Sun Fire™ B100x and B200x Server Blade Installation and Setup Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 817-5625-10
March 2004, Revision A

Send comments about this document to: `docfeedback@sun.com`

Contents

Preface

This manual tells you how to install and set up B100x and B200x server blades for the Sun Fire B1600 blade system chassis.

The manual is intended for experienced system administrators.

Before You Read This Book

Before performing the instructions in this manual, make sure you have installed the blade system chassis into a rack and connected all of the cables required. For information on how to install the chassis hardware, read the *Sun Fire B1600 Blade System Chassis Hardware Installation Guide.*

How This Book Is Organized

Part 1 contains introductory information and tells you how to install the blade:

- Chapter 1 provides an overview of the steps required to install and set up a server blade. It also provides a list of features of the server blade.
- Chapter 2 provides information on system site requirements for a Sun Fire B1600 blade system chassis containing B200x and B100x and B200x blades.
- Chapter 3 tells you how to install or replace a server blade in the Sun Fire B1600 blade system chassis.

Part 2 contains information about running Linux on a blade:

- Chapter 4 tells you how to build a PXE boot install environment.
- Chapter 5 tells you how to power on a server blade and access its console.
- Chapter 6 tells you how to manually install the Linux kernel drivers when you perform a Linux kernel upgrade.
- Chapter 7 tells you how to use the link aggregation and failover to provide redundant network connections for the server blades.
- Chapter 8 provides information on using the `memdiag` utility and the `biosupdate` utility with Linux blades.
- Chapter 9 provides information on problems that can occur during or after a PXE boot installation of the Linux operating system.

Part 3 contains information about running Solaris x86 on a blade:

- Chapter 10 tells you how to set up the Network Install Server and the DHCP Server to install Solaris x86 onto a blade.
- Chapter 11 tells you how to use IPMP to provide redundant network connections for the server blades.
- Chapter 12 tells you how to use IPMP in combination with tagged VLANs to provide redundant virtual connections for the server blades.
- Chapter 13 provides information on testing the memory DIMMS on the Solaris x86 blades.
- Chapter 14 provides information on problems that can occur during or after a PXE boot installation of the Solaris x86 operating system.

Part 4 contains appendixes:

- Appendix A tells you how to upgrade the System Controller firmware and Blade Support Chip firmware.
- Appendix B tells you how to use the monitoring facility to view global information about the chassis and its components.

After You Read This Book

After you read this book you may need to consult two other manuals for the blade system chassis:

- For further information about using the command-line interface to the System Controller on the chassis, refer to the *Sun Fire B1600 Blade System Chassis Administration Guide*.
- For further information about managing the integrated switches on the chassis, refer to the *Sun Fire B1600 Blade System Chassis Switch Administration Guide.* This manual describes the hardware and architecture of the integrated switch (Chapter 1). It also tells you how to perform the initial configuration of the switch (Chapter 2), how to manage the switch using either its web Graphical User Interface or using SNMP (Chapter 3), and how to use all the commands available for managing the switch from the command-line interface (Chapter 4).

Typographic Conventions

Typeface[*]	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

`AaBbCc123`	What you type, when contrasted with on-screen computer output	`%` **`su`**
		`Password:`

AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

[*] The settings on your browser might differ from these settings.

Shell Prompts

Shell	Prompt
C shell	*machine-name*`%`
C shell superuser	*machine-name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`
System Controller shell	sc>
Integrated switch shell	Console#

Related Documentation

Application	Title
Compliance and safety	*Sun Fire B1600 Blade System Chassis* Compliance and Safety Manual
Hardware installation overview (foldout poster)	*Sun Fire B1600 Blade System Chassis* Quick Start
Hardware installation	*Sun Fire B1600 Blade System Chassis* Hardware Installation Guide
Software installation overview (foldout poster)	*Sun Fire B1600 Blade System Chassis* Software Setup Quick Start
Software setup	*Sun Fire B1600 Blade System Chassis* Software Setup Guide
B100x and B200x server blade installation and setup	*Sun Fire B100x and B200x Server Blade* Installation and Setup Guide (this manual)
System chassis administration and component replacement	*Sun Fire B1600 Blade System Chassis* Administration Guide
Switch administration	*Sun Fire B1600 Blade System Chassis* Switch Administration Guide
Late-breaking information	*Sun Fire B1600 Blade System Chassis* Product Notes

Accessing Sun Documentation

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

`http://www.sun.com/documentation`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

```
docfeedback@sun.com
```

Please include the part number (817-5625-10) of your document in the subject line of your email.

PART 1 Installing the Blade Hardware

CHAPTER 1

Preparing to Install and Set Up Server Blades

This chapter provides an overview of the server blades. It contains the following sections:

- Section 1.1, "Blade Hardware Setup Overview" on page 1-2
- Section 1.2, "Blade Software Setup Overview" on page 1-2
- Section 1.3, "Overview of the B100x Server Blade" on page 1-4
- Section 1.4, "Overview of the B200x Server Blade" on page 1-6
- Section 1.5, "Upgrading System Controller Firmware" on page 1-8

1.1 Blade Hardware Setup Overview

1. **Set up and install the system chassis.**

   See the *Sun Fire B1600 Blade System Chassis Hardware Installation Guide* and the *Sun Fire B1600 Blade System Hardware Chassis Quick Start* poster.

   Note – To install B100x or B200x server blades, you must be running System Controller firmware 1.2 or later.

   ---

2. **If you are replacing a blade, disable the existing blade prior to removal.**

   See Section 3.2, "Disabling an Existing Blade Prior to Removal" on page 3-3.

3. **If you are replacing a blade, remove the existing blade.**

   See Section 3.3, "Removing an Existing Blade or Filler Panel" on page 3-3.

4. **Insert the blade.**

   See Section 3.4, "Inserting the New Blade or Filler Panel" on page 3-7.

1.2 Blade Software Setup Overview

1. **Build a PXE boot install environment for the OS (operating system) you are installing.**

   For information on installing Linux, see Chapter 4.

   For information on installing Solaris x86, see Chapter 10.

2. **If you are setting up a blade system chassis for the first time, set up the System Controller (SC) and switches.**

   See the *Sun Fire B1600 Blade System Chassis Software Setup Guide.*

3. **Use the System Controller (SC) to configure the blade to temporarily boot from the network.**

   For Linux, see Section 5.1, "Configuring the Server Blade to Boot From the Network" on page 5-2.

   For Solaris x86, see Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18.

4. **Power on the blade to install the operating system.**

   For Linux, see Section 5.2, “Powering On and Booting the Server Blade” on page 5-3.

   For Solaris x86, see Section 10.7, “Monitoring the Network Booting Process and Starting the Solaris Installation” on page 10-20.

1.3 Overview of the B100x Server Blade

The B100x server blade (FIGURE 1-1) is a single-processor server that fits in a Sun Fire B1600 blade system chassis.

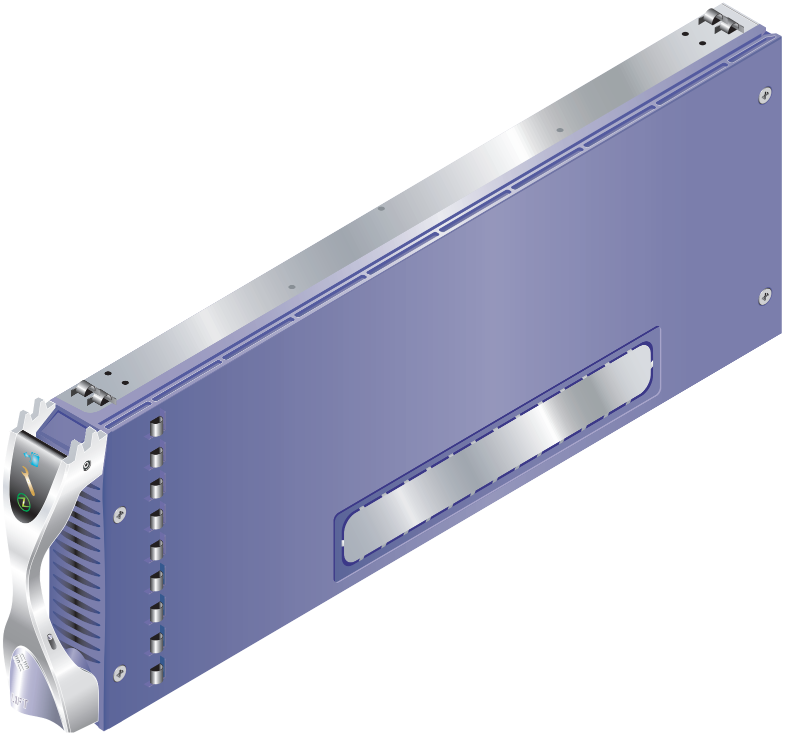


FIGURE 1-1 The B100x server blade

1.3.1 B100x Server Blade Feature Set

The B100x server blade features are listed in TABLE 1-1:

TABLE 1-1 B100x server blade feature set

Feature	Description
CPU architecture	AMD Mobile Athlon processor.
Chipset, front side bus	VIA KT333 (VT8367) North Bridge and VT8233A South Bridge. 266MHz double-clock Front Side Bus (FSB).
Memory architecture	2x 266MHz PC2100 DDR Registered DIMMs with ECC. 2 GByte addressable memory space.
PCI bus architecture	Dual Gbit ethernet MAC with integrated SERDES.
I/O to switch and System Controller (SC)	Two Gbit ethernet SERDES connections. Two serial ports from the Blade Support Chip (BSC) microcontroller to System Controllers (SCs).
Internal I/O	2.5” Ultra DMA100 ATA hard disk 30 GByte. Rated for continuous operation.
Support devices	Blade Support Chip (BSC) microcontroller. 1MB Flash PROM for BIOS. Temperature monitor for CPU and blade board.
Other	“Active”, “Service Required” and “Ready to Remove” indicators.

1.4 Overview of the B200x Server Blade

The B200x server blade (FIGURE 1-2) is a dual-processor server that fits in a Sun Fire B1600 blade system chassis.

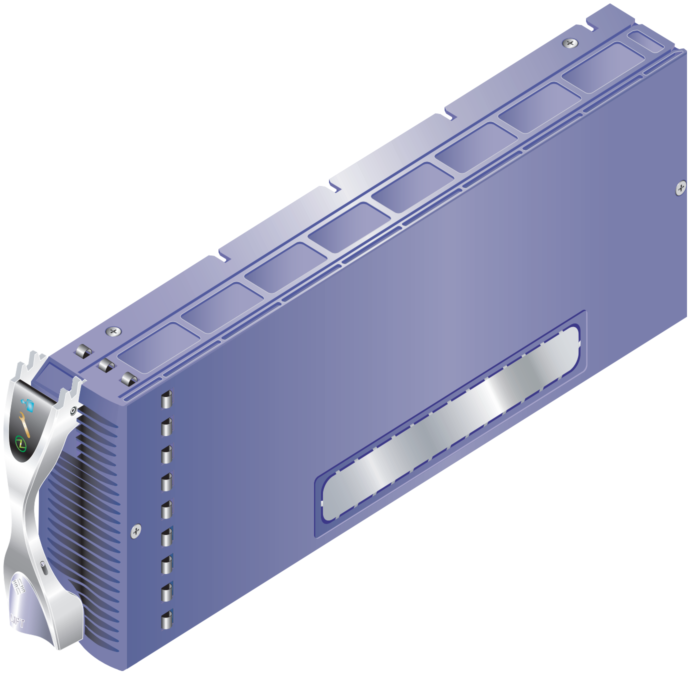


FIGURE 1-2 The B200x server blade

1.4.1 B200x Server Blade Feature Set

The B200x server blade features are listed in TABLE 1-2:

TABLE 1-2 B200x server blade feature set

Feature	Description
CPU architecture	Two Intel LV Xeon processors.
Chipset, front side bus	Intel E7501 chipset. 400/533 MHz Quad-pumped Front Side Bus (FSB).
Memory architecture	Dual-Channel DDR-200/266 Memory Interface. 4x 266MHz PC2100 DDR Registered DIMMS with ECC. 8 GByte addressable memory space.
PCI bus architecture	Two dual Gbit ethernet MAC with integrated SERDES.
I/O to switch and System Controller (SC)	Four Gbit ethernet SERDES connections. Two serial ports from Blade Support Chip (BSC) microcontroller to System Controllers (SCs).
Internal I/O	2.5” Ultra DMA100 ATA hard disk 30 GBytes. Rated for continuous operation.
Support devices	Blade Support Chip (BSC) microcontroller. 1MB Flash PROM for BIOS. Temperature monitor for CPUs and blade board.
Other	“Active”, “Service Required” and “Ready to Remove” indicators. Two fans.

1.5 Upgrading System Controller Firmware

To install these server blades, you must be running System Controller firmware 1.2 or later.

You can check the version of the System Controller firmware by typing `showsc` at the `sc` prompt:

```
sc> showsc

Sun Advanced Lights Out Manager for Blade Servers 1.2
Copyright 2003 Sun Microsystems, Inc. All Rights Reserved.
ALOM-B 1.2

Release: 1.2.1
:
sc>
```

For information on upgrading the System Controller firmware see *Appendix A*.

CHAPTER 2

Site Preparation

This section contains information about the following system site requirements for the Sun Fire B1600 blade system chassis:

- Section 2.1, "System Cooling Requirements" on page 2-2
- Section 2.2, "Operating Power Limits and Ranges" on page 2-6
- Section 2.3, "Estimating Power Consumption" on page 2-6

2.1 System Cooling Requirements

This section provides the general environmental parameters and airflow requirements for the Sun Fire B1600 blade system chassis.

Note – The Sun Fire B1600 blade system chassis uses front-to-back forced air cooling.

2.1.1 General Environmental Parameters

You can operate and store the system safely in the conditions detailed in TABLE 2-1, FIGURE 2-1 and FIGURE 2-2.

TABLE 2-1 Operating and Storage Specifications

Specification	Operating	Storage
Ambient temperature	5˚C to 35˚C maximum ambient temperature is derated by 1˚C per 500m altitude above 500m	-40˚C to 65˚C
Relative humidity	10% to 90% RH non-condensing, 27˚C max wet bulb	up to 93% RH non-condensing, 38˚C max wet bulb
Altitude	-400m up to 3000m	-400m up to 12000m

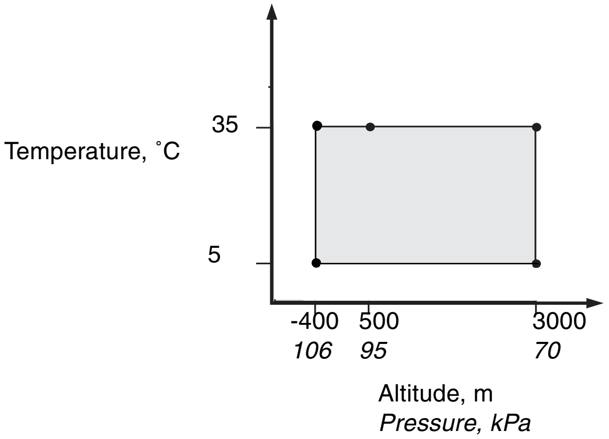


FIGURE 2-1 Temperature and Altitude Operating Ranges

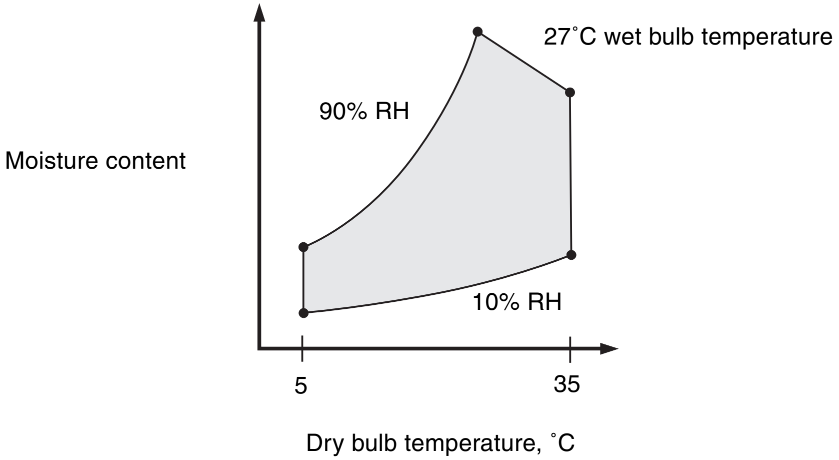


FIGURE 2-2 Temperature and Relative Humidity Ranges

2.1.1.1 Recommended Environment Parameters

Your environmental control system must provide intake air for the server which complies with the limits specified in "General Environmental Parameters" on page 2-2.

To avoid overheating, *do not* direct warmed air:

- towards the front of the cabinet or rack
- towards the server access panels

Note – When you receive your system, leave it in the shipping crate at its final destination for 24 hours in the environment in which you will install it. This is to prevent thermal shock and condensation.

The operating environmental limits in TABLE 2-1 reflect what the systems have been tested to, in order to meet all functional requirements. Operating computer equipment in extremes of temperature or humidity increases the failure rate of hardware components. To minimize the chance of component failure, use the server within the optimal temperature and humidity ranges.

2.1.1.2 Ambient Temperature

An ambient temperature range of 21˚C to 23˚C is optimal for system reliability. At 22˚C it is easy to maintain safe relative humidity levels. Operating in this temperature range provides a buffer in the event of the environmental support systems failing.

2.1.1.3 Ambient Relative Humidity

Ambient relative humidity levels between 45% and 50% are the most suitable for data processing operations in order to:

- prevent corrosion
- provide an operating time buffer in the event of environmental control system failure
- help avoid failures caused by the intermittent interference from static discharges that occur when relative humidity is too low.

Electrostatic discharge (ESD) is easily generated and less easily dissipated in areas where the relative humidity is below 35%, and becomes critical when levels drop below 30%.

2.1.2 Airflow Requirements

The Sun Fire B1600 blade system chassis has been designed to function in a natural convection airflow when mounted in a rack or cabinet and uses front-to-back forced air cooling. To meet the declared environmental specification follow these guidelines:

- The Sun Fire B1600 blade system chassis uses PSU fans that can achieve a maximum airflow of 160 cfm in free air. Ensure that there is sufficient airflow through the rack or cabinet.
- The rack or cabinet in which the system chassis is mounted must provide inlet air at the front of the system chassis. The airflow exhausts horizontally from the PSU and SSC modules located at the back of the system chassis and must be able to leave the cabinet.
- Inlet and exhaust ventilation must both have a minimum open area of 22 in^2 (142 cm^2) for each system chassis.
- The use of perforated or solid door panels must allow adequate airflow to the system chassis when the cabinet doors are closed.

2.1.3 Estimating the Heat Dissipation

To estimate the heat generated by a Sun Fire B1600 blade system chassis convert the figure for the system's power consumption from watts to BTU/hr.

The formula to convert from watts to BTU/hr is to multiply the power in watts by 3.415. For example:

total power consumption of blades + *total power consumption of SSCs* + *total power consumption of PSUs* x 3.415 = *xxxxx* BTU/hr

For power consumption figures for the SSC, the PSU and blades, see "Estimating Power Consumption" on page 2-6

Note – Do not install multiple Sun Fire B1600 blade system chassis in a four-post rack or cabinet unless your cooling system is capable of dissipating in excess of the total thermal load.

2.2 Operating Power Limits and Ranges

TABLE 2-2 Operating Power Limits and Ranges

Description	Operating Limit or Range
Maximum operating current [*]	16A @ 110VAC 8A @ 240VAC
Maximum power supply rating [†]	12A @ 110VAC 6A @ 240VAC
Maximum in-rush current[‡]	20A
Operating input voltage range (auto-ranging)	110 to 240 VAC
Voltage frequency range	47 to 63Hz
Power factor	0.95 to 1.0
BTU/Hr rating	*xxxxx* BTU/Hr. This value will depend on the estimated heat dissipation. See "Estimating the Heat Dissipation" on page 2-5 for more information.

[*] Each power cord provides approximately one half of the input current during normal system operation.

[†] Currents up to the maximum power supply rating might be required for future product upgrades

[‡] The in-rush current decays to the normal operating current in less than 200 milliseconds. Sequencing of power to multiple units is not required, as the peak current is less than seven times the operating current.

2.3 Estimating Power Consumption

To estimate the total power consumption for one or more Sun Fire B1600 blade system chassis installed in a single rack or cabinet, add together the individual power requirement figures for each system chassis you have installed, using the values in TABLE 2-3. A minimum system configuration would be:

One blade + one SSC + two PSUs

TABLE 2-3 Power Consumption

System Chassis Components	Power Consumption (max)
one SSC	Add 65W per SSC
one PSU	Add 110W per PSU
one B100s Blade	Add 35W per blade
one B100x Blade	Add 48W per blade
one B200x Blade	Add 126W per blade

CHAPTER 3

Installing and Replacing Server Blades

This chapter provides the steps required to install and replace B100x blade (single-width) and B200x (double-width) blades in the Sun Fire B1600 blade system chassis. The chapter contains the following sections:

3.1 Introduction

The system chassis contains 16 slots. It can hold a combination of single-width blades, double-width blades and filler panels. Double-width blades occupy two adjacent slots in the system chassis.

FIGURE 3-1 shows a system chassis containing single-width blades and a double-width blade.

Note – Be aware that the system chassis contains three internal dividing walls. Double-width blades must be installed in two available slots between these internal dividing walls.

Caution – Do not leave any slots empty as this can disrupt airflow through the system and compromise EMC performance.

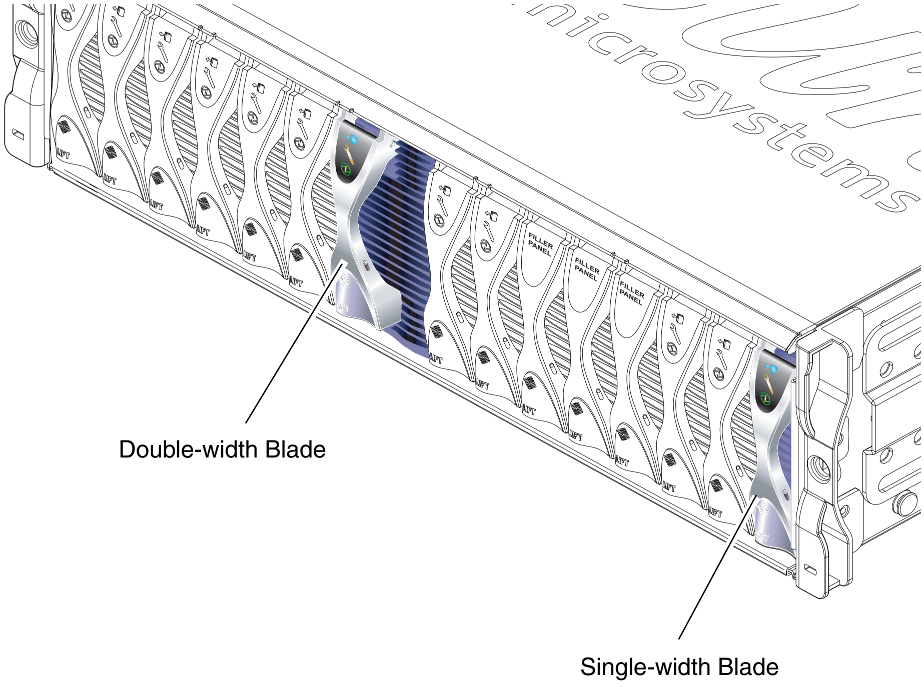


FIGURE 3-1 B1600 System Chassis Containing Single-width and Double-width Blades

3.2 Disabling an Existing Blade Prior to Removal

- **To shut down the blade in preparation for removal, and to cause the blue "Ready to Remove" LED to be lit, type:**

```
sc> removefru sn
```

Where *n* is the number of the slot containing the blade you are removing.

3.3 Removing an Existing Blade or Filler Panel

The steps in this section refer to removal of a single-width blade. The same steps apply when removing a double-width blade or filler panel.

1. **If you are removing a blade, check that the blue "Ready to Remove" LED is lit.**

Note – Do not remove the blade until the blue LED is lit.

2. **Insert your finger in the pull recess located at the bottom front of the blade lever and pull gently to disengage the locking mechanism (FIGURE 3-2).**

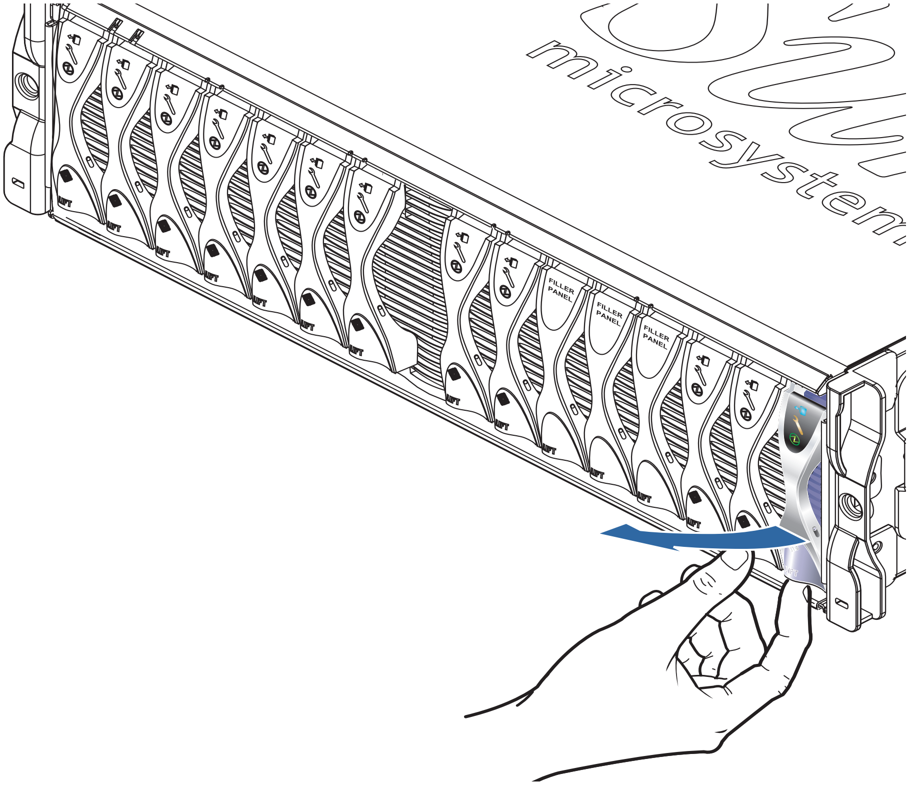


FIGURE 3-2 Disengaging the Blade Locking Mechanism

3. **Pull the lever in a forward and upward motion, causing the blade lever to unlatch and eject the blade partially from the system chassis (FIGURE 3-3).**

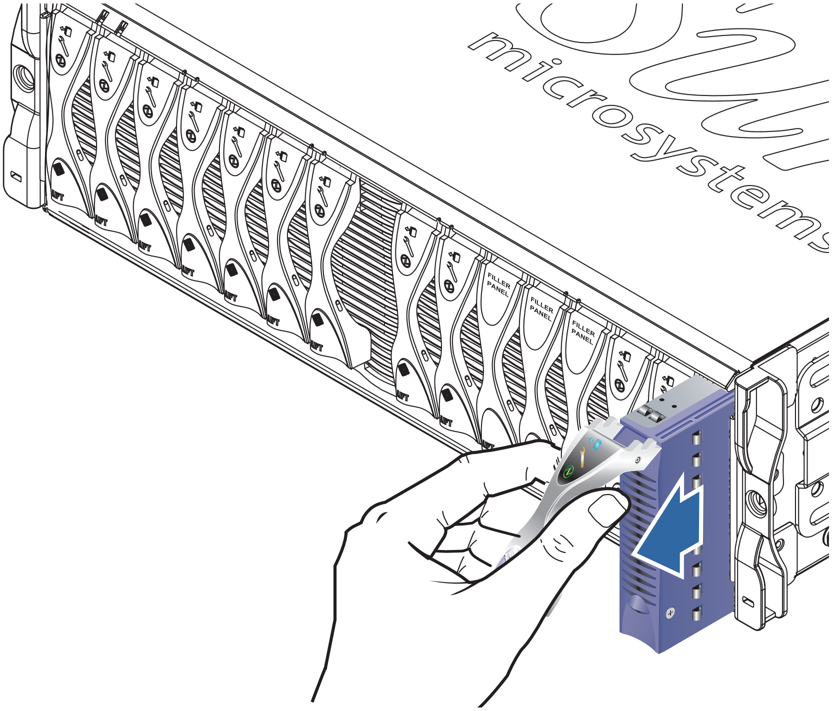


FIGURE 3-3 The Released Blade or Filler Panel Lever Mechanism

4. **Pull the lever to remove the blade from the system chassis (FIGURE 3-4).**

   Support the bottom of the blade with your free hand while lifting the filler panel clear of the system chassis.

