.1.0\crs";  
my $HYPERION_SHARED = "y:/hyperion"
```

- d. Use one of these options to install perl on both nodes of the cluster:
  - ActivePerl from <http://www.activestate.com/activeperl>
  - The perl binaries in `HYPERION_HOME\common\Perl\5.8.3\bin\MSWin32-x86-multi-thread`

If you use the binaries, be sure to install the Win32:Service through CPAN by launching `perl -MCPAN -e shell`, follow the manual configuration instructions, and enter `install 'Win32::Service'`.

- e. Open `y:\crs_action\hss\hss931-pl.bat` and change the path to reflect the location of your `perl.exe`:

```
set PATH=c:\perl\bin;%PATH%
```

- f. Test the action script before registering it in the cluster:
  - i. Stop Hyperion Foundation Shared Services – Web application and Hyperion Foundation OpenLDAP services on the node where the Shared Services VIP is not mounted.

**Tip:** Use `crs_stat -t -v` to determine where the Shared Services VIP is mounted.

- ii. On the remaining node where the Shared Services VIP is mounted, in a command shell, perform these tests:
  - Enter `hss931-pl.bat`. This should return the usage.
  - Enter `hss931-pl.bat check`, then `echo %ERRORLEVEL%`. This should return 0 if Shared Services is running or 1 if Shared Services is not running.
  - If Shared Services is down, try to start it by entering `hss931-pl.bat start`, then `echo %ERRORLEVEL%`. This should return 0 if the start succeeds or 1 if the start fails.

**Note:** The start can take more than 5 minutes, because the script includes a 5 minute delay to wait for OCFS shared disk locks.

- If Shared Services is up, try to stop by entering `hss931-pl.bat stop`, then `echo %ERRORLEVEL%`. This should return 0 if the stop succeeds or 1 if the stop fails.

**Note:** Do not register the Shared Services profile in the cluster until the start stop and check tests succeed.

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

The sample `register_hss.bat` file in the appendix to this document automatically creates an application profile and registers the application profile with Oracle Clusterware.

- Before running `register_hss.bat`, you must open it and ensure that `ACTION_SCRIPT` and `CRS_HOME` settings are correct as follows:

```
o set ACTION_SCRIPT= y:\crs_action\hss\hss931-pl.bat
```

```
o set CRS_HOME=c:\product\11.1.0\crs\bin
```

If you are copying and pasting the script from the appendix. Make sure that the `crs_profile` line is on one line.

- It is only necessary to run `register_hss.bat` on one node of the cluster.

## Postregistration Shared Services Tasks

After installing and configuring Shared Services and registering Shared Services in the cluster, it is important to back up the relational database and Shared Services files, including OpenLDAP, using the `backup.bat` tool. For instructions, see Appendix D, “Shared Services Backup and Recovery,” in the *Hyperion Shared Services Release 9.3.1.0.0x Installation Guide for Windows*: [http://download.oracle.com/docs/cd/E10530\\_01/doc/epm.931/hss\\_931003\\_windows\\_install.pdf](http://download.oracle.com/docs/cd/E10530_01/doc/epm.931/hss_931003_windows_install.pdf).

For example, run this hot backup command from a shell:

