Oracle® Communications Service Broker

Netra 6000 High Availability Manager Administrator's Guide

Release 6.0

E26770-01

March 2012

ORACLE®

Oracle Communications Service Broker Netra 6000 High Availability Manager Administrator's Guide, Release 6.0

E26770-01

Contents

3 About System Administration

4 Connecting HA Manager to the Network

5 Managing and Monitoring Hardware and Processes

6   Managing Statistics

7   Hardware Management and Monitoring Using MBeans

8   Logging

9   Backing Up Files

10   Managing Alarms

11   Upgrading Service Broker Netra 6000 High Availability Manager

12   Replacing Worker Blades

A   Component List

B   Netra HA Manager MBean Hardware Reference

Preface

Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) is a software module providing management of a complete Service Broker deployment, including hardware and software components of a deployment.

The HA Manager supports integration with Sun Netra 6000 and Oracle Enterprise Linux.

This document introduces the following:

- Concepts of Service Broker deployments applied to a combined hardware and software system

- Minimum actions required to connect an HA Manager deployment to your network and get it started

- Management and monitoring of hardware and software components in an HA Manager deployment

Before reading this documentation, you should read the Oracle Communications Service Broker 6.0 documentation for a good understanding of Service Broker deployments.

Audience

This document is intended for system and network administrators who will deploy and manage HA Manager. It can also be used by those who require a general understanding of Service Broker implementation on a combined hardware and software system.

This document is based on the assumption that you are familiar with Service Broker concepts and have proven competence with the following areas:

- Oracle Enterprise Linux

- Sun Netra X6270 M2 Server Module

- Sun Netra 6000 Modular System

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents:

Software Documentation

- Oracle Communications Service Broker documentation set:

    - *Oracle Communications Service Broker Concepts Guide*

    - *Oracle Communications Service Broker System Administrator's Guide*

    - *Oracle Communications Service Broker Processing Domain Configuration Guide*

    - *Oracle Communications Service Broker Signaling Domain Configuration Guide*

    - *Oracle Communications Service Broker Release Notes*

- Oracle JRockit JDK R28.1.4 documentation

    The Oracle JRockit JDK R28 documentation set is available at:

    http://www.oracle.com/technetwork/middleware/jrockit/documentation/index.html

- Oracle Enterprise Linux Release 5.0 documentation

Hardware Documentation

- The Sun Netra 6000 Modular System documentation set

- The Sun Netra X6270 M2 Server Modules documentation set

- The Sun Blade 6000 Ethernet Switched NEM 24p 10GbE documentation set

- The Sun Network 10GbE Switch 72p documentation set

- The Oracle Integrated Lights Out Manager (ILOM) 3.0 documentation set

You can locate these documents through the Oracle Technology Network Web site at:

http://www.oracle.com/technetwork/indexes/documentation/index.html

Downloading Oracle Communications Documentation

Oracle Communications Service Broker documentation is available from the Oracle software delivery Web site:

http://edelivery.oracle.com/

Additional Oracle Communications documentation is available from Oracle Technology Network:

http://www.oracle.com/technetwork/index.html

1

Overview of Service Broker Netra 6000 High Availability Manager

This chapter provides an overview of Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager).

Before you read this chapter, you should be familiar with Oracle Communications Service Broker concepts and architecture. See *Oracle Communications Service Broker Concepts Guide*.

About the Service Broker Netra 6000 High Availability Manager

Service Broker offers service interaction and mediation capabilities, enabling you to control and orchestrate multiple services in real time, across diverse network types, covering legacy SS7 networks, SIP networks, and Diameter networks.

Service Broker is a software product, normally deployed on multiple hardware machines.

HA Manager is a software module that manages a complete Service Broker deployment including the hardware, operating system software, and Service Broker software. The HA Manager consists of the Service Broker software and an integrated software management module that controls the hardware and software processes of a Service Broker deployment.

HA Manager:

- Simplifies and automates the setting up of a Service Broker deployment (hardware, operating system, Service Broker software, and so on)
- Simplifies and automates upgrading of the Service Broker software components
- Automates provisioning new hardware, enabling you to dynamically extend traffic capacity and dynamically replace failed units
- Provides redundancy of the integrated software and hardware components at all levels
- Provides a redundant interconnection for all Service Broker network interfaces
- Provides integrated Operations and Management for all software and hardware components

About the Service Broker Features Supported in This Release

Release 6.0 of the Netra 6000 High Availability Manager supports the Service Broker Service Controller package. It does not currently support the other Service Broker

packages, such as the Policy Controller, Online Mediation Controller, Virtual Private Network or Social Voice Comminatory services.

System Architecture

HA Manager supports one or more Sun Netra 6000 chassis running the HA Manager software. Within each chassis, there are up to ten Sun Netra X6270 blades.

Bootstrap Blades and Worker Blades

Depending on the software that it runs, each blade supports one of the following roles:

- Bootstrap Blade

  Bootstrap Blades run the following system-level services that Worker Blades depend on:

  – Administration Console

  – Disk storage

  – Boot images

  – Logging Server

  – System facilities, such as Dynamic Host Configuration Protocol (DHCP) server and Network Time Protocol (NTP) server

  – Persistent state

  Bootstrap Blades do not process communications traffic and they carry a relatively low load.

  Bootstrap Blades are not required to be online and functional for Worker Blades to operate normally. Bootstrap Blades need to be active only when a Worker Blade boots or a Worker Blade process restarts. However, services provided by Bootstrap Blades are critical for recovering from failures.

- Worker Blade

  Worker Blades run the Service Broker Signaling Server and Processing Server processes. Worker Blades do not have disk storage or any kind of persistent storage. They rely on Bootstrap Blades for startup, after which they run independently. A Worker Blade receives its identity and instance-specific profile based on the chassis slot in which it is running. See "Worker Blade Profiles" for more information.

Figure 1–1 shows the key components of an HA Manager deployment. It shows one chassis with Bootstrap Blades and Worker Blades and the system-level functions running on the Bootstrap Blades.

Figure 1–1   Key Components of an HA Manager Deployment
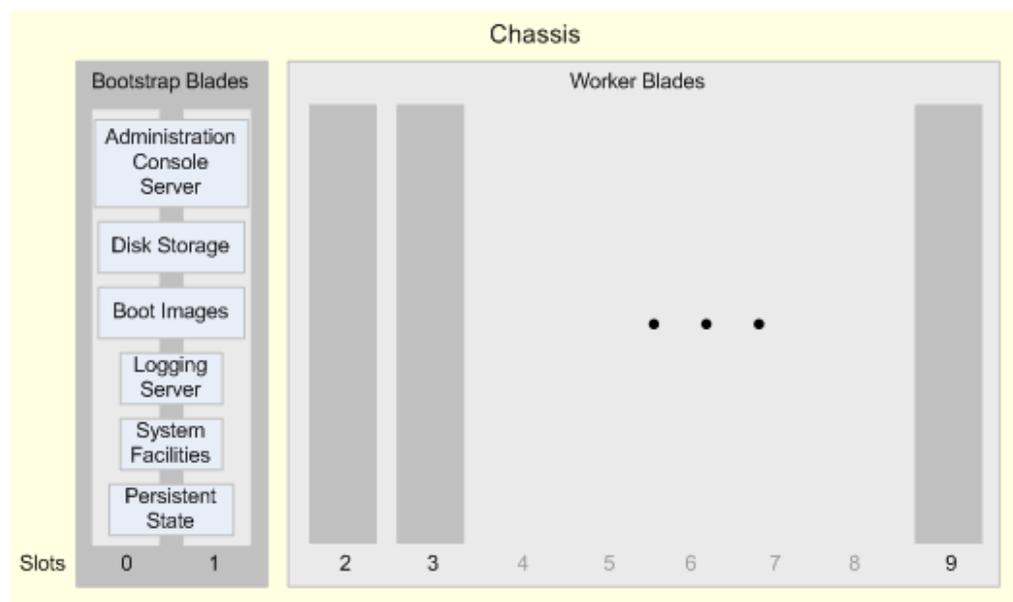
For high availability, an HA Manager deployment includes a pair of Bootstrap Blades, which is sufficient to provide services to Worker Blades on one or more chassis. A single, full chassis consists of two Bootstrap Blades and eight Worker Blades. A minimal deployment consists of two Bootstrap Blades and two Worker Blades - you can add more Worker Blades as required.

Primary and Secondary Bootstrap Blades

Bootstrap Blades run in a primary and secondary configuration; that is, only one Bootstrap Blade, the primary blade, actively provides services at one time. The other blade, the secondary blade, is synchronized and is ready to take over if the primary blade fails or needs to be replaced. The primary and secondary blades share a virtual IP address, which makes the primary-to-secondary transition transparent to processes that use the services running on Bootstrap Blades.

You manage service availability and failover between primary and secondary blades using the Red Hat Cluster Suite, which is part of Oracle Enterprise Linux.

Signaling Servers and Processing Servers on Worker Blades

Worker Blades run Service Broker Signaling Servers and Processing Servers. At the minimum, an HA Manager deployment can include two instances of Signaling Servers and multiple instances of Processing Servers, based on your requirements. A deployment requires fewer instances of Signaling Servers than Processing Servers. Each instance of a Signaling Server runs on a different Worker Blade. Multiple instances of Processing Servers can run on the same Worker Blade. A typical deployment includes a maximum of three Processing Servers per Worker Blade. Figure 1–2 shows an example of an HA Manager minimum deployment.

Figure 1–2   Example of a Minimum HA Manager Deployment
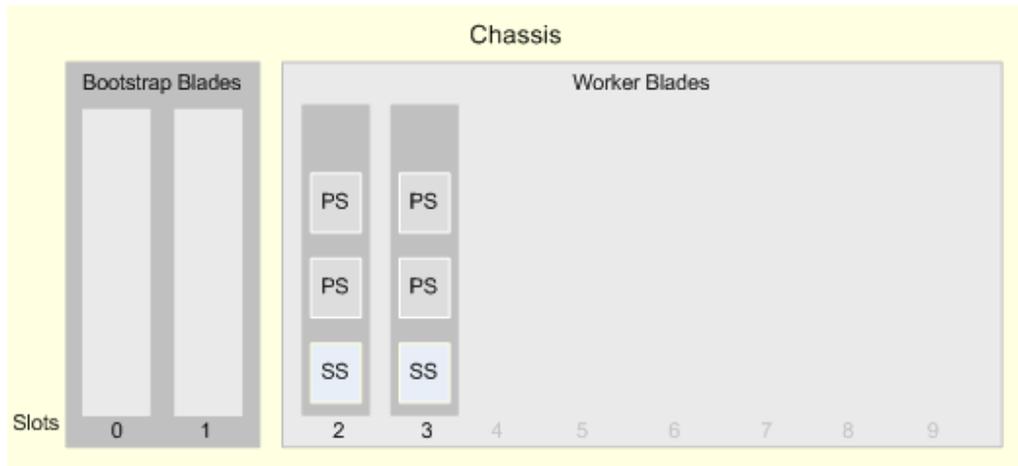
Worker Blades are installed in slots according to the specific processes configured for each slot. Figure 1–3 shows an example of an HA Manager medium deployment.

Figure 1–3   Example of a Medium HA Manager Deployment
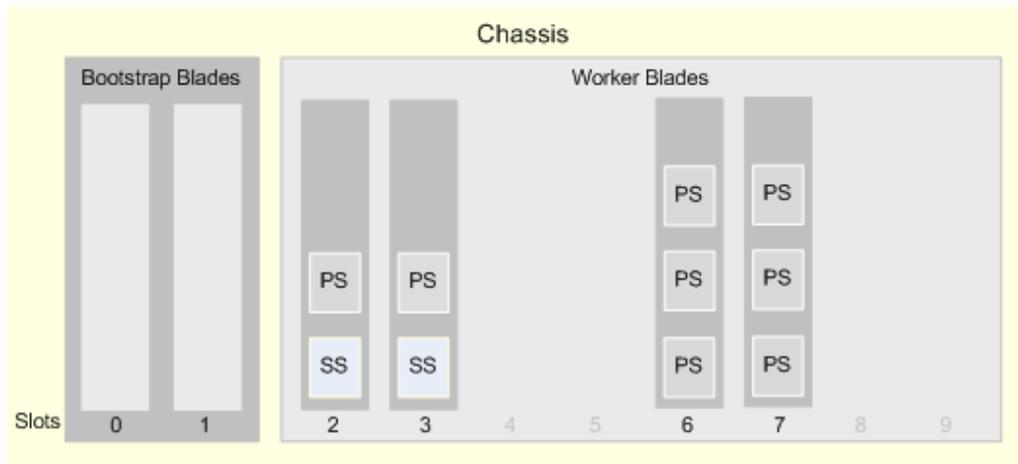
When adding Worker Blades, use a general ratio of one Signaling Server instance to four Processing Server instances. Figure 1–4 shows an HA Manager large deployment that complies with this rule.

Figure 1–4   Example of a Large HA Manager Deployment
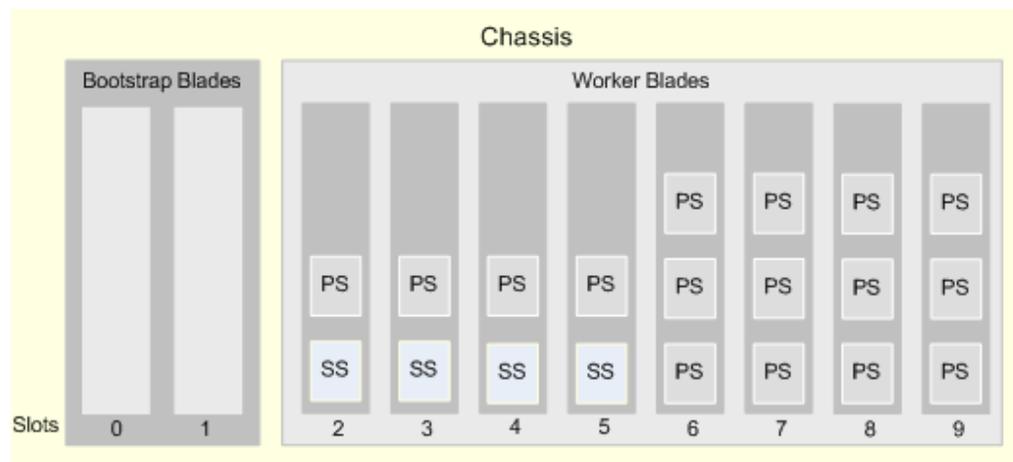
Worker Blade Profiles

A Worker Blade has one of two profiles:

- Processing only

  The processing-only profile means that the Worker Blade runs only Processing Servers. When this profile is assigned, a Worker Blade runs three Processing Server instances.

- Signaling and processing

  The signaling and processing profile means that the Worker Blade runs both Signaling Servers and Processing Servers. When this profile is assigned, a Worker Blade runs one instance of a Signaling Server and two instances of Processing Servers.

Profile assignment is static and depends on the chassis slot into which a Worker Blade is inserted. The signaling-processing profile is assigned to Worker Blades inserted into slots 2 through 5. The processing-only profile is assigned to Worker Blades inserted into slots 6 through 9. If you replace a Worker Blade, the new Worker Blade inherits the same profile, based on the chassis slot.

Each profile of the two different profiles is captured in a Pre-Execution Environment image. The images are stored on the Bootstrap Blades. See "Boot Images" for more information.

Process Instance Identity

Each process running on a blade has a logical identifier within the blade. This identifier is called a Process Instance Identity (PII). A PII is derived from the blade's IP address and the process's fixed order relative to other processes running on the blade.

See "Connecting HA Manager to the Network" for more information on blades' IP addresses.

A PII remains consistent even when the blade or process is restarted, as opposed to operating system processor identifiers (PIDs), which change between processes and blade restarts.

Each Signaling Server instance and Processing Server instance has a PII. PIIs are used internally to reference Signaling Server and Processing Server instances. For example, PIIs are exposed by logs to identify the server that generated a log.

Signaling Domain and Processing Domain

From a management perspective, an HA Manager deployment is a standard Service Broker deployment that includes two domains:

- Signaling Domain: Manages all Signaling Server instances

- Processing Domain: Manages all Processing Server instances

You manage each domain using a different instance of the Administration Console. See "Administration Console" for more information.

Domain Images

Domain software and configuration are bundled together into domain images. A domain image is a group of JAR files and deployment packages containing the software binaries and associated configuration.

See "Upgrading Service Broker Netra 6000 High Availability Manager" for information about deployment packages.

There are two domain images: one for the Signaling Domain and another for the Processing Domain. Domain images are stored on Bootstrap Blades. When a Signaling Server or a Processing Server starts up, it pulls the binaries and related configuration from the corresponding DI.

Service Broker upgrades are upgrades of domain images. You can upgrade domain images using the Administration Console. See "Upgrading Service Broker Netra 6000 High Availability Manager" for more information.

Bootstrap Services

Bootstrap Blades provide the following system-level services:

- Administration Console

- Disk Storage

- Boot Images

- Logging Server

- System Facilities

- State Persistency

Administration Console

HA Manager extends Service Broker Administration Console capabilities to provide integrated management of a deployment's hardware components.

An HA Manager deployment includes three instances of the Administration Console, each enabling different administration tasks as follows:

- System Administration Console

  Use the System Administration Console to get an overall system view of software and hardware components, including blades, processes, alarms, logs and system-level configuration. See "Configuring Network Connectivity" for more information.

- Signaling Servers Administration Console

Use the Signaling Servers Administration Console to manage the Service Broker Signaling Domain. You can configure and upgrade the SIP SSU, Diameter SSU, and SS7 SSU components. See "Configuring Signaling Traffic" for more information.

- Processing Servers Administration Console

Use the Processing Servers Administration Console to manage the Service Broker Processing Domain. You can configure and upgrade IM and SM components. See "Configuring Processing Traffic" for more information.

You access each Administration Console instance through a web browser, using different port numbers. The default ports are 9000 (Processing Servers), 9001 (Signaling Servers), and 9002 (System). You can navigate between the three Administration Console instances from within the Administration Console GUI.

See "About System Administration" for more information about using the Administration Console.

Disk Storage

In an HA Manager deployment, each Bootstrap Blade includes two onboard disks. In total, a deployment requires four disks with 300 GB of space on each disk. However, the effective storage capacity of a system is 300 GB because the four disks are used for mirroring and redundancy.

Within each Bootstrap Blade, the pair of disks is arranged in a software Redundant Array of Independent Disks (RAID), provided by Oracle Enterprise Linux. In addition, disk data is replicated across the primary and secondary Bootstrap Blades. The HA Manager is configured to work with DRBD to accomplish this.

Each disk has to consist of two partitions:

- A local boot/swap/var partition for the Bootstrap Blade itself
- A service partition containing:
  - DHCP server
  - Pre-execution Environment server
  - NTP server
  - Domain Images
  - Logging Server and logs

Boot Images

To boot, Worker Blades use Pre-Execution Environment (PXE) images stored on the Bootstrap Blades.

A PXE image contains the operating system, the external Management Agent, and configuration scripts. There are two PXE images, one for each Worker Blade profile. See "Worker Blade Profiles" for more information.

Logging Server

The Logging Server runs on the Bootstrap Blades and collects logs generated by the Signaling Servers and Processing Servers. Logs are stored on the bootstrap disk storage and can be viewed in the Administration Console's **Log** tab.

Each log contains the PII of the server that generated the log. See "Process Instance Identity" for more information. In the file system, logs for each server are stored in a different directory.

Logging is based on the Apache log4j logging framework. Therefore, logs layout is configured using standard log4j configuration files.

See "Logging" for more information about the Logging Server.

System Facilities

The following are additional standard facilities that must be available on the Bootstrap Blades:

- NTP Server
- DHCP Server
- Network File System (NFS)
- PXE Server

State Persistency

State persistency protects the HA Manager from loss of data if a Processing Servers fails or restarted, if a blade fails or restarted, or if you replace a blade.

State persistency is stored on the Bootstrap Blades.

Hardware and Software Components

See "Component List" for information about the hardware and software components included in an HA Manager deployment.

Network Connectivity

Blades within a chassis communicate using two rack-mounted Sun Blade 6000 Ethernet Switched NEM 24p 10 GbE switches, which are included in every chassis.

Each switch has a single port connection to every blade in the chassis. Accordingly, each blade has one Network Interface Card (NIC) connected to every switch, providing full connection redundancy. However, an HA Manager deployment can function fully with only one operational switch.

Traffic between blades within a chassis, and between blades and network elements outside the chassis, are of the following types:

- SIP and Diameter: SIP and Diameter traffic running between Signaling Servers and network elements outside the chassis
- SIGTRAN: IP-based SS7 traffic running between Signaling Servers and SS7 network elements outside the chassis
- OSS OAM: Management connection associated with Operational Support Systems (OSS) and Operations, Administration and Maintenance (OA&M) activities, running between Bootstrap Blades and Worker Blades, such as JMX and log aggregation.
- SYS ADMIN: Operating System root-level administration connection required for certain system activities such as booting and sending DHCP traffic.

■ Internal: Internal communication between Signaling Servers and Processing Servers.

Inside a chassis, HA Manager uses different Virtual Local Area Networks (VLANs) for each type of traffic. The use of VLANs lets you enforce a different bandwidth for each type of traffic.

Table 1–1 shows the VLANs used by each type of blade inside a chassis.

Table 1–1   VLANs Used by Each Blade

Blade Type	SIP& Diameter	SIGTRAN	OSS OAM	SYS ADMIN	Internal
Bootstrap	No	No	Yes	Yes	No
Workers, running Processing Servers	No	No	Yes	Yes	Yes
Worker, running Signaling Servers and Processing Servers	Yes	Yes	Yes	Yes	Yes

To connect a chassis to an external network or to another chassis, Oracle recommends that you use an external switch, such as the Sun Network 10 GbE Switch 72p Top of Rack switch.

See "Connecting HA Manager to the Network" for more information about how to configure network connectivity.

Process and Hardware Management

You can manage blades and the following processes that run on them:

■ Signaling Servers

■ Processing Servers

■ SS7 stack processes

■ Administration Console instances

■ Logging Server

Process and hardware management is handled internally by two system components:

■ external Management Agent (eMA)

An eMA runs on Worker Blades and Bootstrap Blades. An eMA manages the blade and the processes running on that blade. The eMA can start, stop, and terminate individual processes. It also controls the life cycle of Signaling Servers and Processing Servers. The eMA process is automatically started and stopped as part of the operating system boot and shutdown process.

■ external Management Controller (eMC)

An eMC runs within the same process as the Web Administration Console, on Bootstrap Blades, and controls eMA instances. An eMC manages individual processes running on Worker Blades and Bootstrap Blades.

The eMA and eMC communicate using JMX, over the OSS OAM VLAN. See "Network Connectivity" for more information.

You can manage processes and hardware using the Administration Console GUI. The Administration Console displays the state of the blades and the processes running on each blade. See "Managing and Monitoring Hardware and Processes" for information about managing processes and hardware using the Administration Console.