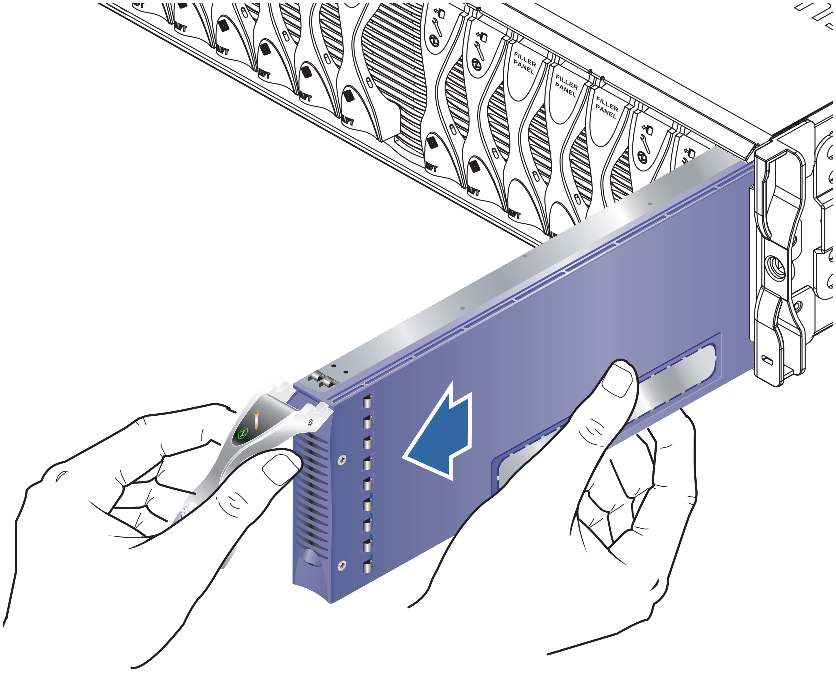


FIGURE 3-4 Removing the Blade or Filler Panel

3.4 Inserting the New Blade or Filler Panel

The system chassis is designed to operate with a total of up to 16 blades and filler panels installed.

Caution – Do not leave any slots empty as this can disrupt airflow through the system and compromise EMC performance.

Note – Be aware that the system chassis contains three internal dividing walls. Double-width blades must be installed in two available slots between these internal dividing walls.

The steps below refer to installation of a single-width blade. The same steps apply when installing a filler panel or double-width blade.

If required, open the blade lever by inserting a finger in the pull recess located in lower portion of the blade lever and pull the lever in a forward and upward motion, causing the lever to unlatch (FIGURE 3-5).

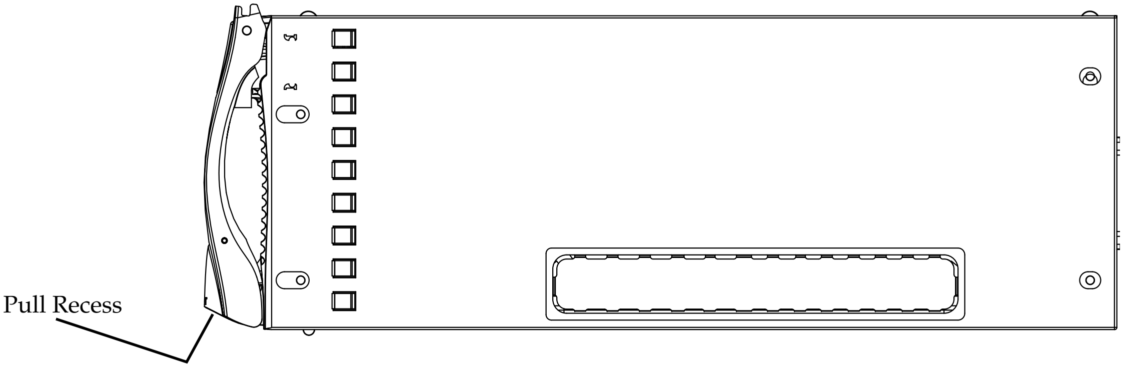


FIGURE 3-5 The Blade Locking Mechanism

5. **Align the blade with the empty slot.**

   Ensure that the blade connector is facing towards the system chassis, with the hinge point of the lever mechanism at the top. Support the bottom of the blade with your free hand while lifting the blade up to the system chassis (FIGURE 3-6).

6. **Insert the blade into the selected system chassis slot (FIGURE 3-6).**

Caution – Ensure that the blade engages with the system chassis guidance system. Failure to align the blade correctly can result in damage to the chassis midplane or the blade connection.

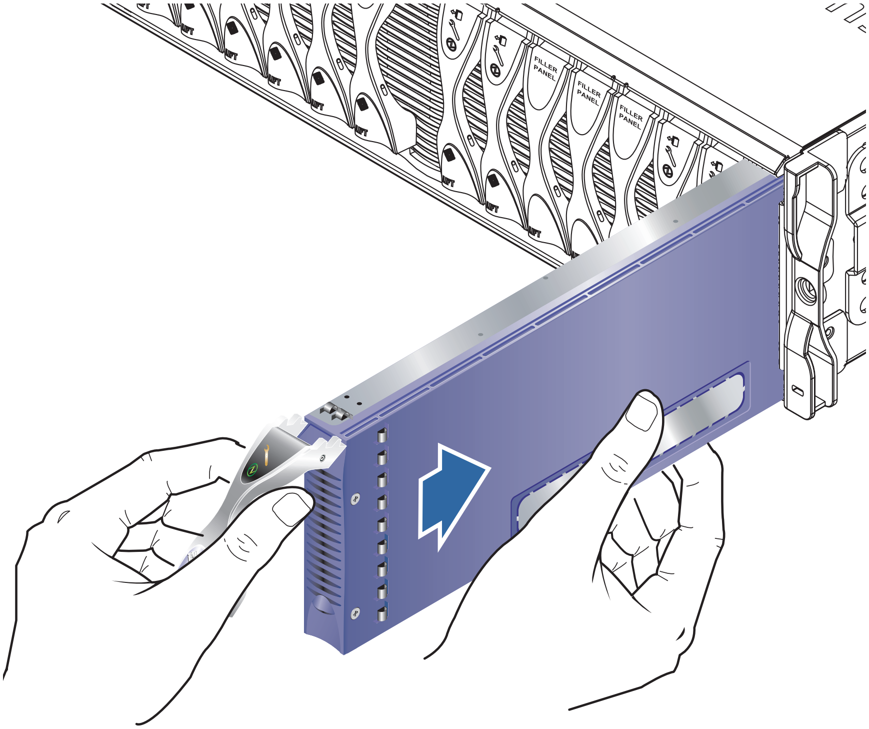


FIGURE 3-6 Aligning and Inserting the Blade

7. **Gently push the blade into the slot until the blade latch ears, on top of the lever, are positioned in the chassis.**

8. **Close the blade lever fully by pushing it down until you feel the latch click in place.**

   This engages the blade with the connectors in the chassis slot (FIGURE 3-7). When you do this, the LEDs on the blade flash several times.

Note – For information interpreting LEDs on a blade, see the *Sun Fire B1600 Blade System Chassis Administration Guide*

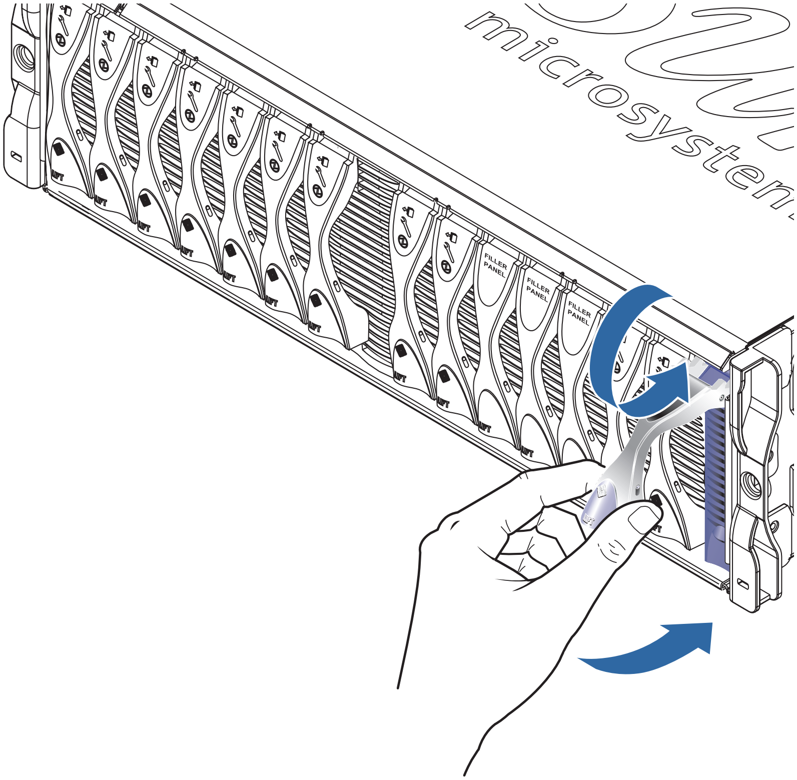


FIGURE 3-7 Closing the Blade Lever Mechanism

PART 2 Installing and Using Linux on a Blade

CHAPTER 4

Installing Linux From a PXE Boot Install Environment

This chapter provides the information you need to install Linux on a B100x or B200x server blade. It contains the following sections:

4.1 PXE Overview

The Preboot Execution Environment (PXE) is a method of network booting blade and cluster systems. It is the core technology for Intel's Wired for Management (WfM) initiative and is supported by most commercial network interfaces. You can install a blade operating system image with minimal effort from a central location by using PXE.

To install Linux onto a server blade using PXE you will need the following:

- A PXE boot server machine. This machine must be running one of the following operating systems:
  - Red Hat Enterprise Linux, Advanced Server 2.1 update 2
  - Red Hat Enterprise Linux, version 3.0
  - SuSE Linux Enterprise Server 8, service pack 3
  - Solaris, version 9 or later
- A server blade (without the operating system installed).
- The *Sun Fire B1600 Platform Documentation, Drivers, and Installation* CD supplied by Sun with the server blade.
- Installation CDs for the version of Linux you are installing. You can install one of the following:
  - Red Hat Enterprise Linux, Advanced Server 2.1 update 2
  - Red Hat Enterprise Linux, version 3.0
  - SuSE Linux Enterprise Server 8, service pack 3

Note – For information on troubleshooting the PXE boot installation see Chapter 9.

Note – If you install a new Linux kernel after the PXE boot installation, you will need to manually install the Linux drivers. For more information, see Chapter 6.

4.1.1 PXE Protocols

PXE comprises three distinct network protocols:

- Dynamic Host Configuration Protocol (DHCP)
- Trivial File Transfer Protocol (TFTP)
- Network File System (NFS)

These protocols allow delivery of system configuration information in addition to system software for blades. See TABLE 4-1 for further details.

TABLE 4-1 Network Protocols Used by the Preboot Execution Environment (PXE)

Protocol	Definition
DHCP	Dynamic Host Configuration Protocol (DHCP) defines a method for delivery of network configuration information to client nodes. This configuration information often includes basic information needed for Internet access, such as the client IP address and netmask. However, RFC1533 defines many advanced DHCP options, which can include packet filter rules and other more obscure networking parameters. In addition, software vendors may extend the protocol by defining their own DHCP options. PXE solutions use DHCP to deliver initial network configuration options to client nodes.
TFTP	Trivial File Transfer Protocol (TFTP) defines a simple UDP protocol for delivering files over a network. PXE solutions can deliver kernels and initial bootstrap software to client nodes using TFTP.
NFS	Network File System. This protocol was developed by Sun Microsystems and is an industry standard for remote file access across a common network.

The PXE standard also specifies a client side BIOS programming interface called UNDI. This API abstracts ethernet devices to allow x86 based systems to implement simple, network-based bootstrap loaders.

Universal Network Driver Interface (UNDI) is a programming API that simplifies network programming. All network interface cards that support PXE network booting can be controlled using the API. This provides the bootstrap mechanism with a universal method for accessing network cards.

4.2 Installing Linux From a Linux PXE Boot Server

This section tells you how to install Linux on a B100x or B200x server blade from a PXE boot server running Linux.

The PXE boot server must be running one of the following versions of Linux:

- Red Hat Enterprise Linux, Advanced Server 2.1 update 2
- Red Hat Enterprise Linux, version 3.0
- SuSE Linux Enterprise Server 8 service pack 3

Note – IMPORTANT: Before installing Linux, ensure that the boot directory on the PXE server (`/tftp`) has enough space to accommodate the version of Linux you are installing. You will require about 6 Gbytes of free space.

4.2.1 Files Relevant to PXE Boot Installation

TABLE 4-2 provides a summary of the files required during the PXE boot installation:

TABLE 4-2 Summary of Files Relevant to PXE Boot Installation

Filename	Purpose
/etc/exports	The NFS server is used by the installation kernel to read the packages necessary to the installation process. The NFS server needs to provide access to the directory structure containing the required packages. During installation you will update the /etc/exports file to provide access to this directory structure.
/tftp/<*Linux_dir*>/sun/install/ks.cfg or: /tftp/sles-8sp3/sun/install/autoyast.xml	The Red Hat PXE boot installation is controlled by the ks.cfg configuration file. The SuSE PXE boot installation is controlled by the autoyast.xml configuration file. During installation you will update this file to use the correct NFS server address. For more information on the ks.cfg or autoyast.xml file, refer to the documentation supplied by your operating system vendor.
/tftp/<*Linux_dir*>/sun/pxelinux.cfg/*	The /tftp/<*Linux_dir*>/sun/pxelinux.cfg/* files control where pxelinux.bin finds a kernel to boot from and how it should boot that kernel. The files in this directory are named based on the IP address that should read them. For example, if the client is given an IP address of 9.10.11.12, pxelinux.bin will attempt to download (using TFTP and the PXE NIC support code) the following files in order: pxelinux.cfg/090A0B0C pxelinux.cfg/090A0B0 pxelinux.cfg/090A0B pxelinux.cfg/090A0 pxelinux.cfg/090A pxelinux.cfg/090 pxelinux.cfg/09 pxelinux.cfg/0 pxelinux.cfg/default The first file downloaded successfully is used to select the kernel image and runtime arguments.
/etc/xinetd.d/tftp or: /etc/inetd.d/tftp	The TFTP server supplies the PXE boot with the stage 1 bootloader image. This image loads the installation kernel that performs the installation on the hard disk.
/etc/dhcpd.conf	The DHCP server supplies the PXE boot plug-in with an IP address and TFTP server address, and the stage 1 image boot-loader name to download and execute.

Note – The Linux directory name (*Linux_dir*) depends on the version of Linux you are installing. Files for Enterprise Linux Advanced Server 2.1 update 2 are in a directory called `as-2.1u2`, the files for Enterprise Linux version 3.0 are in a directory called
`el-3.0`, and the files for SuSE Linux Enterprise Server 8 service pack 3 are in a directory called `sles-8sp3`.

4.2.2 Configuring the PXE Boot Servers

Linux is installed on the server blade using the PXE boot system. Three server processes are required to perform the installation:

- DHCP
- TFTP
- NFS

This section provides information on how to configure the DHCP, TFTP and NFS servers for use with the PXE boot installation.

Note – This chapter assumes that all server processes are running on the same physical host.

4.2.2.1 Configuring the DHCP Server

The DHCP server supplies the PXE boot plug-in with:

- IP address
- TFTP server address
- Stage 1 image boot-loader name from which to download and execute the image.

Note – As the supplied PXE installation environments are non-interactive and will unconditionally reinstall a client machine, you might want to have the client associate its MAC address with a specific OS installation before starting the PXE boot. In other environments, where clients are attached to the network specifically to install one given OS, you might want to have a PXE installation as the default.

Use the `dhcp` package provided with the version of Linux you are installing to provide DHCP services.

1. **Update the** `/etc/dhcpd.conf` **file:**

   a. **Add a subnet section with** `next-server` **referring to your TFTP server.**

   b. **Change the** `filename` **entry to** /*<Linux_dir>*/sun/pxelinux.bin

   where *<Linux_dir>* is either `as-2.1u2`, `el-3.0`, or `sles-8sp3`, depending on the version of Linux you are installing.

Note – You can restrict the use of the `filename` and `next-server` directives in the `dhcpd.conf` file to avoid accidental installations of Linux.

---

   c. **If you are installing Red Hat Enterprise Linux Advanced Server 2.1 update 2, remove the line** `ddns-update-style none;`**. (This line is required when installing all other versions of Linux).**

2. **Enable the DHCP server.**

   For Red Hat, type:

```
/sbin/chkconfig --level 345 dhcpd on
```

   For SuSE, type:

```
chkconfig dhcpd on
```

3. **Restart the DHCP server:**

```
/etc/init.d/dhcpd restart
```

4. **Validate the configuration:**

```
# netstat -an | fgrep -w 67
```

   The output should be:

```
udp        0      0 0.0.0.0:67              0.0.0.0:*
```

Example of the dhcpd.conf *File*

CODE EXAMPLE 4-1 shows a sample /etc/dhcpd.conf file

```
ddns-update-style none;
default-lease-time 1800;
max-lease-time 3600;

option domain-name              "linux.sun.com";
option domain-name-servers      172.16.11.2, 172.16.11.8;
option subnet-mask              255.255.0.0;

allow bootp;
allow booting;

option ip-forwarding    false;  # No IP forwarding
option mask-supplier    false;  # Don't respond to ICMP Mask req
get-lease-hostnames     on;     # DNS lookup hostnames
use-host-decl-names     on;     # And supply them to clients

option routers 172.16.11.6;

# WARNING: This is a default configuration -- any system PXE booting will
#          wipe out all existing data on the first hard disk and install
#          Linux

subnet 172.16.11.0 netmask 255.255.0.0 {
  next-server 172.16.11.8;                   # name of your TFTP server
  filename "/<linux_dir>/sun/pxelinux.bin";  # name of the boot-loader program
  range 172.16.11.100 172.16.11.200;         # dhcp clients IP range
}
```

CODE EXAMPLE 4-1 Sample /etc/dhcpd.conf file

The important areas in this example are the address of the TFTP server (`next-server` **`172.16.11.8`**) and the filename of the stage 1 bootloader image (`filename` “**/***<linux_dir>***/sun/pxelinux.bin**”).

Note – Nameserver and web server software is provided with the Red Hat Enterprise Linux distribution. Installation and configuration of these applications is outside the scope of this document.

Note – If no Nameserver is configured, change `get-lease-hostnames` to **`off`**.

4.2.2.2 Configuring the TFTP Server

The TFTP server supplies the PXE boot with the stage 1 bootloader image. This image loads the installation kernel which performs the actual installation on the hard disk through the use of the custom `initrd.img` supplied by Red Hat.

Use the `tftp-server` package provided with your Linux distribution to provide TFTP services.

1. **Create the TFTP directory. Ensure that all users have read/execute access to the TFTP directory:**

```
umask 022
mkdir /tftp
chmod 755 /tftp
```

2. **Modify the** `/etc/xinetd.d/tftp` file **(for Red Hat) or the** `/etc/inetd.conf` **file (for SuSE) to allow TFTP services:**

   - If you are installing Red Hat, update the `/etc/xinetd.d/tftp` file. You need to change the `server_args` entry to `-s /tftp`. (The `/tftp` path is the directory in which the PXE images are copied.)
   - If you are installing SuSE, update the `/etc/inetd.conf` file by inserting the following line:

```
tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd -s /tftp
```

3. **If you are installing SuSE skip to Step 4. If you are installing Red Hat, configure the TFTP server to be enabled at installation.**

   Change the `disable` entry to `disable= no`.

Note – At installation the TFTP server is disabled by default (`disable= yes`).

4. **Enable the TFTP server.**

   - For Red Hat, type:

```
chkconfig --level 345 xinetd on
```

- For SuSE, type:

```
chkconfig inetd on
```

Note – No output is returned if the command succeeds.

---

5. **Restart** `xinetd` **(for Red Hat) or** `inetd` **(for SuSE):**

- For Red Hat, type:

```
/etc/init.d/xinetd restart
```

- For SuSE, type:

```
/etc/init.d/inetd restart
```

6. **Validate the configuration:**

```
# netstat -an | fgrep -w 69
```

The output should be:

```
udp        0      0 0.0.0.0:69         0.0.0.0:*
```

Example of the `tftp` *File for Red Hat*

shows an example of the `/etc/xinetd.d/tftp` file for Red Hat:

```
# default: off
# description: The tftp server serves files using the trivial file transfer
#       protocol.  The tftp protocol is often used to boot diskless
#       workstations, download configuration files to network-aware printers,
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type= dgram
                              protocol   = udp
        wait          = yes
        user          = root
        server        = /usr/sbin/in.tftpd
        server_args   = -s /tftp
        disable       = no
}
```

Example `/etc/xinetd.d/tftp` file for Red Hat

4.2.2.3 Configuring the NFS Server

The NFS server is used by the installation kernel to read all of the packages necessary to the installation process. The NFS server therefore needs to provide access to the directory structure containing the PXE images.

1. **Update the** `/etc/exports` **file to include the export for the NFS server.**

   Insert the following line into file `/etc/exports`:

```
/tftp  *(ro)
```

2. **Enable the NFS server.**

   - For Red Hat, type:

```
chkconfig --level 2345 nfs on
```

   - For SuSE, type:

```
chkconfig nfslock on
chkconfig nfsserver on
```

Note – No output is returned if the command succeeds.

3. **Restart the NFS server.**

   For Red Hat, type::

```
/etc/init.d/nfs restart
```

   For SuSe, type:

```
/etc/init.d/nfslock restart
/etc/init.d/nfsserver restart
```

4. **Validate the configuration:**

```
showmount -e
```

   The output should include the line:

```
/tftp
```

4.2.3 Installing Linux on a Server Blade from a Linux PXE Boot Server

---

Note – IMPORTANT: Before installing Linux, ensure that the boot directory on the PXE server (`/tftp`) has enough space to accommodate the version of Linux you are installing. You will require about 6 Gbytes of free space.

Note – The PXE boot server should be running Enterprise Linux version AS 2.1 or EL 3.0, or SuSE Linux Enterprise Server 8, service pack 3.

Caution – Installing Linux will overwrite any data already on the destination server blade.

---

1. **If you have configured a firewall, make sure that the TFTP, NFS, and DHCP protocols are not filtered on the server to be used as the PXE boot server.**

2. **Alternatively, disable the firewall and prevent it from running on subsequent reboots.**

   - To do this, for Red Hat, type:

```
chkconfig --level 2345 iptables off
/etc/init.d/iptables stop
```

   - For SuSE, type:

```
chkconfig iptables off
/etc/init.d/iptables stop
```

Note – These examples assume that you are using iptable firewalls. iptable firewalls are not installed by default on SuSE.

---

3. **Ensure that the DHCP server, NFS server and TFTP server have been configured correctly.**

   See Section 4.2.2, "Configuring the PXE Boot Servers" on page 4-6 for more information.

4. **Install the PXE images onto the TFTP server:**

Note – If you are running SuSE on your PXE boot server, replace **/mnt/cdrom** with **/media/cdrom** in the instructions below. For example, **mount /mnt/cdrom** would be **mount /media/cdrom**.

a. **Copy the required Linux directory from the root of the** *Sun Fire B1600 Platform Documentation, Drivers, and Installation* **CD to the** /tftp **directory on your PXE boot server:**

```
umask 022
mount /mnt/cdrom
cd /mnt/cdrom
egrep '^<Linux_dir>' filenames.txt | cpio -pumd /tftp/.
cd /
umount /mnt/cdrom
```

where *<Linux_dir>* is as-2.1u2, el-3.0 or sles-8sp3, depending on the version of Linux you are installing.

Note – The Linux directory contains the files required to perform a PXE installation.

b. **Install the Linux installation CDs to the** /tftp **directory on your PXE boot server.**

- For Red Hat, you need to install the CDs in reverse order. If you have two Red Hat installation CDs, install Disk 2 first; if you have four, install Disk 4 first. After inserting each CD, type the following command:

```
umask 022
mount /mnt/cdrom
cd /mnt/cdrom
tar -cf - . |tar -C /tftp/<Linux_dir> -xf -
cd /
umount /mnt/cdrom
```

where *<Linux_dir>* is as-2.1u2 or el-3.0, depending on the version of Linux you are installing.

- For SuSE Linux Enterprise Server 8 service pack 3, you need to load each image into its own directory rather than into the same directory. This allows the SuSE installer to select the correct packages from each ISO image. Use the following commands:

  After inserting the SLES-8 disk:

```
mount /mnt/cdrom
mkdir /tftp/sles-8sp3/SLES-8-i386-RC5-CD1
cd /mnt/cdrom
pax -rw . /tftp/sles-8sp3/SLES-8-i386-RC5-CD1
cd /
umount /mnt/cdrom
```

  After inserting the first UnitedLinux 1.0 disk:

```
mount /mnt/cdrom
mkdir /tftp/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD1
cd /mnt/cdrom
pax -rw . /tftp/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD1
cd /
umount /mnt/cdrom
```

  After inserting the second UnitedLinux 1.0 disk:

```
mount /mnt/cdrom
mkdir /tftp/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD2
cd /mnt/cdrom
pax -rw . /tftp/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD2
cd /
umount /mnt/cdrom
```

  After inserting the third UnitedLinux 1.0 disk:

```
mount /mnt/cdrom
mkdir /tftp/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD3
cd /mnt/cdrom
pax -rw . /tftp/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD3
cd /
umount /mnt/cdrom
```

After inserting the first United Linux 1.0 SP 3 disk:

```
mount /mnt/cdrom
mkdir /tftp/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD1
cd /mnt/cdrom
pax -rw . /tftp/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD1
cd /
umount /mnt/cdrom
```

Note – The first SP 3 disk contains hard-linked directories. Do not use the `cp, cpio` or `tar` commands to copy this disk as these commands will fail to copy the directories correctly. The directory hierarchy created by `pax` requires about 2Gb of disk space.

After inserting the second UnitedLinux 1.0 SP 3 disk:

```
mount /mnt/cdrom
mkdir /tftp/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD2
cd /mnt/cdrom
pax -rw . /tftp/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD2
cd /
umount /mnt/cdrom
```

When you have copied all the disks, tie the ISO images together:

```
cd /tftp/sles-8sp3
    sh ./create-glue
```

5. **Modify the configuration file to specify the address of your NFS server.**

- For Red Hat, modify the `/tftp/`*<Linux_dir>*`/sun/install/ks.cfg` file. For example:

```
nfs --server 172.16.13.8 --dir /tftp/<Linux_dir>/
mount -t nfs -o nolock 172.16.13.8:/tftp/<Linux_dir> /mnt
```

where *<Linux_dir>* is `as-2.1u2` or `el-3.0`, depending on the version of Red Hat you are installing.

Note – ks.cfg is a read-only file. You must change its permissions to read-write before making modifications.

- For SuSE, modify the /tftp/sles-8sp3/sun/install/autoyast.xml file to set the NFS server address. A sample command is as follows:

```
mount -t nfs -o nolock 172.16.11.8:/tftp/sles-8sp3 $MOUNTPT
install: nfs://172.16.11.8/sles-8sp3
  <server>172.16.11.8</server>
```

6. Set your own root password in the Linux configuration file.

Note – If you do not change the root password, you will be prompted to enter the root password each time you run a PXE boot installation.

- For Red Hat, modify the /tftp/*<Linux_dir>*/sun/install/ks.cfg file by removing the comment symbol (#) in the rootpw entry and then overwriting changeme with your own password:

```
#rootpw changeme
```

For example:

```
rootpw nnnnnnn
```

where *nnnnnnn* is your root password.

- For SuSE, specify the root password in the autoyast.xml file (/tftp/sles-8sp3/sun/install/autoyast.xml) by scrolling to the user password section of the file, removing the existing text between the <user_password> key words, typing the password you want to use:

```
<user>
         <encrypted config:type="boolean">true</encrypted>
         <!-- Define the root password here using the <user_password>    -->
         <!-- tag.  The specified password must be encrypted...  Use     -->
         <!-- the following command to get the encrypted form of (for    -->
         <!-- example) a password of ‘‘changeme’’:                         -->
         <!--    perl -e ’print crypt("changeme", "/."), "\n"’           -->
         <user_password>/.hz7/JN74p1I</user_password>
         <username>root</username>
```

Note – It is only possible to specify passwords for SuSE in an encrypted form.

Note – The default password is `changeme`.

---

7. **Modify the `/tftp/`*`Linux_dir`*`/sun/pxelinux.cfg/default` file to include the path to the kernel to be installed, and the location of the PXE server.**

   The line of the `default` file containing the IP address of the PXE server and the path to the kernel software is the wrapped line beginning with the word "kernel" and ending "`/initrd.img`":

```
serial 0 9600
default Enterprise-Linux-3.0
display pxelinux.cfg/bootinfo.txt
prompt 1
timeout 50
label Enterprise-Linux-3.0
kernel ../images/pxeboot/vmlinuz
append ksdevice=eth0 console=ttyS0,9600n8 load_ramdisk=1 network ks=nfs:
172.16.11.8:/tftp/<Linux_dir>/sun/install/ks.cfg initrd=install/initrd.img
```

   where <*Linux_dir*> is `as-2.1u2` or `el-3.0`, depending on the version of Red Hat you are installing. If you are installing SuSE Linux Enterprise Server 8 service pack 3, the Linux directory will be `sles-8sp3`.

Note – By default the PXE device is eth0 (`ksdevice=eth0`). This means that the PXE boot is performed via the SSC in slot 0. If you want to PXE boot via SSC 1, you can change this parameter to `ksdevice=eth1`.

Note – The `default` file is a read-only file. You must change its permissions to read-write before making modifications.

---

8. **Log into the B1600 System Controller.**

   See the *Sun fire B1600 Blade System Chassis Software Setup Guide* for further details.

Note – The following steps assume that the blade is already installed in the system chassis. For information on installing blades, see Chapter 3.