```
Y:\Hyperion\SharedServices\9.3.1\server\scripts\backup.bat 0:  
\yourbackupfolder\  

```

Also perform a full backup of the Hyperion files on the shared drive with services stopped.

## Reporting and Analysis Installation

Use the Service, RepServices and UIServices installers to install Reporting and Analysis following the instructions in the *Hyperion Reporting and Analysis Release 9.3.1.0 Installation Guide* ([http://download.oracle.com/docs/cd/E10530\\_01/doc/epm.931/hs\\_windows\\_install.pdf](http://download.oracle.com/docs/cd/E10530_01/doc/epm.931/hs_windows_install.pdf)) and these guidelines:

- Before beginning installation on either node, shut down the other node.
- For each node, install Reporting and Analysis on the shared disk `Z:\Hyperion`. If you plan to install other Oracle Hyperion products in the cluster, install the common components (that is, the Hyperion home directory) on a local drive, and the product-specific files such as Hyperion Reporting and Analysis Services on the shared `Z` drive. This allows greater independence of products during failover.

**Note:** If node 1 and node 2 were running, because of an OCFS limitation on Windows, after installing Reporting and Analysis (services, repServices, and UIServices) on node 1 and shutting down the node, you must wait 5 minutes for files on the `Z` drive to become accessible on node 2.

- After every installation, click Finish without launching Hyperion Configuration Utility.
- Perform the same installation steps on the second node. When prompted for overwrite of existing file, click Yes.

## Configuring Reporting and Analysis

**Note:** Before configuring Reporting and Analysis, ensure that the `raccluster` alias points to the virtual IP for Reporting and Analysis, through DNS or host files entries on both cluster nodes. You configure Reporting and Analysis on the node where the Reporting and Analysis VIP is mounted. You can use the `crs_stat -t -v` command to check where the VIP is mounted. This document assumes that the VIP is mounted on node 2.

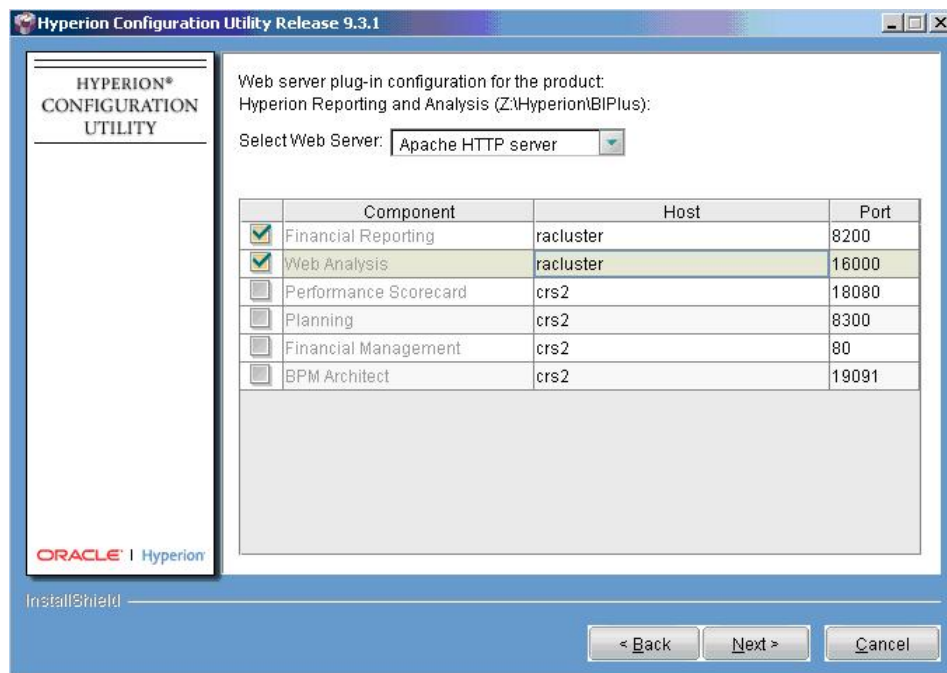
Use the Hyperion Configuration Utility to configure Reporting and Analysis on both nodes of the cluster, starting with node 2. See the *Hyperion Reporting and Analysis Release 9.3.1.0 Installation Guide*. 1 a. b.

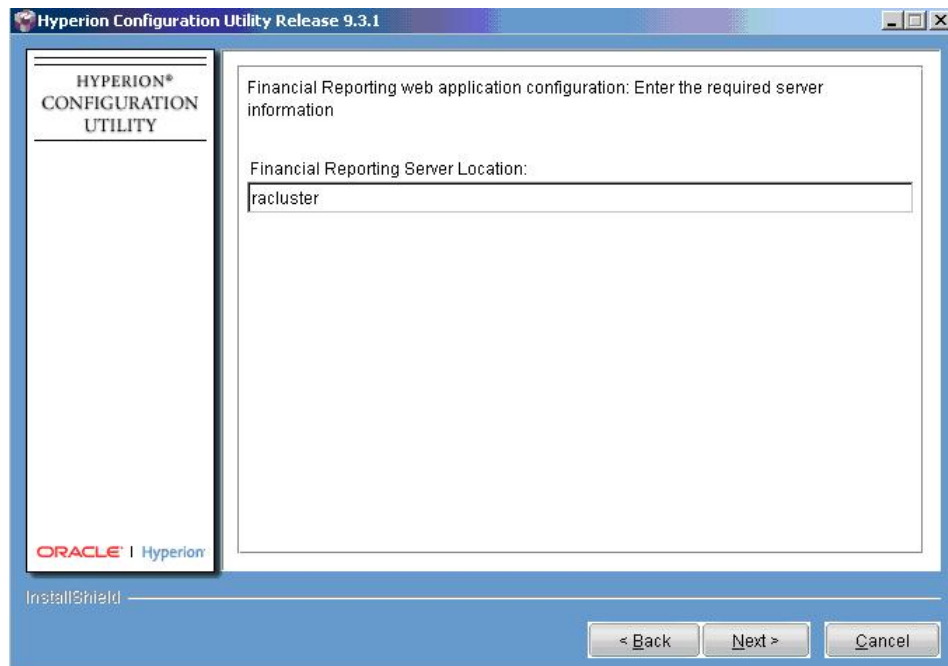
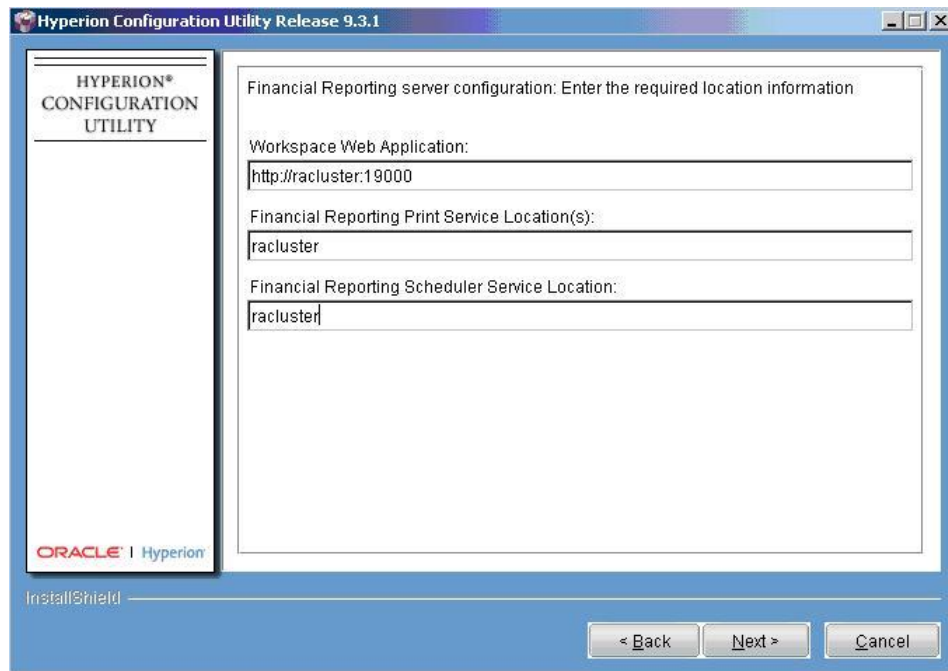
► To configure Reporting and Analysis.

**1 On node 2:**

- a. From the Windows Start menu, select **Programs**, then **Hyperion System 9 Foundation**, and then **Configuration Utility**.
- b. Execute the **Product Options**, **Configure Database**, **Configure Financial Reporting**, **Deploy to Application Server**, and **Configure Web Server** tasks.

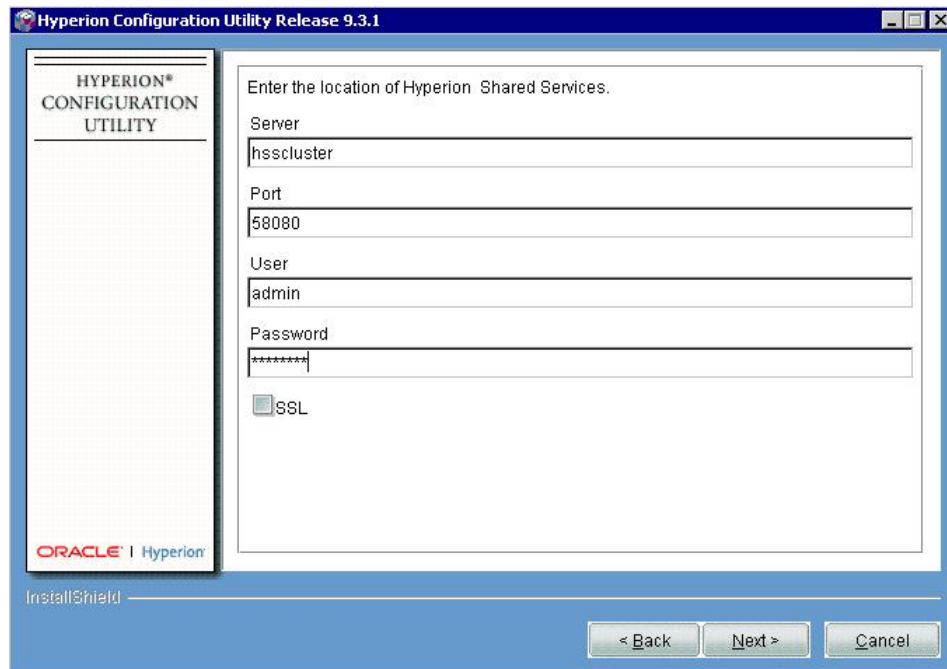
**Note:** During Web Server plug-in, FR Server and Web application, update real hostname with the virtual `racluster` alias:



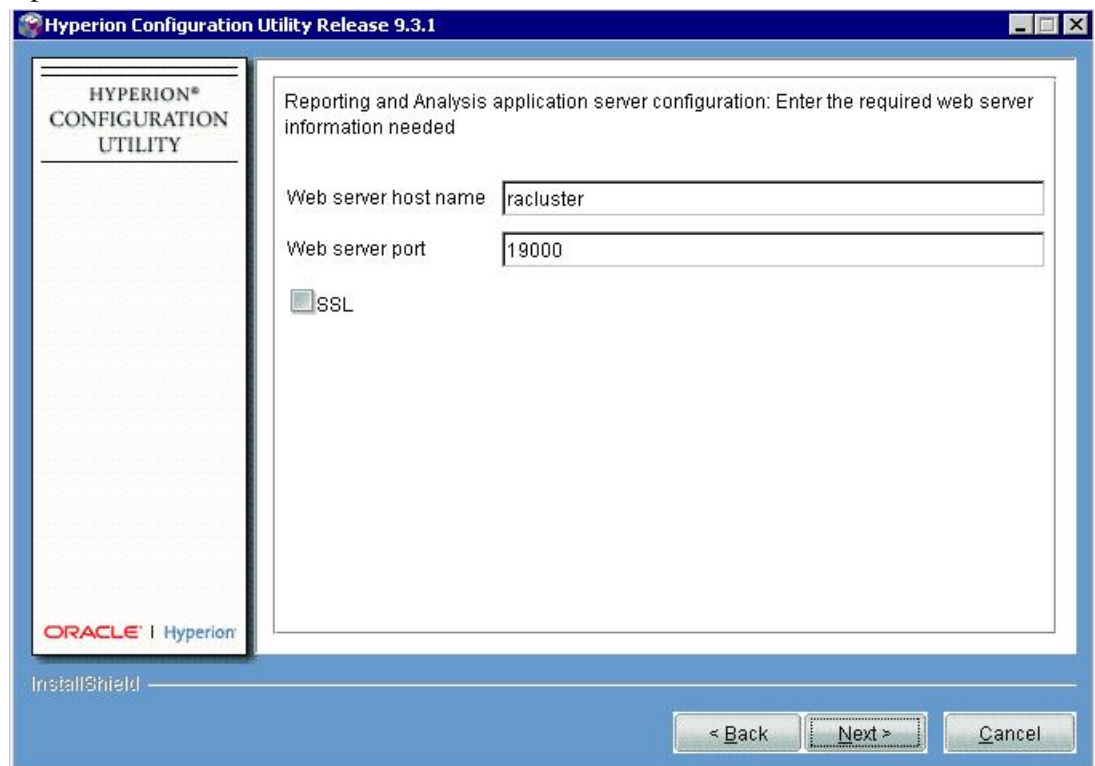


- c. Execute the **Register with Shared Services** task separately.

**Note:** Make sure Shared Services is running in the cluster for this step, using `crs_stat -t -v` and `crs_start -f hss-webapp` if needed. Specify the Shared Services virtual alias (`hssccluster`) as a Shared Services server.



- d. Update Web Server host name with the virtual alias (racluster)



**Note:** The Hyperion System 9 BI+ project name in Shared Services will have physical node name, but this is only a logical name and does not affect functionalities.

- e. After configuration set startup type to Manual for these services:
- Hyperion S9 Apache 2.0



- Hyperion S9 BI+ 9.3 Workspace
  - Hyperion S9 BI+ 9.3 WebAnalysis
  - Hyperion S9 BI+ 9.3 Financial Reporting Java RMI Registry
  - Hyperion S9 BI+ 9.3 Financial Reporting Print Server
  - Hyperion S9 BI+ 9.3 Financial Reporting Report Server
  - Hyperion S9 BI+ 9.3 Financial Reporting Schedule Server
  - Hyperion S9 BI+ 9.3 Financial Reporting Web application
- f. Update `server.xml` and `config.dat` in the `BIPLUS_HOME\common\config` folder:
- Edit the `<host>` tag in `server.xml` as follows:
 