Security

HA Manager imposes the security model described in "Configuring Security" in *Oracle Communications Service Broker Administrator's Guide*.

You can modify the security of the following external system interfaces, if required:

- Operating system users
- Integrated Lights Out Manager (ILOM)
- Web Administration Console

See "Getting Started" for information about changing passwords for these interfaces.

2

Getting Started

This chapter provides an overview of the tasks required to implement the Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager). It also provides information on the physical installation of the system and describes initial tasks, such as starting the system and resetting the Administration Console password.

About Implementing HA Manager

The HA Manager is a complete, self-contained system with integrated internal communication, redundancy, file sharing, and other capabilities.

The HA Manager deployment configuration includes predefined domains and servers that require no manual modification. After the Sun Netra 6000 hardware is physically installed, getting started requires relatively few steps.

An overview of the steps for getting started are:

1. Verify the physical installation and setup. See "HA Manager System Requirements" for more information.

2. Start the system. See "Starting the System" for more information.

3. Reset the Administration Console password. See "Resetting Passwords" for more information.

4. Log in to the Administration Console Web interface. See "Logging In to the Administration Console" for more information.

5. Verify and configure the global network settings. See "Connecting HA Manager to the Network" for more information.

6. Configure the blade-specific network settings. See "Connecting HA Manager to the Network" for more information.

After completing these initial steps, you use the Signaling Servers Administration Console and the Processing Servers Administration Console to configure the traffic brokering properties and the behavior of the system.

HA Manager System Requirements

An HA Manager deployment is a set of four or more Sun Netra X6270 M2 Server Module blades installed on Oracle's Sun Netra 6000 Modular System chassis. The minimum system consists of four server modules, in which two operate as Bootstrap Blades and two as Worker Blades.

While detailed information about the installation and setup of the hardware and system components is beyond the scope of this document, this section provides an overview of the hardware and system setup requirements for an HA Manager deployment.

The hardware and system prerequisites for an HA Manager deployment are:

- A Sun Netra 6000 chassis installed in a rack and Sun Netra X6270 M2 Server Modules installed in the chassis.

- Ethernet cables connecting two rack-mounted Sun Blade 6000 Ethernet Switched NEM 24p 10 GbE switches to the external network. For redundancy, an additional cable connecting the two switches together.

  Note that connectivity between the service modules within the system is automatically enabled. Each blade has a preconfigured dual port, 10 Gigabit Ethernet card that is connected to the rack-mounted switches through internal connectors on the midplane of the Sun Netra 6000 chassis.

- The switches are preconfigured to enable connectivity within HA Manager. Connectivity from the switches to the local network needs to be configured on a site-specific basis.

  For information about configuring the switch for external connectivity, see the Sun Blade 6000 Ethernet Switched NEM 24p 10 GbE switches documentation.

- The Chassis Monitoring Module (CMM) Integrated Lights Out Manager (ILOM) Web interface has to be accessible either on the network or by direct connection to the Ethernet NET MGT port or serial SER MGT port on the chassis. You need to set default username and password.

  For complete information about the modular system, see the Sun Netra 6000 documentation, which is available on the Oracle Technology Network Web site.

See also the *Sun Netra X6270 M2 Server Module* documentation.

Starting the System

Apply power to the system by connecting four power cords from the AC power connectors to your power outlets. Powering on the system starts up the chassis and all blades installed in the chassis.

You can power on or power off individual blades in the chassis using the CMM ILOM Web interface or using the Service Broker System Administration Console. The power-related operations in the System Administration Console are provided as a convenience and do not differ from the equivalent operations in the CMM ILOM Web interface.

Accessing the Bootstrap Blade Operating System Environment

Certain HA Manager administration tasks, such as backing up configuration files or changing user account passwords, require access to the operating system command-line interface on the Bootstrap Blades. Such tasks typically need to be performed on all Bootstrap Blades in your system.

You can access the Bootstrap Blade operating system command-line environment using one of the following methods explained in the following sections:

- CMM ILOM Web interface
- Linux Secure Shell SSH

Accessing the Bootstrap Blade Using the CMM ILOM Web Interface

To access the Bootstrap Blade through CMM ILOM:

1. In the CMM ILOM Web interface, select the Bootstrap Blade from the chassis tree.

2. In the **Remote Control** tab, click the **Launch Remote** console button.

For complete information on using the CMM ILOM Web interface, see the Oracle Integrated Lights Out Manager (ILOM) 3.0 documentation.

Accessing the Bootstrap Blade Using SSH

The active Bootstrap Blade is accessible by an SSH connection to the OSS OAM IP address configured for the system. Note that the OSS OAM IP address is a virtual IP address that is adopted by whichever Bootstrap Blade is serving as the active (primary) Bootstrap. Therefore, extra steps are needed to access the command-line interface on the standby Bootstrap Blade.

To access the standby Bootstrap Blade:

1. Connect by SSH to the active Bootstrap Blade.

2. From the active Bootstrap Blade, make an SSH connection to the standby Bootstrap Blade, using the static internal IP address assigned to each Bootstrap Blade, for example:

    ■ Bootstrap Blade 1 (slot 0): 192.168.1.1

    ■ Bootstrap Blade 2 (slot 1): 192.168.1.2

Resetting Passwords

An HA Manager deployment includes various built-in user accounts. The passwords for these accounts are set individually for each system. When first using the system, you need to change the initial passwords.

■ **root** user on the operating system

■ **ocsb** user on the operating system

■ **admin** user for the System Administration Console

■ **admin** user for the Signaling Servers Administration Console

■ **admin** user for the Processing Servers Administration Console

In addition, the CMM ILOM interface and service module SP ILOM interfaces have their own user accounts. For information on modifying the passwords for ILOM accounts, see the Sun Netra 6000 documentation.

Modifying Operating System Account Passwords

You modify the passwords for the operating system users, **root** and **ocsb**, from the operating system command-line environment on the Bootstrap Blades.

The procedure for changing passwords for user accounts differs between Bootstrap Blades and Worker Blades.

For Bootstrap Blades, you change passwords using standard Linux user account modification mechanisms.

Because the Worker Blade operating systems are dynamically loaded at startup through the pre-boot execution environment (PXE) boot process, changing user

accounts on Worker Blades requires modifying the boot image. Changing the Bootstrap Blade image affects all subsequently loaded Worker Blades.

Changing Bootstrap Blade User Passwords

Follow these steps to change user account passwords on both Bootstrap Blades:

1. As the root user, access the operating system command-line interface for the active Bootstrap Blade.

   See "Accessing the Bootstrap Blade Operating System Environment" for more information.

2. Change the password for the **root** and **ocsb** user accounts using the `passwd` command or other standard Linux password change mechanism.

   For example, to change the password for the **ocsb** user with the `passwd` command, enter the command as follows:

   ```
   passwd ocsb
   Changing password for user ocsb.
   New UNIX password:
   Retype new UNIX password:
   passwd: all authentication tokens updated successfully.
   ```

The change takes effect immediately.

Changing Worker Blade User Passwords

Like Bootstrap Blades, the Worker Blades have two operating system user accounts, **root** and **ocsb**. However, the Worker Blade operating system image is loaded on to the Worker Blade modules dynamically through the PXE boot process. Therefore, to change user passwords on the Worker Blades, you need to modify the image that is loaded by the Bootstrap Blades.

You need to perform the following procedure only once. It modifies the user accounts for all subsequently loaded Worker Blades.

To modify the user account passwords for Worker Blades:

1. As the **root** user, access the operating system command-line interface for the active Bootstrap Blade.

   See "Accessing the Bootstrap Blade Operating System Environment" for more information.

2. At the command line, navigate to the directory that contains the Worker Blade disk image:

   ```
   cd var/ocsb/workerblade-ramdisk/
   ```

3. Start the script that compiles and builds the Worker Blade image used for the PXE boot process as follows:

   ```
   ./build.sh
   ```

   This script builds a new image for the Bootstrap Blade based on the parameters you supply to the script.

4. When prompted, enter a new password for the user accounts.

This change affects only the Worker Blades that are started subsequent to the change. That is, Worker Blades that were running while performing this procedure maintain their former user passwords until the blades are restarted.

Modifying Administration Console Passwords

You need to modify the user name and password for HTTP basic authentication credentials to access Administration Console interfaces.

To modify the user name and password:

1. As the **root** user, access the operating system command-line interface for the active Bootstrap Blade.

   See "Accessing the Bootstrap Blade Operating System Environment" for more information.

2. At the command line, navigate to the directory that contains the password change script:

   ```
   cd /var/ocsb/ocsb/
   ```

3. At the command line, run the Administration Console user modification script:

   ```
   ./set_web_user_and_password.sh
   ```

4. At the following prompt, enter the new user name for the Administration Console user, or enter **admin** to keep the existing username:

   ```
   Please enter the name of the console user
   ```

5. At the following prompt, enter the new password for the Administration Console user:

   ```
   Please enter the password of the console user
   ```

   Note that the values you type do not appear on screen.

The changes take effect immediately.

Logging In to the Administration Console

You configure and administer HA Manager using its Web interface, which includes these three Web applications:

- System Administration Console
- Signaling Servers Administration Console
- Processing Servers Administration Console

Each console is served on a different TCP port number. When you open a console, you initiate a new HTTPS session.You can open each console directly from the System Administration Console interface, which typically serves as the entry point for HA Manager administration.

To log in to the System Administration Console:

1. Open your Web browser.

   The console works with the following browsers:

   - Internet Explorer version 8.0 or later from Microsoft Corporation.
   - Firefox version 3.0 or later from Mozilla Corporation.
   - Safari version 3.0 or later from Apple, Inc.
   - Chrome version 3.0 or later from Google Inc.
   - Opera version 9.0 or later from Opera Software ASA.

2. Enter the URL:

```
https://IP_address:Port_number/console
```

Where:

IP_address is the IP address assigned to the OAS OAM interface for your system. This address is assigned at system initialization time.

Port_number is the listening port number of the console. The default listening ports are:

- **9002** - System Administration Console
- **9001** - Signaling Servers Administration Console
- **9000** - Processing Servers Administration Console

3. If required, add a security exception to continue.

If this is the first time you have logged in to the Administration Console and you have not replaced the default, self-signed security certificate, a connection warning appears and you must add a security exception to continue. The procedure for this differs according to the Web browser you use. If you have installed a certificate that was signed by one of your browser's trusted certificate authorities, no warning appears.

For information on replacing the default security certificate for the Administration Console, see the discussion on configuring security in *Oracle Communications Service Broker System Administrator's Guide*.

4. When prompted, enter the user name and password.

The default username is **admin**. The default password is set at system initialization.

After logging in, the **Overview** tab of the **System** pane appears. See "About System Administration" for overview information on navigating and using the Administration Console.

3

About System Administration

This chapter provides an overview of the Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) configuration and administration tasks.

About Service Broker HA Manager System Administration

You configure and administer the HA Manager using the browser-based GUI consoles. This chapter provides an overview of the configuration and administration tasks you perform by using each console.

HA Manager includes the following consoles:

- System Administration Console: Use to monitor HA Manager performance and configure network connectivity for the HA Manager server modules.

  The following sections describe the tasks you perform with this console

  - Monitoring Your Deployment with the System Administration Console

  - Administering Processes, Servers, and Hardware Components

  - Managing and Configuring Hardware Components

  - Configuring Network Connectivity

- Signaling Servers Administration Console: Use to configure HA Manager Signaling Servers and upgrade them as needed. See "Configuring Signaling Traffic" for information about the tasks you perform with this console.

- Processing Servers Administration Console: Use to configure HA Manager Processing Servers and upgrade Processing Servers as needed. See "Configuring Processing Traffic" for information about the tasks you perform with this console.

You use the System Administration Console to perform HA Manager *external* configuration tasks, and the Signaling Servers Administration Console and Processing Servers Administration Console to perform *internal* configuration tasks. "Getting Started" and "Connecting HA Manager to the Network" describe the default HA Manager configuration. Those chapters also list the configuration steps required to get your HA Manager deployment up and running, and to update it if necessary.

Figure 3–1 shows the **Overview** subtab page that displays CPU performance graphs for blades in Slots 3, 4, 5 and 6. These blades are all currently at 0% of capacity, but the graphs show significant recent CPU usage.

Figure 3–1   System Administration Console
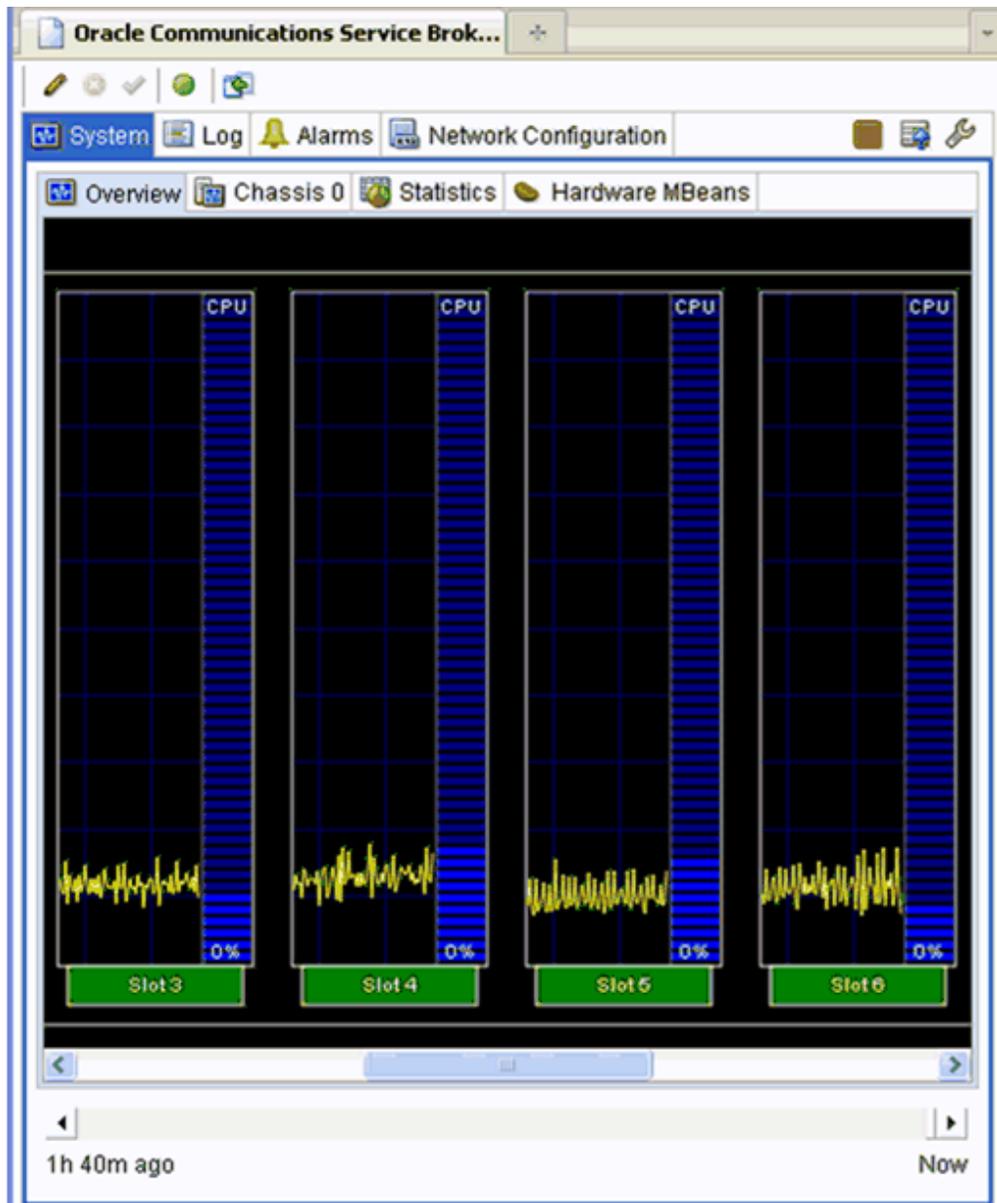
Using the Change Center to Work With Consoles

This section explains the **Change Center** buttons that you use to control changes in HA Manager consoles and navigate among the consoles.

Figure 3–2 shows the following **Change Center** buttons used by all HA Manager consoles:

- **Lock and Edit** (pencil): Use to enable changes in HA Manager consoles.

- **Commit** (check mark): Use to confirm changes in HA Manager consoles.

- **Work Offline/Online** (green/grey circle): Use to take your deployment offline and back online. You use this button primarily for upgrades and server fixes.

- **Open Console** (green plus sign): Use to navigate among the HA Manager consoles.

Figure 3–2   Console Change Center
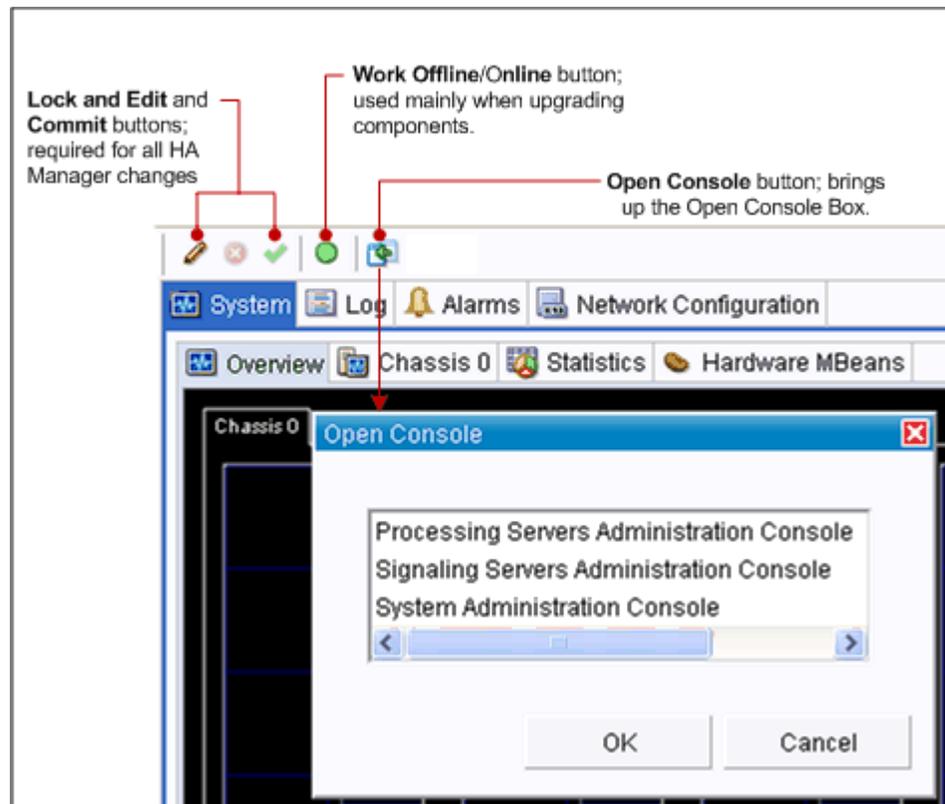
Monitoring Your Deployment with the System Administration Console

The System Administration Console includes monitoring tools you can use to ensure that the HA Manager system is running efficiently. There are two main categories of monitoring tools:

- Real-time monitoring tools that show the state of a running HA Manager deployment. The System Administration Console provides graphs that represent the real-time performance of each blade, server, and process. This console includes a performance statistics page that provides performance information in a numerical format, and a save-to-file option. See "Real-time Monitoring Tools" for more information.

- Additional monitoring tools that you use as needed. These tools include the System Administration Console **Log**, **Alarms**, and **Statistics** tabs. These tabs list important administration messages that you consult as needed. See "Additional Monitoring Tools" for more information.

Real-time Monitoring Tools

You perform system administrative and monitoring tasks as follows:

- Assess the general performance of an HA Manager deployment using the **System** tab's **Overview** subtab. The Overview subtab provides a summary view of the servers running on each blade. See "Configuring How Monitoring Data Is Displayed" to ensure that this subtab displays the metrics useful for your deployment.

■ Administer individual servers on individual blades using the **System** tab's **Chassis** subtab. You can start, shut down, restart individual servers on blades, or set the servers in safe or administrative mode. You can use the **Compare** menu option to select any number of servers and compare their performance. See "Administering Processes, Servers, and Hardware Components" for more information.

■ View recent performance using the **Timer** slider at the bottom of each Domain page tab and subtab. HA Manager saves the last few hours of real-time activity for you to view using the **Timer Slider**, shown in Figure 3–3. To view activity at any point in the session, use the slider to return to the specific time you are interested in.

Figure 3–3 shows the bottom of the Server Console with the **Timer Slider** circled at a point 48 minutes and 43 seconds prior to the present time.

Figure 3–3 Timer Slider
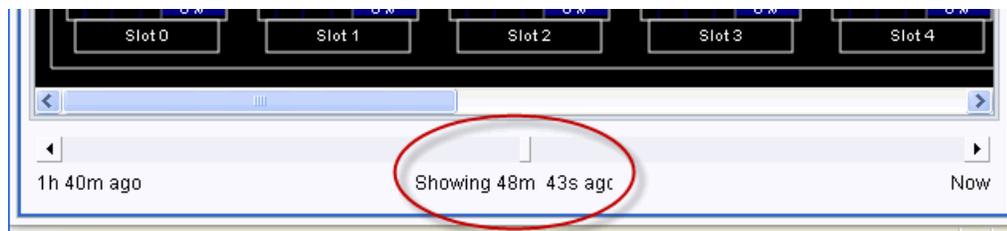
Additional Monitoring Tools

Use the following tabs and subtabs in the System Administration Console to monitor HA Manager performance as needed:

■ **Log**: Displays a list of HA Manager hardware and software events logged by the Worker Blades and HA Manager software. See "Logging" for details.

■ **Alarms**: Displays a list of HA Manager alarm messages. See "Managing Alarms" for details.

■ **Statistics**: Provides system performance information in numerical format. You can save these statistics to a file. See "Managing Statistics" for details.

The Signaling Servers Administration Console and the Processing Servers Administration Console also offer options for saving statistics.

Configuring How Monitoring Data Is Displayed

This section describes how to configure the HA Manager System Administration Console to display the visual metrics applicable to your deployment.

Clicking the **Settings** button (wrench) on the upper right corner of the System Administration Console displays the Settings dialog box.

The Settings dialog box contains the following tab options:

■ **General**: Selects the processing metrics to display, such as whether to display CPU or heap usage for each blade.

■ **Appearance**: Specifies how to display the processing metrics, such as what color to make the CPU or heap usage graphs.

■ **Filters**: Limits the metrics displayed. For example, use this pane to limit the processing display to a single protocol.

Figure 3–4 shows the Settings dialog box and the **Settings** button (circled). The Settings dialog box includes the **General**, **Appearance**, and **Filters** subtabs.

