Configuring Highly Available OracleAS Infrastructure with Foundry Networks ServerIron Hardware Load Balancer

An Oracle-Foundry Networks White Paper
February 2006

ORACLE

FOUNDRY NETWORKS

Configuring Highly Available OracleAS Infrastructure with Foundry Networks ServerIron Hardware Load Balancer

Configuring Highly Available OracleAS Infrastructure with Foundry Networks ServerIron Hardware Load Balancer

EXECUTIVE OVERVIEW

Oracle Application Server Infrastructure provides a centralized security and management platform for deploying business-critical applications. It includes Oracle Identity Management as well as a facility for centralized Product Metadata management and configuration management. The infrastructure was introduced in Oracle9i Application Server 9.0.2 and has continued to provide this role in subsequent releases including the current release Oracle Application Server 10*g* Release 2 (10.1.2.0.0) and Oracle Application Server 10*g* Release 2 (10.1.2.0.2).

A highly available Oracle Application Server deployment requires an Infrastructure service designed for high availability. Oracle Application Server 10*g* supports infrastructure deployments in a variety of Highly Available (HA) architectures each suited to different customer requirements. The primary HA solutions are - Cold Failover Cluster (CFC) and OracleAS Cluster (Identity Management). OracleAS clusters deployed using an Oracle Real Application Clusters (RAC) is active–active in nature. Active-Active implies that there are multiple instances of the infrastructure processes on different servers, concurrently servicing requests. In all active-active deployments, the incoming requests for OID (Oracle Internet Directory), SSO (Single Sign On) & DAS (Delegated Administrative Service) service are distributed across the simultaneously active instances using a hardware load balancer. Failure of any one of the instances simply results in the load balancer directing the subsequent requests to the surviving instances. Thus the hardware load balancer is an integral part of the architecture and provides load balancing as well as failover capabilities.

Foundry Networks ServerIron Load Balancer provides necessary features to work with Highly Available Oracle Application Server Infrastructure deployment. Besides availability, Foundry Server Iron's integrated SSL acceleration hardware can be used to act as an SSL proxy for HTTPS based connections for Single Sign On and DAS to accelerate performance. To ensure that the hardware load balancer is not a single point of failure,, a pair of ServerIron load balancers are always deployed with hitless session failover between them.

This paper has been jointly written by Oracle and Foundry Networks, and describes the configuration and operational best practices associated with using Foundry Networks ServerIron Hardware Load Balancer & Integrated SSL Acceleration in Highly Available OracleAS Infrastructure deployment.

The rest of this paper applies to active-active HA deployments of OracleAS Infrastructure and Identity Management with OracleAS 10.1.2.0.x releases and OracleAS 9.0.4 release.

ACTIVE-ACTIVE INFRASTRUCTURE SOLUTIONS

The primary active-active infrastructure High Availability solution in Oracle Application Server 10g is OracleAS Cluster (Identity Management) deployed on an Oracle Real application cluster backend database. For simplicity, we can assume the deployment to be made of three tiers - database tier, OID tier and the IM Middle Tier (which has the Oracle HTTP server and an OC4J instance running the Oracle Single Sign on and/or the Delegated Administrative Service application). The following block diagrams show the two major deployments for the architectures. It is assumed that the database tier is always Real Application Clusters (RAC) for the active-active solution. SSO and DAS have been shown together in the cases here, but these can be deployed in separate OracleAS instances if necessary.

OracleAS Cluster (Identity Management)

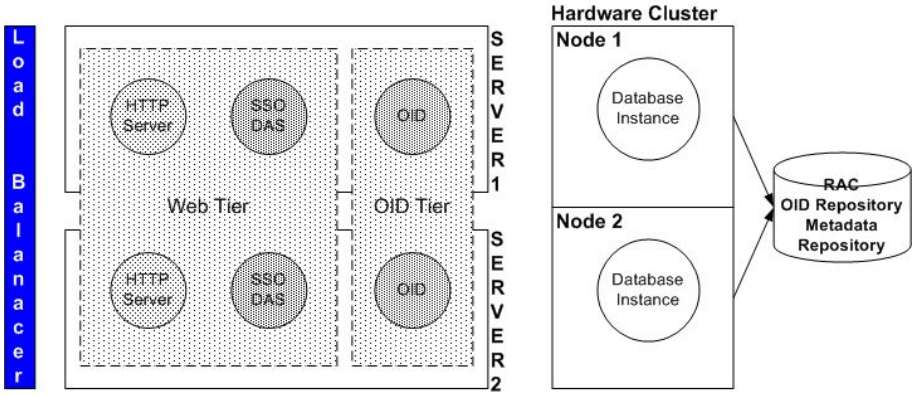


Figure 1 OracleAS Cluster Configuration

The OracleAS Cluster Identity Management HA deployment (Figure 1) has the database tier as a RAC cluster with the OID tier and the IM Middle Tier together on two separate machines (for redundancy). The database tier is created in a pre-existing RAC database using OracleAS MRCA (Metadata Repository Configuration Assistant). The database version can be 9.2.0.6 and above or 10.1.0.4 and above. Or 10.2.x. The OID and IM Middle Tier are installed together in the same Oracle Home on multiple machines. Each install on a machine is a separate install session.

Distributed OracleAS Cluster (Identity Management)

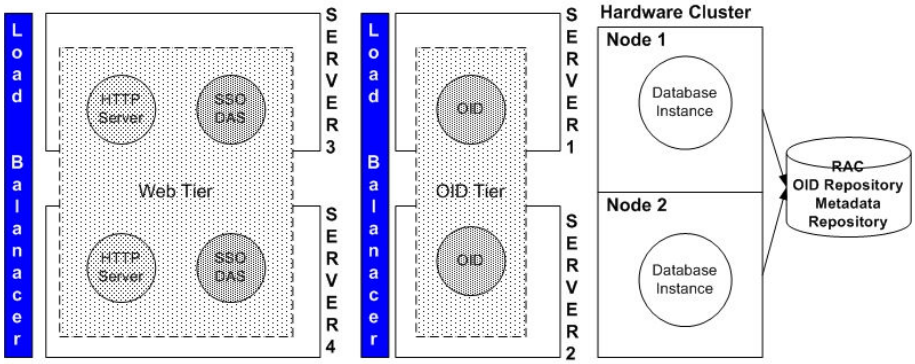


Figure 2 Distributed OracleAS Cluster Configuration

A distributed OracleAS Cluster configuration (Figure 2) has the RAC database tier on a hardware cluster and the OID tier on (at least) two separate machines and the IM Middle Tier on (at least) two separate machines. In this case, the OID tier can also be co-located on the same RAC cluster but in separate Oracle Home from the database installation. The database tier is created using OracleAS repCA (repository configuration assistant) with a pre-existing RAC database. The OID tier is installed separately on at least two separate machines. Each is an individual install. The IM Middle Tier is also installed on multiple machines with each install being a separate install session.