```
<host>racluster</host>
```
  - Edit the `Host` value in `config.dat` as follows:
 

```
Host=racluster
```
- g. Update `WSDEPLOYMENT_HOME\WEB-INF\config\ws.conf` as follows:
- ```
WebClient.UserInterface.Login.RMServer=racluster:6800
WebClient.Diagnostics.Log.LoggingService=racluster:6800
```
- Note:** Check `ws.conf` on both nodes of the cluster.
- h. Update `BIPLUS_HOME\InstallableApps\config\WebAnalysis.properties` as follows:
- ```
ConfigURL=http://racluster:19000/workspace/browse/configURL
```
- i. In `BIPLUS_HOME\bin\startCommonServices.bat`, add `-Drm.ignore_host=true` to the `%BP_JRE%` line.
- j. Update the FR and IR Services sequence in `BIPLUS_HOME\bin\stop_BIPlus.bat` and specify absolutely the same order:
- ```
if exist HyS9FRSched.exe (net stop HyS9FRSched)
if exist HyS9FRReport.exe (net stop HyS9FRReport)
if exist HyS9FRPrint.exe (net stop HyS9FRPrint)
if exist HyS9FRRMI.exe (net stop HyS9FRRMI)

rem service HyS9IRSvc1
net stop "HyS9IRSvc1"