Figure 3–4   Settings Dialog Box
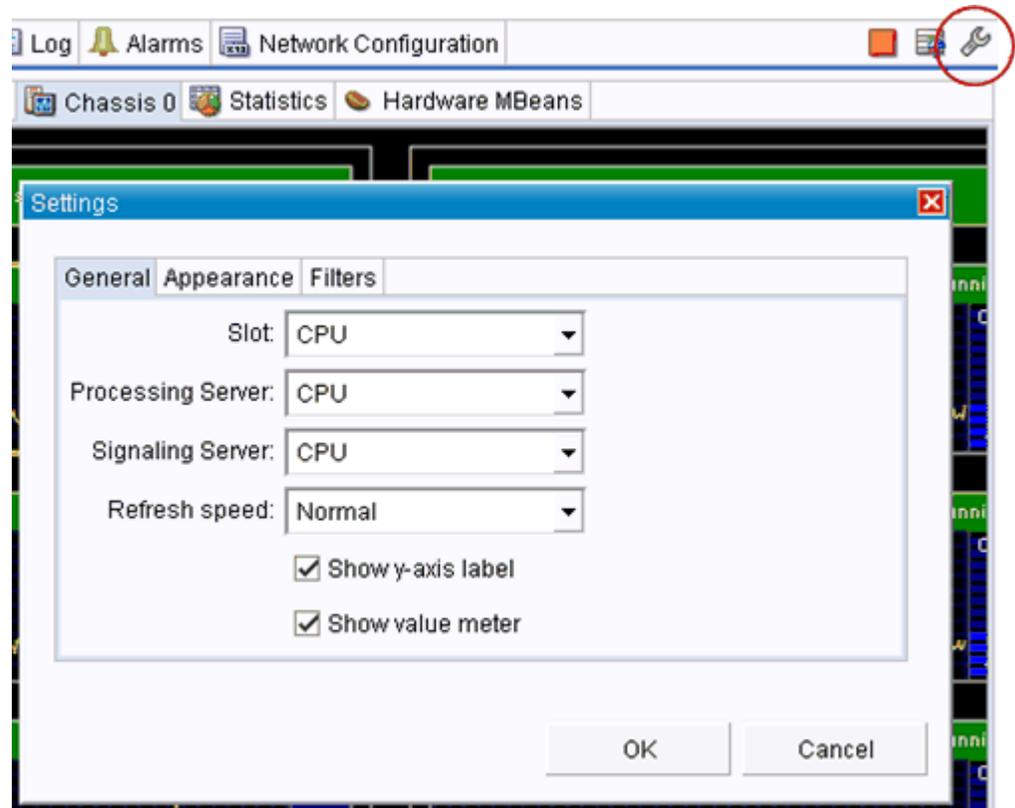
Selecting Metrics to Display

In the Settings dialog box, you use the **General** tab to select the metrics to display in the various System Administration Console tabs and subtabs:

■ **Slot**: Specifies the metric to display on the **System Overview** subtab. The options are:

  ■ **CPU**

  ■ **Sessions**

■ **Processing Server**: Specifies the metrics to display in the Processing Server **System** tab's **Chassis** subtab. The options are:

  ■ **CPU**

  ■ **Heap Usage**

  ■ **Sessions**

■ **Signaling Server**: Specifies the metrics to display in the Signaling Server **System** tab's **Chassis** subtab. The options are:

  ■ **CPU**

- **Heap Usage**

- **Work Manager Queue**

- **Event Rate**

- **Refresh speed**: Specifies how often the display is refreshed. The options are:

    - **Slow** (every 3 seconds)

    - **Normal** (every 1.5 seconds)

    - **Fast** (every .5 seconds)

- **Show y-axis label** and **Show value meter** check boxes: Shows the y-axis label and value meter respectively.

Changing the Display

In the Settings dialog box, you use the **Appearance** tab to specify how the metrics you selected in the **General** tab appear. This tab includes the following subtabs:

- **CPU**

- **Sessions**

- **Work Manager Queue**

- **Heap Usage**

- **Event Rate**

For each of these subtabs:

- Sets the Y-scale to one of the following:

    - Automatic: Automatically calculates the Y-axis data graph to the size of your window. This is the default option.

    - Manual: Resets the Y-axis scale to a value you specify

- **Color**: Displays the standard color editor in which you define RGB values

- **Width**: Changes the width of the graph lines in pixels

Filtering Data for Display

In the Settings dialog box, you use the **Filters** tab for the following:

- **Session**: Adding application MBean names to the MBean Name field limits the Activity Monitor activity displayed to applications for those MBeans.

- **Work Manager Queue**: Adding items to the Protocol Adapter field limits the display to activity for just those protocols.

- **Event Rate**: Setting the Protocol Adapter Bundle and the Event Category

Administering Processes, Servers, and Hardware Components

You administer HA Manager processes from the System Administration Console.

The System Administration Console and its subtabs use color-coding to specify the state of each processes. For example, green indicates a running process and red a stopped process. See "Monitoring" in "Managing and Monitoring Hardware and Processes" for details about the colors and states they represent.

You perform HA Manager administration tasks by selecting a component such as a Processing Server, and right-clicking to select a task from the menu.

Figure 3–5 shows the administration tasks available for the selected Processing Domain (outlined in yellow) in the **System** tab's **Chassis** subtab. The menu displays the following administration tasks:

- **Go to Running**
- **Go to Admin**
- **Go to Safe**
- **Shutdown**
- **Force Shutdown**
- **Restart**
- **Terminate**
- **Compare**

Figure 3–5   Server Menu Options in the Chassis Subtab
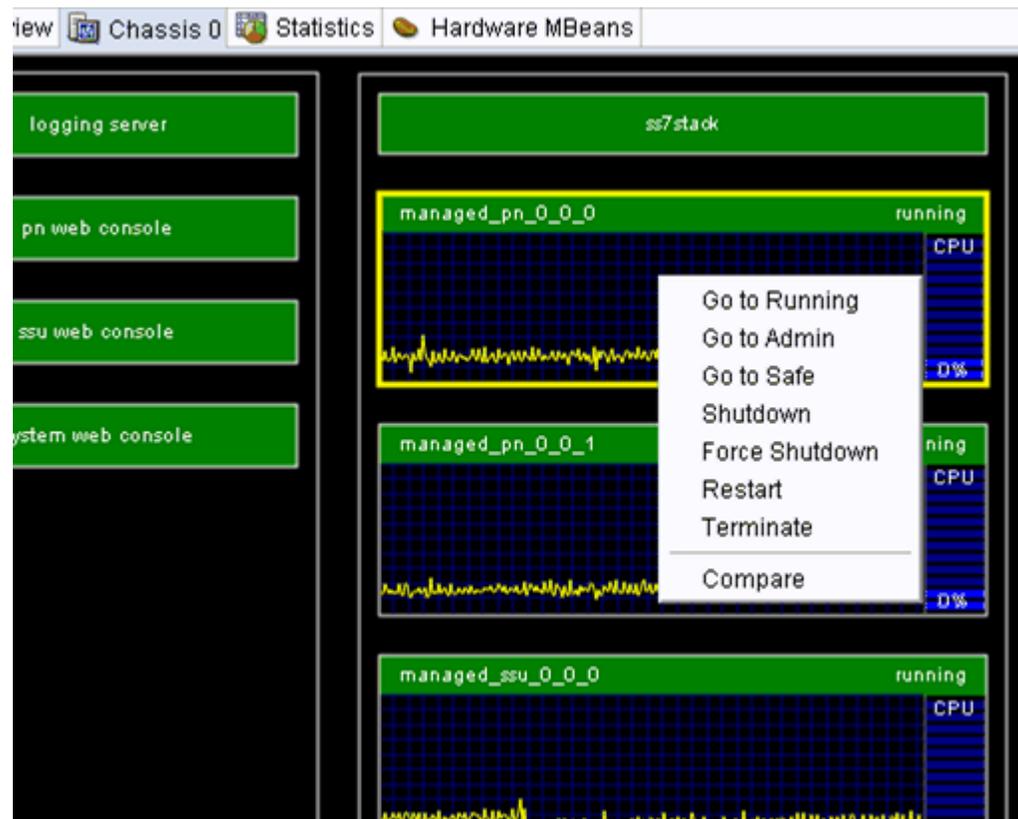
You can perform the following administration actions in the System Administration Console subtabs:

- **System** tab, **Overview** subtab:
    - **Power On**: Starts the selected blades
    - **Power Off**: Shuts down the selected blades
    - **Reset**: Shuts down and restarts the selected blades

- **Compare**: Opens a page displaying only the servers you have selected. You can name and save the comparison for later retrieval.

- **System** tab, **Chassis** subtab, Worker Blades:

    - **Go to Running**: Puts the selected server(s) in Running state

    - **Go to Admin**: Puts the selected server(s) in Administrative state

    - **Go to Safe**: Puts the selected server(s) in Safe mode

    - **Shutdown**: Gracefully shuts down the selected servers

    - **Force Shutdown**: Forces the selected servers to shut down when a graceful shutdown is not possible

    - **Restart**: Shuts down and restarts the server

    - **Terminate**: Uses a `kill` command.

- **System** tab, **Chassis** subtab, Bootstrap Blades:

    - **Stop**: Stops the selected server process

    - **Start**: Starts the selected server process

    - **Terminate**: Uses the a `kill` command

See "Managing and Monitoring Hardware and Processes" for more information about administering your HA Manager processes.

Upgrading Processing Server and Signaling Server Components

Use the Managed Upgrades tab to upgrade both the HA Manager software and firmware on individual blades as upgrades become available. You can do this in one of the following ways:

- Online: Configuration updates are propagated to all servers in the domain as changes are made

- Offline: Updates are saved to the domain configuration directory and propagated to servers only when they are restarted

See "Upgrading Service Broker Netra 6000 High Availability Manager" for details.

Managing and Configuring Hardware Components

You use the System Administration Console to manage and configure hardware components. The JMX MBean interface is provided in the Administration Console's **System** tab, **Hardware MBeans** subtab, that you use to make changes to the HA Manager Hardware controls and settings.

Figure 3–6 shows the System Administration Console with the **Hardware MBeans** subtab highlighted. The left pane shows the **N6000** MBeans navigation tree with the **SUN_NetraCard** with the selected blade's ID **BL8-8;1d-81** highlighted. The right pane shows the selected blade's BL8-8;1d-81 configuration options that you can change, including the **ID** and the **RelativeLocation** fields.

Figure 3–6   Hardware MBeans Subtab Window
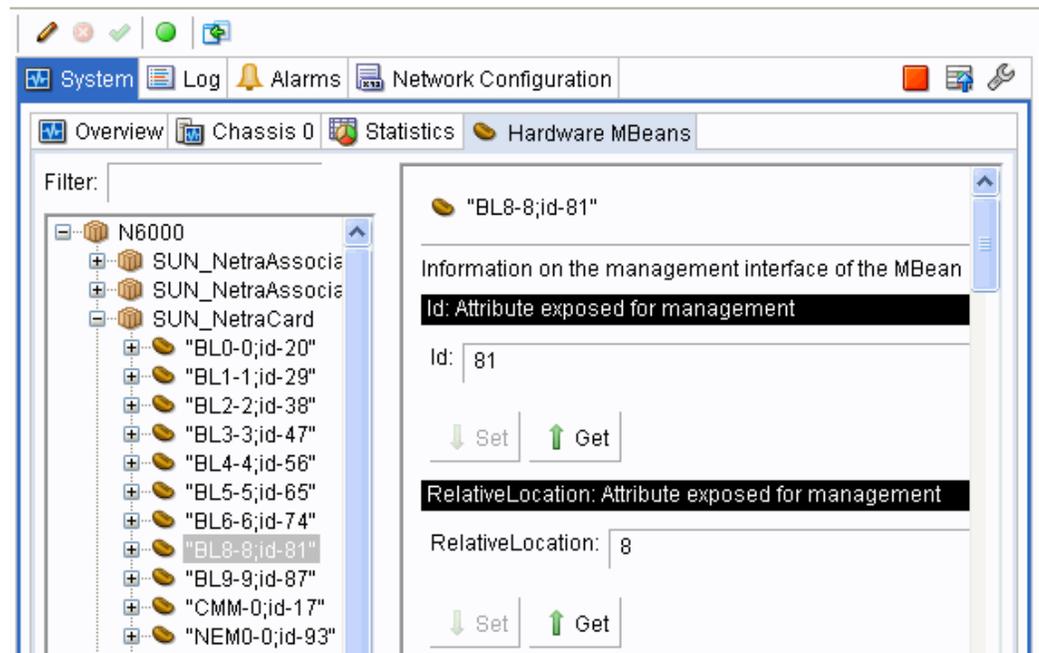
The **Hardware MBeans** subtab offers a subset of available hardware controls and settings. If an ID is listed in the navigation tree but the right side of the pane is empty, you cannot change the settings for the selected blade using the System Administration Console. Instead, use the CMM ILOM Web interface to change the settings.

For details on changing MBean settings using ILOM, see the Oracle Integrated Lights Out Manager (ILOM) 3.0 documentation.

For details on understanding and using Service Broker MBeans, see the discussion about MBean configuration in *Oracle Communications Service Broker System Administrator's Guide*.

Configuring Network Connectivity

You use the **Network Configuration** tab on the System Administration Console to configure parameters for internal and external network traffic. In this tab, you can set configurations for each blade in a chassis or global configurations that apply to all blades. The **Network Configuration** tab contains a navigation pane on the left and a configuration pane on the right. Clicking the **Global** node in the navigation tree displays external network traffic connectivity settings such as NTP, DNS, and OAM.

Expanding the Chassis node and clicking the **Slot** node in the navigation tree displays the internal traffic configuration settings for SIP/Diameter and SIGTRAN traffic.

Figure 3–7 shows the **Network Configuration** tab with Slot 2 selected. The configuration pane on the right shows the SIP/Diameter and SIGTRAN settings with the default IP Address and subnet Mask specified for Slot 2. The traffic **Type** boxes show STATIC.

See "Connecting HA Manager to the Network" for details on configuring network traffic for your HA Manager deployment.

Figure 3–7   Network Configuration Tab
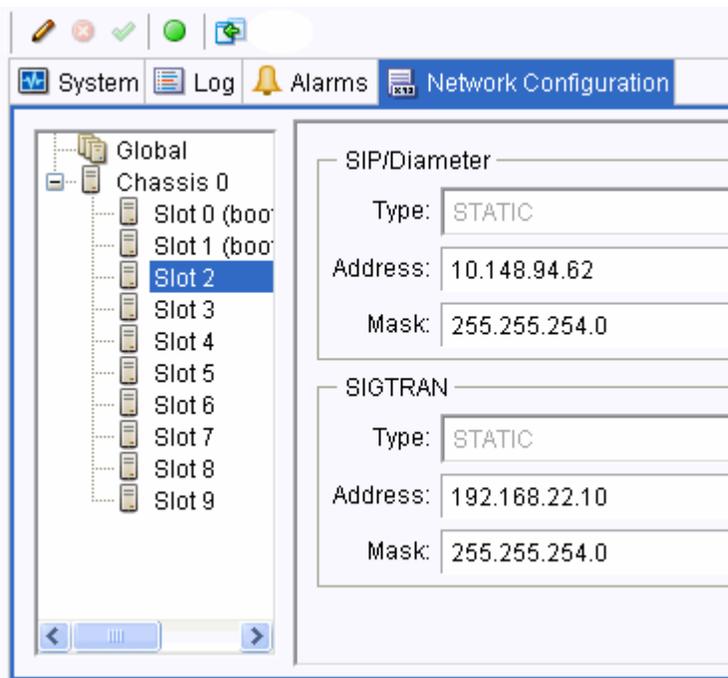
Configuring Signaling Traffic

This section briefly explains the tasks that you perform using the HA Manager Signaling Servers Administration Console. Display this console using the **Open Console** button in the console Change Center. See "Using the Change Center to Work With Consoles" for details.

The Signaling Servers Administration Console includes the OCSB navigation tree that you use to:

- Configure traffic among the Signaling Servers. Signaling Servers (marked as SSUs) are found on the HA Manager Worker Blades.

- Configure traffic between Signaling Servers and your IMS and SS7 networks.

- Create Signaling Server Groups.

- Configure routing rules between Signaling Servers and Diameter nodes.

- Upgrade the various Signaling Server software components.

For details on performing these tasks, see "Connecting HA Manager to the Network" and the *Oracle Communications Service Broker Signaling Domain Configuration Guide*.

Configuring Processing Traffic

This section briefly explains the tasks that you perform using the Processing Servers Administration Console. Display this console using the **Open Console** button in the console Change Center. See "Using the Change Center to Work With Consoles" for details.

Figure 3–8 shows an example HA Manager Processing Servers Administration Console. The OCSB navigation tree is shown in the left pane with the **Orchestration**

Engine highlighted. The Configuration pane on the right shows the **General** subtab, which contains the configuration options that you can change.

Figure 3–8   Processing Servers Administration Console Page
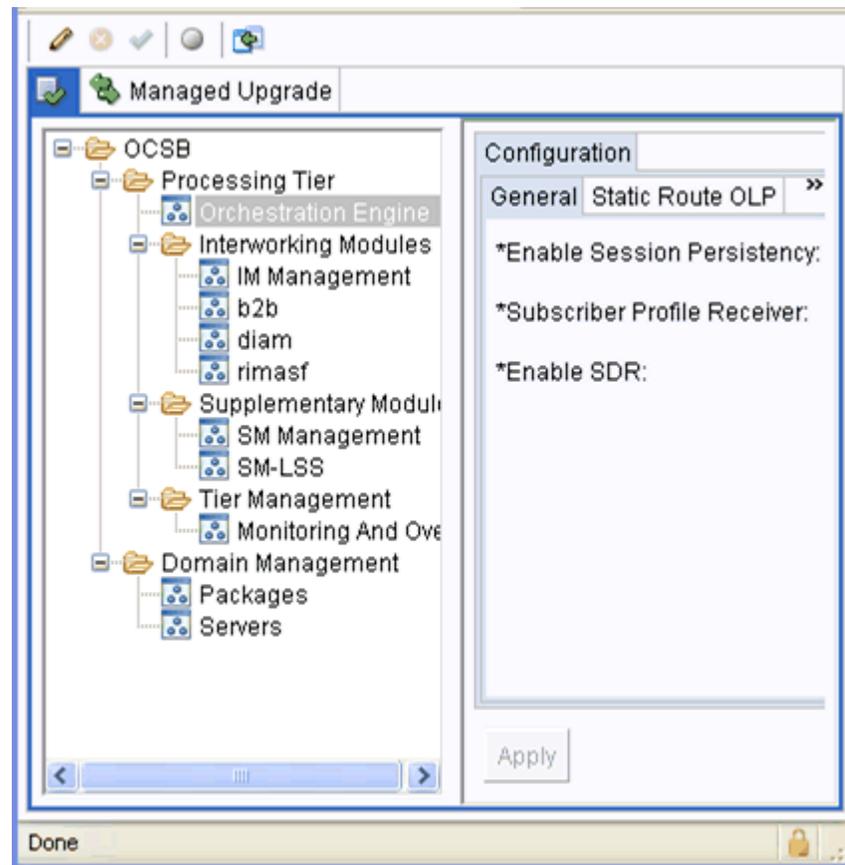
The Processing Servers Administration Console includes OCSB and Managed Upgrade navigation trees that you use to:

- Configure the Orchestration Engine, including orchestration logic and monitoring settings.

- Create and configure the Interworking Modules (IMs) that perform the protocol translation, including the Diameter IM used for charging. You activate and de-activate IMs from this console.

- Create and configure any Supplementary Modules.

- Set Processing Server configuration options, including general management, monitoring, and overload protection settings.

- Manage the Processing Domain, including starting and stopping, uninstalling packages, and configuring Processing Servers.

- Perform software upgrades to the various Processing Domain software components.

For details on performing these tasks, see the *Oracle Communications Service Broker Processing Domain Configuration Guide*.

4

Connecting HA Manager to the Network

This chapter describes how to configure network connectivity for the Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager).

About Networking Configuration

You connect your HA Manager to networks by using the **Network Configuration** tab in the System Administration Console. This tab contains the network settings specific to the hardware platform of the HA Manager deployment. The network configuration parameters include global properties for the system, such as the address of the DNS servers and NTP servers.

The parameters also include Worker Blade-specific settings that enable the blades to connect to the remote SIGTRAN, SIP, and other communications networks for which the system performs traffic brokering.

To implement the HA Manager deployment, you must configure the network parameters in the **Network Configuration** tab. Oracle recommends that you first read "About Predefined VLANs" to become familiar with the networking features that are built into the system.

About Predefined VLANs

Within the HA Manager system, traffic is separated into categories as Virtual LANs (VLANs). This allows the HA Manager to separate for example, internal administrative traffic from SIP or SIGTRAN communications network traffic. Each traffic category is a VLAN, and each blade participates in the VLAN appropriate to its role, whether a Bootstrap Blade or a Worker Blade.

The Bootstrap Blades connect to the Operational Support Systems and Operational Administration and Maintenance (OSS OAM) network. This VLAN is used for administrative activities, such as Administration Console and SSH console access, and access for other clients that OSS OAM functions.

The Worker Blade VLANs carry the communications traffic for which the system performs brokering. These VLANs are:

- SIP/Diameter: Intended for all IP-based communications traffic, including SIP, Diameter, HTTP and SMPP.

- SIGTRAN: Intended for SS7 SIGTRAN traffic.

> **Note:** Additional VLANs exist for internal use. However, they are not exposed to external traffic and therefore not configurable.

When configuring network connectivity for the HA Manager deployment, you assign IP addresses for the blades on the VLANs on which they participate. These are the addresses that an external element, such as a load balancer, would use to send traffic to the system. Not every HA Manager deployment needs connectivity for each type of network. If the interface is not required, its value can remain at the default **0.0.0.0**.

For each VLAN, you also configure static routes and IP rules appropriate for the network. You specify the routes and IP rules using the standard Linux format.

In terms of the physical interface, assigning an IP address to the blade associates the address to one of several logical adapters on the blade's dual bonded port, bond0. In some cases, you might need to know the exact correspondence between the virtual adapters and VLAN. This could be useful, for example, when troubleshooting a connectivity issue with packet capture software. The correspondence between the interfaces and VLAN is as follows:

- bond0.11: SIP/Diameter

- bond0.22: SIGTRAN

- bond0.33 OSS OAM

Signaling Server Configuration Considerations

As described in "About Predefined VLANs", you use the **Network Configuration** tab to assign each blade an IP address on a particular network. However, this task is only one of the steps required to enable connectivity to communications networks. You also need to configure Signaling Server settings.

Signaling Server configuration is common to any Service Broker deployment, whether it uses the HA Manager software or a software-only Service Broker installation. However, configuring the Signaling Servers in an HA Manager deployment imposes certain specific configuration settings.

The following sections list the Signaling Server parameters that have specific values or considerations in the context of an HA Manager implementation. The settings are listed by network type. For complete information on configuring the Signaling Servers, see the *Oracle Communications Service Broker Signaling Domain Configuration Guide*.

SIP Considerations

SIP-related Signaling Server configuration parameters that require a specific value in an HA Manager deployment are:

- Globally routable user agent URI

  You can access this parameter in the SIP Server tab under the SSU SIP node of the Signaling Servers tree. The parameter must be set to the default value, **sip:127.0.0.1:5060**.