A full description of these architectures and their detailed installation is available in the Oracle documentation. Please refer to the following guides.

- **Oracle Application Server High Availability Guide**
- **Oracle Application Server Installation Guide**

USING LOAD BALANCERS WITH ACTIVE-ACTIVE INFRASTRUCTURE

In all variations of the above-mentioned architectures, a hardware load balancer is configured to direct incoming requests for LDAP traffic to the OID tier and the HTTP traffic to the IM Middle Tier. These requests may come from clients in the infrastructure, from the mid-tier nodes or from the user browser. Oracle Net traffic to the database is load balanced using Oracle Net load balancing with Oracle Net connect descriptors and multiple addresses in its address list.

The typical OID LDAP connections are established from SSO/DAS and from Mid-tier components at startup of the components and they remain available throughout the lifetime of the connection and are connection-oriented. Therefore, the load balancer should not timeout these connections. There are also other OID LDAP requests that live only for the moment of the request. The OID connections are established on two ports – one of these is used for SSL encrypted connections to OID and the other is used for non-SSL connections. Each port gets defined at install time. The ports may be automatically selected by the installer or may be pre-defined by the user. It is strongly recommended that the install for the active-active HA infrastructure be done with pre-defined ports using the "Static Ports" feature of the Oracle Installer. When ports are chosen by the installer, the default for non-SSL OID port is 389 (if not in use) and the possible port is a free one from the range 389, 3060-3129; the default for SSL OID port is 636 and the possible port is a free one from the range 636, 3130-3199.

The client connections to Oracle HTTP Server are primarily for the SSO & DAS services. These could be HTTP or HTTPS. Most of these come from outside the infrastructure itself and mainly from the user's browser. The load balancer will balance this traffic to the HTTP server on any of the nodes. The HTTP server forwards the request to the local OC4J instance to fulfill the SSO/DAS requests. The SSO HTTP/HTTPS requests are stateless in nature. There is no stickiness requirement for these connections. The DAS HTTP/HTTPS requests are state oriented and require stickiness. SSO and DAS are two separate applications running in the same OC4J container. A shared set of HTTP processes directs traffic to this OC4J container. The HTTP/HTTPS ports that are used get defined at install time. The ports may be automatically selected by the installer or may be pre-defined by the user. It is strongly recommended that the install for the active-active HA infrastructure be done with pre-defined ports using the "Static Ports" feature of the Oracle Installer. When ports are chosen by the installer, the default for the HTTP port is 7777 and the possible port is a free one from the range 7777-7877; the default for the HTTPS port is 4443 and the possible port is a free one from the range 4443-4543.

In many deployments, it is possible that a firewall separates the load balancer and some of the tiers. In such cases, appropriate ports need to be opened up in the firewall. Depending on where the tiers are placed with respect to the firewall, the following ports need to be opened up – Oracle Net port, oid ports for SSL and non-SSL connections, HTTP and HTTPS ports.

The block diagrams above may show more than one load balancer, but it is likely that the same load balancer device is used for both the OID tier and the IM Middle Tier.

In the OracleAS (Identity Management) configuration, both the OID tier and the IM Middle Tier can use the same load balancer virtual server hostname or use different load balancer virtual server hostname (in OracleAS 10.1.2 only). In the distributed OracleAS (Identity Management) configuration, OID tier and IM Middle Tier each will typically use a different load balancer virtual server and use their respective ports.

FOUNDRY NETWORKS SERVERIRON HARDWARE LOAD BALANCER

Hardware load balancer acts as traffic cop in front of the servers, and uses its application-level intelligence to ensure high service uptime and superior scalability. It efficiently distributes client requests to the "best" server in the pool. The load balancers consider server availability, load, response time, and other performance metrics in server selection to provide the best end-user response time and availability. Using application-specific "health checks" to servers and applications, hardware load balancers identify unavailable resources in real time and switch users to available servers. They also help scale server farms on demand by adjusting server capacity without impacting application availability. When more capacity is

needed, a new server or application instance can simply be added to the pool without impacting the end user.

Foundry Networks ServerIron Load Balancer offers layer 4-7 application traffic management and web optimization capabilities in a dedicated network-centric hardware platform. These devices provide server load balancing, application aware server health checking, layer 7 content switching, cookie persistence and global load balancing across multiple sites. With integration of SSL hardware, the ServerIron load balancer can terminate incoming SSL traffic and accelerate the performance of HTTPS applications by offloading the overhead of SSL processing from the real servers.

The ServerIron load balancers can be configured in several highly available (HA) redundant designs to support highly available OracleAS deployments. In the example described in this document, the two ServerIron load balancers we utilized in a Active-Hot Standby HA design. In this design, the active ServerIron device processes all traffic while the second device idles as it awaits the failure of the active device. TCP sessions are synchronized between the two devices to support hitless failover of traffic when the active ServerIron fails.

This document assumes familiarity with the Foundry Networks ServerIron Load Balancer. Some basic terminologies are explained below to provide the context for implementation and configuration details presented in this document. Readers are encouraged to refer to the Foundry Networks ServerIron L4-7 Software Configuration Guide for complete presentation of the device's capabilities and configuration.

Basic Terminologies

Real server

Real Servers are Application Servers. They usually sit behind the ServerIron hardware load balancer. In a typical network, there would be one or more real servers running the same application instance to service clients/users. Load Balancers allow administrators to add or remove real servers from the available pool without impacting service availability.

Virtual server

Virtual server represents the logical instance of the application residing on the hardware load balancer, which in turn is mapped to a pool of real servers that actually provide content and application services. There can be one ore more virtual server instances defined on the load balancer. Clients connect to the virtual server IP address assigned to the hardware load balancer, and the load balancer distributes these requests among multiple real servers.