rem service HyS9Core1
net stop "HyS9Core1"
```

## 2 Stop node 2.

### 3 On node 1:

- a. From the Windows Start menu, select **Programs**, then **Hyperion System 9 Foundation**, and then **Configuration Utility**.
- b. Complete the **Product Options**, **Configure Financial Reporting**, **Deploy to Application Server** and **Configure Web Server** tasks.

**Note:** During Web Server plug-in, FR Server and Web application update the real host name with the virtual racluster alias

- c. After configuration, set the startup type to **Manual** for these services:
  - Hyperion S9 Apache 2.0
  - Hyperion S9 BI+ 9.3 Core Services 1
  - Hyperion S9 BI+ 9.3 Interactive Reporting Service 1
  - Hyperion S9 BI+ 9.3 Workspace
  - Hyperion S9 BI+ 9.3 WebAnalysis
  - Hyperion S9 BI+ 9.3 Financial Reporting Java RMI Registry
  - Hyperion S9 BI+ 9.3 Financial Reporting Print Server
  - Hyperion S9 BI+ 9.3 Financial Reporting Report Server
  - Hyperion S9 BI+ 9.3 Financial Reporting Schedule Server
  - Hyperion S9 BI+ 9.3 Financial Reporting Web application
- d. When configuration is complete, two services must be created manually, using RMsService8.exe from *BIPLUS\_HOME\bin*:
  - RMsService8 -RegService -L Z:\Hyperion\BIPlus\bin -P startCommonServices.bat -DN HyS9Core1 -MS Foundation
  - RMsService8 -RegService -L Z:\Hyperion\BIPlus\bin -P startIRM.bat -DN HyS9IRSvc1 -MS ProcessMonitor
- e. Update the BIPlus repository database:

```
update V8_PROP_VALUE set VALUE0='racluster' where
PROP_NAME='ConfigurationManager.gsmHost';
update V8_HOST set HOSTNAME='racluster';
update V8_INSTALL_SVC_GSM set GSM_HOST_NAME='racluster';
update V8_INSTALL_SVC_LOG set HOST_NAME='racluster';
update V8_DOMAIN_PROPS set VALUE='racluster' where NAME='GSM HOST';
```

## Registering Reporting and Analysis in the Cluster

- To register Reporting and Analysis in the cluster:

### 1 Create an action program for Reporting and Analysis.

Oracle Clusterware uses this script to start, stop and check the status of Reporting and Analysis. An appendix to this document provides the ra931.pl file as a sample action script for monitoring Reporting and Analysis services.

### 2 Modify ra931.pl to update the values of SCRIPT\_PATH, WEB\_URL,ORA\_CRS\_HOME, BIPLUS\_HOME, or other variables:

- a. Create a z:\crs\_action\ra folder on the shared drive.
- b. Create ra931-pl.bat and ra931.pl files in the z:\crs\_action\ra folder.

You can copy the contents of these files from the Appendix of this document.

- c. Open `z:\crs_action\ra\ra931.pl` in a text editor, and in the following lines change the `SCRIPT_PATH`, `WEB_URL`, `ORA_CRS_HOME`, and `BIPLUS_HOME` values to match your configuration:

```
my $WEB_URL          = "http://racluster:19000";
my $ORA_CRS_HOME     = "C:\product\11.1.0\crs";
my $BIPLUS_HOME      = "Z:/Hyperion/BIPlus";
my $SCRIPT_PATH      = "z:/CRS_ACTION/ra"
```

- d. Use ActivePerl from <http://www.activestate.com/activeperl> to install perl on both nodes of the cluster.
- e. Open `z:\crs_action\ra\ra931-pl.bat` and change the path to reflect the location of your `perl.exe`:

```
set PATH=c:\perl\bin;%PATH%
```

- f. Test the action script before registering it in the cluster:
- i. Stop Hyperion Reporting and Analysis Services on the node where the Shared Services VIP is not mounted.

**Tip:** Use `crs_stat -t -v` to determine where the Reporting and Analysis VIP is mounted.

- ii. On the remaining node where the Reporting and Analysis VIP is mounted, in a command shell, perform these tests:
- Enter `ra931-pl.bat`. This should return the usage.
  - Enter `ra931-pl.bat check`, then `echo %ERRORLEVEL%`. This should return 0 if Reporting and Analysis Services are running or 1 if they are not running.
  - If Reporting and Analysis Services are down, try to start them by entering `ra931-pl.bat start`, then `echo %ERRORLEVEL%`. This should return 0 if the start succeeds or 1 if the start fails.
  - If Reporting and Analysis Services are up, try to stop by entering `ra931-pl.bat stop`, then `echo %ERRORLEVEL%`. This should return 0 if the stop succeeds or 1 if the stop fails.

**Note:** Do not register the Reporting and Analysis profile in the cluster until the start stop and check tests succeed.

### 3 Create and register an application profile for Reporting and Analysis.

The sample `register_ra.bat` file in the Appendix to this document automatically creates an application profile and registers the application profile with Oracle Clusterware. Before running `register_ra.bat`, you must open it and ensure that `ACTION_SCRIPT` and `CRS_HOME` settings are correct as follows:

- `set ACTION_SCRIPT= z:\crs_action\ra\ra931-pl.bat`
- `set CRS_HOME=c:\product\11.1.0\crs\bin`

It is only necessary to run `register_ra.bat` on one node of the cluster.

## Postregistration Reporting and Analysis Tasks

After installing and configuring Reporting and Analysis and registering Reporting and Analysis in the cluster, perform these important tasks:

- If you are installing a service pack, follow the installation notes and these steps
  1. Stop the Reporting and Analysis profile:

```
crs_stop ra-webapp
```
  2. Ensure that Reporting and Analysis services are stopped:

On the node where `raccluster` VIP is running:

    - a. Apply the service pack.
    - b. Run the Application Deployment with WebServer configuration tasks.
    - c. Again update `WSDEPLOYMENT_HOME\WEB-INF\config\ws.conf`:

```
WebClient.UserInterface.Login.RMServer=raccluster:6800
WebClient.Diagnostics.Log.LoggingService= raccluster:6800
```
    - d. Again update `BIPLUS_HOME\InstallableApps\config\WebAnalysis.properties` as follows:

```
ConfigURL=http://raccluster\:19000/workspace/browse/configURL
```
    - e. Again stop and set startup type to Manual for these services:
      - Hyperion S9 Apache 2.0
      - Hyperion S9 BI+ 9.3 Workspace
      - Hyperion S9 BI+ 9.3 WebAnalysis
      - Hyperion S9 BI+ 9.3 Financial Reporting Web application
    - f. Run `crs_relocate ravip` to the second node, and repeat step a through step e.
- Run `crs_start ra-webapp -f`

## Managing the Cluster

You use Oracle Clusterware to manage the Shared Services and Reporting and Analysis cluster. You can control Oracle Clusterware manually with the command line tool `crsctl`. You use `crsctl` commands to start and stop Oracle Clusterware and perform tasks such as enabling online debugging and dynamically adding, removing, and backing up voting disks. These commands must be run from a shell in the `C:\product\11.1.0\crs\BIN` folder.

### 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, Reporting and Analysis, and VIPs.

Perform login checks from a Web browser:

- Use <http://hsscluster:58080/interop/index.jsp>, to ensure that Shared Services runs correctly.
- Use <http://racluster:19000/workspace/index.jsp>, to ensure that Reporting and Analysis runs correctly.

## Monitoring the Cluster

The logical application name registered in the cluster with `register_hss.bat` is `hss-webapp`, and resources monitored include Native Directory and Web application. For Oracle's Hyperion® Reporting and Analysis – System 9, `ra-webapp` is registered

These commands are available to for monitoring a cluster, where *Application\_Name* is `hss-webapp` or `ra-webapp`:

- `crsctl start crs`—Starts these services on the current node:
  - OracleCSService
  - OracleEVMService
  - OracleCRSService
- `crsctl stop crs`—Stops OracleCSService, OracleEVMService, and OracleCRSService on the current node.

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

- `crsctl check crs`—Verifies that the services OracleCRService, OracleCSService, and OracleEVMService are running on the node from which you are checking.
- `crs_start Application_Name`—Starts the service related to the specified application. This causes Oracle Clusterware to call the action programs with the start parameter.

---

**Caution!** The application service must be started 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 virtual IP).

- `crs_stat Application_Name`—Calls the action programs (such as `hss931.bat` or `ra931-pl.bat`) 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 the status of the applications.
- `crs_stop Application_Name`—Stops the application and the service related to the given application, causing Oracle Clusterware to call the action programs with the stop parameter.

---

**Caution!** Use only the `crs_stop Application_Name` command to stop the application service. 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 Oracle's Hyperion® Shared Services virtual IP).

**Note:** In a normal stop, the target status of the application is Offline, and Oracle Clusterware does not restart the application in case of a failure.

When you stop 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 -f Application_Name`—Stops the specified application and starts it on an other node in the cluster.

The `crs_start`, `crs_stop`, and `crs_stat` commands become unavailable when the Cluster Ready Service (OracleCRService) 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.
- Cluster Registry—Manages cluster and Oracle Real Application Clusters database configuration information.

### Voting Disk Backup and Recovery

Oracle recommends backing up the voting disk after the installation. This backup is to be done only after the initial cluster is created.

- To back up the voting disk:
  - 1 Stop Oracle Clusterware on all nodes with the command `crsctl stop crs`.
  - 2 Determine the number of voting disks with the command `crsctl query css votedisk` (on all the nodes).
  - 3 Create a backup script for each voting disk, using the command `c:\product\11.1.0\crs\bin\ocopy \\.votedsk1 c:\backuplocation`, and copy the backup files to a secure location.
  - 4 Restart Oracle Clusterware on all nodes with the command `crsctl start crs`.
- To recover backed up voting disks:
  - 1 Stop Oracle Clusterware on all nodes with the command `crsctl stop crs`.

**2 Enter this command:**

```
c:\product\11.1.0\crs\BIN\ocopy backup_location\backup_votedsk1_datexyz
\.\votedsk1
```

**3 Restart Oracle Clusterware on all nodes with the command `crsctl start crs`.**

## Oracle Clusterware Registry Backup and Recovery

You can back up the Oracle Clusterware registry automatically or manually. For information about automatic backups, see the Oracle Clusterware documentation. You can back up the registry manually at any time. For example, you can obtain a binary backup before making changes to the Oracle Clusterware registry.

► To back up the Oracle Clusterware registry:

- 1 Create the destination directory for the backup file on one node.**
- 2 Check Oracle Clusterware registry information, using the command `ocrcheck`.**
- 3 Enter the command `ocrconfig -manualbackup`.**
- 4 Copy the backup file to the other node.**

If an application fails, restart the application before attempting to restore the Oracle Clusterware registry.

► To determine whether the Oracle Clusterware registry has failed, use the `ocrcheck` command.

If the `ocrcheck` command returns a failure message, then both the primary Oracle Clusterware registry and the Oracle Clusterware registry mirror have failed. Attempt to correct the problem using one of the following platform-specific restoration procedures.

► To recover the Oracle Clusterware registry:

- 1 Enter the command `ocrconfig -showbackup`, and note the name of the latest backup file.**
- 2 On all remaining nodes, using the Service Control Panel, disable and stop these Oracle Clusterware services:**
  - OracleClusterVolumeService
  - OracleCSService
  - OracleCRService
  - OracleEVMSERVICE
- 3 Run the command `ocrconfig -restore path`, where *path* is the location and name of the latest backup file; for example, `ocrconfig -restore C:\product\11.1.0\crs\cdata\cluster\backup00.ocr`.**
- 4 Restart the Oracle Clusterware services and nodes:**
  - Start all of the services that you stopped in [step 2](#).
  - Restart all of the nodes, and resume operations in cluster mode.
- 5 Run this command to verify the Oracle Clusterware registry integrity:**

```
cluvfy comp ocr [-n node_list] | all [-verbose]
```

The `-n node_list | all` argument retrieves a list of all nodes that are configured as part of the cluster.

## Log Files

The global log for all Oracle Clusterware daemons is `c:\product\11.1.0\crs\log\server1>alertserver1.log`. Oracle Clusterware also creates these detail log files:

- All client activities on Oracle Clusterware—`c:\product\11.1.0\crs\log\server1\client`
- Oracle Clusterware `crsd` daemon—`c:\product\11.1.0\crs\log\server1\crsd`
- Oracle Clusterware `cssd` daemon—`c:\product\11.1.0\crs\log\server1\cssd`
- Oracle Clusterware `evmd` daemon—`c:\product\11.1.0\crs\log\server1\evmd`

## Additional Information

You can download more information about high availability Oracle Clusterware from [http://download.oracle.com/docs/cd/B28359\\_01/rac.111/b28255.pdf](http://download.oracle.com/docs/cd/B28359_01/rac.111/b28255.pdf).

## Appendix

### createvip\_hss.bat