---

9. **Boot the blade to begin the PXE boot from the SC prompt.**

```
sc> bootmode bootscript="boot net" sn
sc> poweron sn (if the blade is currently off)
sc> reset sn (if the blade is currently on)
```

where *n* is the slot number of the server blade on which you want to install the operating system.

10. **Access the blade's console to monitor the progress of the installation.**

At the SC prompt, type:

```
sc> console sn
```

where *n* is the number of the slot containing the blade.

Note – If you are installing SuSE, the system will become idle for about 40 seconds during the boot and subsequent reboots. During this idle time a blank screen is displayed. This behavior is due to an old version of the bootloader that ships with SuSE, and does not indicate that there is a problem with booting the blade.

When the installation is complete the blade automatically reboots.

Note – For information on troubleshooting the PXE boot installation see Chapter 9.

4.3 Installing Linux From a Solaris PXE Boot Server

This section tells you how to install Linux on a server blade from a PXE boot server running Solaris.

Note – IMPORTANT: Before installing Linux, ensure that the boot directory on the PXE server (`/tftpboot`) has enough space to accommodate the version of Linux you are installing. You will require about 6 Gbytes of free space.

4.3.1 Files Relevant to PXE Boot Installation

A summary of the files required by the Solaris PXE boot server during PXE boot installation and their purpose is provided in TABLE 4-3.

TABLE 4-3 Summary of Files Relevant to PXE Boot Installation

Filename	Purpose
/etc/dfs/dfstab	The NFS server is used by the installation kernel to read the packages necessary to the installation process. The NFS server needs to provide access to the directory structure containing the required packages. Prior to installation you will update the /etc/dfs/dfstab file to provide access to this directory structure.
/tftpboot/<*Linux_dir*>/sun/install/ks.cfg or: /tftpboot/sles-8sp3/sun/install/autoyast.xml	The Red Hat PXE boot installation is controlled by the ks.cfg configuration file. The SuSE PXE boot installation is controlled by the autoyast.xml file. Prior to installation you will update this file to use the correct NFS server address. For more information on the configuration file for your version of Linux, refer to your Red Hat or SuSE documentation.
/tftpboot/<*Linux_dir*>/sun/pxelinux.cfg/*	The /tftpboot/<*Linux_dir*>/sun/pxelinux.cfg/* files control where pxelinux.bin finds a kernel to boot from and how it should boot that kernel. The files in this directory are named based on the IP address that should read them. For example, if the client is given an IP address of 9.10.11.12, pxelinux.bin will attempt to download (using TFTP and the PXE NIC support code) the following files in order: pxelinux.cfg/090A0B0C pxelinux.cfg/090A0B0 pxelinux.cfg/090A0B pxelinux.cfg/090A0 pxelinux.cfg/090A pxelinux.cfg/090 pxelinux.cfg/09 pxelinux.cfg/0 pxelinux.cfg/default The first file downloaded successfully is used to select the kernel image and runtime arguments.
/etc/inet/inetd.conf	The TFTP server supplies the PXE boot with the stage 1 bootloader image. This image loads the installation kernel that performs the installation on the hard disk. The inetd daemon must be configured to run a TFTP daemon. This TFTP daemon supplies the services necessary to download the PXE loader, the linux kernel and the linux initrd image.
/var/dhcp/*	The DHCP server supplies the PXE boot plug-in with an IP address and TFTP server address, and the stage 1 image boot-loader name to download and execute. The instructions in this chapter tell you how to modify these files using the DHCP Manager utility.

Note – The Linux directory called *<Linux_dir>* depends on the version of Linux you are installing. Files for Enterprise Linux Advanced Server 2.1 update 2 are in a directory called `as-2.1u2`, the files for Enterprise Linux version 3.0 are in a directory called `el-3.0`, and the files for SuSE Linux Enterprise Server 8 service pack 3 are in a directory called `sles-8sp3`.

4.3.2 Preparing to Install Linux

1. **Connect a network port on the SSC to a subnet containing both the Network Install Server you intend to use as the PXE boot server and the DHCP server you intend to use to allocate IP addresses to the server blade.**

   If you have a redundant SSC in the blade system chassis, duplicate this connection on the second SSC.

2. **Find out the MAC address of the first interface on the blade you intend to install Linux onto.**

   To do this, log into the System Controller, and at the `sc>` prompt, type:

```
sc>showplatform -v
:
:

Domain    Status         MAC Address        Hostname
--------  -----------    -----------------  -------------
S1        Standby        00:03:ba:29:e6:28  chatton-s1-0
S2        Standby        00:03:ba:29:f0:de
S6        OS Running     00:03:ba:19:27:e9  chatton-s6-0
S7        OS Stopped     00:03:ba:19:27:bd  chatton-s7-0
S10       Standby        00:03:ba:2d:d1:a8  chatton-s10-0
S12       OS Running     00:03:ba:2d:d4:a0  chatton-s12-0
:
SSC0/SWT  OS Running           00:03:ba:1b:6e:a5
SSC1/SWT  OS Running           00:03:ba:1b:65:4d
SSC0/SC   OS Running (Active)  00:03:ba:1b:6e:be
SSC1/SC   OS Running           00:03:ba:1b:65:66
:
sc>
```

where the : character indicates omitted data. The MAC address listed for each blade is the MAC address of the first interface (by default, `bge0`).

For a basic installation that uses only one active network interface (for example, for setting up a blade to boot Linux from the network), you only need the MAC address of the first network interface.

However, if you are intending to set up redundant connections to the network, you also need to calculate the MAC addresses for `bge1`, `bge2` , and `bge3`.

Make a note of the MAC addresses for each interface on the blade.

3. **Make sure the DHCP server you intend to use is properly set up and functioning.**

   For information about setting up a Solaris DHCP server, refer to the *Solaris DHCP Administration Guid*e.

4. **If you want the DHCP server to allocate IP addresses dynamically to the server blade, then reserve a block of addresses on the DHCP server for this purpose.**

   For information about how to do this, refer to the *Solaris DHCP Administration Guid*e.

4.3.3 Configuring the PXE Boot Servers

Linux is installed on the server blade using the PXE boot system. Three server processes are required to perform the installation:

- DHCP
- TFTP
- NFS

This section provides information on how to configure the DHCP and NFS servers, and how to enable the TFTP server, for use with the PXE boot installation.

Note – This chapter assumes that all server processes are running on the same physical host.

4.3.3.1 Configuring the DHCP Server

PXE booting is supported by DHCP services, and this means that there are a number of setup steps you need to perform involving the DHCP server. The DHCP server needs to be configured for each individual blade otherwise the network installation will not work.

1. **Log into the Network Install Server as** `root`**, and start the DHCP Manager by typing:**

```
# DISPLAY=mydisplay:0.0
# export DISPLAY
# /usr/sadm/admin/bin/dhcpmgr &
```

   where *mydisplay* is the name of the system (for example, a desktop workstation) that you are using to display the DHCP Manager's GUI (Graphical User Interface).

2. **Add the global PXE macro to the DHCP server to enable it to support Linux PXE boot clients.**

   To define the global PXE macro:

   a. **In the main window of DHCP Manager's GUI, click the Macros tab, and select Create from the Edit menu.**

   b. **In the Name field of the Create Macro window, type the name of the global macro that enables the DHCP server to support PXE booting (**`PXEClient:Arch:00000:UNDI:002001`**).**

Note – Step b only needs to be performed once on the DHCP server. If you already have this macro defined correctly, skip this step and go to Step c.

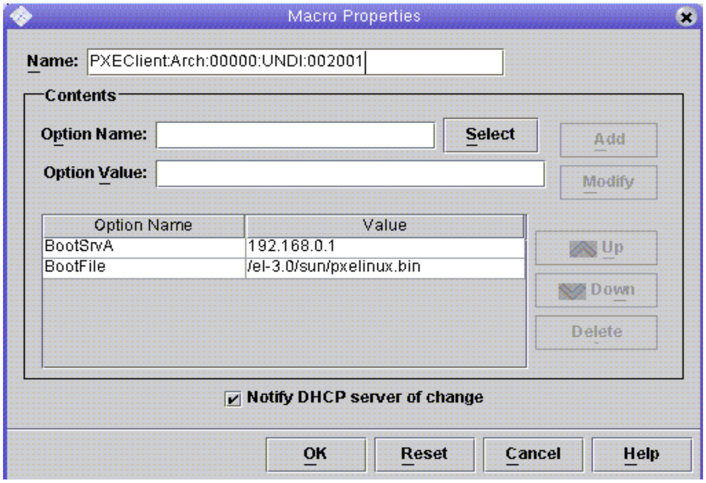
Caution – The global PXE macro is named `PXEClient:Arch:00000:UNDI:002001`. You must ensure that you type this name correctly. If you make a mistake, the blades will not be able to perform a PXE boot of the Linux operating system.

---

   c. **In the Option Name field, type** `BootSrvA`**. And in the Option Value field type the IP address that was listed for the Boot Server (that is, the Network Install Server). Then click Add.**

   d. **In the Option Name field, type** `BootFile`**. And in the Option Value field type the path to the file** `pxelinux.bin`**, for example** `/`*<Linux_dir>*`/sun/pxelinux.bin`**, (where** *<Linux_dir>* **is either** `as-2.1u2`**,** `el-3.0` **or** `sles-8sp3`**, depending on the version of Linux you are installing). Then click Add.**

   To view the properties of the macro you have created, select it from the list of macros displayed on the left of the Macros tab, then select Properties from the Edit menu (see FIGURE 4-1).



Name: PXEClient:Arch:00000:UNDI:002001

Option Name	Value
BootSrvA	192.168.0.1
BootFile	/el-3.0/sun/pxelinux.bin

☑ Notify DHCP server of change

FIGURE 4-1 The Properties Defined for the Global PXE Macro

3. **Click OK to save the settings.**

4.3.3.2 Configuring the NFS Server

The NFS server is used by the installation kernel to read all of the packages necessary to the installation process. The NFS server therefore needs to provide access to the directory structure containing the PXE images.

1. **Make the** `tftpboot` **directory available to all machines running NFS.**

   Update the `/etc/dfs/dfstab` file by adding the following line:

```
share -F nfs -o rw -d "TFTP boot directory" /tftpboot
```

CODE EXAMPLE 4-2 Sample /etc/dfs/dfstab file

```
:
# more dfstab

# Place share(1M) commands here for automatic execution
# on entering init state 3.
#
# Issue the command '/etc/init.d/nfs.server start' to run the NFS
# daemon processes and the share commands, after adding the very
# first entry to this file.
#
# share [-F fstype] [ -o options] [-d "<text>"] <pathname>
[resource]
# .e.g,
# share  -F nfs  -o rw=engineering  -d "home dirs"  /export/home2

share -F nfs -o rw -d "TFTP boot directory" /tftpboot
share -F nfs -o ro,anon=0
/export/install/media/s9u5_cd1combined.s9x_u5wos.08
share -F nfs -o ro,anon=0 /export/install/DVDimages
share -F nfs -o ro,anon=0 /export/install/media/s9u5cd_test
share -F nfs -o ro,anon=0 /export/install/s9u5mis
:
```

2. **Save the** `/etc/dfs/dfstab` **file.**

3. **Share the resources in the** `/etc/dfs/dfstab` **file:**

```
# shareall
```

4. **Validate the configuration by looking in the** `/etc/dfs/sharetab` **file. This file should contain the entry** `/tftpboot`.

4.3.3.3 Enabling the TFTP Server

1. **Modify the** `/etc/inet/inetd.conf` **file to enable the TFTP server.**

Note – `inetd.conf` is a read only file. You must change its permissions to read-write before making modifications.

Remove the comment out symbol (#) from the `tftp` line:

```
# tftp    dgram   udp    wait    root    /usr/sbin/in.tftpd      in.tftpd -s
/tftpboot
```

2. **Save the** `/etc/inet/inetd.conf` **file.**
3. **Restart** `inetd`**:**

```
# pkill -HUP inetd
```

4.3.4 Installing Linux on a Server Blade from a Solaris PXE Boot Server

---

Note – IMPORTANT: Before installing Linux, ensure that the boot directory on the PXE server (`/tftpboot`) has enough space to accommodate the version of Linux you are installing. You will require about 6 Gbytes of free space.

---

1. **Ensure that the DHCP server, NFS server and TFTP server have been configured correctly.**

   See Section 4.3.3, "Configuring the PXE Boot Servers" on page 4-23 for more information.

2. **Install the PXE images onto the TFTP server:**

   a. **Copy the Linux directory from the root of the** *Sun Fire B1600 Platform Documentation, Drivers, and Installation* **CD to the** `/tftpboot` **directory on your PXE boot server:**

Note – The following example assumes that Volume Management is running on the server.

```
# volcheck
# cd /cdrom/cdrom0
# egrep '^<Linux_dir>' filenames.txt | cpio -pumd /tftpboot/.
# cd /
# eject cdrom
```

where *<Linux_dir>* is `as-2.1u2`, `el-3.0`, or `sles-8sp3`, depending on the version of Linux you are installing.

Note – The linux directory contains the files required to perform a PXE installation.

b. Install the Linux installation CDs to the `/tftpboot` directory on your PXE boot server.

- For Red Hat, you need to install the CDs in reverse order. If you have two Red Hat installation CDs, install Disk 2 first; if you have four, install Disk 4 first.

Note – The following example assumes that Volume Management is running on the server.

After inserting each CD, type the following command:

```
# volcheck
# cd /cdrom/cdrom0
# tar -cf - . | (cd /tftpboot/<Linux_dir>; tar xf -)
# cd /
# eject cdrom
```

where *<Linux_dir>* is `as-2.1u2` or `el-3.0`, depending on the version of Linux you are installing.

Note – You only need to copy the installation CDs. Any source RPM, administration, or documentation discs are not used by the PXE server.

- For SuSE Linux Enterprise Server 8 service pack 3, you need to load each image into its own directory rather than into the same directory. This allows the SuSE installer to select the correct packages from each ISO image. Use the following commands:

Note – If you are running SuSE on your PXE boot server, replace **`/mnt/cdrom`** with **`/media/cdrom`** in the instructions below. For example, **`mount /mnt/cdrom`** would be **`mount /media/cdrom`**.

After inserting the SLES-8 disk:

```
mount /mnt/cdrom
mkdir /tftpboot/sles-8sp3/SLES-8-i386-RC5-CD1
cd /mnt/cdrom
pax -rw . /tftpboot/sles-8sp3/SLES-8-i386-RC5-CD1
cd /
umount /mnt/cdrom
```

After inserting the first UnitedLinux 1.0 disk:

```
mount /mnt/cdrom
mkdir /tftpboot/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD1
cd /mnt/cdrom
pax -rw . /tftpboot/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD1
cd /
umount /mnt/cdrom
```

After inserting the second UnitedLinux 1.0 disk:

```
mount /mnt/cdrom
mkdir /tftpboot/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD2
cd /mnt/cdrom
pax -rw . /tftpboot/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD2
cd /
umount /mnt/cdrom
```

After inserting the third UnitedLinux 1.0 disk:

```
mount /mnt/cdrom
mkdir /tftpboot/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD3
cd /mnt/cdrom
pax -rw . /tftpboot/sles-8sp3/UnitedLinux-1.0-i386-RC5-CD3
cd /
umount /mnt/cdrom
```

After inserting the first United Linux 1.0 SP 3 disk:

```
mount /mnt/cdrom
mkdir /tftpboot/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD1
cd /mnt/cdrom
pax -rw . /tftpboot/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD1
cd /
umount /mnt/cdrom
```

After inserting the second UnitedLinux 1.0 SP 3 disk:

```
mount /mnt/cdrom
mkdir /tftpboot/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD2
cd /mnt/cdrom
pax -rw . /tftpboot/sles-8sp3/UnitedLinux-1.0-SP-3-i386-RC4-CD2
cd /
umount /mnt/cdrom
```

Note – The first SP 3 disk contains hard-linked directories. Do not use the `cp`, `cpio` or `tar` commands to copy this disk as these commands will fail to copy the directories correctly. The directory hierarchy created by `pax` requires about 2Gb of disk space.

When you have copied all the disks, tie the ISO images together:

```
cd /tftpboot/sles-8sp3
    ksh ./create-glue
```

Note – The first SP 3 disk contains hard-linked directories. You must not use the `cp, cpio` or `tar` commands to copy this disk as these commands will fail to copy the directories correctly. The directory hierarchy created by `pax` requires about 2Gb of disk space.

---

3. **In the configuration file, replace the directory name `tftp` with `tftpboot`.**

- For Red Hat, modify the `/tftpboot/`*<Linux_dir>*`/sun/install/ks.cfg` file by replacing all instances of `tftp` with `tftpboot`.

Note – `ks.cfg` is a read-only file. You must change its permissions to read-write before making modifications.

- For SuSE, modify the `/tftpboot/sles-8sp3/sun/install/autoyast.xml` file file by replacing all instances of `tftp` with `tftpboot`.

4. **Modify the configuration file to specify the address of your NFS server.**

- For Red Hat, modify the `/tftpboot/`*<Linux_dir>*`/sun/install/ks.cfg` file.

  For example:

```
nfs --server 172.16.13.8 --dir /tftpboot/<Linux_dir>/
mount -t nfs -o nolock 172.16.13.8:/tftpboot/<Linux_dir> /mnt
```

  where *<Linux_dir>* is `as-2.1u2`, `el-3.0` or `sles-8sp3`, depending on the version of Red Hat you are installing.

- For SuSE, modify the `/tftpboot/sles-8sp3/sun/install/autoyast.xml` file to set the NFS server address. Sample configurations are:

```
   mount -t nfs nolock 172.16.11.8:/tftpboot/sles-9 $MOUNTPT
install: nfs://172.16.11.8/tftpboot/sles-8sp3
  <server>172.16.11.8</server>
```

5. **Set your own root password in the Linux configuration file.**

Note – If you do not change the root password, you will be prompted to enter the root password each time you run a PXE boot installation.

- For Red Hat, modify the /tftpboot/<*Linux_dir*>/sun/install/ks.cfg file by removing the comment symbol (#) in the rootpw entry and then overwriting changeme with your own password:

```
#rootpw changeme
```

For example:

```
rootpw nnnnnnn
```

where *nnnnnnn* is your root password.

- For SuSE, specify the root password in encrypted form in the autoyast.xml file (/tftpboot/sles-8sp3/sun/install/autoyast.xml) by doing the following:

a. Generate an encrypted password for the root password:

```
# perl -e ’print crypt("nnnnnnn", "/."), "\n"’
```

where *nnnnnnn* is your root password

b. Scroll to the user password section of the autoyast.xml **file, remove the existing text between the** <user_password> **keywords, and type the encrypted password that you generated in Step a. Sample lines from the** autoyast.xml **file are:**

```
    <user>
          <encrypted config:type="boolean">true</encrypted>
          <!-- Define the root password here using the <user_password>    -->
          <!-- tag.  The specified password must be encrypted...  Use     -->
          <!-- the following command to get the encrypted form of (for    -->
          <!-- example) a password of ‘‘changeme’’:                        -->
          <!--    perl -e ’print crypt("changeme", "/."), "\n"’          -->
          <user_password>/.hz7/JN74p1I</user_password>
          <username>root</username>
```

Note – You can only specify passwords for SuSE in an encrypted form.

Note – The default password is changeme.

6. **Modify the /**`tftpboot/`*<Linux_dir>*`/sun/pxelinux.cfg/default` **file to include the path to the kernel to be installed, and the location of the PXE server.**

Note – The `default` file is a read-only file. You must change its permissions to read-write before making modifications.

For example (Red Hat):

```
kernel ../images/pxeboot/vmlinuz
append ksdevice=eth0 console=ttyS0,9600n8 load_ramdisk=1 network ks=nfs:
172.16.11.8:/tftpboot/<Linux_dir>/sun/install/ks.cfg initrd=
install/initrd.img
```

where *<Linux_dir>* is `as-2.1u2` or `el-3.0`, depending on the version of Red Hat you are installing.

For example (SuSE):

```
kernel ../boot/loader/linux
append insmod=suntg3 load_ramdisk=1 network console=ttyS0,9600n8 initrd=
install/initrd.img install=nfs://172.16.11.8/tftpboot/sles-8sp3 autoyast=
nfs://172.16.11.8/tftpboot/sles-8sp3/sun/install/autoyast.xml
```

Note – The `tftp` directory must be changed to `tftpboot,` as shown in the examples.

Note – By default the PXE device is eth0 (`ksdevice=eth0`). This means that the PXE boot is performed via SSC0. If you want to PXE boot via SSC 1, you can change this parameter to `ksdevice=eth1`.

7. **Log into the B1600 System Controller.**

   See the *Sun fire B1600 Blade System Chassis Software Setup Guide* for further details.

8. **Boot the server blade to begin the PXE boot from the SC prompt.**

```
sc> bootmode bootscript="boot net" sn
sc> poweron sn (if the blade is currently off)
sc> reset sn (if the blade is currently on)
```

where sn is the physical location of the server blade on which you want to install the operating system.

9. **Access the blade's console to monitor the progress of the installation.**

At the SC prompt, type:

```
sc> console sn
```

where sn is the physical location of the server blade.

Note – If you are installing SuSE, the system will become idle for about 40 seconds during the boot and subsequent reboots. During this idle time a blank screen is displayed. This behavior is due to an old version of the bootloader that ships with SuSE, and does not indicate that there is a problem with booting the blade.

When the installation is complete the blade automatically reboots.

Note – For information on troubleshooting the PXE boot installation see Chapter 9.

CHAPTER 5

Setting Up Server Blades

This chapter tells you how to power on a server blade and access its console. The chapter contains the following sections:

- Section 5.1, "Configuring the Server Blade to Boot From the Network" on page 5-2
- Section 5.2, "Powering On and Booting the Server Blade" on page 5-3

Note – Before you set up the server blade, you must build a PXE boot install environment. See Section 4.2.2, "Configuring the PXE Boot Servers" on page 4-6.

5.1 Configuring the Server Blade to Boot From the Network

Before you can use a Linux blade, you need to configure it temporarily to boot from the network. This is to enable it to perform the PXE boot process by which it first receives its operating system.

Type the following command at the System Controller's `sc>` prompt to cause the blade to boot from the network

```
sc> bootmode bootscript="boot net" sn
```

where *n* is the number of the slot containing the blade.

Note – This command is effective for 10 minutes. After that the BIOS reverts to its previous booting behavior. Therefore, to cause the blade to boot from the network you must power it on within 10 minutes of running the bootmode command. If the blade was already powered on when you ran the bootmode command, then to cause it to boot from the network you must reset the blade within 10 minutes by typing:

```
sc> reset sn
```

5.2 Powering On and Booting the Server Blade

When you are ready, power on a server blade and boot it by following the instructions below:

1. **Power on the server blade.**

   Type:

   ```
   sc> poweron sn
   ```

   where *n* is the number of the slot containing the server blade.

2. **Log into the console of the server blade to view (and/or participate in) the booting process.**

   Type the following at the `sc>` prompt to access the blade's console:

   ```
   sc> console sn
   ```

   where *n* is the number of the slot containing the blade.

Note – Whenever you are at a blade console, type #. to return to the active System Controller.

CHAPTER 6

Manually Installing the B100x and B200x Linux Kernel Drivers

This chapter provides information on how to rebuild and reinstall the Linux drivers for a B100x or B200x kernel upgrade. It includes the following sections:

- Section 6.1, "Introduction" on page 6-2
- Section 6.2, "Before Upgrading the Linux Kernel" on page 6-2
- Section 6.3, "After Upgrading the Linux Kernel" on page 6-3

Note – This chapter does not describe how to perform a kernel upgrade. For information on how to upgrade your kernel, refer to the documentation for the version of Linux you have installed.

6.1 Introduction

The Linux kernel provides the underlying services to the rest of a Linux distribution. If you replace the Linux kernel, you must reinstall the blade kernel drivers in the new kernel environment. Failure to reinstall the kernel drivers may result in loss of network connectivity and will result in the loss of other facilities such as network failover and BSC services.

6.2 Before Upgrading the Linux Kernel

Before you upgrade the Linux kernel you must copy the driver source files on to the blade. This is necessary in case the kernel upgrade results in the loss of network connectivity.

When selecting the drivers to be built in the new environment, the latest version of the drivers for the system you are upgrading should be used. Use the following table to determine the driver directory you require:

Installed OS	Driver Directory
Red Hat Enterprise Linux, Advanced Server 2.1	`/src/as-2.1u3`
Red Hat Enterprise Linux, version 3.0	`/src/el-3.0u1`
SuSE Linux Enterprise Server 8	`/src/sles-8sp3`

- **Copy the driver files from a server that has the Sun drivers installed:**

```
mkdir /root/build
cd /root/build
scp server:/src/common/install/memdiag/memdiag-
1.0/driver/highmem.c .
scp server:/src/common/install/bios/mtdbios.c .
scp server:/src/common/install/bsc/*.* .
scp server:/src/common/install/failover/failover.? .
scp server:/src/common/install/pwrbtn/pwrbtn.c .
scp server:/src/common/install/sunecc/sunecc.c .
scp server:<DriverDir>/install/suntg3/suntg3.? .
scp server:<DriverDir>/install/pci_ids.h .
```

where <*DriverDir*> is the required driver directory listed in the table above.

6.3 After Upgrading the Linux Kernel

1. **Ensure that the system compiler is installed:**

```
rpm -q -a | fgrep gcc
```

If gcc is not installed, it must be installed using the `rpm -i` command.

2. **Ensure that the kernel sources are installed:**

```
rpm -q -a | fgrep kernel-sources
```

If the kernel sources are not installed, they must be installed using the `rpm -i` command.

3. **Remove any kernel build files that are not required:**

```
cd /usr/src/linux-<kernel version>
find . -name .depend | xargs rm -f
find include/linux/modules ( -name \*.ver -o -name \*.stamp ) | xargs rm -f
rm -f include/linux/autoconf.h
```

where <*kernel version*> is the version of the kernel you have upgraded to.

4. **Modify the kernel `Makefile` to match your kernel:**

```
sed 's/custom/smp/' Makefile >Makefile.new && mv -f Makefile.new Makefile
```

Note – If you are running on a uniprocessor kernel, change the `sed` argument to `s/custom//`

---

5. **Finish removing build files that are not required, install the configuration and prepare the environment:**

```
make mrproper
cp configs/kernel-<kernel version>-i686-smp.config .config
make oldconfig
make dep
```

where <*kernel version*> is the version of the kernel you have upgraded to.

Note – If you are running on a uniprocessor kernel, change the configuration file name to `kernel-`<*kernel version*>`-athlon.config`

---

6. **Change the directory to the location of the driver files, and build the drivers:**

```
cd /root/build
KINC=/usr/src/linux-<kernel version>/include
INC="-I. -I$KINC -include $KINC/linux/modversions.h"
CFLAGS="$INC -Wall -O2 -D__KERNEL__ -DMODULE -DMODVERSIONS -
DEXPORT_SYMTAB"
rm -f linux
ln -s . linux

cc -c $CFLAGS -o suntg3.o suntg3.c
cc -c $CFLAGS -o bsc.o bsc.c
cc -c $CFLAGS -o sunecc.o sunecc.c
cc -c $CFLAGS -o failover.o failover.c
cc -c $CFLAGS -o highmem.o highmem.c
cc -c $CFLAGS -o pwrbtn.o pwrbtn.c

mtdd=/usr/src/linux-<kernel version>/drivers/mtd
cc -c $CFLAGS -I$mtdd -o mtdcore.o $mtdd/mtdcore.c
cc -c $CFLAGS -I$mtdd -o mtdchar.o $mtdd/mtdchar.c
cc -c $CFLAGS -I$mtdd -o mtdbios.o mtdbios.c
```

where <*kernel version*> is the version of the kernel you have upgraded to.

7. **Install the drivers:**

```
mkdir -p /lib/modules/<kernel version>smp/kernel/misc
mkdir -p /lib/modules/<kernel version>smp/kernel/drivers/mtd
mv -f suntg3.o /lib/modules/<kernel version>smp/kernel/drivers/net/suntg3.o
mv -f bsc.o /lib/modules/<kernel version>smp/kernel/drivers/misc/bsc.o
mv -f sunecc.o /lib/modules/<kernel version>smp/kernel/drivers/char/sunecc.o
mv -f failover.o /lib/modules/<kernel version>smp/kernel/drivers/net/failover.o
mv -f highmem.o /lib/modules/<kernel version>smp/kernel/drivers/char/highmem.o
mv -f mtdcore.o /lib/modules/<kernel version>smp/kernel/drivers/mtd/mtdcore.o
mv -f mtdchar.o /lib/modules/<kernel version>smp/kernel/drivers/mtd/mtdchar.o
mv -f mtdbios.o /lib/modules/<kernel version>smp/kernel/drivers/mtd/mtdbios.o
mv -f pwrbtn.o /lib/modules/<kernel version>smp/kernel/drivers/misc/pwrbtn.o
```

where <*kernel version*> is the version of the kernel you have upgraded to.

Note – If you are running on a uniprocessor kernel, remove the `smp` part of the path name.

---

8. **Recreate the** `initrd` **file:**

- For Red Hat, type:

```
mkinitrd -f --with=suntg3 --with=bsc --with=sunecc --with=pwrbtn \
    /boot/initrd-<kernel version>smp.img <kernel version>smp
```

where <*kernel version*> is the version of the kernel you have upgraded to.

Note – If you are running on a uniprocessor kernel, remove the `smp` part of the path name.

- For SuSE, type:

```
mkinitrd
lilo
```

The `lilo` command is required only if you are using the LILO bootloader. If you are using the GRUB bootloader, only the `mkinitrd` command is required.