Server Load Balancing (SLB)

SLB offers numerous benefits such as ease of overall server farm administration for TCP/UDP applications, and increases service availability, performance and reliability with efficient load distribution and automatic failover. SLB allows IT managers to achieve greater efficiency in server resource management. Server resources can be added or removed depending on traffic requirements without disrupting service to end-users.

Health Checks

Health check is a mechanism by which the hardware load balancer verifies that an application instance or the server is capable of delivering appropriate service in response to end-user client requests. The ServerIron load balancer uses layer 3, layer 4, layer 7 and scripted health checks to verify availability of real servers and application instances on the real servers. If a real server or an application on a real server fails health checks, then the load balancer will take that instance out of rotation.

Load Balancing Predictor

The predictor algorithm determines how to balance client requests across multiple real servers. ServerIron offers following predictor methods:

- Least Connections
- Round Robin
- Weighted
- Server Response Time
- Least Connections + Server Response Time
- Least Local Connections

Sticky Connections

If an application requires series of sequential TCP/UDP port connections to be serviced by the same real server, then sticky feature can be enabled for that virtual application port. By default sticky is disabled for virtual server ports except for SSL connections where it's enabled by default.

Global Server Load Balancing (GSLB)

As reliance on IP and Web applications for business operations increases, organizations need to deliver application services from multiple locations for redundancy and scalability. GSLB uses DNS-based approach to distribute client load to multiple locations and offers automatic failover between sites for disaster recovery and multi-site scalability. GSLB is transparent to OracleAS application implementation and deployment, and is therefore not covered in the following

sections in detail. It can be added as an additional solution to offer redundancy across sites.

SSL Acceleration

The Secure Sockets Layer (SSL) was developed to provide security and privacy over the Internet. Today, most secure applications over the Internet use SSL. SSL provides secure pipe and allows protocols such as http, ftp and LDAP to run inside it.

SSL handshake involves server and optional client authentication using Public Key Infrastructure (PKI). SSL negotiates encryption keys and carries out authentication prior to exchanging data.

ServerIron SSL acceleration comes in two flavors, Management module with integrated SSL and a separate SSL service blade. Hardware SSL acceleration offloads SSL handshake and encryption/decryption from the real server and accelerates end-to-end application performance. It also helps offload the real servers from the overhead of SSL processing, which increases server farm infrastructure and application scalability.

CONFIGURING ACTIVE-ACTIVE INFRASTRUCTURE USING SERVERIRON LOAD BALANCER

This document considers the most generic configuration case for further discussion. Note that it is assumed here that basic setup of the load balancer has already been done. The load balancer configuration discussed further is limited to what is required for the active-active OracleAS configuration.

- Decide on the
    - OID virtual server name (oid.orcl.com in this example)
    - HTTP virtual server name (login.orcl.com in this example)
    - OIDPort (port for OID)
    - OIDSSLPort (port for OID over SSL)
    - VsHTTPPort (Virtual server HTTP port)
    - VsHTTPSPort. (Virtual server HTTPS port)
    - RsHTTPPort (Real server HTTP port)
    - RsHTTPSPort (Real server HTTP port)
- Get IP addresses assigned to the virtual servers and ensure that they are part of your Domain Name Server (DNS).
- Do the Foundry Networks ServerIron Load Balancer configuration based on the guidelines below.

- Create the appropriate staticports.ini file for the OracleAS infrastructure install and use the ports decided earlier and consumed in the load balancer configuration.
- If your load balancer and the servers/cluster nodes are in different security zones (separated by one or more firewalls), ensure that appropriate ports are open for the two way traffic across the firewall.
- Do the OracleAS infrastructures install(s). Use the staticports.ini file created above.

Definitions

Port numbers

The following table defines the ports used for reference in this document. For the sake of simplicity, it is assumed that the port used for a given service will be same on all instances of the service. For example, if the port used for OID on server1 is OIDPort, then the assumption is that the same port is used for OID on the redundant instances of OID. In many cases, OracleAS install enforces this automatically.

Service	**Port**
OID port	OIDPort
OID port for SSL connection	OIDSSLPort
Http Port. This is the virtual server port and is used in URL to access SSO and DAS.	VsHTTPPort
Https Port. This is the virtual server port and is used in URL to access SSO and DAS.	VsHTTPSPort
Http Listen port on the real server. OHS (Oracle HTTP server) on the real server is listening on this port.	RsHTTPPort
Http Listen port on the real server. OHS (Oracle HTTP server) on the real server is listening on this port.	RsHTTPSPort

Real servers

The following table lists physical hostnames assumed for the discussion in this document. The list is broken this into three tiers: the database tier, the OID tier and the IM Middle Tier (SSO & DAS).

Server type	**Hostname**
Hosts for the database tier	db1.orcl.com & db2.orcl.com
Hosts for the OID tier	oid1.orcl.com & oid2.orcl.com
Hosts for IM Middle Tier	login1.orcl.com & login2.orcl.com

For the OracleAS cluster Identity Management HA install, the OID tier and the IM Middle Tier are co-located on the same hardware server and are installed together in the same Oracle Home. This implies oid1=login1 and oid2=login2. For the Distributed OracleAS cluster Identity Management HA install, typically all the servers are different.

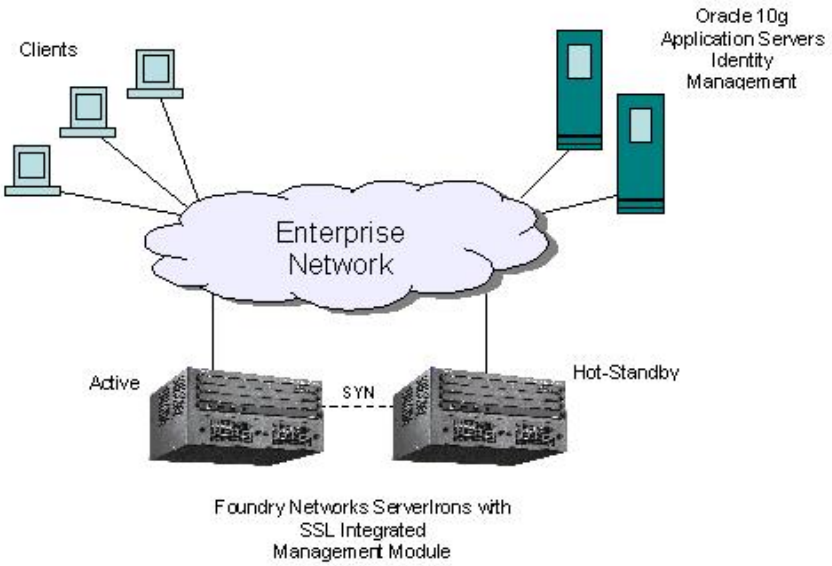
Virtual servers

This table lists the virtual servers for the OID tier and the IM Middle Tier. The virtual server hostname(s) should be part of your Domain Name Server.