```
set APPVIPNAME="hssvip"
set ADAPTER="Public"
set VIPIP=143.46.43.30
set VIPSUBNET=255.255.0.0
set START_TIMEOUT=300
set STOP_TIMEOUT=300
set SCRIPT_TIMEOUT=300
set RESTART_ATTEMPTS=2
set CRS_HOME=c:\product\11.1.0\crs\bin
rem profile the uservip
%CRS_HOME%\crs_profile -create %APPVIPNAME% -t application -a %CRS_HOME%
\usrvip.bat -o rt=%START_TIMEOUT%,pt=%STOP_TIMEOUT%,st=%SCRIPT_TIMEOUT%,ra=
%RESTART_ATTEMPTS%,oi=%ADAPTER%,ov=%VIPIP%,on=%VIPSUBNET%
rem register the usrvip
%CRS_HOME%\crs_register %APPVIPNAME%
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPVIPNAME%
rem Now we need to set the permission on the create application vip.
%CRS_HOME%\crs_setperm %APPVIPNAME% -o system
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPVIPNAME%
rem =====
rem NOW you should be able to start the resource using crs_start %APPVIPNAME
```



```
%
%CRS_HOME%\crs_start %APPVIPNAME%
Pause
```

## Createvip\_ra.bat

```
set APPVIPNAME="ravip"
set ADAPTER="Public"
set VIPIP=143.46.43.31
set VIPSUBNET=255.255.0.0
set START_TIMEOUT=300
set STOP_TIMEOUT=300
set SCRIPT_TIMEOUT=300
set RESTART_ATTEMPTS=2
set CRS_HOME=c:\product\11.1.0\crs\bin
rem profile the uservip
%CRS_HOME%\crs_profile -create %APPVIPNAME% -t application -a %CRS_HOME%
\usrvip.bat -o rt=%START_TIMEOUT%,pt=%STOP_TIMEOUT%,st=%SCRIPT_TIMEOUT%,ra=
%RESTART_ATTEMPTS%,oi=%ADAPTER%,ov=%VIPIP%,on=%VIPSUBNET%
rem register the usrvip
%CRS_HOME%\crs_register %APPVIPNAME%
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPVIPNAME%
rem Now we need to set the permission on the create application vip.
%CRS_HOME%\crs_setperm %APPVIPNAME% -o system
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPVIPNAME%
rem =====
rem NOW you should be able to start the resource using crs_start
%APPVIPNAME
%
%CRS_HOME%\crs_start %APPVIPNAME%
Pause
```

## register\_hss.bat

```
rem =====
set APPNAME="hss-webapp"
set ACTION_SCRIPT=y:\CRS_ACTION\HSS\hss931-pl.bat
set DESCRIPTION=HyperionSharedServices
set REQUIRED_RESSOURCES="hssvip"
set FAILOVER_DELAY=0
set FAILURE_THRESHOLD=5
set FAILURE_INTERVAL=600
set START_TIMEOUT=600
set STOP_TIMEOUT=600
set SCRIPT_TIMEOUT=600
set RESTART_ATTEMPTS=2
set CRS_HOME=c:\product\11.1.0\crs\bin
rem profile the uservip
%CRS_HOME%\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%
rem register the application
```

```

%CRS_HOME%\crs_register %APPNAME%
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPNAME%
rem Now we need to set the permission on the create application.
%CRS_HOME%\crs_setperm %APPNAME% -o system
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPNAME%
rem =====
rem NOW you should be able to start the resource using crs_start %APPNAME%
%CRS_HOME%\crs_start %APPNAME%
Pause

```

## register\_ra.bat

```

rem =====
set APPNAME="ra-webapp"
set ACTION_SCRIPT=z:\CRS_ACTION\ra\ra931-pl.bat
set DESCRIPTION=ReportingAndAnalysis
set REQUIRED_RESSOURCES="ravip"
set FAILOVER_DELAY=0
set FAILURE_THRESHOLD=5
set FAILURE_INTERVAL=600
set START_TIMEOUT=600
set STOP_TIMEOUT=600
set SCRIPT_TIMEOUT=600
set RESTART_ATTEMPTS=2
set CRS_HOME=c:\product\11.1.0\crs\bin
rem profile the uservip
%CRS_HOME%\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%
rem register the application
%CRS_HOME%\crs_register %APPNAME%
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPNAME%
rem Now we need to set the permission on the create application.
%CRS_HOME%\crs_setperm %APPNAME% -o system
rem see who owns the resource
%CRS_HOME%\crs_getperm %APPNAME%
rem =====
rem NOW you should be able to start the resource using crs_start %APPNAME%
%CRS_HOME%\crs_start %APPNAME%
Pause

```

## hss931-pl.bat

```

@echo off
set PATH=c:\perl\bin;%PATH%
perl hss931.pl %1

```

## ra931-pl.bat

```
@echo off
set PATH=c:\perl\bin;%PATH%
perl ra931.pl %1
```

## hss931.pl

```
#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
# hss931.pl
# This perl script is the action script for start / stop / check
# the Oracle Shared Services web application and native directory
# in a cold failover configuration.
#
# This is a sample script to be adapted to your configuration
# Runs on windows platforms

use strict;
use Win32::Service;
use LWP;

# Environment settings please adapt

my $SCRIPT_PATH = "y:/CRS_ACTION/HSS";
my $VER          = "1.0";
my $DEBUG       = "false";
my $HSS_URL     = "http://hsscluster:58080/interop/index.jsp";
my $RESP_TIMEOUT = "40"; #If $HSS_URL does not respond in RESP_TIMEOUT
my $HSS_WEBAPP_SERVICENAME="HYS9SharedServices";
my $HSS_NATIVEDIR_SERVICENAME="OpenLDAP-slapd";
my $CRS_ACTION_LOG = "$SCRIPT_PATH/hssaction.log";
my $ORA_CRS_HOME   = "C:\product\11.1.0\crs";
my $HYPERION_SHARED = "y:/hyperion";
my $SLEEP_INTERVAL = 30;
my $LOG_ENABLED    = "true";

#Service status definition
my %service_status_hash = (
    '1' => 'Stopped',
    '2' => 'Starting',
    '3' => 'Stop pending',
    '4' => 'Running',
    '5' => 'Continue pending',
    '6' => 'Pause pending',
    '7' => 'Paused'
);
if ($#ARGV != 0 ) {
    print "Usage: hss931.pl [start|stop|check] \n";
    exit;
}
if ($^O !~ /MSWin(.+)/) {
    print "Please use unix version instead. This script is for $^O
platforms \n";
    exit;
}
```