9. **Restart your system and select the new kernel from the boot menu.**

CHAPTER 7

Using Linux Blades in Separated Data and Management Networks

This chapter contains the following sections:

7.1 SunFire B1600 Network Topology Overview

This chapter tells you how to set up the Sun Fire B1600 blade system chassis for use in an environment that separates the data and management networks. If you have dual SSCs installed in the chassis, the instructions enable you to take advantage of the presence of two switches to give the server blades two connections to each of your networks.

Note – If you have dual SSCs installed, then, when you are considering how to integrate the chassis into your network environment, you need to remember that the chassis contains two switches. Although only one of its System Controllers is active at any one time, both of its switches are active all the time. This means that, in a system chassis that is working normally, both switches are providing the server blades with continuous network connectivity. However, if for any reason one switch fails, the other switch continues to provide network connectivity. (Also, if either System Controller fails, the switch inside the same SSC module continues to provide network connectivity; the switches operate independently of the System Controllers even though they are physically located in the same enclosure.)

This chapter also explains how to take advantage of the presence of two switches by using failover and link aggregation to provide fully redundant connections from Linux server blades to the data and management networks.

To take advantage of the redundancy offered by the second switch inside the system chassis, we recommend you to:

- Operate the system chassis always with two SSCs installed.
- Make sure that the cable connections from the eight uplink ports to the subnets on your wider network are exactly duplicated on the eight uplink ports of the second switch.
- Copy the configuration file of the first switch you configure over to the redundant switch before setting the IP address, netmask, and default gateway for the switch. For information about how to do this, see the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.
- Specify an IP address (in the `/etc/hosts` file) suitable for a Failover interface configuration that supports redundant interfaces to the data network and the management network from each server blade.

7.1.1 Preparing the Network Environment Using DHCP

If you are using DHCP, make sure that the DHCP server for the System Controllers and switches is on the management network, and that the DHCP server for the blades is on the data network.

Note – The example in "Preparing the Network Environment Using DHCP" on page 7-3 uses static IP addresses, not DHCP.

For information on setting up the `etc/dhcp.conf` file, see Chapter 4.

7.1.2 Sun Fire B1600 Network Environment Using Static IP Addresses

FIGURE 7-1 shows a sample network configuration with the 100Mbps network management port (NETMGT) on both SSCs connected to a different switch from the data uplink ports. This external switch is on a different subnet than the switch that the data uplink ports on the chassis are connected to. It is a subnet dedicated to network management traffic and it therefore also contains both of the System Controllers and switches in the chassis. A management VLAN (VLAN 2) contains both System Controller interfaces and both switch management ports. All the server blades and uplink ports are on the untagged VLAN 1.

FIGURE 7-1 shows the connection of the `snet0` interface on B100x blades to the switch in SSC0, and the connection of the `snet1` interface on B100x blades to the switch in SSC1. It also shows the connection of `snet0` and `snet2` interfaces a B200x blade to the switch in SSC0, and the connection of the `snet1` and `snet3` interfaces on a B200x blade to the switch in SSC1. The IP address of the blade is used by the Failover interfaces to enable failover and link aggregation (see Section 7.4.1, "Setting up Linux Server Blades Using the Failover Interface Driver for Network Resiliency" on page 7-20).

One or more of the eight uplink ports on each switch in FIGURE 7-1 are connected to an external switch that has an Install Server connected to it. This external switch also has a router (with IP address 192.168.1.1) connected to it that acts as the default gateway from the chassis to the wider network.

Note – Note that there is no direct network connection in FIGURE 7-1 from the management port (NETMGT) in the switch to the server blade ports. This means that, by default, you cannot manage the server blades directly from the management network. This is a security feature to protect the management network from the possibility of hostile attack from the data network. For information about permitting specified traffic from the server blades to the management port, see the example in "Example Network Configuration" on page 7-24.

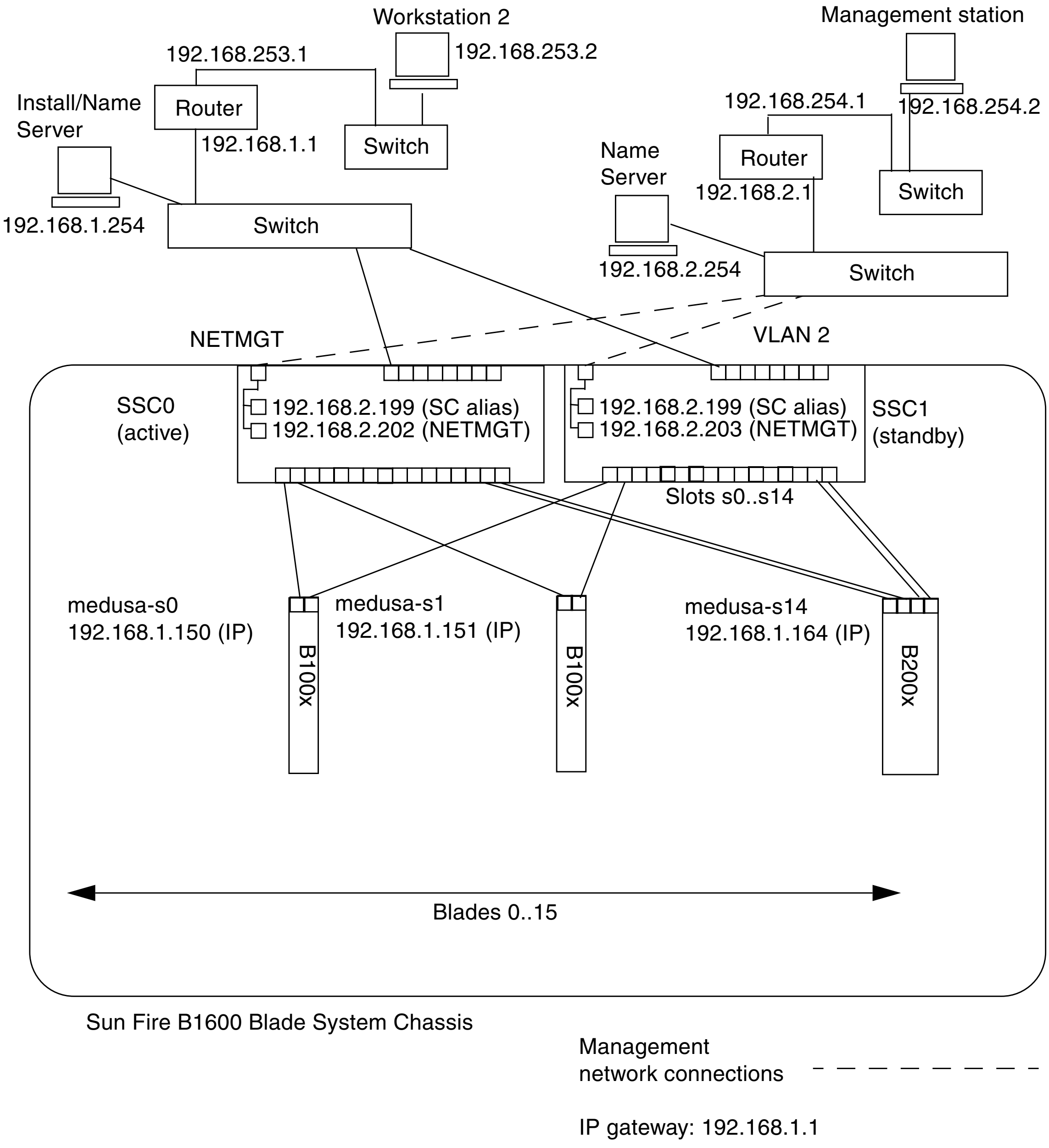


FIGURE 7-1 Sample Network Configuration Using a Management VLAN

7.1.3 Configuring the System Controllers and Switches

To configure the System Controllers and switches for the type of configuration illustrated in FIGURE 7-1, follow the instructions in the Software Setup Guide. However, remember that the IP addresses you assign to the System Controllers and switches need to be on the management subnet.

7.1.4 Configuring Network Interfaces

To set up a fully configured blade that provides redundant connections to the data and management networks, you will need to configure a number of interfaces.

There are four types of network interface:

- Physical interfaces

  These are the standard physical Gigabit Ethernet interfaces on the blade. On a B100x blade, these are `snet0` and `snet1`. On a B200x blade these are `snet0` and `snet1`, and `snet2` and `snet3`.

  To provide consistency with interface order, the standard physical Ethernet interfaces have been renamed from "`eth`" to "`snet`".

- Bonding interfaces (B200x blades only)

  Bonding interfaces use link aggregation to combine the four ethernet interfaces on a B200x blade into two pairs of interfaces, each with a single MAC address. Link aggregation provides 802.3ad interfaces called BOND0 and BOND1.

- VLAN interfaces

  VLAN interfaces are the virtual interfaces that can be configured on top of physical interfaces or bond interfaces. VLAN support is provided by the sun8021q driver.

- Failover interfaces

  Failover support for the switches in SSC0 and SSC1 is provided through failover redundancy interfaces called fail0 and fail1.

It is useful to think of these interfaces as layers, with the physical interface as the bottom layer, and the failover interface as the top layer. The example configurations in the next section demonstrate how these layered interfaces can be configured to provide failover.

Note – Only the top most interfaces in your configuration should have IP addresses configured (using either static IP or DHCP). Also, in the configuration files only the top most interface should have `ONBOOT` set to "`yes`" (when using Red Hat) or `startmode` set to "`ONBOOT`" (when using SuSE).

7.1.5 Example Network Interface Configurations

This section provides sample network interface configurations for server blades.

7.1.5.1 Failover Between the Physical Interfaces on a Blade

FIGURE 7-2 shows a failover interface (fail0) configured to provide redundancy between the physical interfaces `snet0` and `snet1` on a B100x server blade.

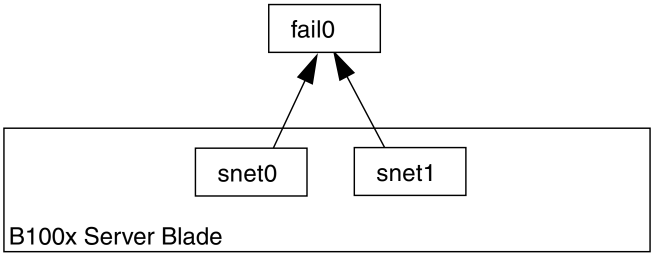


FIGURE 7-2 B100x Server Blade with `snet0` and `snet1` Configured for Failover

FIGURE 7-3 shows two failover interfaces (fail0 and fail1) configured to provide redundancy between two pairs of physical interfaces on a B200x server blade. Fail0 provides redundancy between `snet0` and `snet1`, and fail1 provides redundancy between `snet2` and `snet3`.

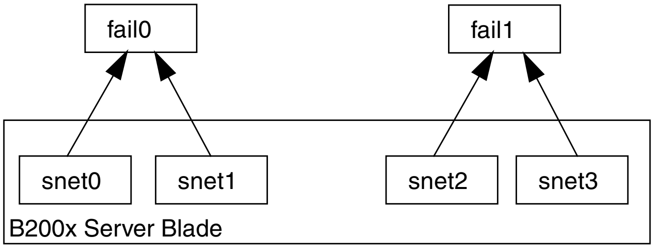


FIGURE 7-3 B200x Blade With `snet0` and `snet1`, and `snet2` and `snet3` Configured for Failover

7.1.5.2 Failover Between Bonding Interfaces

FIGURE 7-4 shows a B200x blade with a bonding interface layer configured to combine the four ethernet interfaces on the blade into two pairs of interfaces, each with a single MAC address. In the bonding interface layer, snet0 and snet2 become a single interface (BOND0), and snet1 and snet3 become a single interface (BOND1).

To enable failover between the two switches, a failover interface (fail0) has been configured on top of the bonding interface. Fail0 provides redundancy between BOND0 and BOND1.

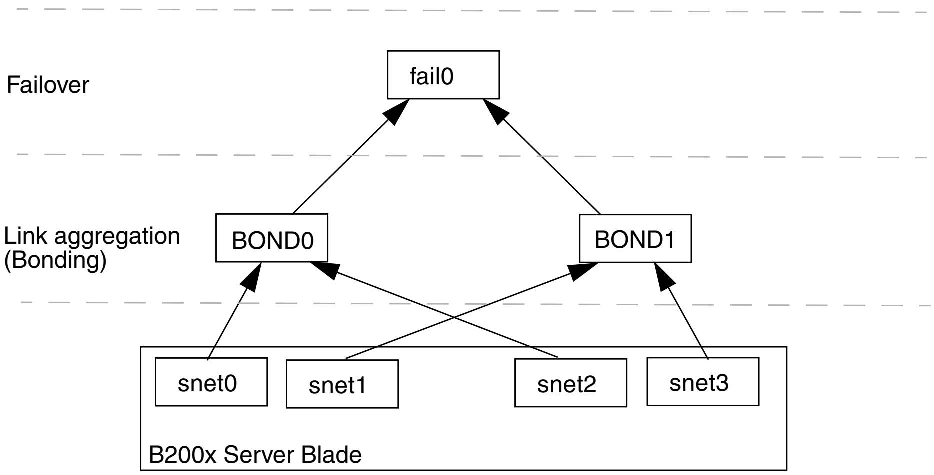


FIGURE 7-4 B200x Blade With Bonding Configured for Failover

7.1.5.3 VLAN Configured on a Physical Interface

FIGURE 7-5 shows a B100x blade with a VLAN3 interface configured on a physical interface (snet0). Note that the VLAN interface name comprises the name of the physical interface (snet0), followed by the VLAN number (.3). Therefore, in this example, the VLAN interface name is snet0.3.

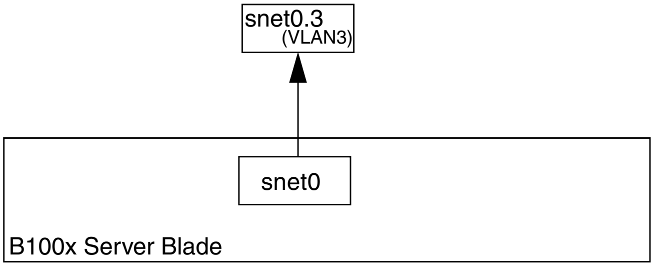


FIGURE 7-5 B100x Blade with `snet0.3` (VLAN3) configured on `snet0`

7.1.5.4 Failover Between VLAN Interfaces

shows a B100x server blade with two VLAN interfaces (`snet0.3` and `snet1.3`) configured on top of the physical interfaces (`snet0` and `snet1`). A failover interface (fail0) has been configured on top of the VLAN interface. Fail0 provides redundancy between `snet0.3` and `snet1.3`.

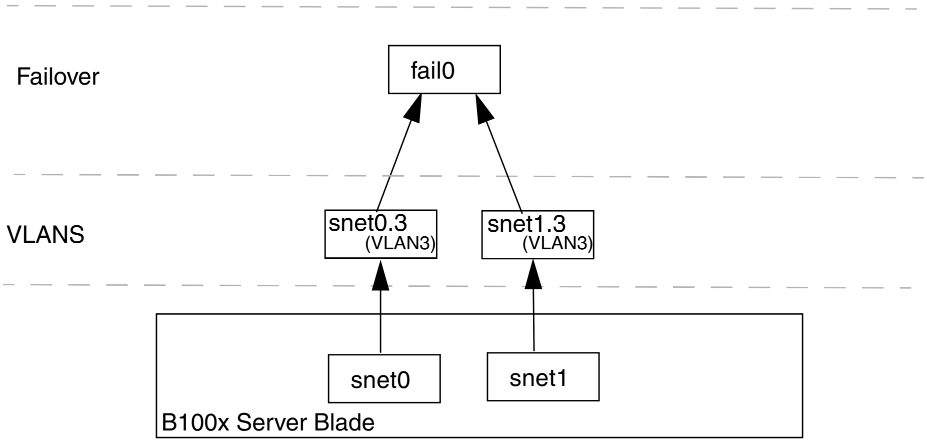


FIGURE 7-6 B100x Blade With Failover Between two VLAN interfaces

FIGURE 7-7FIGURE 7-7 shows a B200x server blade with four VLAN3 interfaces (snet0.3, snet1.3, snet2.3 and snet3.3) configured on top of four physical interfaces. Failover interface fail0 has been configured on top of snet0.3 and snet1.3, and fail1 has been configured on top of snet2.3 and snet3.3.

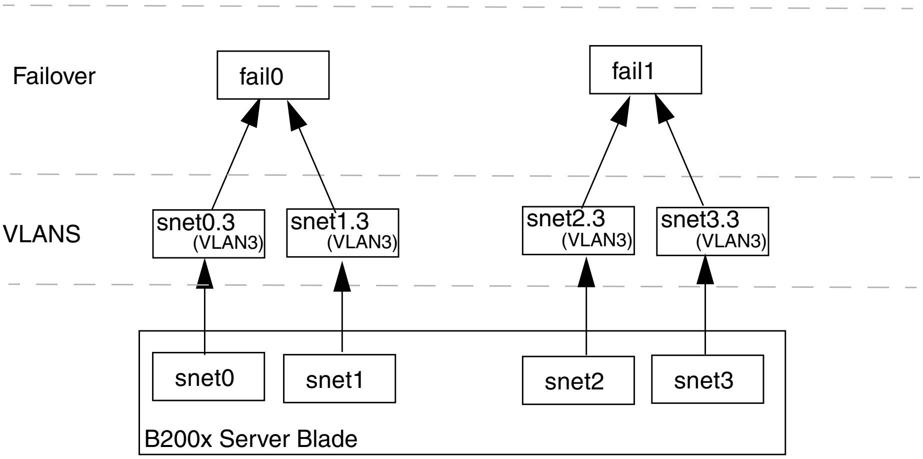


FIGURE 7-7 B200x Blade With Failover Between two VLANS

FIGURE 7-8 shows a B200x blade with failover between two VLAN interfaces that are configured on aggregated links.

A bonding interface layer has been configured to combine the four ethernet interfaces on the blade into two pairs of interfaces, each with a single MAC address. Therefore, in the bonding interface layer, snet0 and snet2 become a single interface (BOND0), and snet1 and snet3 become a single interface (BOND1).

A VLAN3 interface layer has been configured on top of the bonding interface layer to provide two VLAN interfaces called BOND0.3 and BOND1.3.

To enable failover between the two switches, a failover interface (fail0) has been configured on top of the VLAN interfaces. Fail0 provides redundancy between BOND0.3 and BOND1.3.

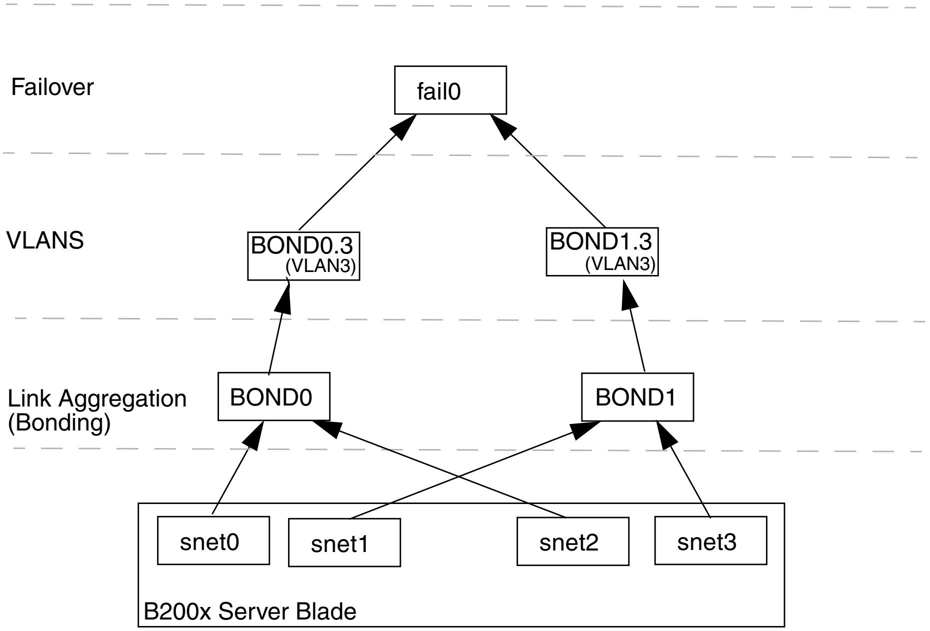


FIGURE 7-8 B200x Blade With Failover Between two Aggregated Links, Using VLANs

7.2 Configuring Bonding Interfaces

Bonding interfaces are used to provide link aggregation for B200x server blades. Link aggregation allows you to combine the four ethernet interfaces on the blade into two pairs of interfaces, each with a single MAC address. Therefore, `snet0` and `snet2` become a single interface to SSC0, and `snet1` and `snet3` become a single interface to SSC1. When the Sun Fire B1600 blade system chassis is fully operational, both switches are constantly active.

Link aggregation is achieved by using the Bonding driver to set up two bonding interfaces to enslave each pair of ethernet interfaces. In Red Hat el-3.0 the full 802.3ad specification is supported. In other versions of Linux, a simple active-backup protocol is used. Note that the Bonding driver can only be configured on top of physical interfaces.

To use link aggregation you must also configure the switches to accept the aggregated links. You do this by either enabling LACP (link aggregation control protocol, which is available with Red Hat el-3.0 only) or by setting up port-channels for the blade that uses aggregated links to the switch. For more information, see "Configuring the Switch for Link Aggregation" on page 7-14.

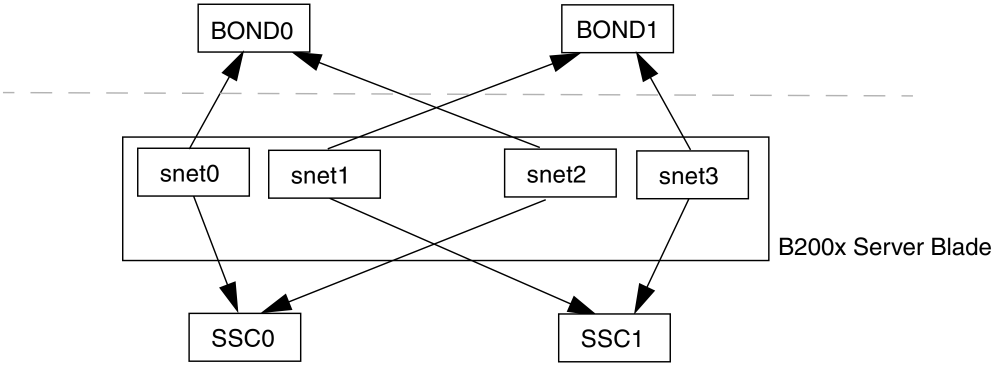


FIGURE 7-9 A B200x Server Blade With Two Bond Interfaces

7.2.1 Configuring the B200x Blade for Link Aggregation

The Bonding driver is used to provide Link aggregation, and is initially configured using module parameters when the driver is loaded. You must then associate physical interfaces to the bond interfaces manually using the ifenslave utility.

The module parameters configure the number of bond interfaces and their behavior. These module parameters are set in the `/etc/modules.conf` file. The parameters are:

```
alias bond0 bonding
alias bond1 bonding
options bonding max_bonds=2 mode=4 miimon=1000
```

- The alias commands associate the interface to the driver.
- `max_bonds` is the maximum number of bonding interfaces that are created.
- `mode` is the behavior of the bonding interface. For Red Hat el-3.0 this value should be 4. For other versions of Linux this value should be 3 for active-backup.
- `miimon` is the period, in milliseconds, to check for link status by MII (Media Independent Information).

You need to associate physical interfaces to the bond interfaces using the ifenslave utility. The ifenslave utility enslaves physical interfaces as slaves to the bond master. For example,

```
ifenslave bond0 snet0 snet2
```

enslaves `snet0` and `snet2` to `bond0`.

Note – In this configuration, the interfaces to enslave must be attached to the same switch, as this will create a virtual point-to-point link from the blade to the switch. Therefore, `snet0` and `snet2` are enslaved together and `snet1` and `snet3` are enslaved together.

7.2.1.1 Example `ifcfg` file on a B200x Blade

The location of `ifcfg` files depends on the version of Linux you are running:

- With Red Hat, `ifcfg` files are located in `/etc/sysconfig/network-scripts/`
- With SuSE, `ifcfg` files are located in `/etc/sysconfig/network/`

CODE EXAMPLE 7-3 shows a bond interface (`ifcfg-bond0`) that enslaves `snet0` and `snet2` to provide link aggregation.

CODE EXAMPLE 7-1 `/ifcfg-bond0`

```
DEVICE=bond0
CHILDREN="snet0 snet2"
ONBOOT=yes
BOOTPROTO=none
[ $ONBOOT = no ] || . ifinit
```

TABLE 7-1 `ifcfg-bond0`

Bonding Interface Driver Configuration	Explanation
`DEVICE=bond0`	Provides the name of the Bond interface driver.
`CHILDREN="snet0 snet2"`	Provides the Ethernet interfaces to be enslaved.
`ONBOOT=yes`	ONBOOT must be set to "yes". This means that the interface is configured at boot time.

7.2.2 Configuring the Switch for Link Aggregation

The instructions in this section tell you how to configure the two switches to accept aggregated links from a B200x blade. The method you use to configure the switches will depend on the version of Linux you are running. For Red Hat el-3.0, which supports 802.3AD, follow the instructions in "Configuring the Switch for Link Aggregation with Red Hat el-3.0 (Using LACP)" on page 7-14. For earlier versions of Red Hat, and SuSE, follow the instructions in "Configuring the Switch for Link Aggregation Using Active-Backup" on page 7-15.

7.2.2.1 Configuring the Switch for Link Aggregation with Red Hat el-3.0 (Using LACP)

The following steps tell you how to configure the switch for link aggregation if you are using Red Hat el-3.0. They use an example B200x server blade in slots 14 and 15.

1. **To log into the switch in SSC0, type:**

```
SC> console ssc0/swt
```

2. **When prompted, type the username and password for the switch.**

3. **Enable LACP on slot 14.**

```
# configure
# interface ethernet snp14
# lacp
# exit
```

4. **Enable LACP on slot 15.**

```
# interface ethernet snp15
# lacp
# exit
# exit
```

5. **Repeat Step 1 through Step 4 for the switch in SSC1.**

7.2.2.2 Configuring the Switch for Link Aggregation Using Active-Backup

The following steps tell you how to configure the switch for link aggregation if you are using active-backup. Active-backup is used with SuSE, and releases of Red Hat earlier than Red Hat el-3.0. The instructions use an example B200x server blade in slots 14 and 15.

1. **To log into the switch in SSC0, type:**

```
SC> console ssc0/swt
```

2. **When prompted, type the username and password for the switch.**
3. **Set up a port-channel for the default configuration.**

```
# configure
# interface port-channel 1
# switchport allowed vlan add 1 untagged
# exit
```

4. **Bind the Ethernet interface for slot 14 into the port-channel.**

```
# interface ethernet snp14
# channel-group 1
# exit
```

5. **Bind the Ethernet interface for slot 15 into the port-channel.**

```
# interface ethernet snp15
# channel-group 1
# exit
# exit
```

6. **Repeat Step 1 through Step 5 for the switch in SSC1.**

7.3 Configuring VLAN Interfaces

VLANs are a virtual interface that can be configured on physical interfaces or on bonding interfaces. For example, you can configure a VLAN interface on Ethernet snet0 (a physical interface), or on BOND0 (a virtual interface). VLAN support is provided by the sun8021q driver.

For VLANs to work correctly, both the blade and the switch ports for that blade need to be configured. The VLAN interfaces are configured using the sunvconfig utility.

7.3.1 Configuring Tagged VLANs

This section tells you how to configure a server blade so that the Ethernet interface provides an active logical interface to a VLAN. In the example shown, `snet0` provides an interface to VLAN 3.

To create VLAN 3 on top of `snet0` use the sunvconfig utility.

```
#sunvconfig add SNET0 3
```

This creates a VLAN3 interface which is configured on `snet0`. All network packets sent through this interface will have a VLAN tag of 3 added.

You can ensure that VLAN settings are maintained after a reboot by editing the `ifcfg-snet0.3` file.

The location of `ifcfg` files depends on the version of Linux you are running:

- With Red Hat, `ifcfg` files are located in `/etc/sysconfig/network-scripts/`
- With SuSE, `ifcfg` files are located in `/etc/sysconfig/network/`

CODE EXAMPLE 7-2 shows an example `ifcfg-snet0.3` file.

CODE EXAMPLE 7-2 `ifcfg-snet0.3`

```
DEVICE=snet0.3
PHYSDEVICE=snet0
ONBOOT=no
DRIVER=sunvlan
```

TABLE 7-2 `ifcfg-sunvlan2`

Master Interface Driver Configuration Variable	Explanation
`DEVICE=snet0.3`	Provides the name of the VLAN interface
`PHYSDEVICE=snet0`	Provides the name of the physical device or master interface on which the VLAN is configured.
`ONBOOT=no`	When set to “no”, the interface is not configured at boot time. Note: If you are running SuSE, replace “`ONBOOT=no`” with “`STARTMODE=manual`”
`DRIVER=sunvlan`	Specifies the initialization script to use to initialize the script.

7.3.2 Adding the Server Blades to a VLAN on the Switches in SSC0 and SSC1

The switch must also be configured to accept tagged VLAN traffic from the blades. The instructions in this section tell you how to add the server blades to the VLAN 3. If you are configuring for switch failover, you must add server blades to the switches in both SSC0 and in SSC1.