The virtual server instance will be created in the hardware load balancer on a virtualhost:port (service) basis.

Virtual Server type	**Virtual Server name**
Virtual server for the OID	oid.orcl.com:OIDPort
Virtual server for the OID (SSL)	oid.orcl.com:OIDSSLPort
Virtual server for IM Middle Tier (SSL)	login.orcl.com:VsHTTPSPort

Network Diagram



Configuration Steps

The following steps are needed for configuring the Foundry ServerIron hardware load balancer for SSL terminating https traffic from the client and forwarding clear-text http requests to the Oracle http server deployed in the IM. The flow is from the Browser →*https* → LBR → *http*→OracleIM. Note that the keywords in ***bold italic*** are environment dependant and should be specified accordingly.

1) Enter the load balancer (ServerIron) configuration mode

Connect to the device through console connector or telnet into the device and then issue following commands:

SI> `enable`

SI# `config term`

SI(config)#

All further configuration changes below are made from **config** mode.

2) Define the Real Servers in the LBR.

Configure real servers using following commands. Define real servers for the web traffic and for the LDAP/LDAPS traffic.

```
(config)# server real rsl1 IP_of_login1.orcl.com
          port RsHTTPPort
          port RsHTTPPort group-id startid# endid#
          port RsHTTPPort server-id serverid#_rsl1


(config)# server real rsl2 IP_of_login2.orcl.com
          port RsHTTPPort
          port RsHTTPPort group-id startid# endid#
          port RsHTTPPort server-id serverid#_rsl2


(config)# server real rso1 IP_of_oid1.orcl.com
          port OIDPort
          port OIDSSLPort

(config)# server real rso2 IP_of_oid2.orcl.com
          port OIDPort
          port OIDSSLPort
```

3) Define Virtual Servers

Configure the virtual servers that will be used in the Identity Management install using the following commands. Define the virtual servers for web traffic and for LDAP/LDAPS traffic, and then bind these to the respective real servers.

```
(config)# server virtual vsl IP_of_login.orcl.com
```

```
          port VsHTTPSPort
          bind VsHTTPSPort rsl1 RsHTTPPort rsl2
RsHTTPPort


(config)# server virtual vso IP_of_oid.orcl.com
          port OIDPort
          bind OIDPort rso1 OIDPort rso2 OIDPort
          port OIDSSLPort
          bind OIDSSLPort rso1 OIDSSLPort rso2
OIDSSLPort
```

4) Map the http and ldap/ldaps ports used in this application to well-known ports for these protocols.

```
(config)# server port RsHTTPPort
             tcp
             tcp keepalive protocol http
(config)# server port OIDPort
             tcp
             tcp keepalive protocol ldap
(config)# server port OIDSSLPort
             tcp
```

5) Define scripted layer 7 server and application health checks for SSO/DAS tier.

Define scripted layer 7 health check to consider a given instance of the SSO/DAS tier as healthy only when a response to the http request "GET sso/status" contains the string "OC4J_SECURITY is running." Repeat these steps for each real web server.

```
(config)# http match-list oc4j
          default down
          up simple "OC4J_SECURITY is running."


(config)# server real rsl1 IP_of_login1.orcl.com
          port RsHTTPPort url "GET /sso/status"
          port RsHTTPPort content-match oc4j
(config)# server real rsl2 IP_of_login2.orcl.com
          port RsHTTPPort url "GET /sso/status"
          port RsHTTPPort content-match oc4j
```

6) Define custom health check for the LDAPS protocol and base its health on the health of LDAP. Define as many custom health checks as the real servers.

```
(config)# healthck ldaps1 tcp
          dest-ip IP_of_oid1.orcl.com
          port OIDPort
          protocol ldap
          l7-check
(config)# server real rso1 IP_of_oid1.orcl.com
          port OIDSSLPort healthck ldaps1
```

```
(config)# healthck ldaps2 tcp
          dest-ip IP_of_oid2.orcl.com
          port OIDPort
          protocol ldap
          l7-check
(config)# server real rso1 IP_of_oid2.orcl.com
          port OIDSSLPort healthck ldaps2
```

7) Define SSL traffic profile and associate it with the virtual server

Prior to defining a SSL traffic profile, import a SSL certificate to the ServerIron device. This certificate may be self signed or obtained from a third party vendor. This certificate is presented to the clients when they connect to the virtual IP address. Please refer to Foundry Networks ServerIron SSL Release Notes on the procedure to import the SSL certificate.

```
(config)# ssl profile sslprofile1
          keypair-file rsakeyfile1
          certificate-file certfile1
          cipher-suite all-cipher-suites
(config)# virtual vs1 IP_of_login.orcl.com
          port VsHTTPSPort ssl-terminate sslprofile1
```

8) Configure client connection persistence for HTTP traffic using cookie insertion and cookie match. In this example, the ServerIron will be set to search for a pre-defined cookie name inside each client request message. If the cookie is not present, then cookie "*ServerID=*" is inserted in the server response message going to the client. This cookie string and value are then used for persistence of a client connection to a given real server instance.

```
(config)# csw-rule "cswrule1" header "cookie" search
"ServerID="
```

```
(config)# csw-policy "cswpolicy1"
          match "cswrule1" persist offset 0 length
<length_of_serverid_in_rs1/2_definition>  group-or-
server-id
          default forward <group-id>
          default rewrite insert-cookie

(config)# virtual vs1 IP_of_login.orcl.com
          port VsHTTPSPort
          port VsHTTPSPort cookie-name "ServerID"
          port VsHTTPSPort csw-policy "cswpolicy1"
          port VsHTTPSPort csw
```

9) Enable source address NAT (optional based on deployment and topology design choices)

Source-nat configuration ensures that return traffic flows through the ServerIron device in a one-arm topology design.

```
(config)# server source-nat
(config)# server source-nat-ip IP-address-LBR1 subnet-mask default-gateway port-range 1
(config)# server source-nat-ip IP-address-LBR1 subnet-mask default-gateway port-range 1 for-ssl
```

10) Configure load balancer (ServerIron) redundancy (optional – but highly recommended for maximum high availability implementation)

For Active-Hot Standby ServerIron design, configure load balancer redundancy as follows:

On the primary,

```
(config)# server backup ethernet mod/port ServerIron-MAC-address vlan-id sync-vlan-#
(config)# server backup-preference delay-time-in-minutes
```

The 'server backup-preference' command is required only on the active load balancer.