```

my $command = $ARGV[0];

# Listener start / stop check

# start listener
if ($command eq "start" || $command eq "\"start\"") {
    my %service_status_nativedir;
    my %service_status_webapp;

    &log("");
    &log("---- Begin START ----");
    sleeping(360, "Waiting shared disk");
    # start native directory service
    Win32::Service::StartService(Win32::NodeName( ),
    $HSS_NATIVEDIR_SERVICENAME);
    sleeping ($SLEEP_INTERVAL, "Starting OpenLDAP service");

    my $res = 0;
    $res = service_check($HSS_NATIVEDIR_SERVICENAME, 4); #Running
    if ($res) {#openLDAP is not Running
        # run recover openldap
        # --- shut down hss web app and native directory
        # --- db_recover -c
        Win32::Service::StopService(Win32::NodeName( ),
    $HSS_NATIVEDIR_SERVICENAME);
        Win32::Service::StopService(Win32::NodeName( ),
    $HSS_WEBAPP_SERVICENAME);
        sleeping($SLEEP_INTERVAL, "Stopping HSS services to perform
    OpenLDAP recovery");

        $res = service_check($HSS_NATIVEDIR_SERVICENAME, 1); #Stopped
        if ($res) { # if service is stopped
            &log("Performing OpenLDAP recovery");
            chdir("$HYPERION_SHARED/SharedServices/931/openLDAP/var/
    openldap-data");
            system("$HYPERION_SHARED/SharedServices/931/openLDAP/bdb/bin/
    db_recover ");

            chdir($SCRIPT_PATH);
        }

        Win32::Service::StartService(Win32::NodeName( ),
    $HSS_NATIVEDIR_SERVICENAME);
        sleeping($SLEEP_INTERVAL, "Starting OpenLDAP Service");

    }

    Win32::Service::StartService(Win32::NodeName( ),
    $HSS_WEBAPP_SERVICENAME);
    sleeping($SLEEP_INTERVAL, "Starting HSS Web Application service");
    sleeping($SLEEP_INTERVAL, "Waiting for HSS Web application to be up");

    $res = services_check(4); #Running

    if ($res) {&log("HSS Services weren't started!");}
    else {&log("HSS Services were started successfully.");}
}

```

```

        &log("----- End START -----");
        exit $res;
    }

# stop listener
if ($command eq "stop" || $command eq "\"stop\"") {
    &log("");
    &log("----- Begin STOP -----");

    Win32::Service::StopService(Win32::NodeName( ),
    $HSS_WEBAPP_SERVICENAME);
    sleeping($SLEEP_INTERVAL, "Stopping HSS Web Application");
    Win32::Service::StopService(Win32::NodeName( ),
    $HSS_NATIVELDAP_SERVICENAME);
    sleeping($SLEEP_INTERVAL, "Stopping openLDAP service");

    my $res = services_check(1); #Stopped
    if ($res) {&log("HSS Services weren't stopped!");}
    else {&log("HSS Services were stopped successfully.");}

    &log("----- End STOP -----");
    exit $res;
}

# check listener
if ($command eq "check" || $command eq "\"check\"") {
    if ($DEBUG eq "true") {
        &log("");
        &log("----- Begin CHECK -----");
    }

    my $res = 0;

    $res = http_check("$HSS_URL");

    $res |= services_check(4); #Running

    if ($res) {
        if ($DEBUG eq "true") { &log("Not all HSS Services are running!");}
    } else {
        if ($DEBUG eq "true") { &log("HSS Services are running.");}
    }

    if ($DEBUG eq "true") { &log("----- End CHECK -----");}
    exit $res;
}

#sleeping procedure sleeps and writes into log file dots...
sub sleeping() {
    my $timeout = $_[0];
    my $action = $_[1];
    my $interval = 30;

    open LOGFILE , ">>$CRS_ACTION_LOG";

```

```

my $now=localtime(time);
print LOGFILE "$now - $action";
close LOGFILE;

my($i) = 0;
for($i = 1; $i <= $interval; $i++) {
    open LOGFILE , ">>$CRS_ACTION_LOG";
    print LOGFILE ".";
    sleep ($timeout/$interval);
    close LOGFILE;
}
open LOGFILE , ">>$CRS_ACTION_LOG";
print LOGFILE "\n";
close LOGFILE;
}

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

sub http_check() {
    my $URL = $_[0];

    my $client = LWP::UserAgent->new;
    $client->timeout($RESP_TIMEOUT);
    my $resp = $client->post( $URL,
        [ 'param1' => 'value1', # any post information that may be added
          'param2' => 'value2' # dummy example
        ]
    );
    if ($resp->is_success) {
        if ($DEBUG eq "true") { &log("URL check successful on
'$URL'"); }
        return 0;
    } else {
        my $resp_status=$resp->status_line;
        if ($DEBUG eq "true") { &log("URL check failed on '$URL' with status
'$resp_status'"); }
        return 1;
    }
}

sub services_check() {
    my $STATUS = $_[0]; #'4' for running and '1' for Stopped

    # query HSS services status
    my $res = 0;
    $res = service_check($HSS_NATIVEDIR_SERVICENAME, $STATUS);
    $res |= service_check($HSS_WEBAPP_SERVICENAME, $STATUS);
}

```

```

    return $res;
}

sub service_check() {
    my $SERVICE = $_[0];
    my $STATUS = $_[1];
    my $res = 0;
    my %service_status;
    Win32::Service::GetStatus( '', $SERVICE, \%service_status);
    if ($service_status{"CurrentState"} != $STATUS) {
        if ($DEBUG eq "true") { &log("$SERVICE status:
$service_status_hash{$service_status{"CurrentState"}}"); }

        $res = 1;
    }
    return $res;
}

```

## ra931.pl

```

#!/usr/bin/perl
# Copyright (c) 2002, 2006, Oracle. All rights reserved.
# ra931.pl
# This perl script is the action script for start / stop / check
# the Oracle Hyperion Reporting and Analysis services
# in a cold failover configuration.
#
# This is a sample script to be adapted to your configuration
# Runs on windows platforms

use strict;
use Win32::Service;
use LWP;