Note – If you reset the switch while you are performing the instructions in this section, you must save the configuration first. If you do not, you will lose all of your changes.

1. **From the `sc>` prompt, log into the console to configure the switch in SSC0.**

   To log into the switch in SSC0, type:

```
sc> console ssc0/swt
```

2. **When prompted, type your user name and password.**
3. **At the `Console#` prompt on the switch's command line, type:**

```
Console#configure
```

4. **Enter the switch's VLAN database by typing:**

```
Console(config)#vlan database
```

5. **Set up the VLAN by typing:**

```
Console(config-vlan)#vlan 3 name Data media ethernet
```

6. **Exit the vlan database by typing:**

```
Console(config-vlan)#end
```

7. **Add the server blade port `SNP0` to the data VLAN (VLAN 3).**

   To do this, type the following commands:

```
Console#configure
Console(config)#interface ethernet SNP0
Console(config-if)#switchport allowed vlan add 3 tagged
Console(config-if)#exit
Console(config)#
```

The meaning of this sequence is as follows:

- The `interface ethernet SNP0` command specifies the blade port you are configuring (in the example, the interface is blade port `SNP0`).
- The `switchport allowed vlan add 3 tagged` command makes the port a member of VLAN 3 (the new data network) and allows it to pass tagged traffic to the data network.

Repeat Step 7 for all the remaining server blade ports (`SNP1` through `SNP15`). All of these ports need to be included in both the management network and the data network.

To inspect the port you have configured, type:

```
Console#show interfaces switchport ethernet SNP0
Information of SNP0
 Broadcast threshold: Enabled, 256 packets/second
 Lacp status: Disabled
 VLAN membership mode: Hybrid
 Ingress rule: Disabled
 Acceptable frame type: All frames
 Native VLAN: 1
 Priority for untagged traffic: 0
 Gvrp status: Disabled
 Allowed Vlan: 3(t), 1(u)
 Forbidden Vlan:
Console#
```

8. **If required, copy the configuration of the switch in SSC0 on to the switch in SSC1.**

   Follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

7.4 Configuring Failover Interfaces

Network resiliency is provided using the Failover interface driver. The failover interface can be used with physical interfaces, and virtual interfaces, such as bond interfaces (used with link aggregation) or VLAN interfaces.

The Failover interface driver enslaves two interfaces. These two interfaces should each provide a path to a different switch in the chassis. For example, for failover between physical interfaces on a B100x blade, `snet0` and `snet1` can be enslaved. On the B200x blade, `snet0` and `snet1` can be enslaved, and so can `snet2` and `snet3`.

When providing failover between virtual interfaces such as VLANs or aggregated links, these interfaces must also provide a path to different switches. Therefore, the physical interfaces underlying the virtual interfaces must be configured so that each enslaved interface has a path to a different switch in the chassis.

7.4.1 Setting up Linux Server Blades Using the Failover Interface Driver for Network Resiliency

The instructions in this section tell you how to use the Failover interface driver to take advantage of the redundant connections from each Linux server blade to the two switches in the chassis.

The Failover interface driver works by enslaving the network interfaces on a server blade. It detects link availability by periodically arping the arp targets from the Ethernet interfaces. This means that if for any reason all of the arps fail on a given interface (indicating that the path to the network is no longer available on the interface that was used to perform the arp) the failover interface ensures that network traffic uses only the interface that remains valid.

The targets used for arping should be the default gateways for the Ethernet interfaces. You can configure arp targets using the failarp utility. The failarp utility looks in the routing table for gateways which it sets as the targets for the failover interface. Alternatively you can specify arp targets manually when you set up the failover interface.

You can configure failover interfaces manually using the failctl utility. Alternatively, you can edit the `ifcfg` files provided in`/etc/sysconfig/network-scripts/`.

7.4.1.1 Failover Support for Server Blades

To enable failover between the two switches you must configure a failover interface (`fail0` in FIGURE 7-10). The failover interface works by enslaving `snet0` and `snet1` and detecting link availability by periodically arping the arp targets through the Ethernet interfaces. If arps fail on `snet0` the failover interface ensures that network traffic uses `snet1`, and vice versa.

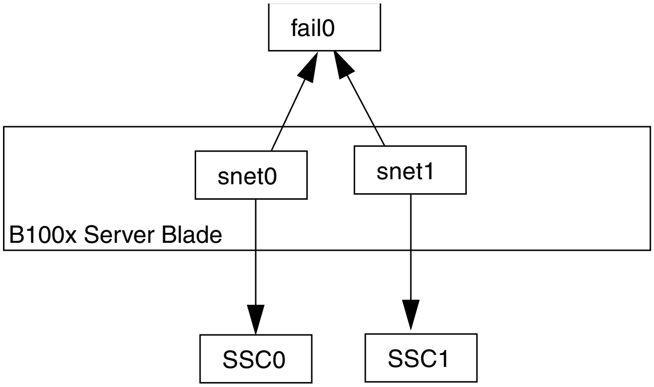


FIGURE 7-10 B100x Server Blade with `fail0` Configured for Failover

7.4.1.2 Configuring Failover for a Server Blade

You can configure failover interfaces manually using the failctl utility. The steps in this section tell you how to configure `fail0` to provide failover between the two switches (as shown in FIGURE 7-10). For the purposes of illustration the instructions use a sample configuration input from the network scenario illustrated in the section "Preparing the Network Environment Using DHCP" on page 7-3.

Note – You need to perform the instructions in this section on each B100x server blade that requires a redundant connection to the network.

TABLE 7-3 summarizes the information you would need to give to the Failover Interface Driver on the server blade as illustrated in FIGURE 7-1.

TABLE 7-3 Sample Failover Interface Driver Configuration for a B100x Server Blade

Failover Interface Driver Configuration Variable	Value
Failover interface	fail0
Physical interfaces	snet0 snet1
Failover interface IP address	192.168.1.150
Arp target IP address	192.168.1.1
Netmask	255.255.255.0

1. **Log into the console of the server blade whose interfaces you want to configure. Type the following at the `sc>` prompt:**

```
sc> console sn
```

where *n* is the number of the slot containing the server blade you want to log into.

2. **Enslave the two Ethernet devices on the blade using the `failctl` command.**

```
$ failctl fail0 snet0 snet1
```

3. **Configure static arp targets for `fail0`.**

```
$ failctl -t fail0 arp_target=192.168.1.1
```

Note – If you do not configure static arp targets, you can use the failarp utility to supply arp targets. The command `failarp -i fail0` will check the routing table for gateways to use for arp targets on fail0.

4. **Configure the arp interval used to check link availability. The arp interval is measured in milliseconds (ms).**

```
$ failctl -t fail0 arp_interval=nnnnn
```

where *nnnnn* is the number of milliseconds required for the arp intervals.

5. **Set up a static IP address for fail0.**

```
$ ifconfig fail0 192.168.1.150
```

Note – Alternatively you can configure the failover interface to obtain IP addresses using DHCP.

Note – You can maintain the failover interface configurations after rebooting by editing the ifcfg-fail files in /etc/sysconfig/network-scripts (or /etc/sysconfig/network-scripts, if you are running SuSE.) For more information, see "Example ifcfg-fail0 File for a B100x Server Blade" on page 7-23.

7.4.1.3 Example ifcfg-fail0 File for a B100x Server Blade

CODE EXAMPLE 7-3 shows an ifcfg-fail0 file that provides failover between the two switches.

CODE EXAMPLE 7-3 ifcfg-fail0

```
DEVICE=fail0
CHILDREN="snet0 snet1"
ONBOOT=yes
BOOTPROTO=none
IPADRR=192.168.1.150
NETMASK=255.255.255.0
ARP_INTERVAL=10000
#ARP_TARGET=192.168.1.1 #failarp(8) is used if ARP_TARGET isn't
specified.
```

TABLE 7-4 `ifcfg-fail0`

Failover Interface Driver Configuration Variable	Explanation
`DEVICE=fail0`	Provides the name of the failover interface.
`CHILDREN="snet0 snet1"`	Provides the Ethernet interfaces to be enslaved.
`ONBOOT=yes`	ONBOOT must be set to “yes”. This means that the interface is configured at boot time. Note: If you are running SuSE, replace “`ONBOOT=yes`” with “`STARTMODE=onboot`”
`BOOTPROTO=none`	Set BOOTPROTO to “none” if you have specified a static IP address for `fail0`. NOTE: If you set BOOTPROTO to DHCP, `fail0` will receive its IP address using DHCP.
`IPADDR=192.168.1.150`	Provides a static IP address for `fail0`.
`NETMASK=255.255.255.0`	Provides netmask for the IP address.
`ARP_INTERVAL=10000`	Checks link availability every 10 seconds.
`#ARP_TARGET=192.168.1.1`	If the arp target is commented out, the `fail0` uses `failarp` to supply arp targets.

7.5 Example Network Configuration

The example in this section (FIGURE 7-11) shows a network configuration where server blades are added to the management VLAN, which is VLAN 2 by default. VLAN 1 is also set up by default on the switch. This VLAN contains all the switch’s server blade and uplink ports. However, to demonstrate the use of the switch’s VLAN configuration facilities, the example uses VLAN 3 instead of VLAN 1 for the data network.

In this example the management VLAN (VLAN 2) and the data VLAN (VLAN 3) are tagged. However, the example also shows an additional VLAN for blade booting (VLAN 4). This handles untagged traffic generated by the blades during the PXE boot installation process.

This traffic on the boot VLAN (VLAN 4) can be tagged or untagged when it leaves the system chassis. In the sample commands in this section it is tagged. (The instructions in this section assume that the devices outside the chassis are VLAN-aware, and VLAN 4 is assumed to contain the PXE boot installation server used by the server blades.)

The example in this section uses full redundancy to the switches in SSC0 and SSC1, and link aggregation.

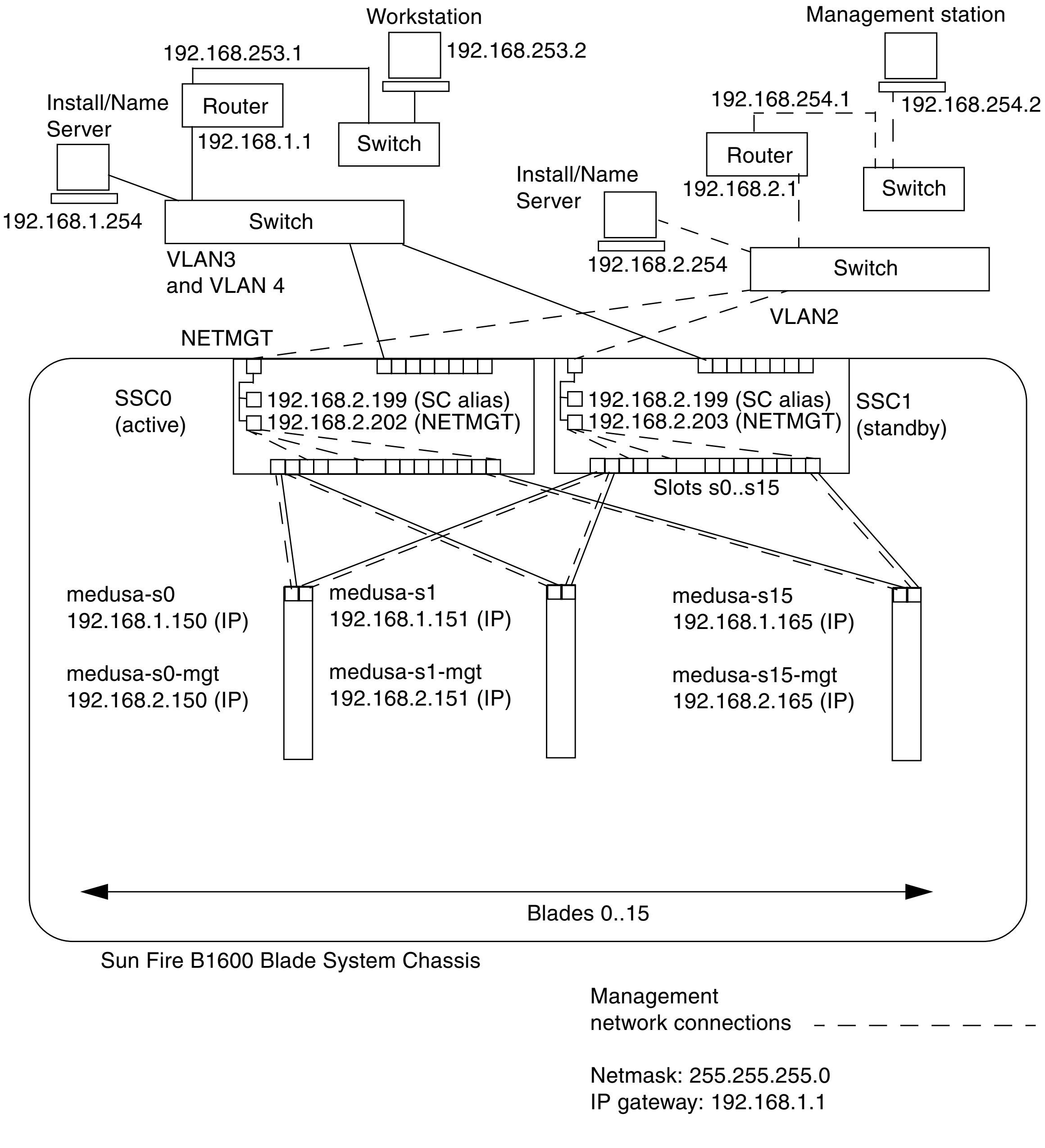


FIGURE 7-11 Sample Network Configuration With a Management VLAN that Includes Server Blades

CODE EXAMPLE 7-4 Sample /etc/hosts file on the Name Server (on the Management Network)

```
# Internet host table
# This is the sample /etc/hosts file for the name-server on the management
# network.

192.168.2.1      mgtnet-router-1    # Management network router
#                                    (default gateway)
192.168.2.254   mgtnet-nameserver  # Management network install/name server
192.168.254.1   mgtnet-router-254  # Management network router (client side)
192.168.254.2   mgtnet-ws          # Management network workstation

192.168.2.199   medusa-sc          # Medusa - alias IP address for active SC
192.168.2.200   medusa-ssc0        # Medusa - ssc0/sc
192.168.2.201   medusa-ssc1        # Medusa - ssc1/sc
192.168.2.202   medusa-swt0        # Medusa - ssc0/swt
192.168.2.203   medusa-swt1        # Medusa - ssc1/swt

# 192.168.2.100 -> 192.168.2.131 are reserved for private use by the
# Sun Fire B1600 Blade System Chassis called medusa. They are test addresses for
# the Master interface driver on each server blade.

192.168.2.150   medusa-s0-mgt
:
192.168.2.165   medusa-s15-mgt
192.168.1.150   medusa-s0
:
192.168.1.165   medusa-s15
```

7.5.1 Configuring the Network Interfaces on a B200x Sever Blade

To support the configuration in FIGURE 7-11 on a B200x blade, you must configure three network interface layers, as illustrated in FIGURE 7-12.

- Layer 1 - Bonding interfaces

  Two bonding interfaces must be configured to provide aggregated links that combine the four ethernet interfaces on a B200x blade into two pairs of interfaces BOND0 provides link aggregation for the physical interfaces snet0 and snet2, and BOND1 provides link aggregation for the physical interfaces snet1 and snet3.

- Layer 2 - VLAN interfaces

  Two VLAN3 interfaces (BOND0.3 and BOND1.3) are configured on top of the two aggregated links (BOND0 and BOND1), and two VLAN2 interfaces (BOND0.2 and BOND1.2) are configured on top of the same two aggregated links.

- Layer 3 - Failover interfaces

  To provide redundancy between the two switches, two failover interfaces must be configured on top of the VLAN interface layer. The fail1 interface provides failover for the two VLAN3 interfaces (BOND0.3 and BOND1.3). The fail2 interface provides failover for two VLAN2 interfaces (BOND0.2 and BOND1.2).

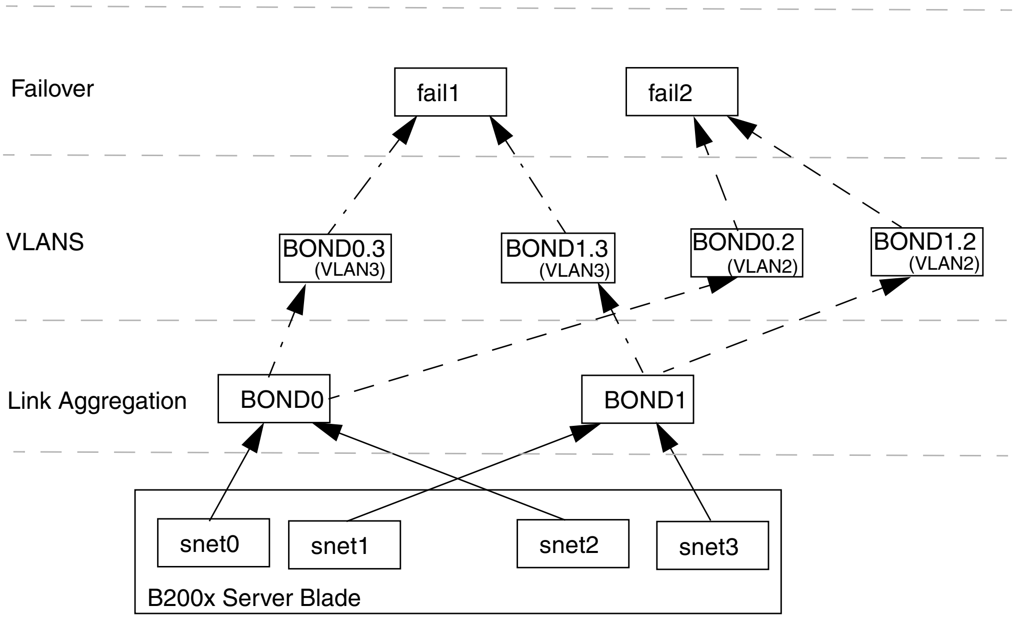


FIGURE 7-12 B200x Blade With Failover Between two Bonding Interfaces

You configure these network interfaces by editing `ifcfg` files for snet0, snet1, snet2, snet3, BOND0, BOND1, BOND0.2, BOND1.2, BOND0.3, BOND1.3, fail1 and fail2.

Note – Only the top most interfaces in your configuration should have IP addresses configured (using either static IP or DHCP). Also, in the configuration files only the top most interface should have `ONBOOT` set to “yes” (when using Red Hat) or `startmode` set to “`ONBOOT`” (when using SuSE).

Refer to the following code examples for information on editing the `ifcfg` files. The location of `ifcfg` files depends on the version of Linux you are running:

- With Red Hat, `ifcfg` files are located in `/etc/sysconfig/network-scripts/`
- With SuSE, `ifcfg` files are located in `/etc/sysconfig/network/`

`ifcfg-snet0`

```
DEVICE=snet0
ONBOOT=no
```

`ifcfg-snet1`

```
DEVICE=snet1
ONBOOT=no
```

`ifcfg-snet2`

```
DEVICE=snet2
ONBOOT=no
```

`ifcfg-snet3`

```
DEVICE=snet3
ONBOOT=no
```

`ifcfg-bond0`

```
DEVICE=bond0
CHILDREN="snet0 snet2"
ONBOOT=no
[ $ONBOOT = no ] || . ifinit
```

`ifcfg-bond1`

```
DEVICE=bond1
CHILDREN="snet1 snet3"
ONBOOT=no
[ $ONBOOT = no ] || . ifinit
```

ifcfg-bond0.2

```
DEVICE=bond0.2
PHYSDEVICE=bond0
DRIVER=sunvlan
ONBOOT=no
[ $ONBOOT = no ] || . ifinit
```

ifcfg-bond1.2

```
DEVICE=bond1.2
PHYSDEVICE=bond1
DRIVER=sunvlan
ONBOOT=no
[ $ONBOOT = no ] || . ifinit
```

ifcfg-bond0.3

```
DEVICE=bond0.3
PHYSDEVICE=bond0
DRIVER=sunvlan
ONBOOT=no
[ $ONBOOT = no ] || . ifinit
```

ifcfg-bond1.3

```
DEVICE=bond1.3
PHYSDEVICE=bond1
DRIVER=sunvlan
ONBOOT=no
[ $ONBOOT = no ] || . ifinit
```

ifcfg-fail1

```
DEVICE=fail1
CHILDREN="bond0.3 bond1.3"
ONBOOT=yes
IPADDR=192.168.1.164
[ $ONBOOT = no ] || . ifinit
```

`ifcfg-fail2`

```
DEVICE=fail2
CHILDREN="bond0.2 bond1.2"
ONBOOT=yes
IPADDR=192.168.2.164
[ $ONBOOT = no ] || . ifinit
```

7.5.2 Adding the Server Blades to the Management and Data VLANs on the Switches in SSC0 and SSC1

To support the configuration in FIGURE 7-11, you will need to add the server blades to the management and data VLANs on the switches in SSC0 and SSC1.

Note – If you reset the switch while you are performing the instructions in this section, you must save the configuration first. If you do not, you will lose all of your changes. To save the configuration, follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

---

1. **From the** `sc>` **prompt, log into the console to configure the switch in SSC0.**

   To log into the switch in SSC0, type:

```
sc> console ssc0/swt
```

2. **When prompted, type your user name and password.**
3. **At the** `Console#` **prompt on the switch's command line, type:**

```
Console#configure
```

4. **Enter the switch's VLAN database by typing:**

```
Console(config)#vlan database
```

5. **Set up the VLAN for the data network and for the boot network by typing:**

```
Console(config-vlan)#vlan 3 name Data media ethernet
Console(config-vlan)#vlan 4 name Boot media ethernet
```

6. **Exit the vlan database by typing:**

```
Console(config-vlan)#end
```

7. **Add the server blade port `SNP0` to the management VLAN (VLAN 2), the data VLAN (VLAN 3), and to the VLAN that you are using for booting (VLAN 4).**

   To do this, type the following commands:

```
Console#configure
Console(config)#interface ethernet SNP0
Console(config-if)#switchport allowed vlan add 2 tagged
Console(config-if)#switchport allowed vlan add 3 tagged
Console(config-if)#switchport allowed vlan add 4
Console(config-if)#switchport native vlan 4
Console(config-if)#switchport allowed vlan remove 1
Console(config-if)#exit
Console(config)#
```

   The meaning of this sequence is as follows:

   - The `interface ethernet SNP0` command specifies the blade port you are configuring (in the example, the interface is blade port `SNP0`).
   - The `switchport allowed vlan add 2 tagged` command makes this blade port a member of VLAN 2 (the management network), and allows it to pass tagged traffic to the management network.
   - The `switchport allowed vlan add 3 tagged` command makes the port a member of VLAN 3 (the new data network) and allows it to pass tagged traffic to the data network.
   - The `switchport allowed vlan add 4` command makes the port a member of VLAN 4. It causes the port to accept untagged packets and to tag them as members of VLAN 4. By doing this, you are providing a path for untagged traffic generated by the blade (during booting) to reach the Network Install Server. In the next command, you will make this the native VLAN, in other words, the VLAN onto which all untagged frames are forwarded.
   - The `switchport native vlan 4` command makes the port put any untagged frames it receives onto VLAN 4. (OBP, Jumpstart and PXE involve server blades in sending untagged frames.)

- The `switchport allowed vlan remove 1` command removes the port from VLAN 1 (the default VLAN on the switch for all the server blade ports and uplink ports).

Repeat Step 7 for all the remaining server blade ports (`SNP1` through `SNP15`). All of these ports need to be included in both the management network and the data network.

To inspect the port you have configured, type:

```
Console#show interfaces switchport ethernet SNP0
Information of SNP0
 Broadcast threshold: Enabled, 256 packets/second
 Lacp status: Disabled
 VLAN membership mode: Hybrid
 Ingress rule: Disabled
 Acceptable frame type: All frames
 Native VLAN: 4
 Priority for untagged traffic: 0
 Gvrp status: Disabled
 Allowed Vlan:    2(t), 3(t), 4(u)
 Forbidden Vlan:
Console#
```

8. **If you intend to combine any of the data uplink ports into aggregated links, do this now.**

   Follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, appendix A.

9. **Add any data uplink ports (that are not aggregated links) to the data VLAN (that is, VLAN 3) and to the boot VLAN (VLAN 4) by typing the following commands:**

```
Console#configure
Console(config)#interface ethernet NETP0
Console(config-if)#switchport allowed vlan add 3 tagged
Console(config-if)#switchport allowed vlan add 4
Console(config-if)#switchport native vlan 4
Console(config-if)#switchport allowed vlan remove 1
Console(config-if)#switchport ingress-filtering
Console(config-if)#switchport mode trunk
Console(config-if)#switchport acceptable-frame-types tagged
Console(config-if)#no switchport gvrp
Console(config-if)#switchport forbidden vlan add 2
Console(config-if)#end
Console(config)#
```

- The `interface ethernet NETP0` command specifies the uplink port you are configuring.
- The `switchport allowed vlan add 3 tagged` command adds this uplink port to the data network (VLAN 3).
- The `switchport allowed vlan add 4` command adds this uplink port to an untagged VLAN you are using for blade booting (VLAN 4). In the next command, you will make this the native VLAN (in other words, the VLAN onto which any untagged frames are forwarded by this data port).
- The `switchport native vlan 4` command makes the external data port put any untagged frames it receives onto VLAN 4. (The effect of this command is temporary; the subsequent commands will prevent the port from accepting untagged frames. The reason you need to type it is that the switch requires a native VLAN to be available until the `switchport mode trunk` command has been executed.)
- The `switchport allowed vlan remove 1` command removes this uplink port from VLAN 1 (the default VLAN). This VLAN can only be removed at this point, (that is, after VLAN 4 - the native, untagged VLAN - has been created).
- The `switchport ingress-filtering` command, the `switchport mode trunk` command, and the `switchport acceptable-frame-types tagged` command cause the port to reject any frames that are not tagged for the particular VLAN or VLANs that it is a member of.
- The `no switchport gvrp` command prevents the port from using GVRP to advertise the VLANs it is a member of (in this case, VLAN 3) to another switch that it is connected to.
- The `switchport forbidden vlan add 2` command prevents the uplink port from being added to vlan 2 in response to a GVRP request from another switch on the network.

To inspect a port that you have configured, type:

```
Console#show interfaces switchport ethernet NETP0
Information of NETP0
 Broadcast threshold: Enabled, 256 packets/second
 Lacp status: Disabled
 VLAN membership mode: Trunk
 Ingress rule: Enabled
 Acceptable frame type: Tagged frames only
 Native VLAN: 4
 Priority for untagged traffic: 0
 Gvrp status: Disabled
 Allowed Vlan:    3(t), 4(t)
 Forbidden Vlan:    2,
Console#
```

10. **Add any external aggregated links to the data VLAN (VLAN 3) by typing the commands below.**

    For more information about using aggregated link connections, see the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

    In the example below, the aggregated link is called port-channel 1. The interface port-channel 1 command specifies the aggregated link you are about to configure.

```
Console(config)#interface port-channel 1
Console(config-if)#switchport allowed vlan add 3 tagged
Console(config-if)#switchport allowed vlan add 4
Console(config-if)#switchport native vlan 4
Console(config-if)#switchport allowed vlan remove 1
Console(config-if)#switchport ingress-filtering
Console(config-if)#switchport mode trunk
Console(config-if)#switchport acceptable-frame-types tagged
Console(config-if)#no switchport gvrp
Console(config-if)#switchport forbidden vlan add 2
Console(config-if)#end
Console(config)#
```

11. **Add any internal aggregated links to the data VLAN (VLAN 3) by typing the commands below.**

    For internal aggregated links the uplink port is added to the data network (VLAN 3).

    For more information about using aggregated link connections, the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

    In the example below, the aggregated link is called port-channel 1. The interface port-channel 1 command specifies the aggregated link you are about to configure..

```
Console(config)#interface port-channel 1
Console(config-if)#switchport allowed vlan add 2 tagged
Console(config-if)#switchport allowed vlan add 3 tagged
Console(config-if)#switchport allowed vlan add 4
Console(config-if)#switchport native vlan 4
Console(config-if)#switchport allowed vlan remove 1
Console(config-if)#switchport ingress-filtering
Console(config-if)#switchport mode trunk
Console(config-if)#switchport acceptable-frame-types tagged
Console(config-if)#no switchport gvrp
Console(config-if)#end
Console(config)#
```

12. **Configure the aggregated links for the server blade.**

    In the example below, `SNP0` is added to port-channel 1.

    ```
    Console(config)#interface ethernet SNP0
    Console(config-if)#channel-group 1
    Console(config-if)#end
    ```

13. **Add all uplink ports to VLAN 3 either individually or as aggregated links (see Step 9 and Step 10).**

    For example, if ports NETP1, NETP2, and NETP3 are combined into aggregated link 1, and NETP4, and NETP5 are combined into aggregated link 2, you will need to add ports NETP0, NETP6, and NETP7 plus aggregated link 1 and aggregated link 2 to VLAN 3.

14. **Follow the instructions in** the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, **Appendix A**.