The configuration of the second LBR in the Active-Hot Standby ServerIron design is similar to that on the primary LBR except for the following:

a. The secondary LBR obviously has a separate host name & management IP address

b. The 'server backup-preference' command is not run on the secondary

c. Use the second half of port range with 'server source-nat-ip' command. For 'server source-nat-ip' command described above, use port range value of 2 instead of 1 on standby load balancer.

d. Ensure that same MAC address is used on both LBRs for 'server backup' command.

ServerIron Load Balancer Configuration (Example)

Appendix B shows the ServerIron configuration file for a distributed IMHA install with the following specific port and IP addresses. The configuration file output shown in Appendix B is the console output from the ***"show run"*** command available in the ServerIron CLI menu.

OIDPort	3060
OIDSSLPort	4030
VsHTTPSPort	443
RsHTTPPort	7785
login1.orcl.com	144.25.145.38

login2.orcl.com	144.25.145.39
login.orcl.com	144.25.145.20
oid1.orcl.com	144.25.145.40
oid2.orcl.com	144.25.145.41
oid.orcl.com	144.25.145.19

Appendix C shows the sample ServerIron configuration for a co-located IMHA install for the same ports and IP addresses listed above. The configuration file output shown in Appendix C is the console output from the ***"show run"*** command available in the ServerIron CLI menu.

OracleAS Cluster (Identity Management) Installation

Once the load balancer has been configured as described above, follow the procedure outlined in the **Oracle Application Server Installation Guide** to install either Distributed or co-located OracleAS Cluster (Identity Management) install. Use oid.orcl.com and login.orcl.com as the OID load balancer virtual server and the OHS load balancer virtual server respectively. A sample staticports.ini file that can be used is shown in the Appendix A.

Validation Steps

- Access https://*login.orcl.com:VsHTTPPort*/oiddas from a browser

  Multiple times and validate by navigating through various pages.

- Access https://*login.orcl.com:VsHTTPPort*/pls/orasso multiple times and validate.

  Multiple times and validate by navigating through various pages.

- From any of the IM tier Oracle Home –
  - $ ldapbind –h oid.orcl.com –p OIDPort

    Bind should be successful multiple times.
  - $ ldapbind –h oid.orcl.com –p OIDSSLPort –U 1

    Bind should be successful multiple times.
- From the Load balancer Console
  - # show server real

    Verify the configuration and the state.
  - # show server virtual

    Verify the configuration and the state.

APPENDIX A – STATICPORTS.INI TEMPLATES

For OracleAS Cluster (Identity Management) install

Template staticports.ini file

```
Oracle Internet Directory port = OIDPort
Oracle Internet Directory (SSL) port = OIDSSLPort
Oracle HTTP Server port = VsHTTPPort
Oracle HTTP Server Listen port = RsHTTPPort
Oracle HTTP Server SSL port = VsHTTPSPort
Oracle HTTP Server Listen (SSL) port = RsHTTPSPort
Oracle HTTP Server Diagnostic port = port_num
Java Object Cache port = port_num
DCM Discovery port = port_num
Oracle Notification Server Request port = port_num
Oracle Notification Server Local port = port_num
Oracle Notification Server Remote port = port_num
Application Server Control port = port_num
Application Server Control RMI port = port_num
Oracle Management Agent port = port_num
Log Loader port = port_num
Oracle Certificate Authority SSL Server Authentication port = port_num
Oracle Certificate Authority SSL Mutual Authentication port = port_num
Ultra Search HTTP port number = port_num
```

For Distributed OracleAS Cluster (Identity Management) install

Template for OID tier - staticports.ini.oid file

```
Oracle Internet Directory port = OIDPort
Oracle Internet Directory (SSL) port = OIDSSLPort
Oracle Notification Server Request port = port_num
Oracle Notification Server Local port = port_num
Oracle Notification Server Remote port = port_num
Application Server Control port = port_num
Application Server Control RMI port = port_num
Oracle Management Agent port = port_num
Log Loader port = port_num
Oracle Certificate Authority SSL Server Authentication port = port_num
Oracle Certificate Authority SSL Mutual Authentication port = port_num
Ultra Search HTTP port number = port_num
```

Template for IM Middle Tier - staticports.ini.sso file

```
Oracle HTTP Server port = VsHTTPPort
Oracle HTTP Server Listen port = RsHTTPPort
Oracle HTTP Server SSL port = VsHTTPSPort
Oracle HTTP Server Listen (SSL) port = RsHTTPSPort
Oracle HTTP Server Diagnostic port = nnnn
Oracle Notification Server Request port = nnnn
Oracle Notification Server Local port = nnnn
Oracle Notification Server Remote port = nnnn
Application Server Control RMI port = nnnn
Application Server Control port = emPort
Oracle Management Agent port = nnnn

Oracle HTTP Server port = VsHTTPPort
Oracle HTTP Server Listen port = RsHTTPPort
Oracle HTTP Server SSL port = VsHTTPSPort
Oracle HTTP Server Listen (SSL) port = RsHTTPSPort
Oracle HTTP Server Diagnostic port = port_num
Java Object Cache port = port_num
DCM Discovery port = port_num
Oracle Notification Server Request port = port_num
```

```
Oracle Notification Server Local port = port_num
Oracle Notification Server Remote port = port_num
Application Server Control port = port_num
Application Server Control RMI port = port_num
Oracle Management Agent port = port_num
```