- Network access point listening address for each SSU instance.

  The listening address for each instance must be set to the corresponding blade's SIP/Diameter IP address as specified in the Network Configuration page.

In the **Network Access Points** tab, Signaling Server instances built-into the HA Manager system are named in the form of **ssu**n, where n is the instance number of the SSU.

To view the listening address for each, select the SSU identifier from the Network Access point list and then click the **Listen Address** tab.

Table 4–1 shows the mapping of SSU instance IDs to chassis slot in a deployment with four Worker Blades.

Table 4–1 SSU Instance to Blade Correspondence in a Single Chassis System

SSU Instance ID	Chassis Slot
ssu0	Slot 2
ssu1	Slot 3
ssu2	Slot 4
ssu3	Slot 5

Note that in a single chassis system, slots 0 and 1 are reserved for Bootstrap Blades.

SS7 SIGTRAN Considerations

SIGTRAN Signaling Server configuration parameters that require a specific value in an HA Manager deployment are:

■ Local system connectivity parameter for the M3UA module.

For each SSU instance, you must set the primary IP address to the value of the corresponding blade's SIGTRAN IP address, found in the **Network Configuration** tab. Note that the configuration parameters include a secondary IP address (**IP Address 2**). The secondary address can remain empty because connection redundancy is achieved on the HA Manager using a dual bonded port.

■ SS7 Stack configuration

In the HA Manager deployment, each Worker Blade runs an instance of SS7 stack software. Therefore, the **SS7 Stack IP** and **SS7 Stack Port** parameters on the **General SSU SS7 SIGTRAN** tab can remain at their default value, **127.0.0.1** and **20004**.

Diameter Considerations

The **Address** field in the Diameter node configuration is the Diameter-related Signaling Server configuration parameter that requires a specific value in an HA Manager deployment.

This value should match the SIP/Diameter network address for the corresponding Worker Blade. You need to create a Diameter node for each Worker Blade on the SIP/Diameter VLAN.

Configuring Global Network Parameters

Global parameters define the common network parameters for the system. They include, for example, the address of the NTP and DNS servers on the local network.

Certain parameter values, such as those related to the OSS OAM network, are populated with values specified at system installation time. You need to specify other parameters manually.

In addition to general system parameters, global network settings include common settings used by Worker Blades to connect to the SIP/Diameter and SIGTRAN traffic networks.

> **Note:** If the existing configuration parameters render the
> Administration Console inaccessible on the local network, you can
> connect a laptop directly to the NEM switch in the chassis using an
> Ethernet cable, and access the Administration Console from a Web
> browser on the laptop.

To configure global network parameters:

1. From a Web browser, click the **Network Configuration** tab in the System
   Administration Console.

2. If it is not already selected, click the **Global** node in the domain navigation pane.

   The Global Parameters pane appears.

3. In the Global Parameters configuration pane, configure the parameters that are
   appropriate for your network environment.

   Table 4–2 describes the parameters in the **Network Configuration** tab. Unless
   otherwise noted, changes to a setting take effect on system restart.

Table 4–2   Global Network Parameters

Parameter	Description
NTP	The IP address of the Network Time Protocol (NTP) server on the local network to be used for clock synchronization.
	You enter from one to three NTP server addresses in the text fields labeled **1**, **2** and **3**. When it needs to perform clock synchronization, the Bootstrap Blade contacts all NTP servers on the list and determines the clock time based on an evaluation of all responses.
DNS	The IP address of the DNS (Domain Name System) server on the local network to which the system should send DNS queries.
	You enter from one to three DNS server addresses in order of contact priority in the text fields labeled **1**, **2** and **3**. When it needs to perform a name query, the Bootstrap Blade tries to contact the first DNS server in the list. If that attempt fails, it then attempts to contact the next DNS server in the list.
OSS OAM	The IP address for the Bootstrap Blades on the VLAN used for administrative traffic, including Web access to the Administration Console.
	This address serves as a floating IP address for system administration. It is taken by the primary Bootstrap Blade. In a failover event, the standby Bootstrap Blade claims this address and assumes the primary role.
	The configuration fields are:
	■ **Type**: The mode in which the virtual adapter acquires an IP address. It can be Static, in which case the address is specified in the **Address** field, or DHCP, in which case the address is acquired dynamically from the default gateway.
	■ **Address**: If **Type** is static, the IP address to use.
	■ **Mask**: The subnet mask that identifies the network portion of the address, such as **255.255.254.0**.

Table 4–2   (Cont.) Global Network Parameters

Parameter	Description
Time Zone	The time zone used for time display purposes. Choose from the following options:
	■ Coordinated Universal Time by selecting the **UTC** check box.
	■ A specific time zone by selecting a location or region from the **Zone** list.
	Changes to this setting take effect immediately on applying configuration modifications.
SIP/Diameter	The static routes and routing table rules for the IP protocol VLAN, which applies to SIP, Diameter, and other types of IP-based communications traffic. The configuration should be entered in the **Route** and **Rule** fields.
	See "Configuring Static Routes and IP Rules" for more information.
SIGTRAN	The static routes and routing table rules for the SIGTRAN VLAN. The configuration should be entered in the **Route** and **Rule** fields.
	See "Configuring Static Routes and IP Rules" for more information.
OSS OAM	The static routes and routing table rules for the OSS OAM VLAN. The configuration should be entered in the **Route** and **Rule** fields.
	See "Configuring Static Routes and IP Rules" for more information.

Configuring Static Routes and IP Rules

The communication networks to which the blades connect are likely to be separate in terms of their network services and topology, and therefore, have different connectivity requirements.

The **Network Configuration** tab enables you to configure static routes and IP rules per VLAN. These VLAN-specific settings apply to all Worker Blade adapters on the same network. That is, a route specified for the SIGTRAN network applies to the SIGTRAN network adapters on all Worker Blades.

You configure the network-specific settings in the **Route** and **Rule** fields in the **Network Configuration** tab. The content of each field is used to compose the corresponding rule-*interface* and route-*interface* files on disk, which are conventional Linux interface configuration files. Therefore, the content of the field follows standard interface file format.

The **Route** value for the SIGTRAN network, for example, would be written to a file on a disk named `route-bond0.22`. (See "About Predefined VLANs" for the interface identifiers for each VLAN.) A sample value for the **Route** field that uses network/netmask directive style is:

```
ADDRESS0=0.0.0.0
NETMASK0=0.0.0.0
GATEWAY0=10.148.94.1
ADDRESS1=10.148.124.0
NETMASK1=255.255.252.0
GATEWAY1=10.148.94.1
ADDRESS3=10.148.104.0
```

```
NETMASK3=255.255.254.0
GATEWAY3=10.148.94.1
```

The example shows three static route definitions, each of which is made up of the network address, the network mask, and the gateway to use for that network address.

By default, the **Route** and **Rule** text fields are empty, except for the **Route** field for the OAM OSS network, where route information is derived from the configuration at system initialization time.

For more information on static routes, see Red Hat documentation on static routes:

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_
Linux/5/html/Deployment_
Guide/s1-networkscripts-static-routes.html

The **Rule** field enables you to specify traffic routing policies for packets based on various packet attributes. The contents of the field should conform to standard Linux `ip rule` statements. For more information on the `ip rule` command format, see the `ip rule` subsection for the Linux `ip` command `man` page.

Configuring Worker Blade Network Settings

To enable connectivity between the HA Manager Worker Blades and remote communications networks, you need to configure the network settings for the Worker Blades (the VLAN interface parameters). The interface parameters specify the local IP address for each blade on the SIP/Diameter or SIGTRAN network. This is the address to which systems, external to HA Manager (such as a load balancer), direct traffic to the Worker Blades.

Note: Blade-specific settings are actually associated with a slot in the chassis rather than a specific blade. This makes the blades in the chassis hot-swappable. That is, if you change blades in a particular slot, the configuration settings applicable to that slot are assigned to the new blade.

Depending on your network topology, you may also need to specify static routes for each network. See "Configuring Static Routes and IP Rules" for more information.

To configure the Worker Blade network settings:

1. From a Web browser, click the **Network Configuration** tab in the System Console.

   The domain navigation pane shows the chassis slots that are occupied by server modules, identified by its slot number, such as **Slot 2**.

2. Under the **Chassis** node in the domain navigation pane, select the slot corresponding to the blade for which you want to configure network settings.

   The network settings for that slot appear in the configuration pane. The page defines settings for two network adapters:

   ■ SIP/Diameter: Intended for IP-based traffic, such as SIP and Diameter.

   ■ SIGTRAN: Intended for SS7 SIGTRAN traffic.

3. For the network to which you want to enable connectivity, choose the address allocation mode from the **Type** list, from the following options:

- **DHCP**: Specifies dynamic address allocation for the adapter. In this case, the blade acquires an address from the default gateway specified in the global settings.

- **Static**: Specifies a static IP address for the adapter. If selected, the **Address** and **Mask** field are enabled. In the fields, enter the IP address and subnet mask for the adapter on the network.

4. Reboot the blade to effect the changes.

To complete the network connectivity configuration for the HA Manager, configure the Signaling Server, as described in the *Oracle Communications Service Broker Signaling Domain Configuration Guide*.

Keeping a Record of Network Settings for Worker Blades

It is imperative that you keep a record of your Worker Blade network settings. If an HA Manager Worker Blade fails, you need to replace it and configure the new one with the exact same network settings.

To keep a record of network settings of a Worker Blade:

1. In the CMM ILOM Web interface, in the chassis tree, select a blade whose network settings you want to record.

The **System Overview** tab appears.

2. Click the **Configuration** tab and then click the **Network** tab.

The **Network** tab shows the MAC address and network settings for the selected Worker Blade.

3. Keep a record of the network settings for future reference.

4. Repeat steps 1 through 3 for each Worker Blade in the deployment.

5

Managing and Monitoring Hardware and Processes

This chapter describes how to manage and monitor hardware and processes for Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager).

Managing and Monitoring Hardware and Processes

Process management relates to the life cycle management of:

- Service Broker Signaling Servers
- Service Broker Processing Servers
- SS7 SIGTRAN processes
- Web Administration Console Servers
- Logging processes

Process Management

Process management controls the life cycle of Signaling Servers and Processing Servers. These servers can have the following states:

- Shut down
- Safe mode
- Admin
- Running

For a description of the life cycle states and transitions, see *Oracle Communications Service Broker System Administrator's Guide*.

Process management includes functions for:

- Starting a process
- Stopping a process
- Terminating a process

The term process refers to:

- Signaling Servers
- Processing Servers
- Logging Server

- SS7 SIGTRAN

- Web Administration Console Servers

When you restart a Signaling or Processing Server, it is gracefully shut down according to the HA Manager life cycle, started, and then brought to the state it was in prior to the restart.

When you stop or terminate a process that is not a Signaling or Processing Server, the process state is managed at the operating system level.

Process management is handled by two components:

- external Management Controller (eMC)

- external Management Agent (eMA)

The eMC controls all Signaling Servers and Processing Servers in a deployment and spans multiple Service Broker domains.

eMA runs as a separate process on the same Worker Blade as the process or processes it controls. The eMA process is automatically started and stopped as a part of the operating system boot and shut down process. Only one eMA process runs on each Worker Blade.

A combined eMC and eMA process runs on the Bootstrap Blade. You use the Administration Console GUI to control this process.

Hardware Management

The blades in a chassis can be:

- Powered on

- Powered off

- Restarted

Monitoring

You can monitor the state of the different processes and the blades using the Administration Console. The states are color coded as described in Table 5–1.

Table 5–1   Color Codes for Monitored Components

Color	Monitored Component	Description
Grey	Signaling Servers Processing Servers Logging Server SS7 SIGTRAN stack Administration Console Web Servers	The eMC in the Administration Console is disconnected from the eMA on the blade. The state is unknown.
Grey	Blade	The blade in the slot is not monitored. The state is unknown.
Black	Signaling Servers Processing Servers	The Signaling Server or Processing Server is in the Shut down state.

Table 5–1   (Cont.) Color Codes for Monitored Components

Color	Monitored Component	Description
Black	Logging Server SS7 stack Administration Console Web Servers	The process is not running.
Black	Blade	The blade in the slot is powered off.
Red	Signaling Servers Processing Servers Logging Server SS7 stack Administration Console Web Servers	The process is in the Error state.
Orange	Signaling Servers Processing Servers	The process is in the Safe state.
Blue	Signaling Servers Processing Servers	The process is in the Admin state.
Green	Signaling Servers Processing Servers	The process is in the Running state.
Green	Logging Server SS7 stack Administration Console Web Servers	The process is running.
Green	Blade	The blade in the slot is powered on.

Metrics available for Worker Blades are:

- CPU utilization (percentage)
- Number of HA Manager sessions handled

Metrics available for Signaling Servers and Process Servers are:

- CPU utilization (percentage)
- Heap Usage (MB)
- Size of the Work Manager thread pool

Signaling Servers also provide the number of processed HA Manager events per second as a metric.

About Process Distribution

HA Manager Signaling Servers and Processing Servers are distributed over the available blades in a chassis. See Table 5–2 for a description of this distribution.

The servers are named using the following convention:

- Signaling Servers: managed_ssu_*chassisID_slotID_processID*

- Processing Servers: managed_pn_*chassisID_slotID_processID*

  Where:

  - *chassisID* is the ID of the chassis

  - *slotID* is the ID of the slot within the chassis

  - *processID* is the unique identifier for the process running on the blade inserted in the slot

Table 5–2   *Process Distribution in a single Chassis Deployment*

Slots	Type	Processes
0 and 1	Bootstrap Blade	Signaling Servers Administration Console Web Server (1)
		Processing Servers Administration Console Web Server (1)
		System Administration Console Web Server (1)
		Logging Server (1)
2, 3, 4, 5	Worker Blade	SS7 SIGTRAN (1)
		Signaling Servers (1)
		Processing Servers (2)
6, 7, 8, 9	Worker Blade	Processing Servers (3)

In a multi-chassis setup, there is only one Bootstrap Blade in each chassis. The Bootstrap Blade is inserted in to slot 0. The Worker Blades are inserted into slots 1 through 9. This means that there is one more Worker Blade per chassis in a multi-chassis setup than there is in a single-chassis setup. This additional Worker Blade runs three Processing Servers.

About Process and Blade States

The state of each process and blade is indicated by its color and corresponding text in the Administration Console. See Table 5–1 for details.

The state of the blades is shown in the **System** tab **Overview** subtab.

The state for blades and processes are visible in the following subtab:

- **Chassis** *n*

  n is the ID of the chassis

- *Comparison_View*

  Comparison_View represents the name of a comparison view that you created. See "Comparing Blades" and "Comparing Signaling and Processing Server Processes" for more information.

Activating and Deactivating Monitoring

By default, the deployment is not monitored and all monitored-component icons in the console are grey. See Table 5–1. Statistics continue to be collected even when monitoring is stopped.

To start monitoring, click the **Start Monitoring** button:

To stop monitoring, click the **Stop Monitoring** button:

Setting the Metrics to Monitor

To set the metrics to monitor:

1. In the System Administration Console, click the **System** tab.

2. On the top right corner of the window, click the **Settings** icon (the wrench).

    The Settings dialog box opens.

3. In the **Slot** list, select one of the following values to set the metric to monitor for the blades:

    - **CPU**: To monitor the CPU utilization on the blade

    - **Sessions**: To monitor the number of HA Manager sessions handled by the blade

4. In the **Signaling Server** list, select one of the following values to set the metric to monitor for a Signaling Server:

    - **CPU**: To monitor the process's CPU utilization

    - **Sessions**: To monitor the number of HA Manager sessions handled by the process

    - **Heap Usage**: To monitor the heap usage of the process

    - **Event Rate**: To monitor the number of Service Broker events per second that is handled by the process

5. In the **Processing Server** list, select one of the following values to set the metric to monitor for a Processing Server:

    - **CPU**: To monitor the process's CPU utilization

    - **Sessions**: To monitor the number of HA Manager sessions handled by the process

    - **Heap Usage**: To monitor the heap usage of the process

6. In the **Refresh speed** list, select one of the following values to set how often the statistics are refreshed in the GUI:

    - **Slow**: Every 3 seconds

    - **Normal**: Every 1.5 seconds

    - **Fast**: Every 0.5 second

7. Click **OK**.

Monitoring a Worker Blade

To monitor a Worker Blade:

1. In the System Administration Console, click the **System** tab and then the **Overview** tab.

2. Locate the slot where the blade is inserted.

   The metric is displayed as a bar graph and in numeric format. A graph showing historical data is also displayed.

Monitoring CPU Use for Processing or Signaling Servers

To monitor the CPU utilization of a Signaling Server or Processing Server:

1. In the System Administration Console, click the **System** tab and then the **Chassis** *n* tab.

2. Locate the slot where the blade is inserted.

   The metric is displayed as a bar graph and in numeric format. A graph showing historical data is also displayed.

Comparing Blades

To compare the metrics of set of blades:

1. In the System Administration Console, click the **System** tab and then the **Overview** tab.

   The **Overview** tab for deployment is displayed.

2. Select the slots in which the blades you want to compare are inserted while holding down the **Shift** key.

   Selected slots are outlined in yellow.

3. Right-click the selected slots.

   A context menu opens.

4. Select **Compare** from the menu.

   The Enter Name dialog box appears.

5. Enter a name for the comparison view and click **OK**.

   A new tab is added for the comparison view. The tab is labeled with the name you entered. This tab is referred to as *Comparison_View*.

   The tab displays the metrics for each slot.

Comparing Signaling and Processing Server Processes

To compare a set of Signaling Servers or a set of Processing Servers:

1. In the System Administration Console, click the **System** tab and then the **Chassis** tab of the chassis you want to monitor.

2. Select the server processes to compare while holding down the **Shift** key.

   Selected server processes are outlined in yellow.

3. Right-click the selected processes.

A context menu for process management opens.

4. Select **Compare** from the menu.

   The Enter Name dialog box appears.

5. Enter the name of the comparison view and click **OK**.

   A new tab is added for the comparison view. The tab is labeled with the name you entered. This tab is referred to as *Comparison_View*.

   The tab displays the process for each slot.

Changing Processing or Signaling Server States

To change the state of a Processing Server or a Signaling Server:

1. In the System Administration Console, click the **System** tab and then the **Chassis** tab of the chassis you want to monitor.

   The tab for the corresponding chassis is displayed.

2. Click the server process to select it. Select multiple server processes while holding down the **Shift** key.

   Selected server processes are outlined in yellow.

3. Right-click the processes.

   A context menu for process management opens.

4. Select one of the following options from the menu:

   - **Go to Running**

     Transitions the server process to Running state.

   - **Go to Admin**

     Transitions the server process to Admin state.

   - **Go to Safe**

     Transitions the server process to Safe state.

   - **Shutdown**

     Gracefully transitions the server process to Shutdown state.

   - **Force Shutdown**

     Forces the server process to Shutdown state.

   - **Restart**

     Gracefully shuts down the process according to the HA Manager life cycle, restarts it at the operating system level using the UNIX `kill` command, and then transitions it to the state it was in before it was restarted.

   - **Terminate**

     Terminates the process on an operating system level using the UNIX `kill -9` command.

   - **Compare**

     Compares two or more processes. See "Comparing Signaling and Processing Server Processes" for more information.

Changing Process States

To change the state of an Administration Console Web Server, Logging Server, or SS7 SIGTRAN process:

1. In the System Administration Console, click the **System** tab and then the **Chassis** tab of the chassis you want to monitor.

2. Click the process to select it. Select multiple processes while holding down the **Shift** key.

   Selected processes are outlined in yellow.

3. Right-click the processes.

   A context menu for process management opens.

4. Select one of the following options from the menu:

   - **Start**

     Starts the selected processes.

   - **Stop**

     Stops the selected processes at the operating system level using the UNIX `kill` command.

   - **Terminate**

     Terminates the selected processes at the operating system level using the UNIX `kill -9` command.

Changing Blade States

To change the state of a blade:

1. In the System Administration Console, click the **System** tab and then the **Overview** tab.

   The **Overview** tab for the deployment is displayed.

2. Click the slot in which the blade you want to change the state for is inserted. Select multiple slots while holding down the **Shift** key.

   Selected slots are outlined in yellow.

3. Right-click the slots.

   A context menu opens.

4. Choose one of the following options from the menu:

   - **Power On**

     Powers on the blade.

   - **Power Off**

     Powers off the blade.

   - **Reset**

     Resets the blade by first powering it off and then on.

6

Managing Statistics

This chapter describes how to view statistics for the hardware and software components of Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager).

About Statistics

HA Manager provides performance statistics that you can use to monitor the performance of HA Manager's components. For example, you can monitor the percentage of time that a hardware component used the CPU and the amount of server memory this component consumes.

You can view statistics in real-time, review them at any point within the previous 100 minutes, and save the statistics to a file.

HA Manager gathers statistics on the following components:

- System, including all installed chassis, slots, and servers
- Individual chassis, including all installed slots and servers
- Individual slots, including all installed servers
- Individual servers

About the Statistics Window

Figure 6–1 shows the Statistics window in the Administration Console.

Figure 6–1   Statistics Window
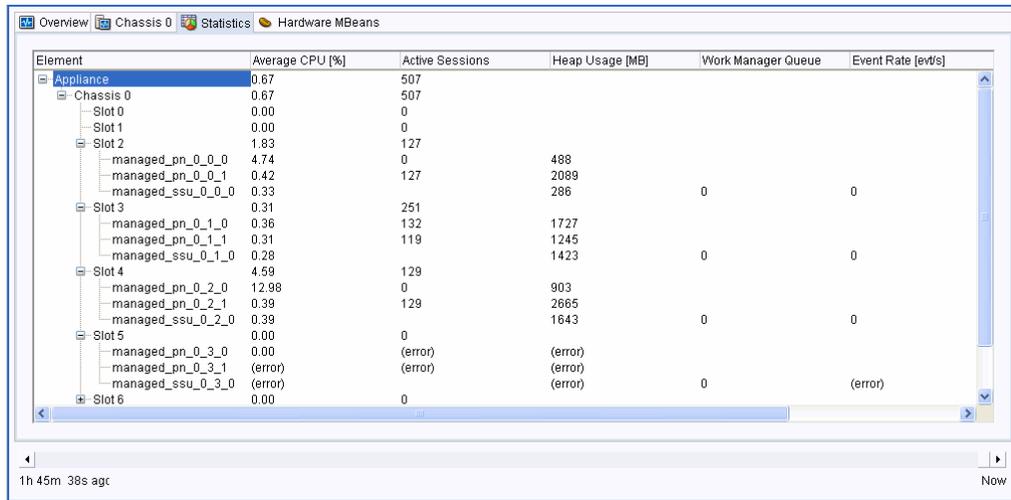
The Statistics window provides information about the performance metrics described in Table 6–1. To ensure that the latest statistics are available to view, HA Manager periodically updates the Statistics window. See for more information about setting up the frequency of the window updates.