15. **Save the changes you have made to the configuration of the switch in SSC0.**

    To do this, follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

16. **Copy the configuration of the switch in SSC0 on to the switch in SSC1.**

    Follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

17. **Type `#.` to exit the switch's command-line interface and return to the System Controller.**

18. **From the `sc>` prompt, log into the switch in SSC1 by typing:**

    ```
    sc> console ssc1/swt
    ```

19. **Type your user name and password.**

20. **Set the IP address, netmask, and default gateway for the switch in SSC1.**

    To do this, follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

21. **Save the changes you have made to the configuration of the switch in SSC1.**

    To do this, follow the instructions in the *Sun Fire B1600 Blade System Chassis Software Setup Guide*, Appendix A.

22. **Type `#.` to exit the switch command-line interface and return to the `sc>` prompt.**

CHAPTER 8

Using Linux Server Blade Utilities

This chapter provides information on using the following utilities with Linux server blades:

- The `memdiag` utility.

  Use this utility to detect memory problems on a server blade. See "Performing Memory Diagnostics on a Server Blade" on page 8-2.

- The `biosupdate` utility.

  Use this utility to upgrade the BIOS. See "Upgrading the BIOS" on page 8-4.

8.1 Performing Memory Diagnostics on a Server Blade

This section tells you how to use the `memdiag` utility to detect memory problems on a server blade.

The `memdiag` utility uses ECC functionality to report any errors on DIMMs installed in the server blade. If a fault is reported by `memdiag`, you may need to replace the faulty DIMM. It is recommended that you run `memdiag` on any server blade that is experiencing problems.

Note – The `memdiag` utility is installed on the server blade as part of the PXE boot installation process. See Chapter 4 for information on performing a PXE boot installation.

8.1.1 Running a Memory Test on a Server Blade

1. **Log into the blade for which you want to perform a memory test.**

   At the `SC` prompt, type:

```
sc> console sn
```

   where *n* is the number of the slot containing the blade.

2. **Run** `memdiag` **from the** `/usr/local/bin` **directory:**

```
/usr/local/bin/memdiag
Starting Tests
        Starting Memory Test
                Testing 512M
        PASS    Memory Test
        Starting ECC Test
                Testing 512M
        PASS    ECC Test
Ending Tests
```

   In this example no ECC errors were reported on the server blade.

3. **Check the output for memory and ECC failures.**

8.1.2 Example `memdiag` Output for Faulty DIMMs

CODE EXAMPLE 8-1 Output for a Dual-processor server blade

```
/usr/local/bin/memdiag
Starting Tests
        Starting Memory Test
                Testing 1536M
        PASS Memory Test
        Starting ECC Test
                Testing 1536M
Warning: Errors were found in Bank 0 this may be an indication that
this item is defective
Please Check DIMM Pair 1
        FAIL ECC Test
Ending Tests
```

CODE EXAMPLE 8-1 shows output for a dual-processor server blade. In this example DIMM pair 1 is faulty and should be replaced.

Note – For information on replacing DIMMs in a B200x Server Blade, see the *Sun Fire B200x Server Blade DIMM Replacement Guide.*

CODE EXAMPLE 1 Output for a Single-processor server blade

```
/usr/local/bin/memdiag
Starting Tests
        Starting Memory Test
                Testing 768M
        PASS Memory Test
        Starting ECC Test
                Testing 768M
Warning: Errors were found in Bank 0 this may be an indication that
this item is defective
Please Check DIMM 0
        FAIL ECC Test
Ending Tests
```

CODE EXAMPLE 1 shows output for a single-processor server blade. In this example DIMM 0 is faulty and should be replaced.

Note – For information on replacing DIMMs in a B100x Server Blade, see the *Sun Fire B100x Server Blade DIMM Replacement Guide.*

8.2 Upgrading the BIOS

This section tells you how to use the `biosupdate` utility to upgrade the BIOS on a server blade. For information on where to find the latest BIOS images, contact your Sun support engineer.

Note – The `biosupdate` utility is installed on the server blade as part of the PXE boot installation process. See Chapter 4 for information on performing a PXE boot installation.

Caution – When upgrading the BIOS, do not interrupt the process by resetting or powering down the blade. Interrupting the upgrade will permanently damage the blade.

8.2.1 To Upgrade the BIOS

1. **Log into the blade for which you want to update the BIOS.**

   At the `SC` prompt, type:

```
sc> console sn
```

   where *n* is the number of the slot containing the blade.

2. **Check the version of the BIOS currently running on the blade, to establish whether the upgrade is necessary:**

```
modprobe mtdbios
cat /proc/BIOS
rmmod mtdbios
BIOS Vendor: AMI
BIOS Version: P1.1.32
BIOS Date: 01/19/2004
Manufacturer: Sun Microsystems
Product: Sun Fire B200x
```

3. **Copy the BIOS image to a known location on the blade.**

4. **Run the** `biosupdate` **command:**

```
biosupdate biosimage
```

where *biosimage* is the BIOS image.

The blade prompt returns when the update is complete.

Note – Do not restart the blade while the update is in progress.

Note – When the update is complete, you can check the BIOS version next time you restart the blade.

CHAPTER 9

Troubleshooting the Linux PXE Boot Installation

This appendix provides information on common problems that may occur during or after a PXE boot installation.

Errors During Startup

The following errors appear at startup when PXE booting the blade:

```
PXE-E51: No DHCP or proxyDHCP offers were received.
PXE-M0F: Exiting Broadcom ROM.
```

Cause

The DHCP service is not configured correctly.

Solution

To ensure that the DHCP service is running on the DHCP server and monitoring the correct port, use the following `netstat` command:

```
$ netstat -an | fgrep -w 67
udp        0      0 0.0.0.0:67              0.0.0.0:*
```

If no listening socket is shown, check your DHCP setup and configuration. If a listening socket is shown, this may indicate another problem such as firewall filtering or cabling issues.

Errors After Obtaining IP Address (Issue 1)

During a PXE boot installation, the following errors appear after obtaining the IP address:

```
PXE-E53: No boot filename received
PXE-M0F: Exiting Broadcom PXE ROM.
```

Cause

The DHCP service did not provide the name of a boot file.

Solution

Ensure that the `filename` command is correctly specified in the `/etc/dhcpd.conf` file on the PXE server.

This problem may also occur if the DHCP lease is received from a different machine. Normally, only one DHCP server should be configured on a single network segment.

Errors After Obtaining IP Address (Issue 2)

During a PXE boot installation, the following errors appear after obtaining the IP number:

```
PXE-E32: TFTP Open timeout
```

Cause

The TFTP service is not configured correctly.

Solution

To ensure that the TFTP service is running and monitoring the correct port, use the following `netstat` command:

```
$ netstat -an | fgrep -w 69
udp        0      0 0.0.0.0:69              0.0.0.0:*
```

If no listening socket is shown, check your TFTP setup and configuration. If a listening socket is shown, this may indicate another problem such as firewall filtering or cabling issues.

To test the TFTP service, try installing a TFTP client on a different machine and attempt to download the `pxelinux.bin` file:

```
# cd /tmp
# tftp PXE-server
tftp> get /as-2.1/sun/pxelinux.bin
Received 10960 bytes in 0.1 seconds
tftp> quit
```

Errors After Obtaining IP Address (Issue 3)

During a PXE boot installation, the following errors appear after obtaining the IP address:

```
PXE-T01: File not found
PXE-E3B: TFTP Error - File Not found
PXE-M0F: Exiting Broadcom PXE ROM.
```

Cause

The boot file name does not exist on the PXE server.

Solution

In the `/etc/xinetd.d/tftp` file on the PXE server:

- Check that the correct arguments are used.

  It is recommended that you use -s /tftp, and ensure that the TFTP service uses `chroot(1)` to change its top level directory to `/tftp`. This means that the `dhcp filename` argument is relative to the top level directory (and does not include the section `/tftp`).
- Check that the `filename` argument has been spelled correctly.
- Check that the `next-server` IP number has been specified correctly.

To test the TFTP service, try installing a TFTP client on a different machine and attempt to download a file:

```
# cd /tmp
# tftp PXE-server
tftp> get /as-2.1/sun/pxelinux.bin
Received 10960 bytes in 0.1 seconds
tftp> quit
```

Error After Installing the Linux Kernel (Issue 1)

During a PXE boot installation, the following error appears after loading the Linux kernel:

```
------+ Kickstart Error +-------+
|                                 |
| Error opening: kickstart file   |
| /tmp/ks.cfg: No such file or    |
| directory                       |
|                                 |
|            +-----+              |
|            | OK |              |
|            +-----+              |
|                                 |
|                                 |
+---------------------------------+
```

Cause

NFS is not working correctly on the PXE server.

Solution

Validate your NFS configuration by doing one or both of the following:

- On the PXE server, run the `showmount -e` command.
- On another machine (not the PXE server), run the `showmount -e` *PXE-server* command, where *PXE-server* is the name or IP address of the PXE server. Ensure that the output includes the tftp path:

```
# showmount -e
Export list for PXE-server:
/tftp           (everyone)
```

If this path is not in the output, check your NFS setup and configuration.

This problem may also occur if the blade is not correctly connected to the PXE server. If you have only one switch and system controller (SSC) installed on the chassis, ensure that the SSC is installed in position `0`. See the *Sun Fire B1600 Chassis Administration Guide* for information on installing the SSC.

If the NFS services are working normally and can be used from other machines on the network, it is likely that the PXE server has provided the wrong kernel to the blade. This occurs if the linux distribution installed on the PXE server does not exactly match the linux distribution against which the supplemental CD (supplied with the Linux blade) was built. An exact match is necessary to ensure that module versioning does not cause the 5704 network driver (suntg3) to fail to load.

Root Password Message After Installing the Linux Kernel

During a PXE boot installation, the following message appears after loading the Linux kernel:

```
+--------------+ Root Password +---------------+
|                                              |
|  Pick a root password. You must type it      |
|  twice to ensure you know what it is and     |
|  didn’t make a mistake in typing. Remember   |
|  that the root password is a critical part   |
|  of system security!                         |
|                                              |
| Password:           ________________________ |
| Password (confirm): ________________________ |
|                                              |
|        +-----+              +-------+        |
|        | OK |               | Back |         |
|        +-----+              +-------+        |
|                                              |
|                                              |
+----------------------------------------------+
```

Cause

No default root password has been specified in ks.cfg.

Solution

In the sun/install/ks.cfg file, ensure that the rootpw command is not commented out, and that you have specified a root password. See Chapter 4 for information on entering a root password.

Error After Rebooting

After completing a PXE boot installation and rebooting, the following screen appears:

```
GRUB  version 0.92  (634K lower / 522176K upper memory)

[ Minimal BASH-like line editing is supported.  For the first word,
TAB lists possible command completions.  Anywhere else TAB lists
the possible completions of a device/filename. ]

grub>
```

Cause

The PXE boot installation did not complete.

Solution

This problem may occur if the blade is removed or powered off during installation. You must re-install the blade.

Blade Does Not Boot From The Disk

After successfully completing a PXE boot installation, the blade continues to boot from the network instead of the disk.

Cause

The BIOS is configured to boot from the network by default.

Solution

At the `SC` prompt, use the `bootmode reset_nvram sn` command to reset the BIOS to boot from the disk by default.

First Boot From Disk Runs `fsck`

When booting the blade from the disk for the first time, the blade runs `fsck` to fix filesystems.

Cause

The blade has not unmounted fileystems.

Solution

To unmount all file systems and enable the blade to reboot correctly, ensure that you press `Enter` at the final `OK` prompt during the PXE boot installation. See Chapter 4 for more information.

Installer Hangs or Fails During PXE Boot Installation

When PXE installing a blade, the installer does one of the following:

- Hangs after the OS requests an IP address from the PXE server.
- Fails with an error message indicating that the signal 11 was received.

Cause

The PXE server may be using the eepro100 driver.

Solution

1. **Check if the PXE server is using the eepro100 driver by examining the** `/etc/modules.conf` **file for a line equivalent to:**

```
alias eth0 eepro100
```

Note – The `eth` instance may be different depending on your hardware setup.

---

2. **Change the line to:**

```
alias eth0 e100
```

This avoids a known interaction issue between the i82557/i82558 10/100 Ethernet hardware and the Broadcom 5704.

Prompted to Insert Module Disks During PXE Boot (SUSE only)

When booting a blade during a SuSE installation, the blade does not boot automatically and you are prompted to perform an interactive installation:

```
Please insert modules disk 3.
```

```
You’ll find instructions on how to create it in boot/README on
CD1 or DVD.
```

Cause

SuSE expects a default router to be supplied by the DHCP server, otherwise it assumes that the interface is not functional.

Solution

Ensure thst you have specified a default router in the `dhcpd.conf` file. For example:

```
ddns-update-style none;
default-lease-time 1800;
max-lease-time 3600;
:
option routers 172.16.11.6;
:
subnet 172.16.11.0 netmask 255.255.0.0 {
  next-server 172.16.11.8;                   # name of your TFTP server
  filename "/<linux_dir>/sun/pxelinux.bin";    # name of the boot-loader program
  range 172.16.11.100 172.16.11.200;         # dhcp clients IP range
}
```

PART

3 Installing and Using Solaris x86 on a Blade

CHAPTER 10

Installing Solaris x86

This chapter tells you how to install Solaris x86 onto a Sun Fire B100x or B200x server blade. It contains the following sections:

10.1 Overview of the Solaris x86 Installation Procedures

The B100x and B200x blades use a PXE-based network installation method to receive the Solaris x86 operating system. PXE booting is supported by DHCP services, and this means that there are a number of setup steps you need to perform involving the DHCP server. Also, the Network Install Server and the DHCP server need to be configured for each individual blade, otherwise the network installation will not work. The instructions in this chapter tell you what to do to get to a point where you can inititiate an interactive Solaris installation or a Jumpstart installation on the blade. The chapter refers you to the *Solaris 9 Installation Guide* for instructions about the interactive part of the Solaris installation.

Caution – Depending on the version of Solaris 9 x86 that you are installing, you might need to perform a procedure to patch the network install image on your Solaris Network Install Server so that it contains the required platform software support for the B100x and B200x blades. If patches are required, the Product Notes provide instructions for dowloading these and running the script that applies them to your Solaris x86 image on the Network Install Server. View the Product Notes at: `http://www.sun.com/products-n-solutions/hardware/docs/Servers/`

The tasks that you will perform in this chapter are as follows:

- General preparation (Section 10.2, "Preparing to Install Solaris x86" on page 10-3).
- Configuration of the DHCP option strings, and of the global PXE boot macro, if these are not already configured on the DHCP server (Section 10.3, "Configuring Global Settings for Solaris x86 Blades on the DHCP Server" on page 10-5).

For each blade you are installing, you will also perform the following steps in Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10:

- Find out and note down the blade's MAC address.
- Run the `add_install_client` script.
- Configure a client-specific DHCP macro on the DHCP server.
- Configure the IP address for the client on the DHCP server
- Configure the blade temporarily to boot from the network (the instructions for this are in Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18)

- Reset or power on the blade and monitor its booting processes (the instructions for these tasks are in Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18, and Section 10.7, "Monitoring the Network Booting Process and Starting the Solaris Installation" on page 10-20)

10.2 Preparing to Install Solaris x86

Note – If you are intending to create your Solaris x86 install image by using the Solaris 9 CD media (instead of the DVD media), you need to have a system running Solaris x86 available. This is because a SPARC Solaris system will not be able to read the Solaris x86 CD media. For instructions about how to create a Solaris x86 Network Install Server on a SPARC system using the x86 CD media, refer to Chapter 12 of the *Solaris 9 Installation Guide*.

---

1. **Connect a network port on the SSC to a subnet containing both the Network Install Server you intend to use and the DHCP server you intend to use to allocate IP addresses to the B100x or B200x server blade.**

   If you have a redundant SSC in the blade system chassis, duplicate this connection on the second SSC.

2. **Find out the MAC address of the first interface on the blade you intend to install Solaris x86 onto.**

   To do this:

   a. **Log into the active System Controller by following the instructions in Chapter 2 of the** *Sun Fire B1600 Blade System Chassis Software Setup Guide* **if you are logging into a brand new chassis in its factory default state.**

   Otherwise log in using the user name and password assigned to you by your system administrator.

b. At the `sc>` prompt, type:

```
sc>showplatform -v
:
:

Domain     Status         MAC Address         Hostname
--------   -----------    -----------------   -------------
S1         Standby        00:03:ba:29:e6:28   chatton-s1-0
S2         Standby        00:03:ba:29:f0:de
S6         OS Running     00:03:ba:19:27:e9   chatton-s6-0
S7         OS Stopped     00:03:ba:19:27:bd   chatton-s7-0
S10        Standby        00:03:ba:2d:d1:a8   chatton-s10-0
S12        OS Running     00:03:ba:2d:d4:a0   chatton-s12-0
:
SSC0/SWT   OS Running           00:03:ba:1b:6e:a5
SSC1/SWT   OS Running           00:03:ba:1b:65:4d
SSC0/SC    OS Running (Active)  00:03:ba:1b:6e:be
SSC1/SC    OS Running           00:03:ba:1b:65:66
:
sc>
```

where the : character (in the leftmost column) indicates omitted data. The MAC address listed for each blade is the MAC address of the first interface (by default, `bge0`).

For an installation that uses the first network interface on the blade, you only need to know the MAC address of the first network interface. Make a note of this MAC address.

If you intend to use the second, third, or fourth interface instead, you need to calculate the MAC address for that interface (see Section 10.12, "Installing Solaris x86 Onto a Blade by Using the Second, Third, or Fourth Network Interface" on page 10-47).

3. **Set up a Network Install Server for Solaris x86 by following the instructions in the *Solaris 9 Installation Guide*.**

   Make a note of the IP address of the Network Install Server that your blades will install their x86 operating system from.

4. **Make sure the DHCP server you intend to use is properly set up and functioning.**

   For information about setting up a Solaris DHCP server, refer to the *Solaris DHCP Administration Guid*e.

Note – Make sure you have updated your DHCP server with the latest DHCP patches available at: `http://sunsolve.sun.com`.

---

5. **If you want the DHCP server to allocate IP addresses dynamically to the server blade, then reserve a block of addresses on the DHCP server for this purpose.**

   For information about how to do this, refer to the *Solaris DHCP Administration Guid*e.

6. **Read the latest Product Notes for the chassis and blades to find out whether you need to download any patches for the version of Solaris x86 that you intend to install onto the blade.**

   Check the following location on the web:

   `http://www.sun.com/servers/entry/b100x/`

   The information you require is in the section of the Product Notes entitled "Installing the Solaris x86 Operating System Onto a Server Blade".

10.3 Configuring Global Settings for Solaris x86 Blades on the DHCP Server

This section tells you how to configure the option strings that are required on the DHCP server to support booting of the B100x and B200x blades. It also tells you how to configure the global PXE boot client. If the required options strings are already defined on the DHCP server and the PXE boot client is already correctly specified, proceed to Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10.

10.3.1 Adding the Required Option Strings to the DHCP Server

1. **Log into the Network Install Server as** `root`**, and start the DHCP Manager GUI by typing:**

```
# DISPLAY=mydisplay:0.0
# export DISPLAY
# /usr/sadm/admin/bin/dhcpmgr &
```

   where *mydisplay* is the name of the system (for example, a desktop workstation) that you are using to display the DHCP Manager's GUI (Graphical User Interface).

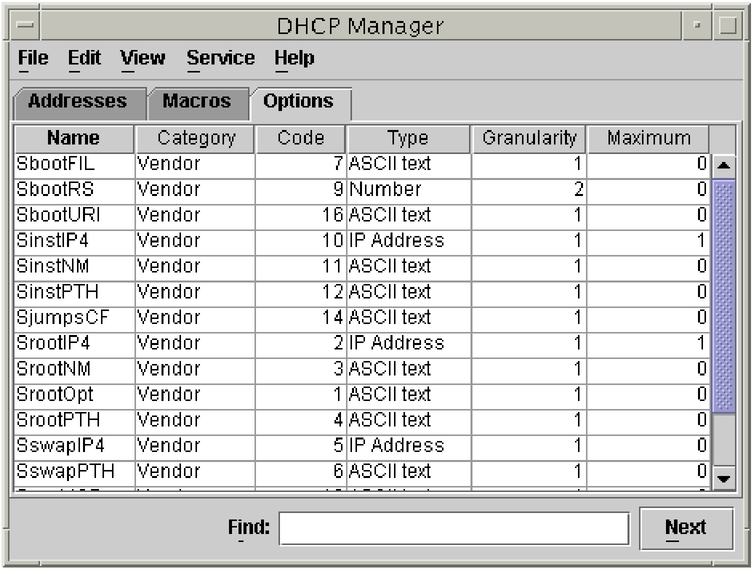
2. **If the following option names are not already defined in the DHCP server, add them:**

   `SinstNM`, `SinstIP4`, `SinstPTH`, `SrootNM`, `SrootIP4`, `SrootPTH`, `BootFile`, `SbootURI`, `BootSrvA`

Note – If you intend to perform a Jumpstart installation of Solarix x86, you also need to add definitions for `SsysidCF` and `SjumpsCF`.

---

   a. **Find out which option names are already defined in the DHCP server by clicking the Options tab in the DHCP Manager's main window (see FIGURE 10-1).**



Name	Category	Code	Type	Granularity	Maximum
SbootFIL	Vendor	7	ASCII text	1	0
SbootRS	Vendor	9	Number	2	0
SbootURI	Vendor	16	ASCII text	1	0
SinstIP4	Vendor	10	IP Address	1	1
SinstNM	Vendor	11	ASCII text	1	0
SinstPTH	Vendor	12	ASCII text	1	0
SjumpsCF	Vendor	14	ASCII text	1	0
SrootIP4	Vendor	2	IP Address	1	1
SrootNM	Vendor	3	ASCII text	1	0
SrootOpt	Vendor	1	ASCII text	1	0
SrootPTH	Vendor	4	ASCII text	1	0
SswapIP4	Vendor	5	IP Address	1	0
SswapPTH	Vendor	6	ASCII text	1	0

FIGURE 10-1 The DHCP Manager 'Options' Tab

   b. **Use the command line to add (using `-A`, as shown below) or modify (using `-M` instead of `-A`) the required option strings.**

   To do this, continue as `root` on the Network Install Server, and in a terminal window type the command for each option you require. The full list of required options is shown in FIGURE 10-2.

Note – Note that, although some of the required DHCP options strings might already have been defined on your DHCP server, SbootURI is a new option string that has not been used before on Sun platforms.

```
# dhtadm -A -s SrootIP4 -d ’Vendor=SUNW.i86pc,2,IP,1,1’
# dhtadm -A -s SrootNM -d ’Vendor=SUNW.i86pc,3,ASCII,1,0’
# dhtadm -A -s SrootPTH -d ’Vendor=SUNW.i86pc,4,ASCII,1,0’
# dhtadm -A -s SinstIP4 -d ’Vendor=SUNW.i86pc,10,IP,1,1’
# dhtadm -A -s SinstNM -d ’Vendor=SUNW.i86pc,11,ASCII,1,0’
# dhtadm -A -s SinstPTH -d ’Vendor=SUNW.i86pc,12,ASCII,1,0’
# dhtadm -A -s SsysidCF -d ’Vendor=SUNW.i86pc,13,ASCII,1,0’
# dhtadm -A -s SjumpsCF -d ’Vendor=SUNW.i86pc,14,ASCII,1,0’
# dhtadm -A -s SbootURI -d ’Vendor=SUNW.i86pc,16,ASCII,1,0’
```

FIGURE 10-2 Commands for Configuring the Option Strings

Caution – When you are configuring the DHCP option strings, make sure you allocate the option string code correctly for each option. These values are used by the network bootstrap process and the process will fail if the values are not specified correctly. The option code is the fourth value from the right on the command line. For example, the code for SbootURI is 16 (see FIGURE 10-2). If you specify values that are different from the values in FIGURE 10-2, the blades will not be bootable from the network.

3. **Verify that you have specified the DHCP option strings correctly.**

   Type:

```
# dhtadm -P
:

SrootIP4      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,2,IP,1,1
SinstPTH      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,12,ASCII,1,0
SinstNM       Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,11,ASCII,1,0
SinstIP4      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,10,IP,1,0
SbootURI      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,16,ASCII,1,0
SjumpsCF      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,14,ASCII,1,0
SsysidCF      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,13,ASCII,1,0
SrootPTH      Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,4,ASCII,1,0
SrootNM       Symbol       Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,3,ASCII,1,0
#
```

FIGURE 10-3 Sample `dhtadm -P` Output For Checking the Option Strings Are Correct

The : character beneath the first user prompt in FIGURE 10-3 indicates omitted data.

Note – FIGURE 10-3 shows output relating to the DHCP options strings (output relating to the macros has been omitted and the omission is indicated by the : character). Note that different vendor names (for example, `SUNW.Ultra-1`, `SUNW.Ultra-30`, `SUNW.i86pc`) might be associated with each option string in your configuration, but that the user-specified values for the other fields of the command line must be exactly as printed in FIGURE 10-3. For example, the last four values for the `SbootURI` option need to be `16`, `ASCII`, `1`, `0`.

For further information about adding options, see refer to the *Solaris DHCP Administration Guide*.

4. **Proceed to Section 10.3.2, "Adding the Global PXE Macro for Solaris x86 to the DHCP Server" on page 10-8.**

10.3.2 Adding the Global PXE Macro for Solaris x86 to the DHCP Server

---

Note – The instructions in this section only need to be performed once on the DHCP server. If you already have the PXE macro correctly defined for Solaris x86, you can skip this section and proceed to Section 10.1, "Overview of the Solaris x86 Installation Procedures" on page 10-2. However, it is critical that the macro is defined correctly, so if you are in any doubt follow the instructions in this section. For equivalent CLI (command-line interface) commands, refer to Section 10.11.3, "Using the DHCP Manager's Command-line Interface Instead of the GUI" on page 10-46.

To define the global PXE macro:

1. **In the main window of DHCP Manager's GUI, click the Macros tab, and select Create from the Edit menu.**

2. **In the Name field of the Create Macro window, type the name of the PXE macro:** `PXEClient:Arch:00000:UNDI:002001`

Caution – The global PXE macro is named `PXEClient:Arch:00000:UNDI:002001`. Make sure you type this name correctly. If you make a mistake, the blades will not be able to perform a PXE boot installation of the Solaris x86 operating system.

---

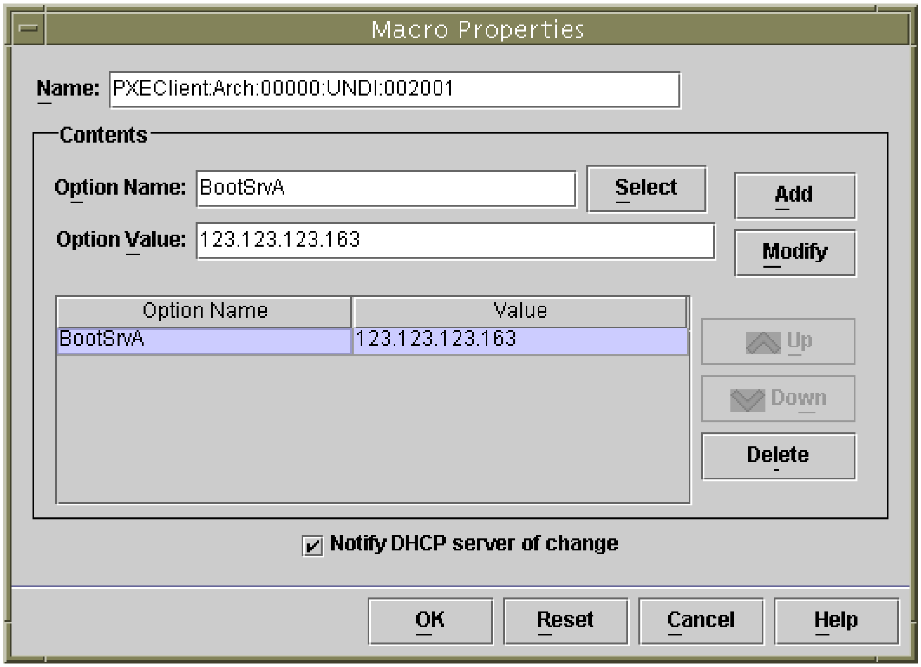
3. **Complete the other fields in the Create Macro window**

   a. **In the Option Name field, type** `BootSrvA`**.**

b. In the Option Value field, type the IP address of your Network Install Server.

c. Click Add, and click OK.

To view the properties of the macro you have created, select it from the list of macros displayed on the left of the Macros tab, then select Properties from the Edit menu (see FIGURE 10-4).



Macro Properties

Name: PXEClient:Arch:00000:UNDI:002001

Contents

Option Name: BootSrvA

Option Value: 123.123.123.163

Option Name	Value
BootSrvA	123.123.123.163

☑ Notify DHCP server of change

FIGURE 10-4 The Property Defined for the Global PXE Macro

Note – The global PXE macro has only a single property defined: `BootSrvA`.

---

4. **Proceed to Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10.**

10.4 Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade

Before following the instructions in this section, make sure you have completed all the steps in the previous sections of this chapter, and that you have performed any steps relating to the Solaris x86 installation in the latest Product Notes.

The tasks in this section need to be performed for every blade that you intend to install Solaris x86 onto. They are as follows:

- Find out and note down the blade's MAC address (Step 1).
- Run the `add_install_client` script on the Network Install Server (Step 2, Step 3).
- Configure a client-specific DHCP macro on the DHCP server (Step 4, Step 5, Step 6).
- Configure the IP address for the client on the DHCP server (Step 7)

After performing Step 7, you will need to perform the following tasks:

- Configure the blade temporarily to boot from the network (the instructions for this are in Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18
- Reset or power on the blade and monitor its booting processes (the instructions for these tasks are in Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18 and Section 10.7, "Monitoring the Network Booting Process and Starting the Solaris Installation" on page 10-20)

1. **Make a note of the MAC address of the blade you are intending to install Solaris x86 onto (see Section 10.1, "Overview of the Solaris x86 Installation Procedures" on page 10-2).**

   If you are intending to use an interface other than the first network interface on the blade, see Section 10.12, "Installing Solaris x86 Onto a Blade by Using the Second, Third, or Fourth Network Interface" on page 10-47.