APPENDIX B – SERVERIRON CONFIGURATION FOR DISTRIBUTED ORACLEAS CLUSTER (IDENTITY MANAGEMENT) INSTALL

```
ServerIron 1 (Active):
! Building configuration...
! Current configuration: 2408 bytes
!
ver 09.4.50b222TD2
!
module 1 bi-0-port-wsm6-management-module
module 2 bi-jc-16-port-gig-copper-module
!
! Define custom Layer-7 health checks to determine LDAPS health on two real servers.
! LDAPS health is based on LDAP health.
! Custom health check for Oracle Application Server 1
healthck ldaps1 tcp
  dest-ip 144.25.145.40
  port 3060
  protocol ldap
  l7-check
!
! Custom health check for Oracle Application Server 2
healthck ldaps2 tcp
  dest-ip 144.25.145.41
  port 3060
  protocol ldap
  l7-check
!
! Define SSL profile: SSL key-file, certificate file and cipher-suites
ssl profile sslprofile1
  keypair-file rsakeyfile1
  certificate-file certfile1
  cipher-suite all-cipher-suites
  session-cache off
!
! Define Sync Link to the other ServerIron for ServerIron redundancy
server backup ethe 2/16 0012.f233.e400 vlan-id 99
!
! Give higher priority to this ServerIron over the other. This ServerIron will assume the role of
! active from the other ServerIron in 5 minutes
server backup-preference 5
!
! Use pre-defined application port 7785 for http process
server port 7785
  connection-log src-nat
  tcp
  tcp keepalive protocol 80
!
! Use pre-defined application port 3060 for LDAP traffic
server port 3060
  connection-log src-nat
  tcp
  tcp keepalive protocol 389
!
! Use pre-defined application port 4030 for LDAPS traffic
server port 4030
  connection-log src-nat
  tcp
!
! The one-arm network setup requires source-nat configuration to ensure that return traffic (from
! real servers to client) flows through ServerIron
! Separate source-nat-ip required to handle both SSL and non-SSL traffic
! Port range 1 designates the first half port range for this ServerIron
server source-nat
server source-nat-ip 144.25.145.56 255.255.252.0 0.0.0.0 port-range 1 for-ssl
server source-nat-ip 144.25.145.55 255.255.252.0 0.0.0.0 port-range 1
!
! Define Layer-7 cookie insert/switching rule to insert cookie in the server response packet and
! then persist on cookie value
```

```
csw-rule "cswrule1" header "cookie" search "ServerID="
csw-policy "cswpolicy1"
  match "cswrule1" persist offset 0 length 4 group-or-server-id
  default forward 1
  default rewrite insert-cookie
!
! Define Real Servers and configure scripted Layer-7 http healtchecks. Also associate server-id
! and group-id with the servers
server real rsl1 144.25.145.38
  port 7785
  port 7785 server-id 2001
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
server real rso1 144.25.145.40
  port 3060
  port 4030
  port 4030 healthck ldaps1
!
server real rsl2 144.25.145.39
  port 7785
  port 7785 server-id 2002
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
server real rso2 144.25.145.41
  port 3060
  port 4030
  port 4030 healthck ldaps2
!
! Configure Virtual IP for incoming SSL traffic. Bind SSL profile with virtual IP and enable
! cookie switching
server virtual vsl 144.25.145.20
  port ssl sticky
  port ssl ssl-terminate sslprofile1
  port ssl cookie-name "ServerID"
  port ssl csw-policy "cswpolicy1"
  port ssl csw
  port ssl keep-alive
  bind ssl rsl1 7785 rsl2 7785
!
! Configure Virtual IP for LDAP and LDAPS traffic
server virtual vso 144.25.145.19
  port 3060
  port 4030
  bind 3060 rso1 3060 rso2 3060
  bind 4030 rso1 4030 rso2 4030
!
vlan 1 name DEFAULT-VLAN by port
!
! Define Sync-VLAN for sync traffic between two ServerIrons
vlan 99 name sync-vlan by port
  untagged ethe 2/16
  no spanning-tree
!
! Configure scripted Layer-7 http healthcheck
http match-list oc4j
  default down
  up simple "OC4J_SECURITY is running."
!
! Define ServerIron hostname, IP address and default gateway
hostname HAFNSI01
ip address 144.25.144.59 255.255.252.0
ip default-gateway 144.25.144.1
!
! DNS definitions
ip dns domain-name us.oracle.com
ip dns server-address 130.35.249.52 130.25.249.41
!
clock timezone us Pacific
sntp server 144.25.144.1
```

```
sntp poll-interval 5
auto-cam-repaint
pram-write-retry
!
end

ServerIron 2 (Hot-Standby):
! Building configuration...
! Current configuration: 2408 bytes
!
ver 09.4.50b222TD2
!
module 1 bi-0-port-wsm6-management-module
module 2 bi-jc-16-port-gig-copper-module
!
! Define custom Layer-7 health checks to determine LDAPS health on two real servers.
! LDAPS health is based on LDAP health.
! Custom health check for Oracle Application Server 1
healthck ldaps1 tcp
  dest-ip 144.25.145.40
  port 3060
  protocol ldap
  l7-check
!
! Custom health check for Oracle Application Server 2
healthck ldaps2 tcp
  dest-ip 144.25.145.41
  port 3060
  protocol ldap
  l7-check
!
! Define SSL profile: SSL key-file, certificate file and cipher-suites
ssl profile sslprofile1
  keypair-file rsakeyfile1
  certificate-file certfile1
  cipher-suite all-cipher-suites
  session-cache off
!
! Define Sync Link to the other ServerIron for ServerIron redundancy
server backup ethe 2/16 0012.f233.e400 vlan-id 99
!
! Use pre-defined application port 7785 for http process
server port 7785
  connection-log src-nat
  tcp
  tcp keepalive protocol 80
!
! Use pre-defined application port 3060 for LDAP traffic
server port 3060
  connection-log src-nat
  tcp
  tcp keepalive protocol 389
!
! Use pre-defined application port 4030 for LDAPS traffic
server port 4030
  connection-log src-nat
  tcp
!
! The one-arm network setup requires source-nat configuration to ensure that return traffic (from
! real servers to client) flows through ServerIron
! Separate source-nat-ip required to handle both SSL and non-SSL traffic
! Port range 2 designates the second half port range for this ServerIron
server source-nat
server source-nat-ip 144.25.145.56 255.255.252.0 0.0.0.0 port-range 2 for-ssl
server source-nat-ip 144.25.145.55 255.255.252.0 0.0.0.0 port-range 2
!
! Define Layer-7 cookie insert/switching rule to insert cookie in the server response packet and
! then persist on cookie value
csw-rule "cswrule1" header "cookie" search "ServerID="
csw-policy "cswpolicy1"
  match "cswrule1" persist offset 0 length 4 group-or-server-id
  default forward 1
  default rewrite insert-cookie
```

```
!
! Define Real Servers and configure scripted Layer-7 http healtchecks. Also associate server-id
! and group-id with the servers
server real rsl1 144.25.145.38
  port 7785
  port 7785 server-id 2001
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
server real rso1 144.25.145.40
  port 3060
  port 4030
  port 4030 healthck ldaps1
!
server real rsl2 144.25.145.39
  port 7785
  port 7785 server-id 2002
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
server real rso2 144.25.145.41
  port 3060
  port 4030
  port 4030 healthck ldaps2
!
! Configure Virtual IP for incoming SSL traffic. Bind SSL profile with virtual IP and enable
! cookie switching
server virtual vsl 144.25.145.20
  port ssl sticky
  port ssl ssl-terminate sslprofile1
  port ssl cookie-name "ServerID"
  port ssl csw-policy "cswpolicy1"
  port ssl csw
  port ssl keep-alive
  bind ssl rsl1 7785 rsl2 7785
!
! Configure Virtual IP for LDAP and LDAPS traffic
server virtual vso 144.25.145.19
  port 3060
  port 4030
  bind 3060 rso1 3060 rso2 3060
  bind 4030 rso1 4030 rso2 4030
!
vlan 1 name DEFAULT-VLAN by port
!
! Define Sync-VLAN for sync traffic between two ServerIrons
vlan 99 name sync-vlan by port
  untagged ethe 2/16
  no spanning-tree
!
! Configure scripted Layer-7 http healthcheck
http match-list oc4j
  default down
  up simple "OC4J_SECURITY is running."
!
! Define ServerIron hostname, IP address and default gateway
hostname HAFNSI02
ip address 144.25.144.60 255.255.252.0
ip default-gateway 144.25.144.1
!
! DNS definitions
ip dns domain-name us.oracle.com
ip dns server-address 130.35.249.52 130.25.249.41
!
clock timezone us Pacific
sntp server 144.25.144.1
sntp poll-interval 5
auto-cam-repaint
pram-write-retry
!
end
```