Table 6–1   Statistics Window Fields

Field	Description	Applicable to
Element	Specifies the name of a component. Each component is represented by a node that you can expand or collapse.	Blade Signaling Servers Processing Servers
Average CPU	Specifies the average percentage of CPU capability that the component used.	Blades Processing Servers
Active Sessions	Specifies the number of HA Manager active sessions handled by the component.	Signaling Servers Processing Servers
Heap Usage	Specifies the amount of server memory, in megabytes, that the component used.	Blades Signaling Servers Processing Servers
Work Manager Queue	Specifies the number of threads in the pool.	Signaling Servers Processing Servers
Event Rate	Specifies the number of events created by an individual protocol adapter.	Processing Servers
Up Time	Specifies the amount of time a server has been running.	Signaling Servers Processing Servers

Setting the Statistics Window Refresh Rate

HA Manager periodically updates the Statistics window with the latest statistics. The default refresh rate is 1.5 seconds. You can change this rate using the Settings dialog box.

To change the Statistics window refresh rate:

1. In the System Administration Console, click the **System** tab.

2. On the top right corner of the window, click the **Settings** icon.

   The Settings dialog box appears.

3. In the Refresh list, select one of the following refresh rates:

   - **Slow**: Every 3 seconds

   - **Normal**: Every 1.5 seconds

   - **Fast**: Every 0.5 second

Viewing Statistics History

In addition to viewing the statistics in real-time, you can view statistics at any point within the previous 100 minutes. For example, you can view statistics gathered an hour earlier.

To view statistics history:

1. In the System Administration Console, click the **System** tab.

2. Click the **Statistics** subtab.

3. At the bottom of the Statistics window, move the **Timer** slider to the specific point in time at which you want to view the statistics. Figure 6–2 shows the bottom of the Server Console with the **Timer Slider** circled at a point 48 minutes and 43 seconds prior to the present time.

Figure 6–2   Timer Slider

HA Manager updates the data in the Statistics window.

Exporting Statistics

You can export the statistics gathered during the last 100 minutes to a file and download this file to a local drive. You can use third party software to view the statistics file and manipulate the statistics data.

You can specify the information to be included in the file as follows:

- Export statistics for a specific metric.

- Export statistics for a specific component and specify metrics to be exported for this component.

Exporting Statistics by Metric

You can generate a file containing the statistics for a specified metric. In this case, the first column of the file contains timestamps of when the measurements were taken for the specified metric. Other columns represent the components for which the statistics were gathered.

Figure 6–3 shows an example of a file opened in a spreadsheet program. The file contains statistics for CPU usage.

Figure 6–3   Example of Statistics Gathered by Metric

	A	B	C	D	E	F	G	H	I
1	Average CPU [%]	Appliance	Chassis 0	Slot 0	Slot 1	Slot 2	managed_pn_0_0_0	managed_pn_0_0_1	managed_ssu_0_0_0
2	01/04/2011 20:57	0	0	0	0	0	0	0	0
3	01/04/2011 20:58	0	0	0	0	0	0	0	0
4	01/04/2011 20:58	0	0	0	0	0	0	0	0
5	01/04/2011 20:58	0.01	0.01	0	0	0	0	0	0
6	01/04/2011 20:58	0.07	0.07	0	0	0.23	0.21	0.23	0.24
7	01/04/2011 20:58	0.07	0.07	0	0	0.15	0.19	0.15	0.11
8	01/04/2011 20:58	0.08	0.08	0	0	0.15	0.19	0.15	0.11
9	01/04/2011 20:58	0.1	0.1	0	0	0.32	0.28	0.33	0.36
10	01/04/2011 20:58	0.07	0.07	0	0	0.13	0.14	0.14	0.11
11	01/04/2011 20:58	0.09	0.09	0	0	0.28	0.39	0.14	0.3
12	01/04/2011 20:58	0.1	0.1	0	0	0.36	0.28	0.44	0.36
13	01/04/2011 20:58	0.08	0.08	0	0	0.22	0.28	0.15	0.22
14	01/04/2011 20:58	0.09	0.09	0	0	0.32	0.36	0.33	0.28
15	01/04/2011 20:58	0.09	0.09	0	0	0.29	0.25	0.39	0.22

To export statistics for a specific metric:

1. In the System Administration Console, click the **System** tab.

2. Click the **Statistics** subtab.

3. If there is no data in the Statistics window, click **Start Monitoring** to start gathering statistics.

   > **Note:** If there is no data, the **Export Statistics** button is disabled.

4. Click **Export Statistics**.

   The Export Statistics dialog box appears.

5. Click the **By Metric** tab.

6. In the **Metric** list, select the metric whose statistics you want to export.

7. Click **Export**.

   HA Manager generates the file and displays the Download File dialog box.

8. Click the statistics' file link to download the file.

9. Specify where on your local drive you want to save the file.

   HA Manager downloads and saves the file to the specified location.

10. To close the Download File dialog box, click **Close**.

Exporting Statistics by Component

You can generate a file containing the statistics for a specific component. In this case, the first column of the file contains timestamps of when the measurements were taken for the specified component. Other columns represent the metrics for the specified component.

Figure 6–4 shows an example of the .csv file that contains statistics on CPU usage and active sessions handled by Chassis 0.

Figure 6–4   Example of Statistics Gathered by Component

	A	B	C
1	Chassis 0	Average CPU [%]	Active Sessions
2	01/04/2011 20:57	0	0
3	01/04/2011 20:58	0	0
4	01/04/2011 20:58	0	252
5	01/04/2011 20:58	0.01	252
6	01/04/2011 20:58	0.07	252
7	01/04/2011 20:58	0.07	252
8	01/04/2011 20:58	0.08	252
9	01/04/2011 20:58	0.1	252
10	01/04/2011 20:58	0.07	252
11	01/04/2011 20:58	0.09	252
12	01/04/2011 20:58	0.1	252
13	01/04/2011 20:58	0.08	252
14	01/04/2011 20:58	0.09	252
15	01/04/2011 20:58	0.09	252
16	01/04/2011 20:58	0.09	252
17	01/04/2011 20:58	0.1	252

To export statistics for a specific component:

1.   In the System Administration Console, click the **System** tab.

2.   Click the **Statistics** subtab.

3.   If there is no data in the Statistics window, click **Start Monitoring** to begin gathering statistics.

> **Note:**   If there is no data, the **Export Statistics** button is disabled.

4.   Click **Export Statistics**.

5.   Click the **By Element** tab. In the

6.   In the **Element** list, select the component whose statistics you want to export.

7.   Select the metrics that you want to export for the specified component.

8.   Click **Export**.

   HA Manager generates the file and displays the Download File dialog box.

9.   Click the statistics file link to download the file.

   Specify where on your local drive you want to save the file.

   HA Manager downloads and saves the file to the specified location.

10.   To close the Download File dialog box, click **Close**.

7

Hardware Management and Monitoring Using MBeans

This chapter describes how Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) displays Netra hardware MBeans. See "Netra HA Manager MBean Hardware Reference" for a list of the MBeans and their attributes and operations.

About Hardware MBeans

HA Manager includes hardware MBeans that you use to manage and monitor hardware equipment. Each MBeans is associated with a specific hardware component.

The HA Manager organizes MBeans into a hierarchy as defined in the Common Information Model (CIM). You navigate through the hardware MBeans hierarchy using the Administration Console.

For more information about the CIM, see:

http://dmtf.org/standards/cim

You can perform the following actions by using hardware MBeans:

- View their hierarchy

- Set MBean attribute values

- Get an MBean attribute value

- Execute an MBean operation

See "Netra HA Manager MBean Hardware Reference" for a complete list of the HA Manager hardware MBeans.

Navigating the Hardware MBeans Hierarchy

To navigate through the hardware MBeans hierarchy:

1. Open the Administration Console.

2. In the **System** tab, click the **Hardware MBeans** tab.

   The **Hardware MBeans** appears. Table 7–1 describes the icons that represent the different types of nodes in the MBean tree.

Table 7–1   Nodes Icons

Icon	Node
	CIM element
	Hardware MBean
	Hardware MBean attribute
	Hardware MBean operation

3. Navigate to the MBean by clicking the plus sign on its left.

   The attributes and operations of the selected MBean appear.

4. Select the attribute you want to set or get, or the operation you want to invoke.

   One of the following appears in the right pane:

   ■ If you select an attribute, the field that displays the attribute value appears.

   ■ If you select an operation, the fields for entering parameters and displaying the return value appear.

Setting an Attribute

You can set the value of an attribute. Depending on the attribute, the type of the value can be either boolean or a string.

To set an attribute:

1. Navigate to the MBean whose attribute you want to set.

   See "Navigating the Hardware MBeans Hierarchy" for more information.

2. Select the attribute.

   One of the following appears in the configuration pane:

   ■ If the attribute is boolean, a check box appears.

   ■ If the attribute is a string or integer, a field appears.

3. Modify the value as required and click the **Set** button.

Getting an Attribute

When you select an attribute in the MBeans tree, and the value of this attribute is already set, the current value is displayed. However, while you are reviewing the attribute value, another user might change it. To verify that the displayed value is the most updated one, you can ask HA Manager to get the current value.

To get the value of an attribute:

1. Navigate to the MBean whose attribute you want to get.

   See "Navigating the Hardware MBeans Hierarchy" for more information.

2. Select the attribute.

   One of the following appears in the configuration pane:

- If the attribute is boolean, a check box appears. The state of the check box represents the value of the attribute.

- If the attribute is a string or integer, a field appears. The value stored in the field represents the value of the attribute.

3. To get the updated value, click the **Get** button.

   The updated value is displayed in the field.

Executing an Operation

To execute an operation:

1. Navigate to the MBean whose operation you want to execute.

   See "Navigating the Hardware MBeans Hierarchy" for more information.

2. Select the operation that you want to execute.

   The fields for entering parameters and displaying the return value appear.

3. In the parameter fields, provide information as required.

4. Click the **Execute** button.

   The return value is displayed in the **Output** field.

8

Logging

This chapter describes how to view Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) logs.

About Logging

The HA Manager Logging Server, located on the primary Bootstrap Blade, collects the logs generated by the Signaling Servers and the Processing Servers. It also groups log messages into files, assigns file names, and writes the log files to the file system.

Logging is based on a standard log4j configuration. See "log4j Configuration" for more information.

Figure 8–1 shows the flow of log messages to the Logging Server, where they are managed and stored.

Figure 8–1   Logging Server Log Aggregation
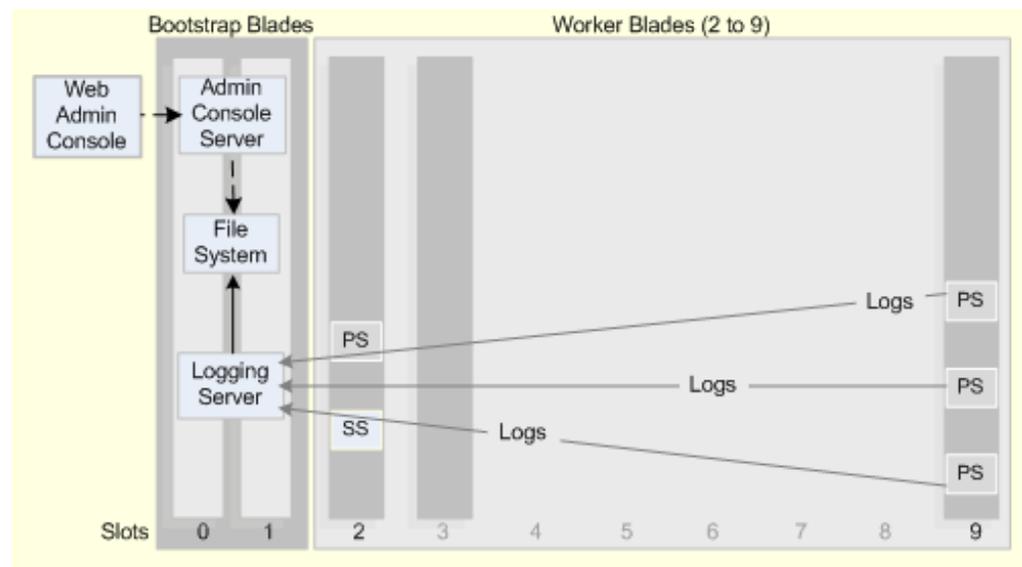
Logging Server Management

The Logging Server process is started automatically when the HA Manager is started. You can manage Logging Server life cycle tasks through the System Administration Console, **System** tab. See "Managing and Monitoring Hardware and Processes" for more information.

Log Directory and File Naming Conventions

The Logging Server writes logs collected from the Signaling Servers and the Processing Servers to a shared file system on the two Bootstrap Blades.

Log files are stored in separate directories dedicated to the server from which the logs originated. Directory names are based on the slot number and server identity of the Worker Blade from which the log events were generated.

See "Viewing Log Files from the File System" for more information.

When logs from a specific server reach a preconfigured size of 10 MB, the logs are rolled over into a single file. A default maximum of 90 MB of log files from the source server are stored at any one time.

The first 10-MB file in the directory is automatically assigned the name **server.log.1**. When the second 10-MB file is rolled over, the numeric suffix to the first file is changed to **2** and the newer file is assigned the suffix **1**.

This rollover of files continues until by default, the ninth file is created, that is, **server.log.9**, and the oldest (or tenth) file in the chain is removed from the system.

Logging Server Configuration

Logging Server and file aggregation processes are controlled by the following:

- Logging Server Configuration
- log4j Configuration

Logging Server Configuration

Table 8–1 describes the parameters and default values in the Logging Server properties file.

Table 8–1   Logging Server Properties Files

Parameter	Description	Default Value
axia.loggingserver.listenhost	Specifies the host or IP address that listens for incoming logging events.	**None**. Defaults to all local address.
axia.loggingserver.port	Specifies the port the Logging Server uses to listen for incoming logging events.	**4560**
axia.loggingserver.log4j.configuration	Points to the log4j configuration file.	**properties/loggingserver-configuration.xml**
axia.loggingserver.root.log.directory	Specifies the root log directory for directories and files. Defaults to the logs directory in the Logging Server directory.	**Logs**
log4j.configuration	log4j configuration file that the Logging Server uses for its own log events.	**log4j.xml**

Table 8–1   (Cont.) Logging Server Properties Files

Parameter	Description	Default Value
axia.platform	Specifies an internal property used for the Logging Server process setup.	Used for platform properties setup
axia.loggingserver.aggregate.logs	Specifies how logs are aggregated.	**False** Setting this value to **True** would collate logs from different servers into a single file.

log4j Configuration

Logging is based on the standard log4j logging framework.

The log4j configuration for the Logging Server and each of the Signaling Servers and Processing Servers is as follows:

■ Logging Server: Configured in the **loggingserver-configuration.xml** file

■ Signaling Servers and Processing Servers: Logs of each server are configured in a standard **log4j.xml** configuration file, located under directories dedicated to specific servers

> **Note:**   It is recommended not to modify the logging configuration files. If it is necessary for you to change any element in the files, please first contact Oracle Consulting for assistance.

For detailed information about log4J and the configuration files, see log4J documentation at:

http://logging.apache.org/log4j

Viewing Log Files

You can view the logs in either of the following ways:

■ Viewing Files Using the System Administration Console

■ Viewing Log Files from the File System

Viewing Files Using the System Administration Console

You can view the log files using the System Administration Console. Up to 100 log events are displayed per page.

To view log files using the System Administration Console:

1. In the System Administration Console main screen, click the **Log** tab to display a list of log messages.

   Figure 8–2 shows how log messages are displayed in the **Log** tab.

Figure 8–2   List of Log Messages

The active logs are displayed in the Log window with the newest message at the top of the page and the oldest at the end of the page, under the following columns:

- **Date**: The date the event was logged

- **Source**: The server from which the log message originated

- **Severity**: The level of the log event

- **Message**: The message contained in the log event

2. You can filter the log messages that are displayed by typing a search string in the **Search** field and selecting any of the following search criteria:

- **Case sensitive**

- **Exact search**

- **Allow wildcards**: The following wild cards are supported:

  - * Any number of characters from 0 up

  - ? One character only

- **Run search continuously**: The system re-reads the log files every two seconds and turns off pagination.

You can search for specific logs based on date and time, using the formats for date and time used in the **log4j** configuration file.

You can also specify a severity to display all logs matching the specified severity level.

3. Click **Update**.

The filtered view you specified is now displayed.

Viewing Log Files from the File System

You can view the log files directly by connecting to the Bootstrap Blade to the following default location:

/var/ocsb/ocsb/logging_server/logs

Under this directory, the log files are split among sub-directories with names based on the Signaling Server or the Processing Server that generated the logs. For example, log messages generated by a Processing Server on Worker Blade 2 would appear under the directory named **managed_pn_0_1_0**.

See "Bootstrap Blades and Worker Blades" for information on identifying Worker Blade slots and servers.

You can also access the Bootstrap Blade using Secure Shell (SSH). See "Accessing the Bootstrap Blade Using SSH" for information.

Error Handling

If the Logging Server fails, the HA Manager process management system detects if the Logging Server terminates and attempts to restart it automatically. Alternatively, you can restart the Logging Server using the System Administration Console.

If a hard drive or system error jeopardizes the ability of the Logging Server to handle the logs, an ERROR log message is printed to **System.err**.

Archiving Log Files

When the log files stored in the system exceed the size configured for each Signaling Server and Processing Server, the older files are deleted. A default aggregate of 90 MBs of the most current log events per server are stored at any time.

If you want to back up older log files destined for deletion, you can use any of the standard archiving methods supported by your individual system.

9

Backing Up Files

This chapter describes backing up files belonging to Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager).

About Backing Up Files

The breakdown of any aspect of an HA Manager deployment can quickly lead to failure in other parts of the system. To minimize the impact of failure on any part of the system, it is important to back up all system files and all configuration files and store them in a location from where they can be easily restored.

The method used to back up and, if necessary, restore these files is left to your discretion.

Oracle recommends backing up the files under the following directories on the Bootstrap Blades:

- **/var/ocsb/ocsb**

  Bootstrap Blade disk storage

- **/var/ocsb/servers**

  Worker Blade boot images

10

Managing Alarms

This chapter describes how you can use the Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) Administration Console to view the notifications that Runtime MBeans generate.

About Alarms

The HA Manager provides a graphical user interface to view JMX notifications, known as alarms. Alarms report changes to the following components:

- HA Manager software, such as crossing the upper threshold that you specified for a counter attribute of a Runtime MBean. See "About HA Manager Software Alarms" for more information.

- The hardware of an HA Manager deployment, such as adding a new hardware component. See "About HA Manager Deployment Hardware Alarms" for more information.

Each alarm provides the following information:

- The HA Manager component, such as a Managed Server or a slot in which an alarm occurred

- The object name of the Runtime MBean that generated the notification

- The time when the notification was generated

- The notification message

> **Note:** You must configure the HA Manager notification mechanism to view alarms. For more information about configuring software notifications, see the *Understanding Notifications* section in the *Monitoring Service Broker* chapter in the *Oracle Communications Service Broker System Administration Guide*.

About HA Manager Software Alarms

The Administration Console enables you to view alarms that report on the following events:

- A software Runtime MBean generated a notification on crossing the upper threshold

- A software Runtime MBean generated a notification on crossing the lower threshold

- A software Runtime MBean cleared a notification on crossing the upper threshold
- A software Runtime MBean cleared a notification on crossing the lower threshold

About HA Manager Deployment Hardware Alarms

The Administration Console enables you to view alarms that report the following events:

- A hardware Runtime MBean generated a notification about adding a hardware component to HA Manager.
- A hardware Runtime MBean generated a notification about removing a hardware component from HA Manager.
- A hardware Runtime MBean generated a notification about changing the availability status of a hardware component.
- A hardware Runtime MBean generated a notification about changing the operational status of a hardware component.
- A hardware Runtime MBean generated a notification about system errors.

About the Alarms Window

Figure 10–1 shows an example Alarms window. You to view and manage alarms using this window.

Figure 10–1   Alarms Window
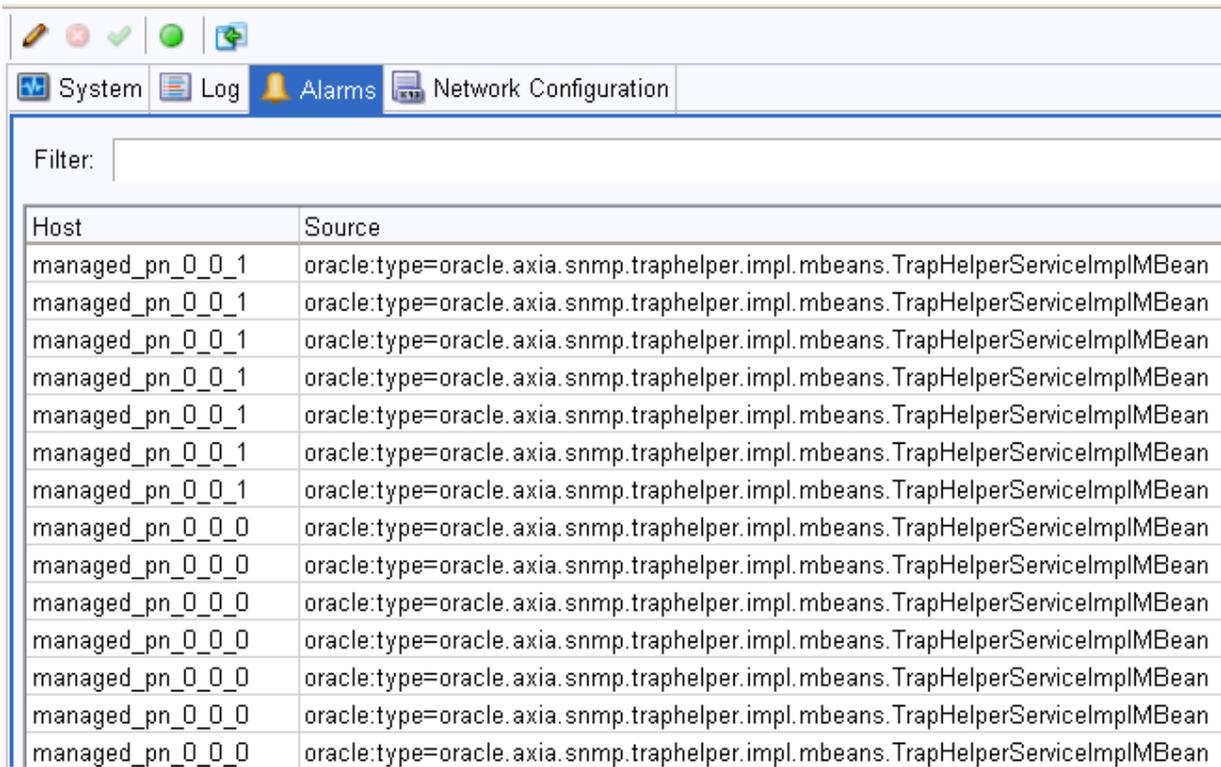
Table 10–1 describes the information displayed for each alarm.