# Environment settings please adapt

my $SCRIPT_PATH = "Z:/CRS_ACTION/ra";
my $VER          = "1.0";
my $WEB_URL      = "http://racluster:19000";
my $WS_URL       = "$WEB_URL/workspace/index.jsp";
my $WA_URL       = "$WEB_URL/WebAnalysis/WebAnalysis.jsp";
my $FR_URL       = "$WEB_URL/hr/status.jsp";

my $APACHE_SERVICE = "HyS9Apache2";

my $RESP_TIMEOUT = "40"; #If URLs do not respond in RESP_TIMEOUT

my $CRS_ACTION_LOG = "$SCRIPT_PATH/ra_action.log";
my $ORA_CRS_HOME   = "C:\product\11.1.0\crs";
my $BIPLUS_HOME    = "Z:/Hyperion/BIPlus";
my $IR_STARTSCRIPT = "$BIPLUS_HOME/bin/start_BIPlus.bat";
my $IR_STOPSCRIPT  = "$BIPLUS_HOME/bin/stop_BIPlus.bat";

```

```

my $SLEEP_INTERVAL = 120;
my $LOG_ENABLED = "true";

#Service status definition
my %service_status_hash = (
    '1' => 'Stopped',
    '2' => 'Starting',
    '3' => 'Stop pending',
    '4' => 'Running',
    '5' => 'Continue pending',
    '6' => 'Pause pending',
    '7' => 'Paused'
);
if ($#ARGV != 0 ) {
    print "Usage: ra931.pl [start|stop|check] \n";
    exit;
}
if ($^O !~ /MSWin(.+)/) {
    print "Please use unix version instead. This script is for $^O
platforms \n";
    exit;
}
my $command = $ARGV[0];

# Listener start / stop check

# start listener
if ($command eq "start" || $command eq "\"start\"") {
    &log("");
    &log("----- Begin START
-----");

    sleeping(360, "Wait shared disk");

    Win32::Service::StartService(Win32::NodeName( ), $APACHE_SERVICE);
    system("$IR_STARTSCRIPT");

    sleeping($SLEEP_INTERVAL, "Starting IR&A Services");
    #verification part for startup

    my $res = services_check(4);

    if ($res) {&log("R&A Services weren't started!");}
    else {&log("R&A Services were started successfully.");}

    &log("----- End START
-----");
    exit $res;
}

# stop listener
if ($command eq "stop" || $command eq "\"stop\"") {
    &log("");
    &log("----- Begin STOP
-----");
}

```



```

-----");

system("$IR_STOPSCRIPT");
Win32::Service::StopService(Win32::NodeName( ), $APACHE_SERVICE);

sleeping($SLEEP_INTERVAL, "Stopping IR&A Services");
#verification part for shutdown
my $res = services_check(1);

if ($res) {@log("R&A Services weren't stopped!");}
else {@log("R&A Services were stopped successfully.");}

&log("----- End STOP
-----");
exit $res;
}

# check listener
if ($command eq "check" || $command eq "\"check\"") {
    &log("");
    &log("----- Begin CHECK
-----");

    my $res = 0;

    $res = http_check("$WS_URL");
    $res |= http_check("$FR_URL");
    $res |= http_check("$WA_URL");

    $res |= services_check(4);

    if ($res) {@log("Not all R&A Services are running!");}
    else {@log("R&A Services are running.");}

    &log("----- End CHECK
-----");
    exit $res;
}

#sleeping procedure sleeps and writes into log file dots...
sub sleeping() {
    my $timeout = $_[0];
    my $action = $_[1];
    my $interval = 30;

    open LOGFILE , ">>$CRS_ACTION_LOG";
    my $now=localtime(time);
    print LOGFILE "$now - $action";
    close LOGFILE;

    my($i) = 0;
    for($i = 1; $i <= $interval; $i++) {
        open LOGFILE , ">>$CRS_ACTION_LOG";
        print LOGFILE ".";
        sleep ($timeout/$interval);
        close LOGFILE;
    }
}

```

```

    }
    open LOGFILE , ">>${CRS_ACTION_LOG}";
    print LOGFILE "\n";
    close LOGFILE;
}

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

sub http_check() {
    my $URL = $_[0];

    my $client = LWP::UserAgent->new;
    $client->timeout($RESP_TIMEOUT);
    my $resp = $client->post( $URL,
        [ 'param1' => 'value1', # any post information that may be added
          'param2' => 'value2' # dummy example
        ]
    );
    if ($resp->is_success) {
        &log("URL check successful on '$URL'");
        return 0;
    } else {
        my $resp_status=$resp->status_line; &log("URL check failed on
'$URL' with status '$resp_status'");
        return 1;
    }
}

sub services_check() {
    my $STATUS = $_[0]; #'4' for running and '1' for Stopped

    my $HyS9Core = "HYS9Core1";
    my $HyS9IR   = "HyS9IRsvcl";

    my $HyS9Workspace = "HyS9Workspace";
    my $HyS9WebAnalysis = "HyS9WebAnalysis";

    my $HyS9FRRMI      = "HyS9FRRMI";
    my $HyS9FRPrint    = "HyS9FRPrint";
    my $HyS9FRReport   = "HyS9FRReport";
    my $HyS9FRSched    = "HyS9FRSched";
    my $HyS9FRWeb      = "HyS9FRWeb";

    # query IR&A services status
    my $res = 0;
    $res = service_check($APACHE_SERVICE, $STATUS);

    $res |= service_check($HyS9FRRMI, $STATUS);
}

```

```

$res |= service_check($HyS9FRPrint, $STATUS);
$res |= service_check($HyS9FRReport, $STATUS);
$res |= service_check($HyS9FRSched, $STATUS);

$res |= service_check($HyS9Core, $STATUS);
$res |= service_check($HyS9IR, $STATUS);

$res |= service_check($HyS9Workspace, $STATUS);
$res |= service_check($HyS9FRWeb, $STATUS);
$res |= service_check($HyS9WebAnalysis, $STATUS);

return $res;
}

sub service_check() {
    my $SERVICE = $_[0];
    my $STATUS = $_[1];
    my $res = 0;
    my %service_status;
    Win32::Service::GetStatus( '', $SERVICE, \%service_status);
    if ($service_status{"CurrentState"} != $STATUS) {
        &log("$SERVICE status:
$service_status_hash{$service_status{"CurrentState"}}");
        $res = 1;
    }
    return $res;
}

```

## COPYRIGHT NOTICE

Shared Services Shared Services and Reporting and Analysis High Availability (Windows Environments), 9.3.1

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

Authors: Eric Belmon, Paula Rubenstein, Cheryl Morrison

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.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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.