APPENDIX C – SERVERIRON CONFIGURATION FOR CO-LOCATED ORACLEAS CLUSTER (IDENTITY MANAGEMENT) INSTALL

In this case, oid1.orcl.com = login1.orcl.com and oid2.orcl.com = login2.orcl.com. The virtual server used (oid.orcl.com and login.orcl.com) are different in this example..

```
ServerIron 1 (Active):
! Building configuration...
! Current configuration: 2408 bytes
!
ver 09.4.50b222TD2
!
module 1 bi-0-port-wsm6-management-module
module 2 bi-jc-16-port-gig-copper-module
!
! Define custom Layer-7 health checks to determine LDAPS health on two real servers.
! LDAPS health is based on LDAP health.
! Custom health check for Oracle Application Server 1
healthck ldaps1 tcp
  dest-ip 144.25.145.38
  port 3060
  protocol ldap
  l7-check
!
! Custom health check for Oracle Application Server 2
healthck ldaps2 tcp
  dest-ip 144.25.145.39
  port 3060
  protocol ldap
  l7-check
!
! Define SSL profile: SSL key-file, certificate file and cipher-suites
ssl profile sslprofile1
  keypair-file rsakeyfile1
  certificate-file certfile1
  cipher-suite all-cipher-suites
  session-cache off
!
! Define Sync Link to the other ServerIron for ServerIron redundancy
server backup ethe 2/16 0012.f233.e400 vlan-id 99
!
! Give higher priority to this ServerIron over the other. This ServerIron will assume the role of
! active from the other ServerIron in 5 minutes
server backup-preference 5
!
! Use pre-defined application port 7785 for http process
server port 7785
  connection-log src-nat
  tcp
  tcp keepalive protocol 80
!
! Use pre-defined application port 3060 for LDAP traffic
server port 3060
  connection-log src-nat
  tcp
  tcp keepalive protocol 389
!
! Use pre-defined application port 4030 for LDAPS traffic
server port 4030
  connection-log src-nat
  tcp
!
! The one-arm network setup requires source-nat configuration to ensure that return traffic (from
! real servers to client) flows through ServerIron
! Separate source-nat-ip required to handle both SSL and non-SSL traffic
! Port range 1 designates the first half port range for this ServerIron
server source-nat
server source-nat-ip 144.25.145.56 255.255.252.0 0.0.0.0 port-range 1 for-ssl
```

```
server source-nat-ip 144.25.145.55 255.255.252.0 0.0.0.0 port-range 1
!
! Define Layer-7 cookie insert/switching rule to insert cookie in the server response packet and
! then persist on cookie value
csw-rule "cswrule1" header "cookie" search "ServerID="
csw-policy "cswpolicy1"
  match "cswrule1" persist offset 0 length 4 group-or-server-id
  default forward 1
  default rewrite insert-cookie
!
! Define Real Servers and configure scripted Layer-7 http healtchecks. Also associate server-id
! and group-id with the servers
server real rs1 144.25.145.38
  port 3060
  port 4030
  port 4030 healthck ldaps1
  port 7785
  port 7785 server-id 2001
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
server real rs2 144.25.145.39
  port 3060
  port 4030
  port 4030 healthck ldaps2
  port 7785
  port 7785 server-id 2002
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
! Configure Virtual IP for incoming SSL traffic. Bind SSL profile with virtual IP and enable
! cookie switching
server virtual vsl 144.25.145.20
  port ssl sticky
  port ssl ssl-terminate sslprofile1
  port ssl cookie-name "ServerID"
  port ssl csw-policy "cswpolicy1"
  port ssl csw
  port ssl keep-alive
  bind ssl rs1 7785 rs2 7785
!
! Configure Virtual IP for LDAP and LDAPS traffic
server virtual vso 144.25.145.19
  port 3060
  port 4030
  bind 3060 rs1 3060 rs2 3060
  bind 4030 rs1 4030 rs2 4030
!
vlan 1 name DEFAULT-VLAN by port
!
! Define Sync-VLAN for sync traffic between two ServerIrons
vlan 99 name sync-vlan by port
  untagged ethe 2/16
  no spanning-tree
!
! Configure scripted Layer-7 http healthcheck
http match-list oc4j
  default down
  up simple "OC4J_SECURITY is running."
!
! Define ServerIron hostname, IP address and default gateway
hostname HAFNSI01
ip address 144.25.144.59 255.255.252.0
ip default-gateway 144.25.144.1
!
! DNS definitions
ip dns domain-name us.oracle.com
ip dns server-address 130.35.249.52 130.25.249.41
!
clock timezone us Pacific
sntp server 144.25.144.1
```

```
sntp poll-interval 5
auto-cam-repaint
pram-write-retry
!
end

ServerIron 2 (Hot-Standby):
! Building configuration...
! Current configuration: 2408 bytes
!
ver 09.4.50b222TD2
!
module 1 bi-0-port-wsm6-management-module
module 2 bi-jc-16-port-gig-copper-module
!
! Define custom Layer-7 health checks to determine LDAPS health on two real servers.
! LDAPS health is based on LDAP health.
! Custom health check for Oracle Application Server 1
healthck ldaps1 tcp
  dest-ip 144.25.145.38
  port 3060
  protocol ldap
  l7-check
!
! Custom health check for Oracle Application Server 2
healthck ldaps2 tcp
  dest-ip 144.25.145.39