Table 10–1   Alarm Window Columns and Fields

Column or Field	Description
Filter	Enables you to enter criteria to limit the results displayed. See "Searching Alarms" for more information.
Host	Specifies the component on which an alarm occurred:
	For software alarms, the Host field displays the name of a Managed Server on which an alarm occurred
	For hardware alarms, the Host field displays the number of the blade slot on which an alarm occurred
Source	Specifies the object name of the Runtime MBean that generated a notification
Time	Specifies the time when the notification was generated
Description	Displays a notification message

Searching Alarms

Depending on how you configured the notifications mechanism, the Alarms window can contain a significant number of alarms. To help you find specific alarms, HA Manager provides the following capabilities:

- Filtering Alarms
- Sorting Alarms

Filtering Alarms

You can narrow the list of alarms displayed in the Alarms window by specifying text that the alarms must include. For example, if you want to view only those alarms that were generated by a particular Runtime MBean, you can specify the object name of this Runtime MBean. HA Manager searches for this text in all fields and displays the alarms that contain the specified text at least in one of the fields.

The filter is not case-sensitive.

To filter alarms:

1. Click the **Alarms** tab.

    The Alarms window appears.

2. In the **Filter** text field, enter text to search for.

    Every time you modify the text in the **Filter** field (for example, enter or delete a character), HA Manager refreshes the list of alarms and displays only those alarms that contain the specified text.

    > **Note:**   Filtering alarms changes only change the view of alarms on a local computer. Filtering does not have any impact on how Runtime MBeans generate notifications or how HA Manager processes these notifications.

Sorting Alarms

You can sort alarms displayed in the Alarms window by any of the fields that appear on the window. For example, if you want to find all alarms generated by a specific

Runtime MBean, you can sort alarms using the **Source** field. In this case, all alarms are displayed according to their object name in alphabetic order. Similarly, you can sort alarms according to the time they were generated.

By default, alarms are sorted at the time they were received by HA Manager.

To resort alarms:

1. Click the **Alarms** tab.

   The Alarms window appears.

2. Select the header of a column in which you want to sort the alarms.

   Alarms are sorted accordingly.

Clearing the List of Alarms

You can clear the list of alarms if the list contains a significant number of alarms that are irrelevant to you.

This action cannot be undone.

To clear the list of alarms:

1. Click the **Alarms** tab.

   The Alarms window appears.

2. Click **Clear**.

   HA Manager deletes all alarms from the Alarms window.

Configuring the Alarms Display

HA Manager updates the Alarms window periodically with these newly generated notifications. The MBean notifications are stored permanently. HA Manager displays the newer alarms at the top of the Alarms window.

You can specify how frequently HA Manager refreshes the list of alarms.

In addition, you can define the maximum number of alarms to be displayed. If the number of generated alarms exceeds your maximum number, the older alarms are removed from the list.

To configure alarms display:

1. Click the **Alarms** tab.

   The Alarms window appears.

2. In the Alarms window, click **Settings**.

   The Settings dialog box appears.

3. Select one of the following **Refresh speed** options:

   - **Slow**: Every 3 seconds

   - **Normal**: Every 1.5 seconds

   - **Fast**: Every 0.5 second

4. In the **Buffer Size** text field, specify the number of alarms to display.

5. Click **OK**.

11

Upgrading Service Broker Netra 6000 High Availability Manager

This chapter describes how to upgrade Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) software components.

About Upgrading Service Broker Netra 600 High Availability Manager Software

You can upgrade software components using the Administration Console. You can update the following components:

- Deployment packages
- Individual bundles

This chapter explains how to upgrade deployment packages only. For information about upgrading individual bundles, see the discussion on managing bundles in the Administration Console in *Oracle Communications Service Broker System Administrator's Guide.*

> **Note:** Instructions on whether to upgrade bundles individually or to upload a new Deployment Package are provided together with an upgrade patch.

About Deployment Packages

To upgrade HA Manager components, the HA Manager implements Open Services Gateway initiative (OSGi) architecture. According to this architecture, the HA Manager software is modular, grouped into replaceable deployment packages.

Each deployment package contains one or more bundles that implement a certain functionality. You manage these bundles as a unit. For example, bundles that implement the Diameter SSU are grouped into a deployment package. This enables you to install Diameter SSU bundles together as a single unit.

Deploying a particular functionality might require you to install multiple deployment packages.

You can install and uninstall deployment packages using the Administration Console. See "Upgrading the HA Manager Software" for more information.

About Installing New Deployment Packages

To add functionality, you upload the Oracle deployment package named Full Package. Full Package contains a complete set of bundles required for the new functionality.

Figure 11–1 shows how installing a Full Package with three new bundles is added to the existing deployment package of four bundles. Following the upgrade, HA Manager supports seven bundles, numbered from 1 to 7.

Figure 11–1 Adding a New Full Package
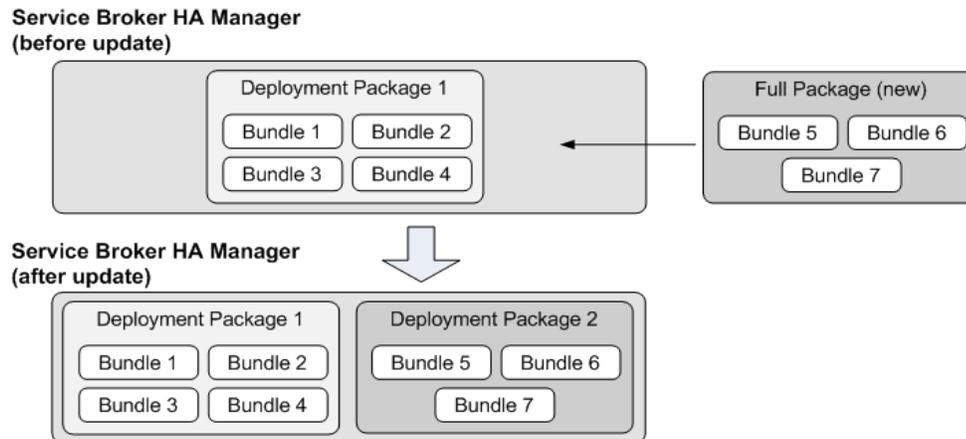
About Upgrading Existing Deployment Packages

You can upgrade deployment packages in the following ways:

- All bundles in an existing deployment package. See "About Upgrading Entire Deployment Packages" for more information.

- Specific bundles in an existing deployment package. See "About Upgrading Specific Bundles in a Deployment Package" for more information.

- New bundles to an existing deployment package. See "About Adding New Bundles to a Deployment Package" for more information.

About Upgrading Entire Deployment Packages

To upgrade all bundles in an existing deployment package, you upload the Full Package that contains new versions of all bundles that the existing deployment package contains.

Figure 11–2 shows how installing a Full Package with three new bundle versions replaces the existing deployment package with earlier-version bundles of the same name. Following the upgrade, HA Manager supports upgraded versions of existing bundles.

Figure 11–2   Upgrading an Entire Deployment Package
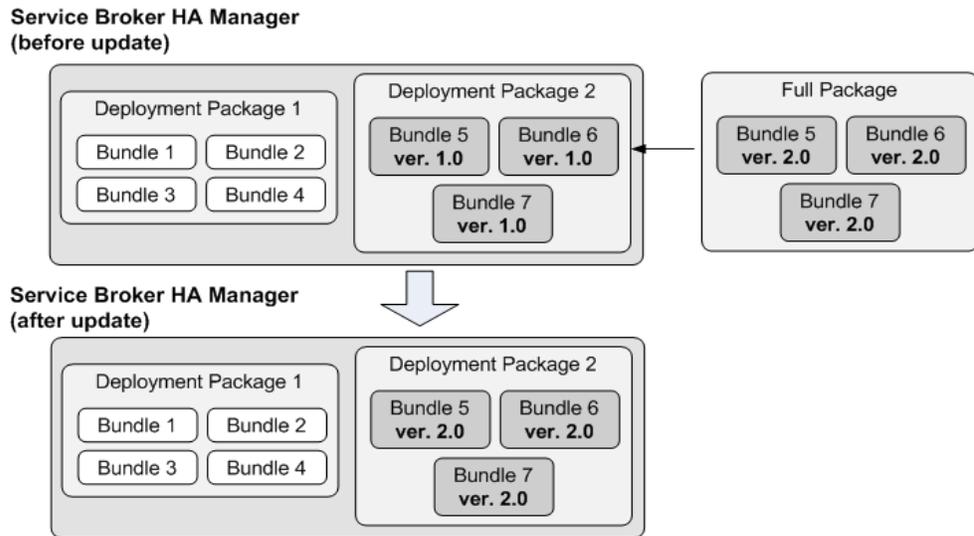
About Upgrading Specific Bundles in a Deployment Package

To upgrade specific bundles in an existing deployment package, you upload the deployment package named Fix Package. Fix Package contains only the upgraded bundles. When you upload Fix Package, bundles not included in the Fix Package remain intact in the original deployment package.

Figure 11–3 shows how installing a Fix Package with new bundle versions replaces existing deployment package with earlier-version bundles of the same name. If the Fix Package does not contain a bundle with the same name as an existing bundle, the bundle remains intact in the deployment package. Following the upgrade, HA Manager supports upgraded versions of specific bundles.

Figure 11–3   Upgrading Specific Bundles in an Existing Deployment Package
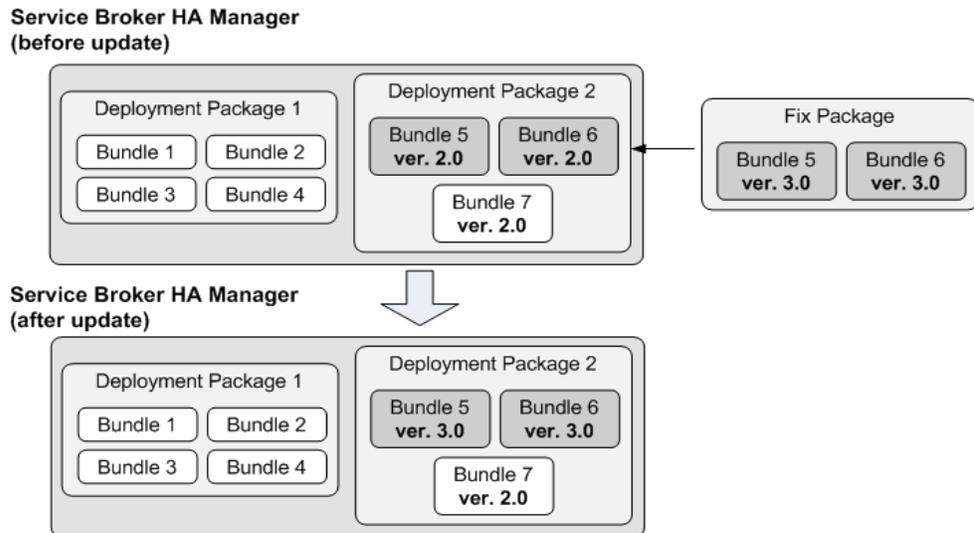
About Adding New Bundles to a Deployment Package

To add new bundles to an existing deployment package, upload the Fix Package that contains only new bundles to be added to an existing deployment package.

Figure 11–4 shows how installing a Fix Package adds two new bundles to the three bundles in the existing deployment package. Following the upgrade, HA Manager supports five bundles in the deployment package compared to three before the upgrade.

Figure 11–4   Adding New Bundles to an Existing Deployment Package
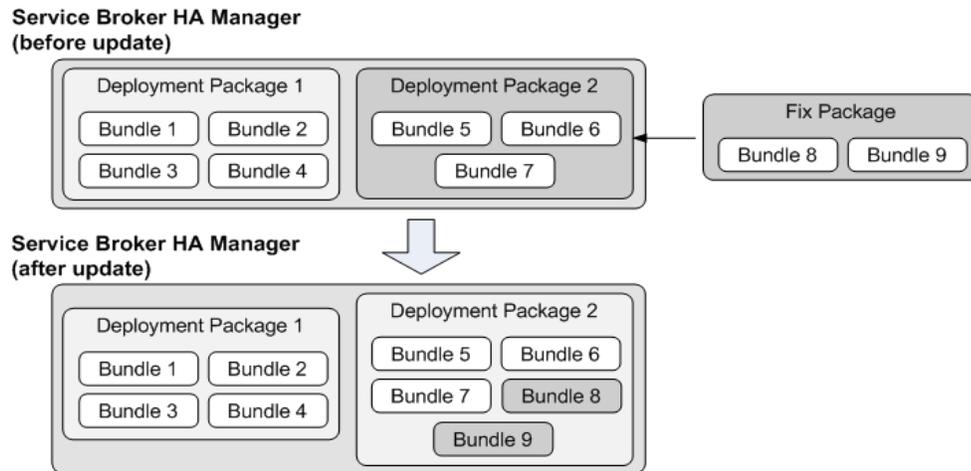
About the Upgrade Process

The deployment package upgrade process works as follows, regardless of the upgrade type:

1. You upload a deployment package to HA Manager using the Administration Console.

2. HA Manager installs the deployment package on Signaling Servers or Processing Servers in the domain.

3. You restart each Signaling Server and Processing Server in the domain for the new deployment package to take effect.

About the Managed Upgrade Window

Figure 11–5 shows the Managed Upgrade window, which you use to upgrade deployment packages.

Figure 11–5   Managed Upgrade Window
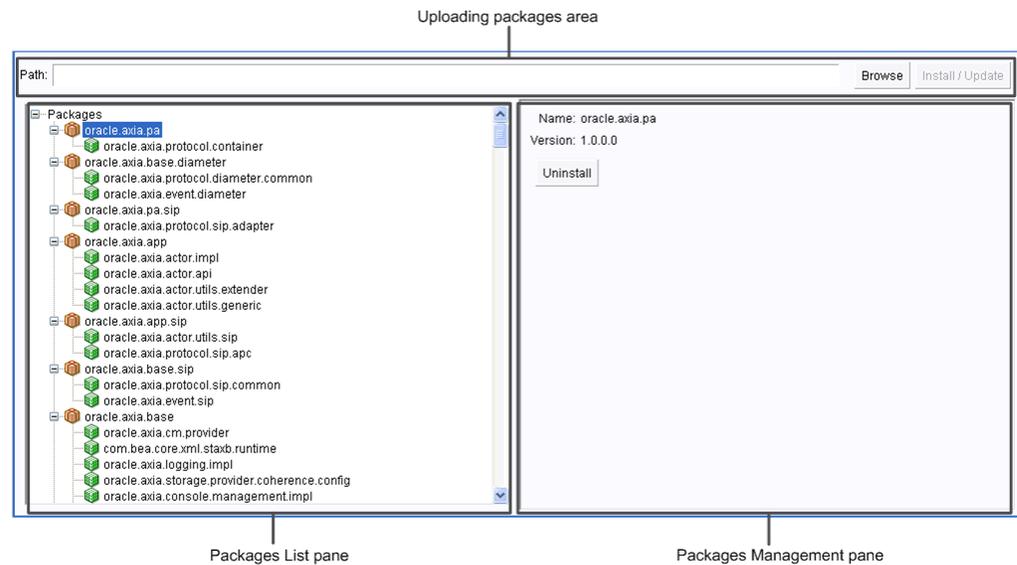
The Managed Upgrade window consists of the following areas:

- Packages List pane: Displays the **Packages** navigation tree which lists the installed packages.

- Package Management pane: Displays the name and version of the bundle or package selected in the Packages List pane.

- Uploading Packages area: Use this field to locate a package on a local drive and install the package on HA Manager.

Upgrading the HA Manager Software

You can manage HA Manager software upgrades by performing the following actions:

- Installing a Deployment Package
- Uninstalling a Deployment Package

Installing a Deployment Package

You can install new deployment packages and upgrade existing deployment packages by uploading packages to HA Manager using the System Administration Console. The procedure for installing Full Packages and Fix Packages is the same. After a package is installed, you restart the servers for the installation to take effect.

To install a deployment package:

1. On the System Administration Console toolbar, click **Lock & Edit**.

   You can now make changes in HA Manager configuration settings.

2. On the toolbar, click **Switch to Offline Mode** and then click **OK** in the confirmation window.

   Note: You can install deployment packages in offline mode only.

   ---

3. Click the **Managed Upgrade** tab.

4. Click **Browse**.

   The Upload File dialog box appears.

5. Click **Browse**.

   The File Upload dialog box appears.

6. Navigate to the directory that contains the deployment package you want to install and click **Open**.

7. Click **Upload**.

   A message confirming that the package was successfully uploaded appears.

8. Click **OK**.

9. In the Managed Upgrade window, click **Install / Update** to install the package that you uploaded.

   HA Manager installs the deployment package on the servers.

10. Restart the Signaling Servers and Processing Servers for the uploaded deployment package to take effect.

    See "Managing and Monitoring Hardware and Processes" for information about restarting servers.

Upgrading adds or updates packages based on the type of package you uploaded:

- If you installed a Full Package, and a Deployment Package with the same name does not exist, the Deployment Package is installed as a new package.

- If you installed a Full Package, and a Deployment Package with the same name already exists, all bundles of the existing package are replaced with the bundles from the uploaded Full Package.

- If you installed a Fix Package, existing bundles are updated or new bundles are added to an existing package.

See "About Installing New Deployment Packages" and "About Upgrading Existing Deployment Packages" for more information about types of Deployment Package updates.

Uninstalling a Deployment Package

You can uninstall existing deployment packages using the Administration Console. After a package is uninstalled, you must restart servers for the new configuration to take effect.

To uninstall a deployment package:

1. On the Administration Console toolbar, click **Lock & Edit**.

   You can now make changes to HA Manager configuration settings.

2. On the toolbar, click to **Switch to Offline Mode** and then click **OK** in the confirmation window.

   Note: You can uninstall Deployment Packages in offline mode only.

   ---

3. In the Packages navigation tree, select a package to uninstall.

> **Note:** You can use the Managed Upgrade window to uninstall packages only. You cannot uninstall individual bundles. For instructions on uninstalling individual bundles, see the discussion on managing domain bundles in *Oracle Communications Service Broker System Administrator's Guide*.

4. In the Package Management pane, click **Uninstall**.

   The Uninstall Package progress bar appears. After HA Manager uninstalls the package from the Signaling Servers and Processing Servers the Packages tree is refreshed to display the updated list of packages.

5. Restart the Signaling Servers and Processing Servers for the new configuration to take effect.

   See "Managing and Monitoring Hardware and Processes" for information about restarting servers.

Handling Errors During Installation or Uninstallation

If installation or uninstallation fails (for example, when a server on which HA Manager tries to install a deployment package is unavailable), HA Manager displays an error message in the Administration Console and restores the previous version of the deployment package.

Upgrading the Operating System

You can upgrade the operating system on the Bootstrap Blades and the Worker Blades. To automate the upgrade process, Oracle provides upgrade patches that contain all files required for the upgrade:

- Red Hat Package Manager (RPM) files, which contain package files required to upgrade the operating system

- Upgrade script, which manages dependencies to ensure that the required files are installed

- Installation notes, which provide detailed guidelines on how to install the upgrade patch

> **Note:** During the upgrade process, the upgrade script merges existing RPM files with newer versions of these files. If you made any changes in the RPM files, the upgrade script might be unable to merge these files with their new versions. In this case, the script does the following:
>
> - For the new version of the RPM file, the script creates the file with the **rpmnew** extension
>
> - For the file modified by you, the script creates the file with the **rpmsave** extension
>
> Although the upgrade script can automatically detect these cases, you need to manually resolve the conflicts. For more information, see the installation notes provided with the upgrade patch.

Upgrading the Operating System on the Bootstrap Blades

You can upgrade the operating system only on secondary Bootstrap Blades. (See "Primary and Secondary Bootstrap Blades" for more information about primary and secondary Bootstrap Blades. After you complete the upgrade, you can switch the Bootstrap Blade from secondary to primary.

To upgrade the operating system on the Bootstrap Blades:

1. In a Linux command-line interface, log in to the server on which the Bootstrap Blades are installed.

2. Check the current status of the Bootstrap Blades by entering the following command in the shell:

   clustat

   The following information is returned:

   - IP addresses of the Bootstrap Blades that the cluster contains

   - ID and status of each Bootstrap Blade

   - Services on the Bootstrap Blades and the state of these services. The Bootstrap Blade that contains a service with a state of **Started** is the primary Bootstrap Blade.

   For example:

   ```
   [root@bootstrap1 ~]# clustat
   Cluster Status for ocsba @ Tue Dec 28 07:56:28 2010
   Member Status: Quorate
   Member Name                          ID   Status
   ------ ----                          ---- ------
   192.168.1.1                            1 Online, Local, rgmanager
   192.168.1.2                            2 Offline

   Service Name              Owner (Last)              State
   ------- ----              ----- ------              -----
   service:ip                192.168.1.1               started
   ```

3. Make the primary Bootstrap Blade the secondary blade by entering the following command:

   service **rgmanager restart**

   The *service* parameter specifies the name of the service that runs on the primary Bootstrap Blade. For example:

   ip rgmanager restart

   The primary Bootstrap Blade is now in the secondary state.

4. Unpack and run the upgrade patch that you received from Oracle, as described in the upgrade patch documentation.

Upgrading the Operating System on the Worker Blades

You can upgrade the operating system using the upgrade script that you received with the upgrade patch.

To upgrade the operating system on the Worker Blades:

1. Copy the RPM files that the upgrade package contains to the RPM repository at **/var/ocsb/ramdisk/rpm**.

2. Run the RAM disk upgrade script as described in the upgrade patch documentation.

   The script rebuilds the RAM disk image based on the RPM files that you copied to **/var/ocsb/ramdisk/rpm**.

3. Enter your root and ocsb password when prompted.

   The resulting RAM disk image is copied to the TFTP directory.

4. Reboot all Worker Blades as specified in the patch documentation.

   The Worker Blades load the new RAM disk image from the TFTP directory.

12

Replacing Worker Blades

This chapter describes how to identify and replace a failed Worker Blade in a deployment of Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager).

Identifying a Failed Worker Blade

Hardware issues in an HA Manager deployment might cause Worker Blade to fail. The following events might indicate that a Worker Blade has failed:

- The state of the Worker Blade changes to Powered Off.

- One or more processes that run on the Worker Blade fail.

These events alone do not necessarily mean that a Worker Blade failed. For example, a system administrator might intentionally change the state of a Worker Blade to Powered Off.

To ensure that switching to the Powered Off state or process failure occurred because of a failed Worker Blade, you must check the alarms that the HA Manager generates (see "Managing Alarms" for more details). When a Worker Blade fails, the HA Manager generates a notification, indicating that the **OperationalStatus** attribute of the hardware MBean which represents the blade, changed from **OK** to **Error**.

Replacing a Failed Worker Blade

Replacing a Worker Blade involves the following stages:

1. Powering off the failed Worker Blade, if necessary

2. Physical removal of the failed blade and insertion of the new blade

3. Configuration of the new blade using the Chassis Monitoring Module (CMM) Integrated Lights Out Manager (ILOM) Web interface

4. Powering on the new Worker Blade

To replace a failed Worker Blade:

1. If a Worker Blade failed but is still in the **Power On** state, shut down the failed Worker Blade by performing the steps a through d. Otherwise, proceed to step 2.

   a. In the CMM ILOM Web interface, in the chassis tree, select the failed Worker Blade.