2. **Log in as `root` to the system you are using as the Network Install Server, and run the `add_install_client` script.**

   When you run this script make sure that you use the correct `bootpath` parameters for the server blade.

   The correct `bootpath` parameter for a B100x blade is shown in FIGURE 10-5.

   The correct `bootpath` parameter for a B200x blade is shown in FIGURE 10-6.

Note – The -b option for the add_install_client command is new. For information about this option, see Section 10.13, "The New add_install_client -b Option" on page 10-50.

If you are intending to perform a Jumpstart installation, you need to use additional parameters on the command line when you run the add_install_client script.

For information about the parameters to use for Jumpstart, refer to Section 10.9, "Preparatory Steps for Setting up a Jumpstart Installation for a Blade" on page 10-34, and to Section 10.10, "Configuring a Jumpstart Installation" on page 10-39.

- **For a B100x blade with the MAC address** 00:03:ba:29:f0:de**, see the sample command in** FIGURE 10-5**.**

```
# cd install-dir-path/Solaris_9/Tools
# ./add_install_client -d -e "00:03:ba:29:f0:de" \
> -b "input-device=ttya" -b "output-device=ttya" \
> -b "bootpath=/pci@0,0/pci108e,16a8@8" \
> i86pc
```

FIGURE 10-5 Sample Command Showing the bootpath Property for a B100x Blade

where *install-dir-path* is the location of your install image.

Note – In the sample commands in this step, the '\' character tells the operating system that the command is being continued on the next line.

Note – If you are configuring multiple blades, you might want to create a wrapper script to run the add_install_client command for each blade (see Section 10.11.1, "Calling the add_install_client Utility From a Wrapper Shell Script" on page 10-42).

- **For a B200x blade with the MAC address** 00:03:ba:2d:d4:a0**, see the sample command in** FIGURE 10-6**.**

```
# cd /export/s9x/Solaris_9/Tools
# ./add_install_client -d -e "00:03:ba:2d:d4:a0" \
> -b "input-device=ttya" -b "output-device=ttya" \
> -b "bootpath= /pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci108e,16a8@3" \
> i86pc
```

FIGURE 10-6 Sample Command Showing the bootpath Property for a B200x Blade

FIGURE 10-7 shows sample output from the add_install_client script executed with a bootpath for the B100x blade.

```
# cd /export/s9x/Solaris_9/Tools
# ./add_install_client -d -e "00:03:ba:29:f0:de" \
> -b "input-device=ttya" -b "output-device=ttya" \
> -b "bootpath=/pci@0,0/pci108e,16a8@8" \
> i86pc
cleaning up preexisting install client "00:03:ba:29:f0:de"
To disable 00:03:ba:29:f0:de in the DHCP server,
  remove the entry with Client ID 010003BA29F0DE

To enable 010003BA29F0DE in the DHCP server, ensure that
the following Sun vendor-specific options are defined
(SinstNM, SinstIP4, SinstPTH, SrootNM, SrootIP4,
SrootPTH, SbootURI and optionally SjumpCF and SsysidCF),
and add a macro to the server named 010003BA29F0DE,
containing the following option values:

  Install server      (SinstNM)  : cerberus
  Install server IP   (SinstIP4) : 123.123.123.163
  Install server path (SinstPTH) : /export/s9x
  Root server name    (SrootNM)  : cerberus
  Root server IP      (SrootIP4) : 123.123.123.163
  Root server path    (SrootPTH) : /export/s9x/Solaris_9/Tools/Boot
  Boot file           (BootFile) : nbp.010003BA29F0DE
  Solaris boot file   (SbootURI) : tftp://123.123.123.163/010003BA29F0DE

If not already configured, enable PXE boot by creating
a macro called PXEClient:Arch:00000:UNDI:002001
which contains the following values:
  Boot server IP      (BootSrvA) : 123.123.123.163
This macro will be explicitly requested by the PXE boot.
```

FIGURE 10-7 Sample Output From the add_install_client Script

The sample command illustrated in FIGURE 10-7 uses the new (-b) boot option. For information about the arguments taken by this option, and required for the PXE boot process to work on a blade, see Section 10.13, "The New add_install_client -b Option" on page 10-50 at the end of this chapter.

3. **Make a note of the options listed in the output from the add_install_client script (see FIGURE 10-7).**

   You need to note the option names and their values.

   The output from the add_install_client script is displayed in three sections. The first contains text explaining that the previous install configurations associated with the specified client are being cleaned up in preparation for the new install configuration. The second contains a list of options that are specific to the client. These are the options that you need to write down; you will need to add them as properties (in later steps) to the client-specific DHCP macro. Finally, the third contains information concerning the global PXE boot macro (including the name of the global macro).

4. **Make sure the required option names are defined in the DHCP server.**

   You defined these in Section 10.3.1, "Adding the Required Option Strings to the DHCP Server" on page 10-5.

5. **Make sure the global PXE macro for Solaris x86 has been correctly added to the DHCP server.**

   You added this in Section 10.3.2, "Adding the Global PXE Macro for Solaris x86 to the DHCP Server" on page 10-8.

6. **Create the client-specific macro for the blade you are intending to install Solaris x86 onto.**

   To use the command-line interface, see Section 10.11.3, "Using the DHCP Manager's Command-line Interface Instead of the GUI" on page 10-46.

   To use the GUI, do the following:

   a. **If you are not already running the DHCP Manager GUI, log into the Network Install Server as** `root`**, and start the DHCP Manager GUI by typing:**

   ```
   # DISPLAY=mydisplay:0.0
   # export DISPLAY
   # /usr/sadm/admin/bin/dhcpmgr &
   ```

   where *mydisplay* is the name of the system (for example, a desktop workstation) that you are using to display the DHCP Manager's GUI (Graphical User Interface).

   b. **In the DHCP Manager main window, click the Macros tab, and select Create from the Edit menu.**

   The blades are indentified to the DHCP server by a client identifier (ID) string. This string contains the digits 01 followed by the MAC address of the blade's network interface (however, the string does not include any colons). In the example we have been using this MAC address is `00:03:ba:29:f0:de`. The client ID for the blade is therefore `010003BA29F0DE` (see FIGURE 10-7).

   c. **With the Create Macro window open:**

      i. **In the Name field of the Create Macro window, type the client ID for your blade.**

      In the example we have been using (see FIGURE 10-7), the client ID is `010003BA29F0DE`, therefore the name of the macro for this sample client is `010003BA29F0DE`.

ii. **In the Contents section of the Create Macro window, click the Select button.**

iii. **From the drop-down Category list, select Vendor.**

iv. **Select** `SinstNM` **and click OK.**

v. **Delete any existing information in the Option Value field.**

vi. **Using the data you wrote down in Step 3 (in this section), type the correct Option Value for** `SinstNM`**.**

vii. **Click Add.**

viii. **Repeat Step iv through Step vii for** `SinstIP4`, `SinstPTH`, `SrootNM`, `SrootIP4`, `SrootPTH`, **and** `SbootURI`**.**

ix. **When you have configured the seven Vendor options for the client, click Select in the Create Macro window and, from the drop-down Category list, select Standard.**

x. **Select** `BootFile` **and click OK.**

xi. **Delete any existing information in the Option Value field.**

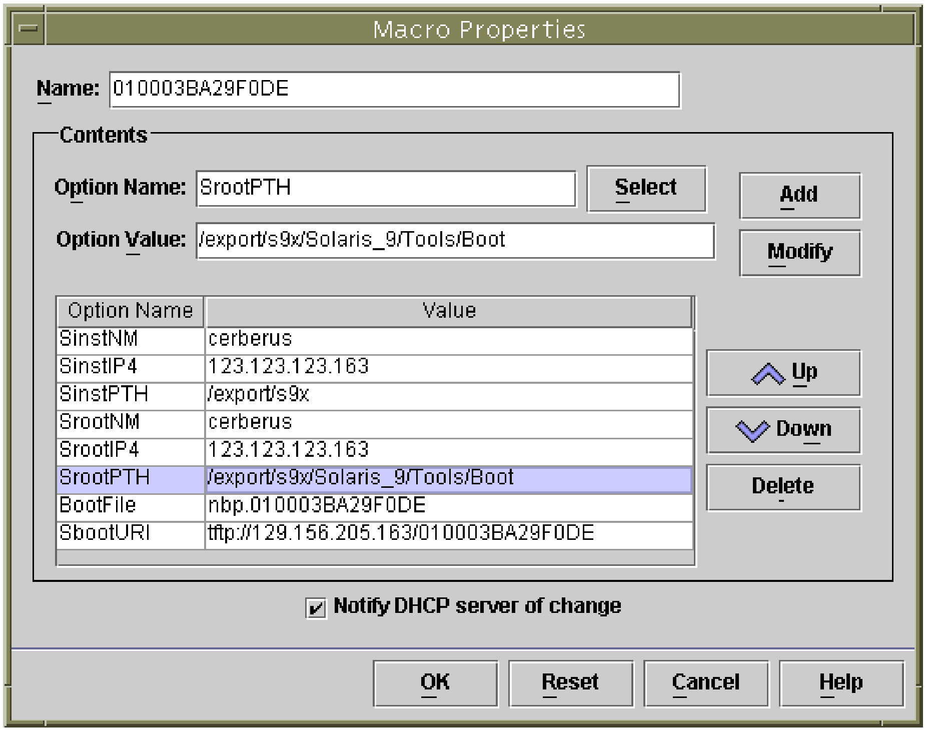
xii. **Using the data you wrote down in Step 3 (in this section), type the correct Option Value for** `BootFile`**.**

xiii. **Click Add.**

xiv. **Repeat Step iv through Step x through Step xiii for the** `BootSrvA` option.

xv. **When you have configured the client-specific macro with each of the options that were listed in the output from the** `add_install_client` **script (see Step 2 and FIGURE 10-7), click OK.**

For information about the extra configuration you need to perform at this stage if you are intending to perform a Jumpstart installation, see Section 10.10, "Configuring a Jumpstart Installation" on page 10-39.



Macro Properties

Name: 010003BA29F0DE

Contents

Option Name: SrootPTH

Option Value: /export/s9x/Solaris_9/Tools/Boot

Option Name	Value
SinstNM	cerberus
SinstIP4	123.123.123.163
SinstPTH	/export/s9x
SrootNM	cerberus
SrootIP4	123.123.123.163
SrootPTH	/export/s9x/Solaris_9/Tools/Boot
BootFile	nbp.010003BA29F0DE
SbootURI	tftp://129.156.205.163/010003BA29F0DE

☑ Notify DHCP server of change

FIGURE 10-8 Sample Property Window for a B100x Blade's Client-specific Macro

7. **Assign an IP address for the blade in the DHCP server.**

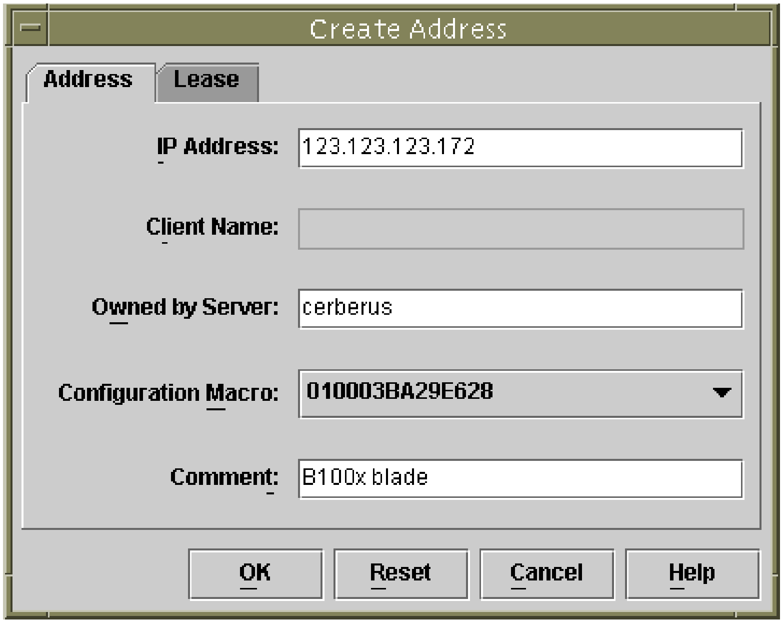
   To use the command-line interface, see Section 10.11.3, "Using the DHCP Manager's Command-line Interface Instead of the GUI" on page 10-46.

   To use the GUI, do the following:

   a. **In the main DHCP Manager window, click on the Addresses tab.**

   b. **Select and double-click the IP address that you want the blade to use.**

      The address you choose will be from the block that you reserved (in Section 10.1, "Overview of the Solaris x86 Installation Procedures" on page 10-2) for the server blades in the chassis.



IP Address: 123.123.123.172

Client Name:

Owned by Server: cerberus

Configuration Macro: 010003BA29E628

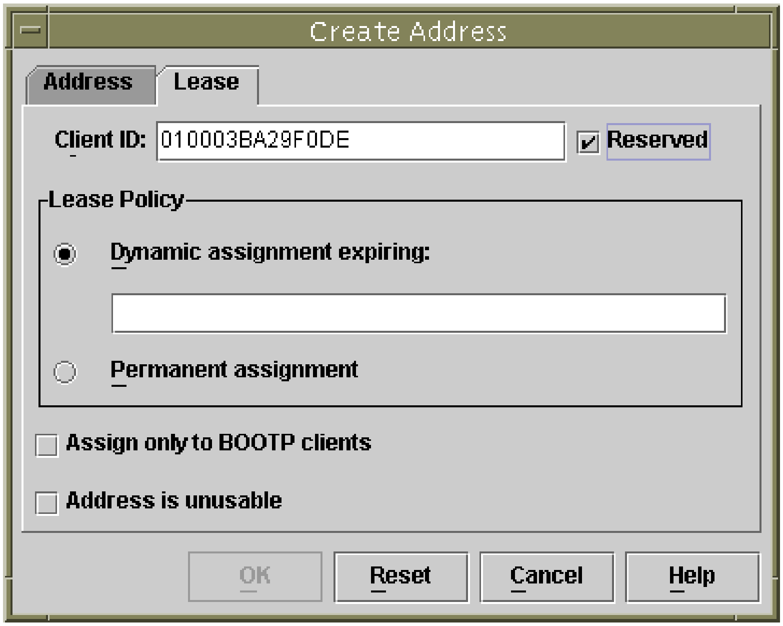
Comment: B100x blade

FIGURE 10-9 Creating an IP Address for the Blade to Use

c. **From the drop down selection list labeled Configuration Macro, select the name of the the client-specific macro that you set up in Step 6.**

d. **In the Create Address window, click the Lease tab (see FIGURE 10-10).**

In the Client ID field, type the Client ID for the blade (that is, 01 followed by the blade’s MAC address, with all alphabetic characters in uppercase, and without any colons; see Step 6 on page 10-13). Click OK.



Client ID: 010003BA29F0DE

☑ Reserved

Lease Policy

● Dynamic assignment expiring:

○ Permanent assignment

□ Assign only to BOOTP clients

□ Address is unusable

FIGURE 10-10 Associating the Blade's Client ID With the IP Address

8. **If you are installing Solaris x86 onto a blade that previously ran Linux, proceed to Section 10.5, "Re-initializing the Hard Disk On a Blade That Previously Ran Linux" on page 10-17.**

   Otherwise skip this step.

9. **Proceed to Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18.**

10. **Power on the Blade by following the instructions also in Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18.**

10.5 Re-initializing the Hard Disk On a Blade That Previously Ran Linux

The Solaris x86 and Linux operating systems use different methods to lay out the disk partition table. Therefore when Solaris x86 first installs onto a blade that has previously been installed with Linux, it prompts you to run the fdisk utility to set up a Solaris disk partition table. This prompt requires user input and therefore causes a potential interruption to a Jumpstart installation. To avoid this problem, if you want to perform a completely automated custom Jumpstart on a B100x or B200x blade that has previously had Linux installed, you must first delete the partition table by using the command below. However, read the following caution before executing this command.

Caution – When you have deleted the disk partition table, any data stored on the hard disk is no longer accessible. Also, when you have done this, you can no longer boot Linux from the blade s hard disk. If you want to run Linux on the blade again, you must install it from the network by following the instructions in Chapter 4.

```
# dd if=/dev/zero of=/dev/hda count=512
512+0 records in 512+0 records out
```

You can automate this task within the Jumpstart configuration by using the `fdisk` keyword in the `x86-class` script. For more information, see Section 10.9, "Preparatory Steps for Setting up a Jumpstart Installation for a Blade" on page 10-34.

10.6 Configuring a Blade to Boot Temporarily From the Network

---

Note – To install Solaris x86 from a network install image onto a blade, you need to configure the blade temporarily to boot from the network. The System Controller command that you type in Step 2 below to configure the blade to do this is effective for 10 minutes. After that the blade's BIOS reverts to its previous booting behavior. Therefore, to cause the blade to boot from the network you must power it on within 10 minutes of running the `bootmode` command. (If the blade was already powered on when you ran the `bootmode` command, then to cause it to boot from the network you must reset the blade within 10 minutes. Follow the instructions below.)

---

1. **Log into the active System Controller by following the instructions in Chapter 2 of the** ***Sun Fire B1600 Blade System Chassis Software Setup Guide*** **if you are logging into a brand new chassis in its factory default state.**

   Otherwise log in using the user name and password assigned to you by your system administrator.

2. **Type the following command at the System Controller’s `sc>` prompt to cause the blade to boot from the network:**

```
sc> bootmode bootscript=”boot net” sn
```

where *n* is the number of the slot containing the blade.

Alternatively, if you want to install the blade by using a differnet network interface, see Section 10.12, “Installing Solaris x86 Onto a Blade by Using the Second, Third, or Fourth Network Interface” on page 10-47.

3. **Power on the blade by typing:**

```
sc> poweron sn
```

or, if the blade is already powered on, type:

```
sc> reset sn
```

where *n* is the number of the slot containing the blade.

4. **Connect to the blade console by typing:**

```
sc> console -f sn
```

Note – The `-f` parameter is optional, but it is sometimes useful. The ‘f’ stands for ‘force’, and this option forces you onto a blade console even if someone else is using that console (the other person is not forced out of the console but will be granted read-only access for the rest of the session).

---

5. **Proceed to Section 10.7, “Monitoring the Network Booting Process and Starting the Solaris Installation” on page 10-20.**

Note – If you are performing an interactive installation, you must make sure that separate Boot and Solaris partitions are defined during the installation procedure. The way in which you need to do this depends upon the install media you are using and whether your blade is in its factory default state. Instructions for how to define the partitions correctly are provided in Section 10.8, “Specifying Disk Partitioning During an Interactive Installation” on page 10-23.

10.7 Monitoring the Network Booting Process and Starting the Solaris Installation

When you have booted a blade (by following the instructions in Section 10.6, “Configuring a Blade to Boot Temporarily From the Network” on page 10-18), you can monitor the booting processes to check that no problems occur.

At the end of these booting processes, the server blade will prompt you to select the Solaris interactive installation or the Jumpstart installation.

1. **Connect to the blade console by typing:**

```
sc> console -f sn
```

where *n* is the number of the slot containing the blade.

2. **View the output displayed during the booting process:**

After displaying the BIOS initialization screens, the blade will start to PXE boot from the network. At this point you will see the following information, which includes the MAC address that the blade is using for the boot process:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000
000000000000
DHCP./
```

After a few seconds the blade will pick up the primary bootstrap program from the network install image and the following message will be displayed.

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000
000000000000
CLIENT IP: 123.123.123.172 MASK: 255.255.255.0
DHCP IP: 123.123.123.163  GATEWAY IP: 123.123.123.8



Solaris network boot ...
```

After a few more seconds the primary bootstrap will load and execute the secondary bootstrap program.

The following screen illustrates this point in the booting process:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000
000000000000
CLIENT IP: 123.123.123.172  MASK: 255.255.255.0
DHCP IP: 123.123.123.163
SunOS Secondary Boot version 3.00


Solaris network boot ...
```

After a few more seconds again a screen will appear prompting you to specify whether you want to perform a Solaris interactive or a Jumpstart installation.

3. **Press 1 and press [RETURN] to perform the interactive installation.**

```
             <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci108e,16a8@8
Boot args:


Select the type of installation you want to perform:

        1 Solaris Interactive
        2 Custom JumpStart

Enter the number of your choice followed by the <ENTER> key.
Alternatively, enter custom boot arguments directly.

If you wait for 30 seconds without typing anything,
an interactive installation will be started.

Select type of installation:1
```

When you have specified the type of installation you require, the blade begins to boot the Solaris operating system:

```
          <<< starting interactive installation >>>

Booting kernel/unix...
SunOS Release 5.9 Version Generic_112234-11 32-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
```

The interactive installation program then begins:

```
Select a Language

   0. English
   1. French
   2. German
   3. Italian
   4. Japanese
   5. Korean
   6. Simplified Chinese
   7. Spanish
   8. Swedish
   9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help:
```

4. **Select the language you require.**
5. **Proceed to section Section 10.8, "Specifying Disk Partitioning During an Interactive Installation" on page 10-23.**

10.8 Specifying Disk Partitioning During an Interactive Installation

If you are performing an interactive installation of Solaris x86, you need to make sure that separate Boot and Solaris partitions are defined on the blade's hard disk. This enables the blade to identify its boot device correctly during reboots performed after the operating system has been installed from the network.

If you are performing a Jumpstart installation, skip this section. For blades that use Jumpstart, the boot device is set by the custom `x86-finish` script after the installation has completed and regardless of the disk partitions defined. For information about the x86-finish script, see Section 10.9, "Preparatory Steps for Setting up a Jumpstart Installation for a Blade" on page 10-34.

If you do not define separate Boot and Solaris partitions during an interactive installation, you may encounter the problem described in Chapter 14, "Synopsis: Blade Boots to Device Configuration Assistant on Every Reboot After an Interactive Network Installation" on page 14-14.

The actions you need to perform in this section depend on the install media (CDs or DVD) you have used to build the install image on your Network Install Server.

- For CD installation, see Section 10.8.1, “Disk Partitioning for an Install Image Created From the Solaris CD Media” on page 10-24
- For DVD installation, see Section 10.8.2, “Disk Partitioning for an Install Image Created From the Solaris DVD Media” on page 10-24

10.8.1 Disk Partitioning for an Install Image Created From the Solaris CD Media

If you are installing Solaris x86 onto:

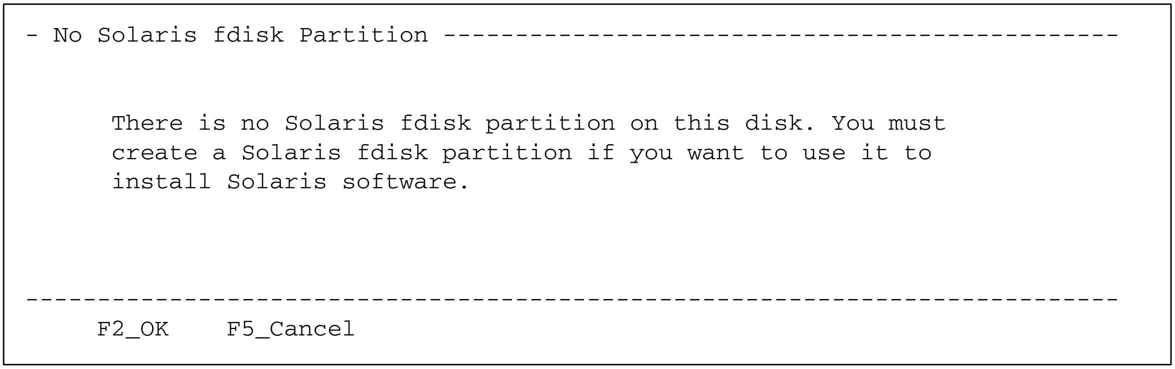
- A blade in its factory default state, you will be prompted by the Solaris installation utility to create a Solaris fdisk partition on the hard disk (In their factory default state the blades have no partition table defined). To create the correct disk partition table, follow the instructions in Section 10.8.3, “Creating a Solaris fdisk Partition Using the Solaris Installation Utility” on page 10-25.
- A previously used blade whose disk partition table contains more than one disk partition, you will be prompted to decide whether to re-use the existing partition layout or to abort the installation utility. If the existing table contains separate Solaris and Boot partitions, you can use the existing table. Otherwise you need to cancel the installation and remove the existing partition table. For instructions, see Section 10.8.4, “Re-using or Deciding to Remove an Existing Partition Table” on page 10-26.
- A previously used blade whose disk partition table contains only a single partition, you will not receive any prompts or messages concerning the disk partition table, but nevertheless you must remove the existing partition table. For instructions, see Section 10.8.5, “Aborting the Installation for a Used Blade Whose Disk Contains only a Single Partition” on page 10-27

10.8.2 Disk Partitioning for an Install Image Created From the Solaris DVD Media

During a Webstart installation, select the ’Custom Install’ option and specify separate Boot and Solaris partitions (see Section 10.8.7, “Specifying Separate Boot and Solaris Partitions During a Manual Webstart Installation” on page 10-31.

10.8.3 Creating a Solaris fdisk Partition Using the Solaris Installation Utility

If you are installing Solaris x86 onto a blade in its factory default state, you will receive the following message from the Solaris installation utility:



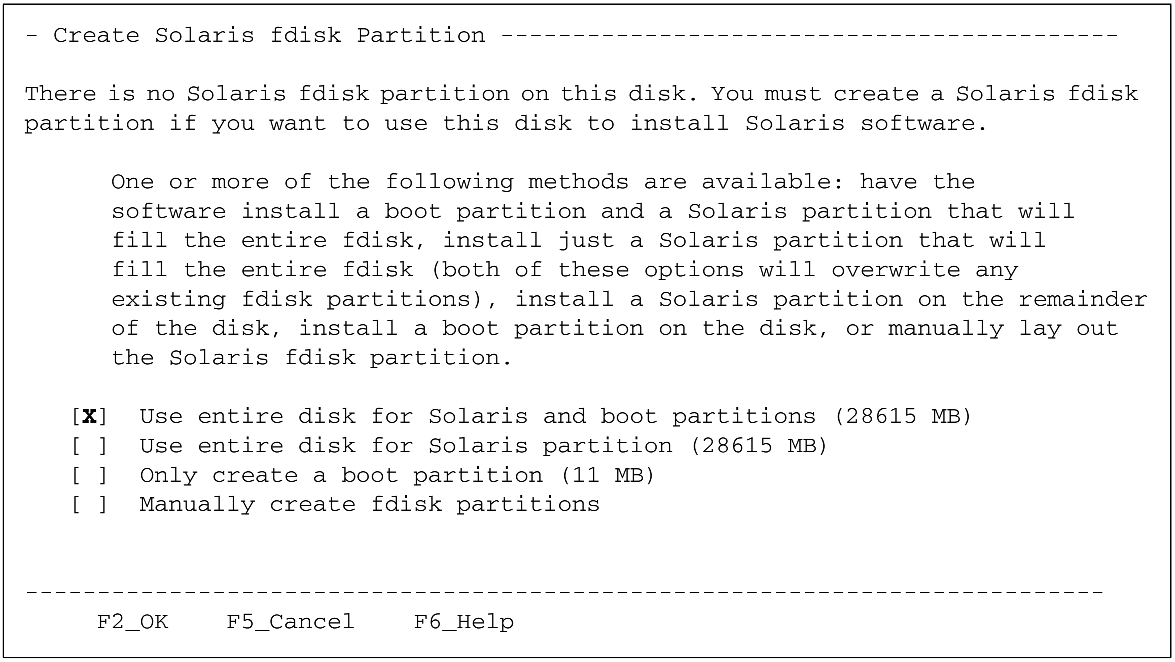
```
- No Solaris fdisk Partition ------------------------------------------------


      There is no Solaris fdisk partition on this disk. You must
      create a Solaris fdisk partition if you want to use it to
      install Solaris software.




----------------------------------------------------------------------------
     F2_OK    F5_Cancel
```

1. **Press [F2].**
2. **In the screen for creating a Solaris fdisk partition, select** “`Use entire disk for Solaris and boot partitions (28615MB)`”**.**



```
- Create Solaris fdisk Partition -------------------------------------------

There is no Solaris fdisk partition on this disk. You must create a Solaris fdisk
partition if you want to use this disk to install Solaris software.

      One or more of the following methods are available: have the
      software install a boot partition and a Solaris partition that will
      fill the entire fdisk, install just a Solaris partition that will
      fill the entire fdisk (both of these options will overwrite any
      existing fdisk partitions), install a Solaris partition on the remainder
      of the disk, install a boot partition on the disk, or manually lay out
      the Solaris fdisk partition.