  port 3060
  protocol ldap
  l7-check
!
! Define SSL profile: SSL key-file, certificate file and cipher-suites
ssl profile sslprofile1
  keypair-file rsakeyfile1
  certificate-file certfile1
  cipher-suite all-cipher-suites
  session-cache off
!
! Define Sync Link to the other ServerIron for ServerIron redundancy
server backup ethe 2/16 0012.f233.e400 vlan-id 99
!
! Use pre-defined application port 7785 for http process
server port 7785
  connection-log src-nat
  tcp
  tcp keepalive protocol 80
!
! Use pre-defined application port 3060 for LDAP traffic
server port 3060
  connection-log src-nat
  tcp
  tcp keepalive protocol 389
!
! Use pre-defined application port 4030 for LDAPS traffic
server port 4030
  connection-log src-nat
  tcp
!
! The one-arm network setup requires source-nat configuration to ensure that return traffic (from
! real servers to client) flows through ServerIron
! Separate source-nat-ip required to handle both SSL and non-SSL traffic
! Port range 2 designates the second half port range for this ServerIron
server source-nat
server source-nat-ip 144.25.145.56 255.255.252.0 0.0.0.0 port-range 2 for-ssl
server source-nat-ip 144.25.145.55 255.255.252.0 0.0.0.0 port-range 2
!
! Define Layer-7 cookie insert/switching rule to insert cookie in the server response packet and
! then persist on cookie value
csw-rule "cswrule1" header "cookie" search "ServerID="
csw-policy "cswpolicy1"
  match "cswrule1" persist offset 0 length 4 group-or-server-id
  default forward 1
  default rewrite insert-cookie
```

```
!
! Define Real Servers and configure scripted Layer-7 http healtchecks. Also associate server-id
! and group-id with the servers
server real rs1 144.25.145.38
  port 3060
  port 4030
  port 4030 healthck ldaps1
  port 7785
  port 7785 server-id 2001
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
server real rs2 144.25.145.39
  port 3060
  port 4030
  port 4030 healthck ldaps2
  port 7785
  port 7785 server-id 2002
  port 7785 group-id  1 1
  port 7785 url "GET /sso/status"
  port 7785 content-match oc4j
!
! Configure Virtual IP for incoming SSL traffic. Bind SSL profile with virtual IP and enable
! cookie switching
server virtual vsl 144.25.145.20
  port ssl sticky
  port ssl ssl-terminate sslprofile1
  port ssl cookie-name "ServerID"
  port ssl csw-policy "cswpolicy1"
  port ssl csw
  port ssl keep-alive
  bind ssl login1 7785 login2 7785
!
! Configure Virtual IP for LDAP and LDAPS traffic
server virtual vso 144.25.145.19
  port 3060
  port 4030
  bind 3060 rs1 3060 rs2 3060
  bind 4030 rs1 4030 rs2 4030
!
vlan 1 name DEFAULT-VLAN by port
!
! Define Sync-VLAN for sync traffic between two ServerIrons
vlan 99 name sync-vlan by port
  untagged ethe 2/16
  no spanning-tree
!
! Configure scripted Layer-7 http healthcheck
http match-list oc4j
  default down
  up simple "OC4J_SECURITY is running."
!
! Define ServerIron hostname, IP address and default gateway
hostname HAFNSI02
ip address 144.25.144.60 255.255.252.0
ip default-gateway 144.25.144.1
!
! DNS definitions
ip dns domain-name us.oracle.com
ip dns server-address 130.35.249.52 130.25.249.41
!
clock timezone us Pacific
sntp server 144.25.144.1
sntp poll-interval 5
auto-cam-repaint
pram-write-retry
!
end
```

Configuring Highly Available OracleAS infrastructure with Foundry Networks ServerIron Hardware Load Balancer
February, 2006
Author: Pradeep S. Bhat , HA Systems Group, Oracle Corporation; Deepak Kothari, Product Marketing Engineer, Foundry Networks, Inc.

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

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

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2001 Oracle Corporation
All rights reserved.

Foundry Networks, Inc.
World Headquarters
4980 Great America Parkway,
Santa Clara, CA 95054
U.S.A.

Worldwide Inquiries:
Phone: + 1.408-207-1700
Fax: + 1.408-586-1900
Email: info@foundrynet.com
Web: http://www.foundrynet.com

Foundry Networks, BigIron, EdgeIron, FastIron, IronPoint, IronView IronWare, JetCore, NetIron, ServerIron, SecureIron, Terathon, TurboIron, and the “Iron” family of marks are trademarks or registered trademarks of Foundry Networks, Inc. in United States and other countries. All other trademarks are the properties of their respective owners.

Although Foundry has attempted to provide accurate information in these materials, Foundry assumes no legal responsibility for the accuracy or completeness of the information. More specific information is available on request from Foundry. Please note that Foundry's product information does not constitute or contain any guarantee, warranty or legally binding representation, unless expressly identified as such in a duly signed writing.

Copyright © 2006 Foundry Networks, Inc. All rights reserved.