   The **System Overview** tab appears.

   b. Click the **Remote Control** tab and then click the **Remote Power Control** tab.

The **Remote Power Control** tab shows the list from which you can select a state to which you want to switch the specified Worker Blade.

    c. From the list, select the **Graceful Shutdown and Power Off** option.

    d. Click **Save**.

    The failed Worker Blade is now shutting down.

2. Remove the failed Worker Blade from the chassis slot.

3. Insert a new Worker Blade into the same chassis slot.

4. In the CMM ILOM Web interface, in the chassis tree, select the newly inserted Worker Blade.

The **System Overview** tab appears.

5. Click the **Configuration** tab and then click the **Network** tab.

The **Network** tab enables you to configure the MAC address and network settings for the new Worker Blade.

6. Configure the new Worker Blade using the same settings that were used to configure the failed blade.

> **Note:** You must keep a record of the network settings for all Worker Blades. See "Keeping a Record of Network Settings for Worker Blades" for more details.

7. Click **Save**.

8. Click the **Remote Control** tab and then click the **Remote Power Control** tab.

The **Remote Power Control** tab shows the list from which you can select a state to which you want to switch the new Worker Blade.

9. From the list, select the **Power On** option.

10. Click **Save**.

The new Worker Blade is now functional.

A

Component List

This section describes the hardware and software components of an Oracle Communications Service Broker Netra 6000 High Availability Manager (HA Manager) deployment.

Hardware

The HA Manager operates with the following key hardware components:

- Sun Netra 6000 Modular System chassis, either AC or DC version (1)
- Sun Netra X6270 M2 Server Modules blades (4 to 10)
- Sun Blade 6000 Ethernet Switched NEM 24p 10 GbE (2)
- Chassis Monitoring Module (CMM) (1)

The following components are required on each blade:

- Sun Dual 10 GbE PCIe 2.0 FEM (1)
- Intel Xeon Model L5638, 2.00 GHz, Hex-core, 12 MB Cache (2)
- 4GB Memory, DDR3 1333 MHz (12)

The following additional components are required on each Bootstrap Blade:

- 300 GB 10K RPM 2.5" SAS HDD (2)
- RAID Expansion Module

To connect a chassis to an external network or to another chassis, Oracle recommends that you use an external switch, such as the Sun Network 10 GbE Switch 72p Top Of Rack switch.

When deploying multi-chassis, each Top Of Rack switch must be connected to all chassis in the deployment.

Software

The HA Manager includes the following software components:

- Oracle Communications Service Broker Netra 6000 High Availability Manager Release 6.0
- Oracle JRockit R 28.1.4

The HA Manager is engineered to work with:

- Oracle Enterprise Linux Release 5.5

- DRBD 8.3.7

B

Netra HA Manager MBean Hardware Reference

This appendix provides details on the HA Manager hardware configuration MBeans.

Hardware MBean Reference

The following sections provide reference information on Netra hardware MBeans displayed by the HA Manager.

See "Netra HA Manager MBean Hardware Reference" for information on how to manage the HA Manager hardware using these MBeans.

In this reference, the name of an MBean indicates the location of the MBean in the hierarchy as follows:

- The name contains all ancestors of the MBean separated by a full stop (.).

- Ancestors are listed from left to right, from the top-level node to the bottom-level node

- When a hardware element has multiple instances of an MBean, the name of the MBean contains the instance order number and instance ID separated by a semicolon.

    For example: `BladeCardLogDev-i;id-j`

For example, the MBean SUN_NetraLogicalDevice.SUN_NetraCard."BladeCardLogDev-0;id-1".N6000 is described as SUN_NetraLogicalDevice.SUN_NetraCard."BladeCardLogDev-i;id-j".N6000.

Netra HA Manager MBeans

The following sections list the Netra HA Manager MBeans and their attributes and operations.

SUN_NetraLogicalDevice.SUN_NetraCard."BladeCardLogDev-i;id-j".N6000

This MBean defines the logical part of the card. The logical part contains dynamic information about the card (for example, availability states), and operations (for example, powering on the card) that can be performed on the card.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

SystemCreationClassName

Specifies the scoping CreationClassName of the system.

SystemName

Specifies the scoping name of the system.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

DeviceID

Specifies the address or other data that uniquely identifies the device.

Availability

Specifies the availability status of the device.

Table B–1 describes values to which the Availability attribute can be set.

Table B–1   Availability Allowed Values

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded
11	Not Installed
12	Install Error
13	Power Save - Unknown

Table B–1   (Cont.) Availability Allowed Values

Value	Description
14	Power Save - Low Power Mode
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning
18	Paused
29	Not Ready
20	Not Configured
21	Quiesced

TotalPowerOnHours

Specifies the total number of hours that the device is in the Power On state.

Name

Specifies the name of the device.

RequestedState

Specifies the last requested state for the device.

Table B–2 describes values to which the RequestedState attribute can be set.

Table B–2   RequestedState Allowed Values

Value	Description
0	Unknown
2	Enabled
3	Disabled
4	Shut Down
5	No Change. Deprecated. Use the value 0 ("Unknown") instead.
6	Offline
7	Test
8	Deferred
9	Quiesce
10	Reboot
11	Reset
12	Not Applicable
..	DMTF Reserved
32768..65535	Vendor Reserved

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–3 describes values to which the OperationalStatus attribute can be set.

Table B–3   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

NetraSetPowerState()

Sets the power state of the device.

Table B–4 describes values to which the PowerState parameter can be set.

Table B–4   PowerState Allowed Values

Parameter	Description
0	Power Off
1	Power On
2	Power Cycle

Reset()

Resets the device.

Table B–5 describes return values of the operation.

Table B–5   Reset() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

SaveProperties()

Saves the current configuration of the device. To restore the configuration, use the RestoreProperties() operation. Not all the devices support this operation.

Table B–6 describes return values of the operation.

Table B–6   SaveProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RestoreProperties()

Restores the previously saved configuration of the device. See "SaveProperties()" for more information).

Table B–7 describes return values of the operation.

Table B–7   RestoreProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RequestStateChange()

Changes the state of the element as specified in the RequestedState parameter.

Table B–8 describes return values of the operation.

Table B–8   RequestStateChange() Return Values

Value	Description
0	Completed with No Error
1	Not Supported
2	Unknown or Unspecified Error
3	Cannot complete within Timeout Period
4	Failed
5	Invalid Parameter
6	In Use
..	DMTF Reserved
4096	Method Parameters Checked - Job Started
4097	Invalid State Transition
4098	Use of Timeout Parameter Not Supported
4099	Busy
4100..32767	Method Reserved
32768..65535	Vendor Specific

Deprecated Attributes and Operations

The following attributes and operations are deprecated:

- PowerManagementSupported
- PowerManagementCapabilities
- MaxQuiesceTime
- Status

  Use the OperationalStatus attribute instead.

- StatusInfo

  Use the EnabledState attribute instead.

- SetPowerState()
- EnableDevice()

  Use the RequestStateChange() operation instead.

- OnlineDevice()

  Use the RequestStateChange() operation instead.

- QuisceDevice()

  Use the RequestStateChange() operation instead.

Non-Supported Attributes

The following attributes are currently not supported:

- LastErrorCode
- ErrorDescription

- ErrorCleared
- OtherIdentifyingInfo
- PowerOnHours
- IdentifyingDescriptions
- AdditionalAvailability
- EnabledState
- OtherEnabledState
- EnabledDefault
- TimeOfLastStateChange
- AvailableRequestedStates
- TransitioningToState
- InstallDate
- StatusDescriptions
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraLogicalDevice.SUN_NetraCard.""CmmCardLogDev--i;id-j".N6000

The attributes and operations of this MBean are identical to SUN_NetraCard."BladeCardLogDev-i;id-j".N6000. See "SUN_NetraCard."BladeCardLogDev-i;id-j".N6000" for more information.

SUN_NetraLogicalDevice.SUN_NetraCard."NemCardLogDev-i;id-j".N6000

The attributes and operations of this MBean are identical to SUN_ NetraCard."BladeCardLogDev-i;id-j".N6000. See "SUN_ NetraCard."BladeCardLogDev-i;id-j".N6000" for more information.

SUN_NetraLogicalDevice.SUN_NetraChassis."ChassisLogDev--ID;id-j".N6000

The attributes and operations of this MBean are identical to SUN_
NetraCard."BladeCardLogDev-i;id-j".N6000. See "SUN_
NetraCard."BladeCardLogDev-i;id-j".N6000" for more information.

SUN_NetraLogicalDevice.SUN_NetraPhysicalPackage."FemLogDev"--i;id-j".N6000

The attributes and operations of this MBean are identical to SUN_
NetraCard."BladeCardLogDev-i;id-j".N6000. See "SUN_
NetraCard."BladeCardLogDev-i;id-j".N6000" for more information.

SUN_NetraNetworkPort."SUN_NetraNetworkPort"."FEM/NETi-j;id-k".N6000

This MBean defines a network port in the system.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

PortNumber

NetworkPorts are often numbered relative to either a logical module or a network element.

LinkTechnology

Specifies the type of a link.

Table B–9 describes values to which the LinkTechnology attribute can be set.

Table B–9   LinkTechnology Allowed Values

Value	Description
0	Unknown
1	Other
2	Ethernet
3	IB
4	FC
5	FDDI
6	ATM
7	Token Ring
8	Frame Relay
9	Infrared
10	BlueTooth
11	Wireless LAN

PermanentAddress

Specifies the hard-coded network address of the port.

NetworkAddresses

Specifies an array of strings that define the network addresses for the port.

FullDuplex

Specifies whether or not the port operates in full duplex mode.

PortType

Specifies the type of the port.

Table B–10 describes values to which the PortType attribute can be set.

Table B–10   PortType Allowed Values

Value	Description
0	Unknown
1	Other
2	Not Applicable
3..15999	DMTF Reserved
16000..65535	Vendor Reserved

SystemCreationClassName

Specifies the scoping CreationClassName of the system.

SystemName

Specifies the scoping name of the system.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

DeviceID

Specifies the address or other data that uniquely identifies the device.

Availability

Specifies the availability status of the device.

Table B–11 describes values to which the Availability attribute can be set.

Table B–11   Availability Allowed Values

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded
11	Not Installed
12	Install Error

Table B–11   (Cont.) Availability Allowed Values

Value	Description
13	Power Save - Unknown
14	Power Save - Low Power Mode
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning
18	Paused
29	Not Ready
20	Not Configured
21	Quiesced

TotalPowerOnHours

Specifies the total number of hours that the device is in the Power On state.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–12 describes values to which the OperationalStatus attribute can be set.

Table B–12   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted

Table B–12   (Cont.) OperationalStatus Allowed Values

Value	Description
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_ NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

Reset()

Resets the device.

Table B–13 describes return values of the operation.

Table B–13   Reset() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

SaveProperties()

Saves the current configuration of the device. To restore the configuration, use the RestoreProperties() operation. Not all devices support this operation.

Table B–14 describes return values of the operation.

Table B–14   SaveProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RestoreProperties()

Restores the previously saved configuration of the device. See "SaveProperties()" for more information).

Table B–15 describes return values of the operation.

Table B–15   RestoreProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RequestStateChange()

Changes the state of the element as specified in the RequestedState parameter.

Table B–16 describes return values of the operation.

Table B–16   RequestStateChange() Return Values

Value	Description
0	Completed with No Error
1	Not Supported
2	Unknown or Unspecified Error
3	Cannot complete within Timeout Period
4	Failed
5	Invalid Parameter
6	In Use
..	DMTF Reserved
4096	Method Parameters Checked - Job Started
4097	Invalid State Transition
4098	Use of Timeout Parameter Not Supported
4099	Busy
4100..32767	Method Reserved
32768..65535	Vendor Specific

Deprecated Attributes and Operations

The following attributes and operations are deprecated:

- PowerManagementSupported

- PowerManagementCapabilities

- MaxQuiesceTime

- Status

   Use the OperationalStatus attribute instead.

- SetPowerState()

- EnableDevice()

  Use the RequestStateChange() operation instead.

- OnlineDevice()

  Use the RequestStateChange() operation instead.

- QuisceDevice()

  Use the RequestStateChange() operation instead.

- OtherNetworkPortType

  Use the PortType attribute instead.

Non-Supported Attributes

The following attributes are currently not supported:

- Speed
- OtherLinkTechnology
- AutoSense
- SupportedMaximumTransmissionUnit
- ActiveMaximumTransmissionUnit
- MaxSpeed
- RequestedSpeed
- UsageRestriction
- OtherPortType
- StatusInfo
- LastErrorCode
- ErrorDescription
- ErrorCleared
- OtherIdentifyingInfo
- PowerOnHours
- IdentifyingDescriptions
- AdditionalAvailability
- EnabledState
- OtherEnabledState
- RequestedState
- EnabledDefault
- TimeOfLastStateChange
- AvailableRequestedStates
- TransitioningToState
- InstallDate
- StatusDescriptions
- HealthState

- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraNetworkPort >"SUN_NetraNetworkPort">"NETi-j;id-k">N6000

The attributes and operations of this MBean are identical to SUN_
NetraNetworkPort."SUN_NetraNetworkPort"."FEM/NETi-j;id-k".N6000. See "SUN_
NetraNetworkPort."SUN_NetraNetworkPort"."FEM/NETi-j;id-k".N6000" for more
information.

SUN_NetraSensor."SUN_NetraSensor"."ChassisTempSensor-i;id-j".N6000

This MBean defines a sensor that measures or reports characteristics of various physical properties, such as temperature or voltage.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

RelativeLocation

Specifies the number of the slot into which the card is inserted.

SensorType

Specifies the type of the sensor.

Table B–17 describes values to which the SensorType attribute can be set.

Table B–17   SensorType Allowed Values

Value	Description
0	Unknown
1	Other
2	Temperature
3	Voltage
4	Current
5	Tachometer
6	Counter
7	Switch
8	Lock
9	Humidity
10	Smoke Detection
11	Presence
12	Air Flow
13	Power Consumption
14	Power Production
15	Pressure
..	DMTF Reserved
32768..65535	Vendor Reserved

CurrentState

Specifies the current state of the sensor. This is one of the values specified in the PossibleStates attribute.

SystemCreationClassName

Specifies the scoping CreationClassName of the system.

SystemName

Specifies the scoping name of the system.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

DeviceID

Specifies the address or other data that uniquely identifies the device.

Availability

Specifies the availability status of the device.

Table B–18 describes values to which the Availability attribute can be set.

Table B–18   Availability Allowed Values

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded
11	Not Installed
12	Install Error
13	Power Save - Unknown
14	Power Save - Low Power Mode
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning
18	Paused
29	Not Ready
20	Not Configured
21	Quiesced

TotalPowerOnHours

Specifies the total number of hours that the device is in the Power On state.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–19 describes values to which the OperationalStatus attribute can be set.

Table B–19   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: {SUN_NetraLogicalDevice}{SUN_
NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}

Reset()

Resets the device.

Table B–20 describes return values of the operation.

Table B–20   Reset() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

SaveProperties()

Saves the current configuration of the device. To restore the configuration, use the RestoreProperties() operation. Not all the devices support this operation.

Table B–21 describes return values of the operation.

Table B–21   SaveProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RestoreProperties()

Restores the previously saved configuration of the device. See "SaveProperties()" for more information).

Table B–22 describes return values of the operation.

Table B–22   RestoreProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RequestStateChange()

Changes the state of the element as specified in the RequestedState parameter.

Table B–23 describes return values of the operation.

Table B–23   RequestStateChange() Return Values

Value	Description
0	Completed with No Error
1	Not Supported

Table B–23   (Cont.) RequestStateChange() Return Values

Value	Description
2	Unknown or Unspecified Error
3	Cannot complete within Timeout Period
4	Failed
5	Invalid Parameter
6	In Use
..	DMTF Reserved
4096	Method Parameters Checked - Job Started
4097	Invalid State Transition
4098	Use of Timeout Parameter Not Supported
4099	Busy
4100..32767	Method Reserved
32768..65535	Vendor Specific

Deprecated Attributes and Operations

The following attributes and operations are deprecated:

- PowerManagementSupported

- PowerManagementCapabilities

- MaxQuiesceTime

- Status

    Use the OperationalStatus attribute instead.

- EnableDevice()

- OnlineDevice()

- QuisceDevice()

- SetPowerState

Non-Supported Attributes

The following attributes are currently not supported:

- OtherSensorTypeDescription

- PossibleStates

- PollingInterval

- StatusInfo

- LastErrorCode

- ErrorDescription

- ErrorCleared

- OtherIdentifyingInfo
- PowerOnHours
- IdentifyingDescriptions
- AdditionalAvailability
- EnabledState
- OtherEnabledState
- RequestedState
- EnabledDefault
- TimeOfLastStateChange
- AvailableRequestedStates
- TransitioningToState
- InstallDate
- StatusDescriptions
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k"

This MBean defines a cooling device.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

RelativeLocation

Specifies the number of a slot into which the blade card is inserted.

PhysicalPackageLogicalDevice

Specifies the Logical part of the PhysicaPackage.

RemovalConditions

Specifies the conditions when a PhysicalPackage can be removed.

Table B–24 describes values to which the RemovalConditions attribute can be set.

Table B–24   RemovalConditions Possible Values

Value	Description
0	Unknown
2	Not Applicable
3	Removable when off
4	Removable when on or off

Height

Specifies the height of the PhysicalPackage in inches.

Depth

Specifies the depth of the PhysicalPackage in inches.

Width

Specifies the width of the PhysicalPackage in inches.

Weight

Specifies the weight of the PhysicalPackage in pounds.

PackageType

Specifies the type of the PhysicalPackage.

Table B–25 describes values to which the PackageType attribute can be set.

Table B–25   PackageType Allowed Values

Value	Description
0	Unknown
1	Other, which means the package type does not correspond to an existing enumerated value
2	Rack (defined per the Entity-MIB)
3	Chassis/Frame (defined per the Entity-MIB)
4	Cross Connect/Backplane (defined per the Entity-MIB)
5	Container/Frame Slot (defined per the Entity-MIB)
6	Power Supply (defined per the Entity-MIB)
7	Fan (defined per the Entity-MIB)
8	Sensor (defined per the Entity-MIB)
9	Module/Card (defined per the Entity-MIB)
10	Port/Connector (defined per the Entity-MIB)
11	Battery
12	Processor
13	Memory
14	Power Source/Generator
15	Storage Media Package
16	Blade
17	Blade Expansion

OtherPackageType

Specifies a description of the package when the PackageType attribute is set to 1 ("Other").

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_ NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

Tag

Specifies an arbitrary string that uniquely identifies the element and serves as the key of the element. This attribute can contain information, such as asset tag or serial number data.

CreationClassName

Specifies the scoping CreationClassName of the system.

Manufacturer

Specifies the name of the manufacturer that produced the element.

Model

Specifies the name by which the element is known.

SKU

Specifies the stock-keeping unit number for this element.

SerialNumber

Specifies the serial number assigned by the manufacturer to the element.

Version

Specifies the version of the element.

PartNumber

Specifies the part number assigned by the manufacturer to the element.

CanBeFRUed

Specifies whether or not the device can be removed and replaced without affecting other devices.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–26 describes values to which the OperationalStatus attribute can be set.

Table B–26   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service

Table B–26 (Cont.) OperationalStatus Allowed Values

Value	Description
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the object.

Deprecated Attributes

The following attributes are deprecated:

- Removable

  Use RemovalConditions instead

- Replaceable

- HotSwappable

  Use RemovalConditions instead

- IsCompatible

Non-Supported Attributes

The following attributes are currently not supported:

- VendorCompatibilityStrings
- OtherIdentifyingInfo
- PoweredOn
- ManufactureDate
- VendorEquipmentType
- UserTracking
- InstallDate
- StatusDescriptions
- Status
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus

- PrimaryStatus
- InstanceID

SUN_NetraPhysicalPackage."MIDPLANE-;id-i"

The attributes and operations of this MBean are identical to SUN_ NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k". See "SUN_ NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k"" for more information.

SUN_NetraPhysicalPackage."PS-i;id-j"

The attributes and operations of this MBean are identical to SUN_
NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k". See "SUN_
NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k"" for more information.

SUN_NetraPhysicalPackage."RFEM-i;id-j"

The attributes and operations of this MBean are identical to SUN_
NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k". See "SUN_
NetraPhysicalPackage."COOLING_DEVICE-i-j;id-k"" for more information.

SUN_NetraPowerSupply."SUN_ NetraPhysicalPackage"."PowerSupplyLogDev-i;id-j".N6000

This MBean defines the power supply device.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

SystemCreationClassName

Specifies the scoping CreationClassName of the system.

SystemName

Specifies the scoping name of the system.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

DeviceID

Specifies the address or other data that uniquely identifies the device.

Availability

Specifies the availability status of the device.

Table B–27 describes values to which the Availability attribute can be set.

Table B–27   Availability Allowed Values

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded
11	Not Installed
12	Install Error
13	Power Save - Unknown

Table B–27   (Cont.) Availability Allowed Values

Value	Description
14	Power Save - Low Power Mode
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning
18	Paused
29	Not Ready
20	Not Configured
21	Quiesced

TotalPowerOnHours

Specifies the total number of hours that the device is in the Power On state.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–28 describes values to which the OperationalStatus attribute can be set.

Table B–28   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode

Table B–28 (Cont.) OperationalStatus Allowed Values

Value	Description
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

NetraSetPowerState()

Sets the power state of the device.

Table B–29 shows possible values of the PowerState parameter.

Table B–29 PowerState Possible Values

Parameter	Description
0	Power Off
1	Power On
2	Power Cycle

Reset()

Resets the device.

Table B–30 describes return values of the operation.

Table B–30 Reset() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

SaveProperties()

Saves the current configuration of the device. To restore the configuration, use the RestoreProperties() operation. Not all the devices support this operation.

Table B–31 describes return values of the operation.

Table B–31   SaveProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RestoreProperties()

Restores the previously saved configuration of the device. See "SaveProperties()" for more information).

Table B–32 describes return values of the operation.

Table B–32   RestoreProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RequestStateChange()

Changes the state of the element as specified in the RequestedState parameter.

Table B–33 describes return values of the operation.

Table B–33   RequestStateChange() Return Values

Value	Description
0	Completed with No Error
1	Not Supported
2	Unknown or Unspecified Error
3	Cannot complete within Timeout Period
4	Failed
5	Invalid Parameter
6	In Use
..	DMTF Reserved
4096	Method Parameters Checked - Job Started
4097	Invalid State Transition
4098	Use of Timeout Parameter Not Supported
4099	Busy
4100..32767	Method Reserved
32768..65535	Vendor Specific

Deprecated Attributes and Operations

The following attributes and operations are deprecated:

- PowerManagementSupported

- PowerManagementCapabilities

- MaxQuiesceTime

- Status

  Use the OperationalStatus attribute instead.

- SetPowerState()

- EnableDevice()

  Use the RequestStateChange() operation instead.

- OnlineDevice()

  Use the RequestStateChange() operation instead.

- QuisceDevice()

  Use the RequestStateChange() operation instead.

Non-Supported Attributes

The following attributes are currently not supported:

- IsSwitchingSupply

- Range1InputVoltageLow

- Range1InputVoltageHigh

- Range1InputFrequencyLow

- Range1InputFrequencyHigh

- Range2InputVoltageLow

- Range2InputVoltageHigh

- Range2InputFrequencyLow

- Range2InputFrequencyHigh

- ActiveInputVoltage

- TypeOfRangeSwitching

- TotalOutputPower

- StatusInfo

- LastErrorCode

- ErrorDescription

- ErrorCleared

- OtherIdentifyingInfo

- PowerOnHours

- IdentifyingDescriptions

- AdditionalAvailability

- EnabledState

- OtherEnabledState

- RequestedState

- EnabledDefault

- TimeOfLastStateChange

- AvailableRequestedStates

- TransitioningToState

- InstallDate

- StatusDescriptions

- HealthState

- CommunicationStatus

- DetailedStatus

- OperatingStatus

- PrimaryStatus

- InstanceID

SUN_NetraSlot."BLi-j;id-k"

This MBean defines the slot device.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

SlotLogicalDevice

Specifies the name of the MBean that represents the slot.

SupportsHotPlug

Specifies whether or not the slot supports hot-plug of adapter cards.

Number

Specifies the physical slot number, which can be used as an index into a system slot table.

ConnectorElectricalCharacteristics

Specifies the electrical characteristic for this connector.

Table B–34 describes values to which the ConnectorElectricalCharacteristics attribute can be set.

Table B–34   ConnectorElectricalCharacteristics Allowed Values

Value	Description
0	Unknown
1	Other
2	Single Ended
3	Differential
4	Low Voltage Differential
5	Optical
6	Copper
7	Shielded
8	Unshielded

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_
NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

Tag

Specifies an arbitrary string that uniquely identifies the element and serves as the key of the element. This attribute can contain information, such as asset tag or serial number data.

CreationClassName

Specifies the scoping CreationClassName of the system.

Manufacturer

Specifies the name of the manufacturer that produced the element.

Model

Specifies the name by which the element is known.

SKU

Specifies the stock-keeping unit number for this element.

SerialNumber

Specifies the serial number assigned by the manufacturer to the element.

Version

Specifies the version of the element.

SystemName

Specifies the scoping name of the system.

PartNumber

Specifies the part number assigned by the manufacturer to the element.

CanBeFRUed

Specifies whether or not the device can be removed and replaced without affecting other devices.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–35 describes values to which the OperationalStatus attribute can be set.

Table B–35   OperationalStatus Allowed Values

Value	Description
0	Unknown

Table B–35   (Cont.) OperationalStatus Allowed Values

Value	Description
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Deprecated Attributes

The following attributes are deprecated:

■   Status

■   ConnectorType

Non-Supported Attributes and Operations

The following attributes are currently not supported:

■   PoweredOn

■   HeightAllowed

■   LengthAllowed

■   MaxDataWidth

■   VccMixedVoltageSupport

■   VppMixedVoltageSupport

■   ThermalRating

- SpecialPurpose
- PurposeDescription
- Powered
- OpenSwitch
- MaxLinkWidth
- VendorCompatibilityStrings
- ConnectorPinout
- OtherTypeDescription
- ConnectorGender
- OtherElectricalCharacteristics
- NumPhysicalPins
- ConnectorLayout
- ConnectorDescription
- OtherIdentifyingInfo
- ManufactureDate
- VendorEquipmentType
- UserTracking
- StatusDescriptions
- InstallDate
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraSlot."CMM-i;id-j"

The attributes and operations of this MBean are identical to SUN_ NetraSlot."BLi-j;id-k". See "SUN_NetraSlot."BLi-j;id-k"" for more information.

SUN_NetraSlot."COOLING_DEVICE_SLOT-i;id-j"

The attributes and operations of this MBean are identical to SUN_
NetraSlot."BLi-j;id-k". See "SUN_NetraSlot."BLi-j;id-k"" for more information.

SUN_NetraSlot."COOLING_DEVICE_SLOT-i;id-j"

The attributes and operations of this MBean are identical to SUN_
NetraSlot."BLi-j;id-k". See "SUN_NetraSlot."BLi-j;id-k"" for more information.

SUN_NetraCard."BLi-j;id-k"

This MBean defines the card PhysicalElement.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

RelativeLocation

Specifies the number of the slot into which the blade card is inserted.

CardLogicalDevice

Specifies the Logical part of the card.

HostingBoard

Specifies whether or not the card is a baseboard in the chassis.

RemovalConditions

Specifies the conditions when a PhysicalPackage can be removed.

Table B–36 describes values to which the RemovalConditions attribute can be set.

Table B–36   RemovalConditions Possible Values

Value	Description
0	Unknown
2	Not Applicable
3	Removable when off
4	Removable when on or off

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

Tag

Specifies an arbitrary string that uniquely identifies the element and serves as the key of the element. This attribute can contain information, such as asset tag or serial number data.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

SystemName

Specifies the scoping name of the system.

Manufacturer

Specifies the name of the manufacturer that produced the element.

Model

Specifies the name by which the element is known.

SKU

Specifies the stock-keeping unit number for this element.

SerialNumber

Specifies the serial number assigned by the manufacturer to the element.

Version

Specifies the version of the element.

PartNumber

Specifies the part number assigned by the manufacturer to the element.

CanBeFRUed

Specifies whether or not the device can be removed and replaced without affecting other devices.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–37 describes values to which the OperationalStatus attribute can be set.

Table B–37   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error

Table B–37   (Cont.) OperationalStatus Allowed Values

Value	Description
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

ConnectorPower()

Turns the power on and off for the connector on the card.

Deprecated Attributes

The following attributes are deprecated:

■   Removable

Use RemovalConditions instead

■   Replaceable

■   HotSwappable

Use RemovalConditions instead

■   IsCompatible

Non-Supported Attributes

The following attributes are currently not supported:

■   SlotLayout

■   RequiresDaughterBoard

■   SpecialRequirements

■   RequirementsDescription

■   OperatingVoltages

■   Height

- Depth
- Width
- Weight
- PackageType
- OtherPackageType
- VendorCompatibilityStrings
- OtherIdentifyingInfo
- PoweredOn
- OtherIdentifyingInfo
- PoweredOn
- ManufactureDate
- VendorEquipmentType
- UserTracking
- StatusDescriptions
- InstallDate
- Status
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID
- HealthState
- CommunicationStatus

SUN_NetraCard."CMMLi-j;id-k"

The attributes and operations of this MBean are identical to SUN_
NetraCard."BLi-j;id-k". See "SUN_NetraCard."BLi-j;id-k"" for more information.

SUN_NetraCard."NEMi-j;id-k"

The attributes and operations of this MBean are identical to SUN_ NetraCard."CMMLi-j;id-k". See "SUN_NetraCard."CMMLi-j;id-k"" for more information.

SUN_NetraChassis."/ChassisID;id-i"

This MBean defines the chassis element.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

ChassisLogicalDevice

Specifies the Logical part of the chassis.

ChassisPackageType

Specifies the physical form factor for the type of chassis.

Table B–38 describes values to which the ChassisPackageType attribute can be set.

Table B–38   ChassisPackageType Allowed Values

Value	Description
0	Unknown
1	Other
2	SMBIOS Reserved
3	Desktop
4	Low Profile Desktop
5	Pizza Box
6	Mini Tower
7	Tower
8	Portable
9	LapTop
10	Notebook
11	Hand Held
12	Docking Station
13	All in One
14	Sub Notebook
15	Space-Saving
16	Lunch Box
17	Main System Chassis
18	Expansion Chassis
19	SubChassis
20	Bus Expansion Chassis
21	Peripheral Chassis

Table B–38   (Cont.) ChassisPackageType Allowed Values

Value	Description
22	Storage Chassis
23	SMBIOS Reseved
24	Sealed-Case PC
25	SMBIOS Reserved
26	CompactPCI
27	AdvancedTCA
28	Blade Enclosure
..	DMTF Reserved
0x8000..0xFFFF	Vendor Reserved

ChassisTypeDescription

When the ChassisPackageType attribute is set to 1 ("Other"), the ChassisTypeDescription attribute provides a description of the chassis type.

MultipleSystemSupport

Specifies whether or not this chassis supports multiple systems, for example, server blades.

Table B–39 describes values to which the MultipleSystemSupport attribute can be set.

Table B–39   MultipleSystemSupport Allowed Values

Value	Description
0	Unknown
1	True
2	False

RackMountable

Specifies whether or not the chassis is rack mountable.

Table B–40 describes values to which the RackMountable attribute can be set.

Table B–40   RackMountable Possible Values

Value	Description
0	Unknown
1	True
2	False

RemovalConditions

Specifies the conditions when a PhysicalPackage can be removed.

Table B–41 describes values to which the RemovalConditions attribute can be set.

Table B–41   RemovalConditions Possible Values

Value	Description
0	Unknown
2	Not Applicable
3	Removable when off
4	Removable when on or off

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

Tag

Specifies an arbitrary string that uniquely identifies the element and serves as the key of the element. This attribute can contain information, such as asset tag or serial number data.

CreationClassName

Specifies the scoping CreationClassName of the system.

Manufacturer

Specifies the name of the manufacturer that produced the element.

Model

Specifies the name by which the element is known.

SKU

Specifies the stock-keeping unit number for this element.

SerialNumber

Specifies the serial number assigned by the manufacturer to the element.

Version

Specifies the version of the element.

PartNumber

Specifies the part number assigned by the manufacturer to the element.

CanBeFRUed

Specifies whether or not the device can be removed and replaced without affecting other devices.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–42 describes values to which the OperationalStatus attribute can be set.

Table B–42   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Deprecated Attributes

The following attributes are deprecated:

- ChassisTypes

  Use ChassisPackageType instead.

- TypeDescriptions

  Use ChassisPackageType instead.

- Removable

  Use RemovalConditions instead

- Replaceable

- HotSwappable

  Use RemovalConditions instead

- IsCompatible

Non-Supported Attributes

The following attributes are currently not supported:

- NumberOfPowerCords

- CurrentRequiredOrProduced

- HeatGeneration

- CableManagementStrategy

- ServicePhilosophy

- ServiceDescriptions

- LockPresent

- AudibleAlarm

- VisibleAlarm

- SecurityBreach

- BreachDescription

- IsLocked

- Height

- Depth

- Width

- Weight

- PackageType

- OtherPackageType

- VendorCompatibilityStrings

- OtherIdentifyingInfo

- PoweredOn

- ManufactureDate

- VendorEquipmentType

- UserTracking

- StatusDescriptions

- InstallDate

- Status

- HealthState

- CommunicationStatus

- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraIndicatorLED."SUN_NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000"

This MBean defines the LED indicator.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

RelativeLocation

Specifies the number of a slot into which the blade card is inserted.

ElementName

Specifies a user-friendly name for the element.

IndicatedConditions

Specifies the state that the LED indicator shows.

Table B–43 describes values to which the IndicatedConditions attribute can be set.

Table B–43   IndicatedConditions Allowed Values

Value	Description
0	Unknown
1	Other
2	Not Applicable
3	Location
4	Attention
5	Activity
6	Powered On
7	Fault
..	DMTF Reserved
32768..65535	Vendor Reserved

Color

Specifies the current color of the LED indicator.

Table B–44 describes values to which the Color attribute can be set.

Table B–44   Color Allowed Values

Value	Description
0	Unknown
1	Other

Table B–44   (Cont.) Color Allowed Values

Value	Description
2	Not Applicable
3	White
4	Red
5	Green
6	Blue
7	Orange
8	Yellow
9	Black
..	DMTF Reserved
32768..65535	Vendor Reserved

ControlMode

Specifies the current control mode of the LED indicator.

Table B–45 describes values to which the ControlMode attribute can be set.

Table B–45   ControlMode Allowed Values

Value	Description
2	Automatic
3	Manual
4	Test
..	DMTF Reserved
32768..65535	Vendor Reserved

DefaultActivationState

Specifies the default state of the LED indicator.

Table B–46 describes values to which the DefaultActivationState attribute can be set.

Table B–46   DefaultActivationState Allowed Values

Value	Description
2	Lit
3	Blinking
4	Off
5	Control Pattern
..	DMTF Reserved
32768..65535	Vendor Reserved

ActivationState

Specifies the current activity of the LED indicator.

Table B–47 describes values to which the ActivationState attribute can be set.

Table B–47   ActivationState Allowed Values

Value	Description
2	Lit
3	Blinking
4	Off
5	Control Pattern
..	DMTF Reserved
32768..65535	Vendor Reserved

SystemCreationClassName

Specifies the scoping CreationClassName of the system.

SystemName

Specifies the scoping name of the system.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

DeviceID

Specifies the address or other data that uniquely identifies the device.

Availability

Specifies the availability status of the device.

Table B–48 describes values to which the Availability attribute can be set.

Table B–48   Availability Allowed Values

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded
11	Not Installed
12	Install Error
13	Power Save - Unknown
14	Power Save - Low Power Mode

Table B–48   (Cont.) Availability Allowed Values

Value	Description
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning
18	Paused
29	Not Ready
20	Not Configured
21	Quiesced

TotalPowerOnHours

Specifies the total number of hours that the device is in the Power On state.

Name

Specifies the name of the device.

RequestedState

Specifies the last requested state for the device.

Table B–49 describes values to which the RequestedState attribute can be set.

Table B–49   RequestedState Allowed Values

Value	Description
0	Unknown This status indicates the last requested state for the element is unknown.
2	Enabled
3	Disabled
4	Shut Down
5	No Change. Deprecated. Use the value 0 ("Unknown") instead.
6	Offline
7	Test
8	Deferred
9	Quiesce
10	Reboot
11	Reset
12	Not Applicable
..	DMTF Reserved
32768..65535	Vendor Reserved

EnabledDefault

This attribute is not supported.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–50 describes values to which the OperationalStatus attribute can be set.

Table B–50   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Description

Specifies a textual description of the element.

Reset()

Resets the device.

Table B–51 describes return values of the operation.

Table B–51   Reset() Return Values

Value	Description
0	Request was successfully executed

Table B–51   (Cont.) Reset() Return Values

Value	Description
1	Request is not supported
other value	Error occurred while executing the request

SaveProperties()

Saves the current configuration of the device. To restore the configuration, use the RestoreProperties() operation. Not all devices support this operation.

Table B–52 describes return values of the operation.

Table B–52   SaveProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RestoreProperties()

Restores the previously saved configuration of the device. See "SaveProperties()" for more information).

Table B–53 describes return values of the operation.

Table B–53   RestoreProperties() Return Values

Value	Description
0	Request was successfully executed
1	Request is not supported
other value	Error occurred while executing the request

RequestStateChange()

Changes the state of the element as specified in the RequestedState parameter.

Table B–54 describes return values of the operation.

Table B–54   RequestStateChange() Return Values

Value	Description
0	Completed with No Error
1	Not Supported
2	Unknown or Unspecified Error
3	Cannot complete within Timeout Period
4	Failed
5	Invalid Parameter
6	In Use
..	DMTF Reserved

Table B–54   (Cont.) RequestStateChange() Return Values

Value	Description
4096	Method Parameters Checked - Job Started
4097	Invalid State Transition
4098	Use of Timeout Parameter Not Supported
4099	Busy
4100..32767	Method Reserved
32768..65535	Vendor Specific

Deprecated Attributes and Operations

The following attributes and operations are deprecated:

- PowerManagementSupported

- PowerManagementCapabilities

- MaxQuiesceTime

- Status

  Use the OperationalStatus attribute instead.

- SetPowerState()

- EnableDevice()

  Use the RequestStateChange() operation instead.

- OnlineDevice()

  Use the RequestStateChange() operation instead.

- QuisceDevice()

  Use the RequestStateChange() operation instead.

Non-Supported Attributes

The following attributes are currently not supported:

- OtherIndicatedConditionDescription

- OtherColorDescription

- ControlPattern

- StatusInfo

- LastErrorCode

- ErrorDescription

- ErrorCleared

- OtherIdentifyingInfo

- PowerOnHours

- IdentifyingDescriptions

- AdditionalAvailability

- EnabledState

- OtherEnabledState
- TimeOfLastStateChange
- AvailableRequestedStates
- TransitioningToState
- InstallDate
- StatusDescriptions
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraIndicatorLED."SUN_NetraIndicatorLED"."MajorAlarmIndicator-i;id-j"."N6000"

The attributes and operations of this MBean are identical to SUN_NetraIndicatorLED."SUN_NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000". See "SUN_NetraIndicatorLED."SUN_NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000"" for more information.

SUN_NetraIndicatorLED."SUN_ NetraIndicatorLED"."MinorAlarmIndicator-i;id-j"."N6000"

The attributes and operations of this MBean are identical to SUN_ NetraIndicatorLED."SUN_NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000". See "SUN_NetraIndicatorLED."SUN_ NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000"" for more information.

SUN_NetraIndicatorLED."SUN_NetraIndicatorLED"."UserAlarmIndicator-i;id-j"."N6000"

The attributes and operations of this MBean are identical to SUN_
NetraIndicatorLED."SUN_NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000".
See "SUN_NetraIndicatorLED."SUN_
NetraIndicatorLED"."CriticalAlarmIndicator-i;id-j".N6000"" for more information.

SUN_NetraLog.AlarmLog;id-i

This MBean defines the log.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

MaxNumberOfRecords

Specifies the maximum number of records that the log can capture.

CurrentNumberOfRecords

Specifies the current number of records in the log.

RequestedState

Specifies the last requested state for the device.

Table B–55 describes values to which the RequestedState attribute can be set.

Table B–55   RequestedState Allowed Values

Value	Description
0	Unknown
2	Enabled
3	Disabled
4	Shut Down
5	No Change. Deprecated. Use the value **0** (Unknown) instead.
6	Offline
7	Test
8	Deferred
9	Quiesce
10	Reboot
11	Reset
12	Not Applicable
..	DMTF Reserved
32768..65535	Vendor Reserved

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–56 describes values to which the OperationalStatus attribute can be set.

Table B–56   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

Description

Specifies a textual description of the element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

ClearLog()

Clears the log.

Table B–57 shows the return values of the operation.

Table B–57   ClearLog() Return Values

Value	Description
0	Completed with no error
1	Not Supported
2	Unspecified Error
3	Timeout
4	Failed
5	Invalid Parameter
6..0x0FFF	DMTF_Reserved
0x1000..0x7FFF	Method_Reserved
0x8000..	Vendor_Reserved

RequestStateChange()

Changes the state of the element as specified in the RequestedState parameter.

Table B–58 describes return values of the operation.

Table B–58   RequestStateChange() Return Values

Value	Description
0	Completed with No Error
1	Not Supported
2	Unknown or Unspecified Error
3	Cannot complete within Timeout Period
4	Failed
5	Invalid Parameter
6	In Use
..	DMTF Reserved
4096	Method Parameters Checked - Job Started
4097	Invalid State Transition
4098	Use of Timeout Parameter Not Supported
4099	Busy
4100..32767	Method Reserved
32768..65535	Vendor Specific

Non-Supported Attributes

The following attributes are currently not supported:

■   OverwritePolicy

- LogState
- EnabledState
- OtherEnabledState
- EnabledDefault
- TimeOfLastStateChange
- AvailableRequestedStates
- TransitioningToState
- InstallDate
- StatusDescriptions
- Status
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID

SUN_NetraCard."BladeCardLogDev-i;id-j".N6000

This MBean defines the card.

Supported Attributes and Operations

This section contains the attributes and operations that are implemented and supported for this MBean.

Id

Specifies the element ID.

RelativeLocation

Specifies the number of a slot into which the blade card is inserted.

CardLogicalDevice

Specifies the Logical part of the card.

HostingBoard

Specifies whether or not the blade card is a baseboard in the chassis.

RemovalConditions

Specifies the conditions when a PhysicalPackage can be removed.

Table B–59 describes values to which the RemovalConditions attribute can be set.

Table B–59   RemovalConditions Possible Values

Value	Description
0	Unknown
2	Not Applicable
3	Removable when off
4	Removable when on or off

Description

Specifies a textual description of element.

ElementName

Specifies the hierarchy path of the element. The path must include the names of all ancestors of the element enclosed in the curly brackets.

For example: `{SUN_NetraLogicalDevice}{SUN_NetraCard}{"BladeCardLogDev-i;id-j"}{N6000}`

Tag

Specifies an arbitrary string that uniquely identifies the element and serves as the key of the element. This attribute can contain information, such as asset tag or serial number data.

CreationClassName

Specifies the name of the class or subclass that created this instance of the device.

Manufacturer

Specifies the name of the manufacturer that produced the element.

Model

Specifies the name by which the element is known.

SKU

Specifies the stock-keeping unit number for this element.

SerialNumber

Specifies the serial number assigned by the manufacturer to the element.

Version

Specifies the version of the element.

SystemName

Specifies the scoping name of the system.

PartNumber

Specifies the part number assigned by the manufacturer to the element.

CanBeFRUed

Specifies whether or not the device can be removed and replaced without affecting other devices.

Name

Specifies the name of the device.

OperationalStatus

Specifies an array of comma-separated integer values that describe the current status of the element.

Table B–60 describes values to which the OperationalStatus attribute can be set.

Table B–60   OperationalStatus Allowed Values

Value	Description
0	Unknown
1	Other
2	OK
3	Degraded
4	Stressed
5	Predictive Failure
6	Error

Table B–60   (Cont.) OperationalStatus Allowed Values

Value	Description
7	Non-Recoverable Error
8	Starting
9	Stopping
10	Stopped
11	In Service
12	No Contact
13	Lost Communication
14	Aborted
15	Dormant
16	Supporting Entity in Error
17	Completed
18	Power Mode
..	DMTF Reserved
0x8000..	Vendor Reserved

Caption

Specifies a short textual description of the element.

ConnectorPower()

Turns the power the connector on the card on and off.

Deprecated Attributes

The following attributes are deprecated:

- Removable

  Use RemovalConditions instead

- Replaceable

- HotSwappable

  Use RemovalConditions instead

- IsCompatible

Non-Supported Attributes

The following attributes are currently not supported:

- SlotLayout

- RequiresDaughterBoard

- SpecialRequirements

- RequirementsDescription

- OperatingVoltages

- Height

- Depth
- Width
- Weight
- PackageType
- OtherPackageType
- VendorCompatibilityStrings
- OtherIdentifyingInfo
- PoweredOn
- ManufactureDate
- VendorEquipmentType
- UserTracking
- StatusDescriptions
- InstallDate
- Status
- HealthState
- CommunicationStatus
- DetailedStatus
- OperatingStatus
- PrimaryStatus
- InstanceID