   [X]  Use entire disk for Solaris and boot partitions (28615 MB)
   [ ]  Use entire disk for Solaris partition (28615 MB)
   [ ]  Only create a boot partition (11 MB)
   [ ]  Manually create fdisk partitions


---------------------------------------------------------------------------
     F2_OK    F5_Cancel    F6_Help
```

3. **Press [F2].**

4. **Go to Section 10.8.8, "Completing the Solaris x86 Installation" on page 10-33.**

10.8.4 Re-using or Deciding to Remove an Existing Partition Table

If you are installing Solaris x86 onto a previously used blade whose disk partition table contains more than one disk partition, you will be prompted by the Solaris installation utility to decide whether to re-use the existing partition layout or to abort the installation utility:

```
- Use x86boot partition? ----------------------------------------------------

        An x86boot partition has been detected on c0d0p1.  It points to
        a Solaris root filesystem on c0d0s0, though no attempt has been
        made to verify that a valid Solaris system exists at that
        location.  Do you want to use this x86boot partition to be
        reused now when you install the system?

        WARNING: If you elect to reuse this x86boot partition, the
        Solaris system whose root filesystem is on c0d0s0 will be
        rendered unusable.


------------------------------------------------------------------------------
     F2_OK    F5_Cancel
```

- **If you know that the existing disk partition table contains separate Solaris and Boot partitions, continue the installation process by pressing [F2], then go to Section 10.8.8, "Completing the Solaris x86 Installation" on page 10-33.**

Note – For information about what happens if you press [F2] but the disk partition table contains separate Solaris and Boot partitions, see Chapter 14.

If you are not certain that the disk partition table contains separate Solaris and Boot partitions, you need to cancel the installation, remove the entire disk partition table, and then run the Solaris installation program again.

Do the following:

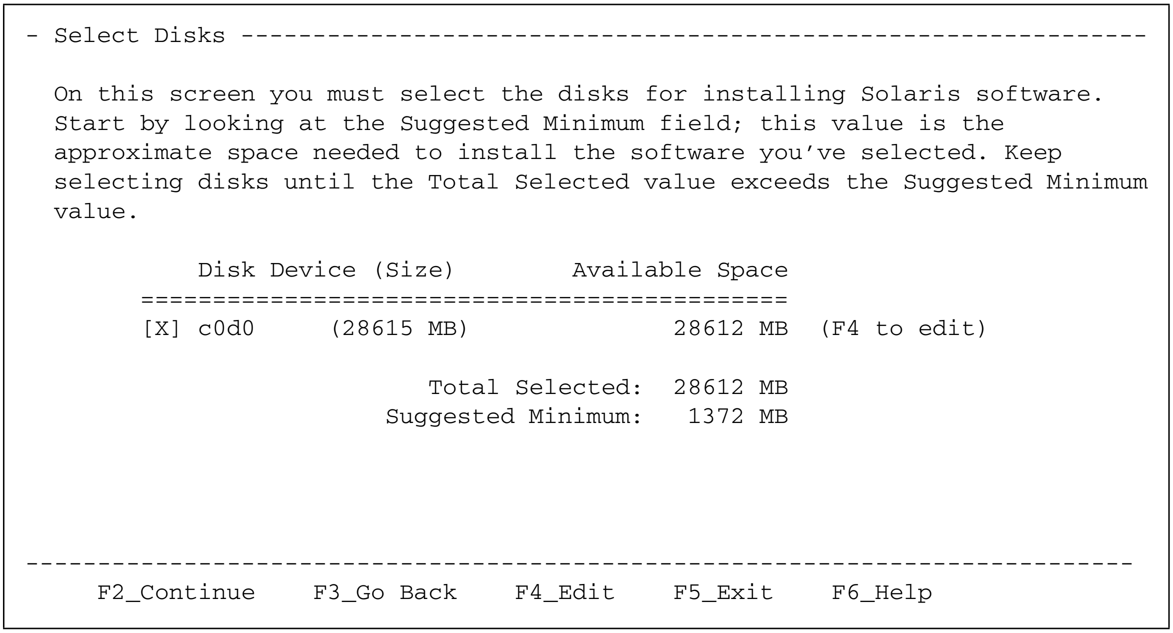
1. **Press [F5] to cancel the installation.**

2. **Follow the instructions in Section 10.8.6, "Removing the Entire Disk Partition Table Before Restarting the Solaris Install Program" on page 10-28.**

10.8.5 Aborting the Installation for a Used Blade Whose Disk Contains only a Single Partition

If you are installing Solaris x86 onto a previously used blade whose disk partition table contains only a single partition (that is, it contains no separate Boot and Solaris partitions), you will not receive an error message to the effect that there is "`No Solaris fdisk Partition`" on the disk, or prompting you to use a particular partition.

Caution – If you arrive at the "Select Disks" screen and you have not received a disk partition error message or prompt, then you must abort the Solaris installation.



```
- Select Disks ---------------------------------------------------------------

  On this screen you must select the disks for installing Solaris software.
  Start by looking at the Suggested Minimum field; this value is the
  approximate space needed to install the software you've selected. Keep
  selecting disks until the Total Selected value exceeds the Suggested Minimum
  value.

            Disk Device (Size)         Available Space
        =============================================
        [X] c0d0     (28615 MB)              28612 MB  (F4 to edit)

                            Total Selected:  28612 MB
                         Suggested Minimum:   1372 MB




-------------------------------------------------------------------------------
     F2_Continue    F3_Go Back    F4_Edit    F5_Exit    F6_Help
```

1. **Press [F5].**

2. **Follow the instructions in Section 10.8.6, "Removing the Entire Disk Partition Table Before Restarting the Solaris Install Program" on page 10-28.**

10.8.6 Removing the Entire Disk Partition Table Before Restarting the Solaris Install Program

This section tells you how to remove an existing disk partition table on a blade so that Solaris will install onto it as if the blade is in its factory default state. You need to do this to prevent the blade from booting to the Device Configuration Assistant each time you reboot after performing an interactive network installation on a blade with a previously existing disk partition table.

Note – If you abort an interactive network installation on a blade, you will remain logged in as root.

---

1. **At the blade's console prompt, run the `format` command:**

```
# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0d0 <DEFAULT cyl 58098 alt 2 hd 16 sec 63>
          /pci@0,0/pci-ide@1f,1/ide@0/cmdk@0,0
Specify disk (enter its number): 0
```

2. **Type** 0 **(to specify the disk you want to format) and press [ENTER].**

3. **At the `format>` prompt, type:**

```
format> fdisk
Total disk size is 58140 cylinders
             Cylinder size is 1008 (512 byte) blocks

                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1       Active    Solaris           1  58100    58100    100




SELECT ONE OF THE FOLLOWING:

   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Exit (update disk configuration and exit)
   5. Cancel (exit without updating disk configuration)
Enter Selection: 3
```

4. **Type `3` (“`Delete a partition`”).**

5. **When prompted, specify the number of the partition to be deleted.**

   In the example in Step 3, the partition to be removed is number 1.

6. **Type `Y` at the next prompt to delete the partition:**

```
Are you sure you want to delete partition 1? This will make all files and
programs in this partition inaccessible (type "y" or "n"). y
```

7. **Repeat Step 4 through Step 6 until there are no longer any partitions defined:**

```
Total disk size is 58140 cylinders
             Cylinder size is 1008 (512 byte) blocks

                                                   Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===


WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:

   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Exit (update disk configuration and exit)
   5. Cancel (exit without updating disk configuration)
Enter Selection:
```

8. **Type** `4` **to exit the** `fdisk` **utility, then type** `q` **to quit the** `format` **utility.**

```
SELECT ONE OF THE FOLLOWING:

   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Exit (update disk configuration and exit)
   5. Cancel (exit without updating disk configuration)
Enter Selection: 4


Solaris fdisk partition not found
No fdisk solaris partition found
format> q
#
```

9. **Now that the blade's hard disk has been restored to its factory default state, restart the Solaris installation procedure.**

   To do this, follow the instructions in Section 10.6, "Configuring a Blade to Boot Temporarily From the Network" on page 10-18, and then repeat Section 10.7, "Monitoring the Network Booting Process and Starting the Solaris Installation" on page 10-20.

10.8.7 Specifying Separate Boot and Solaris Partitions During a Manual Webstart Installation

This section is for network install images created from DVD media resulting in the Webstart Installation utility running. This utility manages the installation of the operating system onto the blade. Follow the instructions in this section to make sure the disk partitions on the blade are correctly defined to enable the blade to reboot after the operating system has installed from the network.

1. **When prompted, type** `2` **to select the option to perform a 'Custom Install':**

```
To install basic Solaris products into their default directory locations,
select Default Install.

Custom install provides a choice of which Solaris products to install. For each
product, it also provides an option to customize the products install.

Types of install available:

1. Default Install
2. Custom Install

   Select the number corresponding to the type of install you would like [1]: 2
```

2. **When prompted to lay out file systems on disk c0d0 (bootdisk), type `y`:**

```
Please indicate if you want the Default Packages for the Entire Group or if you
want to select Custom Packages. Selecting Custom Packages allows you to add or
remove packages from the selected Solaris Software Group.  When selecting which
packages to add or remove, you will need to know about software dependencies
and how Solaris software is packaged.

1. Default Packages
2. Custom Packages

   Default Packages or Custom Packages [1]

Select which disks you want to lay out the file systems on.
Required disk space: 2,459 MB

Available Disks:
   Disk      Size
 c0d0      28615 MB

Enter ’y’ to layout file systems on the specified disk.  This will erase all
existing data on the Solaris fdisk partition.  Enter ’n’ to leave the disk
unmodified.  Enter ’e’ to leave the remaining disks unmodified and continue
with install.

   Layout file systems on disk c0d0 (bootdisk) (y/n) [y]?
```

3. **Follow the on-screen instructions to define partition 1 as the `x86Boot` partition with a size of 10MB, and partition 2 as the Solaris partition using the remaining free disk space.**

   When you have finished configuring the disk partitions you will see a screen summarising them. For example:

```
Customize fdisk Partitions-- Disk c0d0

You can customize the type of the partition and the size of the partition. A
disk can contain only one Solaris partition and one X86Boot partition. Only one
X86Boot disk is allowed per system.

 Partition  Type      Size (MB)
 1          x86Boot   10
 2          Solaris   28604
 3          Unused    0
 4          Unused    0
Capacity: 28615
Allocated: 28614
Free: 1
Rounding Error: 0


Enter b to go back, r to reset original information, d to load the default
layout, or n to go to the next screen.
   To customize a partition, enter partition number here [n]:
```

4. **Press [ENTER] to go to the next screen, and complete the installation the custom installation.**

   There is no more platform-specific configuration required for the blades after this point. Proceed to Section 10.8.8, “Completing the Solaris x86 Installation” on page 10-33.

10.8.8 Completing the Solaris x86 Installation

The procedures you have followed to create a blade-specific install image are complete. The remainder of this chapter contains information supplementary to these procedures.

For documentation describing the interactive or Webstart Solaris installations, refer to the *Solaris 9 Installation Guide*.

10.9 Preparatory Steps for Setting up a Jumpstart Installation for a Blade

The previous sections of this chapter have explained how to configure the DHCP server and network install image so that the B100x and B200x blades can be installed interactively. An interactive installation requires a lot of user input and it is time-consuming to use this process when installing multiple blades.

This section provides the extra steps you need to perform to enable the blades to be installed in a completely hands-free manner. This is known as a Jumpstart installation and is fully documented in the *Solaris 9 Installation Guide*.

Caution – In some circumstances a system administrator might choose to boot a blade from the network to recover from possible errors on its hard disk. If you have configured the blade to perform a Jumpstart installation, *any* subsequent network boot of the blade will by default result in a Jumpstart installation being performed. This will erase the contents of the hard disk. Therefore, to prevent the blade from executing a Jumpstart installation (after the first operating system installation), we recommend you remove the `SjumpsCF` and `SsysidCF` option names from the blade's client-specific macro after the initial Jumpstart installation has completed. (This network booting behavior is different from that of blades running SPARC Solaris.)

---

1. **Log into the Network Install Server as root and create a directory to hold the Jumpstart configuration files.**

```
# mkdir -p /export/jumpstart
# cd /export/jumpstart
```

The instructions in this section assume `/export/jumpstart` as the location of the Jumpstart configuration files.

2. **Copy the sample** `jumpstart` **directory from the install image to your** `jumpstart` **directory.**

```
# cp -r install_dir-path/Solaris_9/Misc/jumpstart_sample/* /export/jumpstart
```

where *install_dir-path* is the location of the install image.

3. **Share the Jumpstart directory.**

   To make the rules file and profiles accessible to systems on the network, you need to share the /export/jumpstart directory. To enable sharing of this directory, add the following line to the /etc/dfs/dfstab file:

```
share -F nfs -o ro,anon=0 /export/jumpstart
```

   Then, at the command line, type:

```
# shareall
```

4. **Modify the file called** rules **to suit your site's requirements.**

   a. **This file contains a lot of information. Comment out every line except the line starting with** arch i386**:**

```
# The following rule matches all x86 systems:
arch i386   x86-begin   x86-class -
```

   b. **Add the keyword** x86-finish **to the end of the line starting with** arch i386. This line will then look as follows:

```
# The following rule matches all x86 systems:
arch i386   x86-begin   x86-class   x86-finish
```

   The rules file dictates which systems will be installed by the Jumpstart configuration. For more information about its function, refer to the *Solaris 9 Installation Guide*.

5. **Edit the file called `x86-class` so that it describes the type of installation you want the Jumpstart to perform.**

```
# Sample profile for an x86 machine. Installation will
# provide default partitioning on a server system.
#
install_type    initial_install
fdisk all       solaris all
system_type     server
partitioning    default
cluster         SUNWCall
```

FIGURE 10-11 Sample `x86-class` File

The `fdisk` key word automates the deletion of any existing disk partition table on the hard disk that may have been created by a previous installation of Solaris x86 or Linux. For more information about defining the `x86-class` file and its associated key words, refer to the *Solaris 9 Installation Guid*e.

6. **Use a text editor to create an `x86-finish` script that will perform the required post-intallation steps.**

   This file is required to ensure that the blades will reboot correctly after the Jumpstart installation has been accomplished. The file must contain the information below:

```
#!/bin/sh

echo "Changing and syncing bootenv.rc"

# clear the boot-args property
echo "setprop boot-args ’’" >> /a/boot/solaris/bootenv.rc

# set the bootpath property to boot from the hard disk
STRING=‘df | grep ’^/a ’  | sed ’s/).*//’ | sed ’s/^.* (//’‘
STRING=‘ls -l ${STRING}‘
MYROOT=‘echo $STRING | sed ’s/.*..\/..\/devices//’‘
echo "setprop bootpath ${MYROOT}" >> /a/boot/solaris/bootenv.rc

# disable kdmconfig from running after the first reboot
sysidconfig -b /a -r /usr/openwin/bin/kdmconfig

sync

# Some x86 systems sometimes do not reboot after a jumpstart
reboot
```

FIGURE 10-12 Sample `x86-finish` Script

The `x86-finish` script file is used for post-installation operations such as the synchronising of `bootenv.rc`. It is also used to ensure that the `kdmconfig` utility does not run on the first reboot.

7. **Run the `check` command to verify the `rules` file and to create a `rules.ok` file.**

```
# ./check
Validating rules...
Validating profile x86-class...
The custom JumpStart configuration is ok.
```

8. **Use a text editor to create a `sysidcfg` file (or to modify the existing `sysidcfg` file) in the directory `/export/jumpstart.`**

   If you have already set up Jumpstart on your Network Install Server, the file will already exist. Otherwise you must create it.

This file contains responses to questions asked during the Jumpstart installation concerning, for example, time zone, terminal type, security, IPv6, time and date , system locale, and root password. The values for some of the keywords in this file will be specific to your local network configuration and its use of different services (for example, NIS).

Note – The root password that you need to specify in the `sysidcfg` is an encrypted one. You can find out a password's encrypted value (to insert into the `sysidcfg`) by setting up a user on a system and looking in that system's `/etc/shadow` file. The user password gets encrypted when a new user is added to a system by the System Administrator. In the sample `sysidcfg` file below (FIGURE 10-13), the password shown is `new.hope`. Choose a password that conforms to your local secure password policy.

```
system_locale=en_US
timezone=US/Pacific
terminal=dtterm
network_interface=primary {protocol_ipv6=no}
name_service=NONE
security_policy=NONE
timeserver=123.123.123.163
keyboard=Unknown
display=Unknown
pointer=Unknown
monitor=Unknown {
        DisplayChksum=0x0
}

root_password=45JhxF3R5G/4k
```

FIGURE 10-13 Sample `sysidcfg` File

Note – For information about creating or editing this file, refer to the *Solaris 9 Installation Guide*. Note that the four parameters printed in bold in FIGURE 10-13 are specific to Solaris x86.

---

9. **Proceed to Section 10.10, "Configuring a Jumpstart Installation" on page 10-39.**

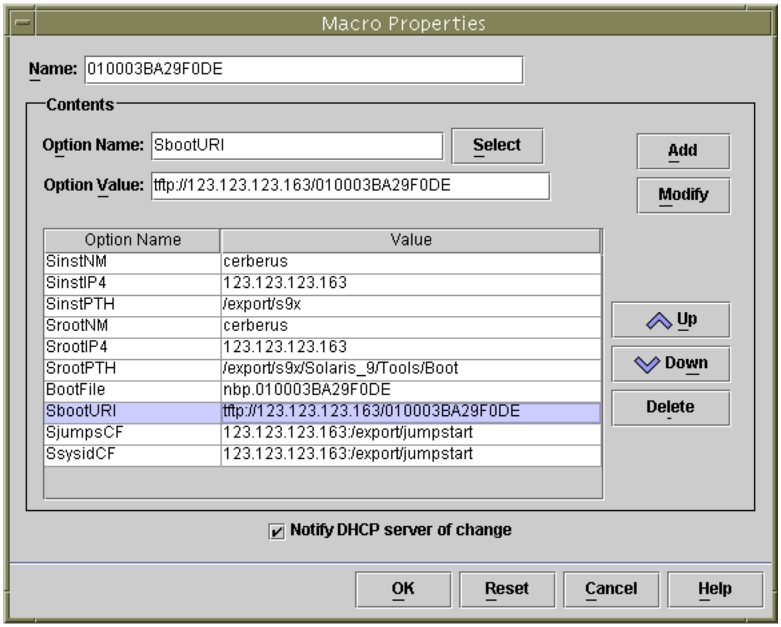
10.10 Configuring a Jumpstart Installation

If you are configuring a blade to perform a Jumpstart installation there are two extensions to the configuration steps in Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10. The extra tasks at Step 2 and at Step 6.)

- **In Step 2 (Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10), when you run the** `add_install_client` **utility you must include the Jumpstart configuration options on the command line. For a sample command, see FIGURE 10-15.**

  The sample command illustrated in FIGURE 10-15 uses the `-b` boot option. For information about the arguments taken by this option and required for the Jumpstart process to work on a blade, see Section 10.13, "The New `add_install_client -b` Option" on page 10-50 at the end of this chapter.

- **In Step 6 (in Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10), when you are configuring the client-specific DHCP macro for the blade, you must add values for the** `SjumpsCF` **and** `SsysidCF` **option strings.**



Name: 010003BA29F0DE

Option Name: SbootURI

Option Value: tftp://123.123.123.163/010003BA29F0DE

Option Name	Value
SinstNM	cerberus
SinstIP4	123.123.123.163
SinstPTH	/export/s9x
SrootNM	cerberus
SrootIP4	123.123.123.163
SrootPTH	/export/s9x/Solaris_9/Tools/Boot
BootFile	nbp.010003BA29F0DE
SbootURI	tftp://123.123.123.163/010003BA29F0DE
SjumpsCF	123.123.123.163:/export/jumpstart
SsysidCF	123.123.123.163:/export/jumpstart

☑ Notify DHCP server of change

FIGURE 10-14 Sample Macro Properties Window (in DHCP Manager) to Support Jumpstart

```
# ./add_install_client -d -e "00:03:ba:29:f0:de" \
> -b "input-device=ttya" -b "output-device=ttya" \
> -b "bootpath=/pci@0,0/pci108e,16a8@8" \
> -b "boot-args=’ - install dhcp’” \
> -c 123.123.123.163:/export/jumpstart \
> -p 123.123.123.163:/export/jumpstart \
> i86pc
cleaning up preexisting install client "00:03:ba:29:f0:de"
To disable 00:03:ba:29:f0:de in the DHCP server,
  remove the entry with Client ID 010003BA29F0DE

To enable 010003BA29F0DE in the DHCP server, ensure that
the following Sun vendor-specific options are defined
(SinstNM, SinstIP4, SinstPTH, SrootNM, SrootIP4,
SrootPTH, SbootURI and optionally SjumpCF and SsysidCF),
and add a macro to the server named 010003BA29F0DE,
containing the following option values:

  Install server      (SinstNM)  : cerberus
  Install server IP   (SinstIP4) : 123.123.123.163
  Install server path (SinstPTH) : /export/s9x
  Root server name    (SrootNM)  : cerberus
  Root server IP      (SrootIP4) : 123.123.123.163
  Root server path    (SrootPTH) : /export/s9x/Solaris_9/Tools/Boot
  Boot file           (BootFile) : nbp.010003BA29F0DE
  Solaris boot file   (SbootURI) : tftp://123.123.123.163/010003BA29F0DE
  Profile location    (SjumpsCF) : 123.123.123.163:/export/jumpstart
  sysidcfg location   (SsysidCF) : 123.123.123.163:/export/jumpstart

If not already configured, enable PXE boot by creating
a macro called PXEClient:Arch:00000:UNDI:002001
which contains the following values:
  Boot server IP      (BootSrvA) : 123.123.123.163
This macro will be explicitly requested by the PXE boot.
```

FIGURE 10-15 Sample `add_install_client` Command and Output for Jumpstart on a B100x Blade

10.11 Useful Tips for Installing Solaris x86 onto Multiple Blades

When setting up multiple blades to install from the same network image you can save time by using the tips in this section.

10.11.1 Calling the `add_install_client` Utility From a Wrapper Shell Script

Most of the arguments taken by the `add_install_client` utility will be the same for each blade; only a blade's MAC address will change. Therefore, you can invoke the utililty from a shell script (see FIGURE 10-12, FIGURE 10-16 and FIGURE 10-17). The example in FIGURE 10-12 assumes the script is stored in

/export/s9x/Solaris_9/Tools and named add-blade-B100x. The example in FIGURE 10-17 assumes the same location for the script, and also assumes it is named add-blade-B200x.

```
#!/bin/sh
[ $# -ne 1 ] && echo "Usage: add-blade-B100x blade-mac-address" && exit 1
MAC="$1"
P1="input-device=ttya"
P2="output-device=ttya"
BP="bootpath=/pci@0,0/pci108e,16a8@8"
BA="boot-args=’ - install dhcp’"
COPT="-c 123.123.123.163:/export/jumpstart"
POPT="-p 123.123.123.163:/export/jumpstart"

set -x
./add_install_client -d -e "$MAC" -b "$P1" -b "$P2" -b "$BP" -b "$BA" \
$COPT $POPT i86pc
```

FIGURE 10-16 Sample Wrapper Script for Installing a B100x Blade

```
#!/bin/sh
[ $# -ne 1 ] && echo "Usage: add-blade-B200x blade-mac-address" && exit 1
MAC="$1"
P1="input-device=ttya"
P2="output-device=ttya"
BP="bootpath=/pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci108e,16a8@3"
BA="boot-args=’ - install dhcp’"
COPT="-c 123.123.205.163:/export/jumpstart"
POPT="-p 123.123.205.163:/export/jumpstart"

set -x
./add_install_client -d -e "$MAC" -b "$P1" -b "$P2" -b "$BP" -b "$BA" \
$COPT $POPT i86pc
```

FIGURE 10-17 Sample Wrapper Script for Installing a B200x Blade

Note – Remember that the boothpath is different for B100x and B200x blades and for different interfaces. Make sure the scripts you use apply to groups of blades of the same type and also groups of blades that use the same network interface. For information about using an interface other than the default one, see Section 10.12, “Installing Solaris x86 Onto a Blade by Using the Second, Third, or Fourth Network Interface” on page 10-47.

When you use wrapper scripts, the command to set up the blade using the `add_install_client` utility becomes:

- For a B100x blade:

```
# cd /export/s9x/Solaris_9/Tools
# ./add-blade-b100x "blade-MAC-address"
```

- For a B200x blade:

```
# cd /export/s9x/Solaris_9/Tools
# ./add-blade-b200x "blade MAC address"
```

A sample command for a B200x blade is:

```
# cd /export/s9x/Solaris_9/Tools
# ./add-blade-b200x "00:03:ba:2d:d4:a0"
```

10.11.2 Speeding Up the Creation of Macros for Installing Multiple Blades

This section tells you how to use the DHCP Manager's Include and Duplicate facilities to speed up the creation of macros when you are installing multiple x86 blades in a chassis.

10.11.2.1 Using the DHCP Manager's Macro Include Facility

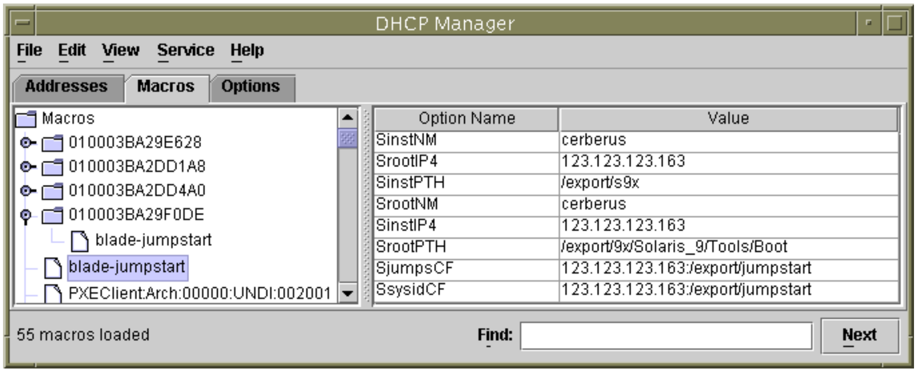
From FIGURE 10-1 and FIGURE 10-15 you can see that a number of the option strings you need to include in a blade's client-specific DHCP macro will be common to all blades that you install from the same network install image. For example in FIGURE 10-15 the following macros are the same for each client, regardless of the client blade's Ethernet address:

Install server (`SinstNM`): cerberus
Install server IP (`SinstIP4`): 123.123.123.163
Install server path (`SinstPTH`): /export/s9x
Root server name (`SrootNM`): cerberus
Root server IP (`SrootIP4`): 123.123.123.163

Root server path (`SrootPTH`): /export/s9x/Solaris_9/Tools/Boot
Profile location (`SjumpsCF`): 123.123.123.163:/export/jumpstart
sysidcfg location (`SsysidCF`): 123.123.123.163:/export/jumpstart

Conveniently the DHCP Manager GUI allows you to set up a named macro and then reference it from more than one client-specific macro by using an option string called '`Include`'.

FIGURE 10-18 illustrates this by showing a macro called '`blade-jumpstart`' that has been created to include by reference all the options associated with a Jumpstart installation. FIGURE 10-19 shows a client-specific macro that *includes* the '`blade-jumpstart`' macro.

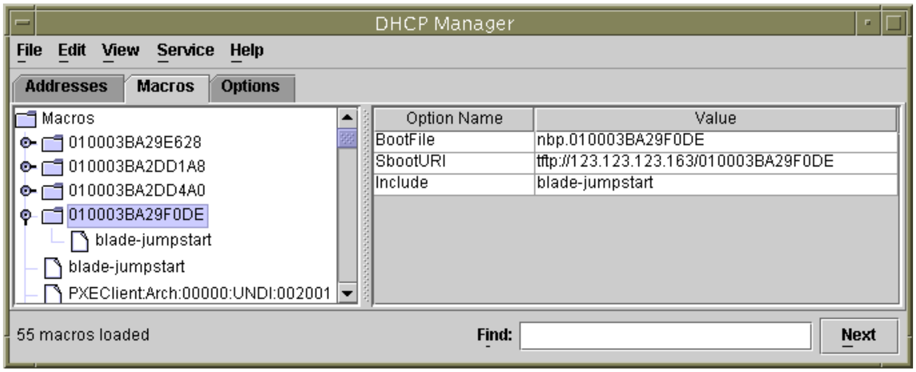


- Macros
  - 010003BA29E628
  - 010003BA2DD1A8
  - 010003BA2DD4A0
  - 010003BA29F0DE
    - blade-jumpstart
  - blade-jumpstart
  - PXEClient:Arch:00000:UNDI:002001

Option Name	Value
SinstNM	cerberus
SrootIP4	123.123.123.163
SinstPTH	/export/s9x
SrootNM	cerberus
SinstIP4	123.123.123.163
SrootPTH	/export/9x/Solaris_9/Tools/Boot
SjumpsCF	123.123.123.163:/export/jumpstart
SsysidCF	123.123.123.163:/export/jumpstart

55 macros loaded

FIGURE 10-18 Creating a Sample 'Include' Macro Called '`blade-jumpstart`'



- Macros
  - 010003BA29E628
  - 010003BA2DD1A8
  - 010003BA2DD4A0
  - 010003BA29F0DE
    - blade-jumpstart
  - blade-jumpstart
  - PXEClient:Arch:00000:UNDI:002001

Option Name	Value
BootFile	nbp.010003BA29F0DE
SbootURI	tftp://123.123.123.163/010003BA29F0DE
Include	blade-jumpstart

55 macros loaded

FIGURE 10-19 Sample Client-specific Macro That Uses the "Include" Facility

10.11.2.2 Using the DHCP Manager’s Macro Duplicate Facility

When you have set up a client-specific macro correctly for one blade, you can use the Duplicate option from the DHCP Manager’s Edit menu, to create a new macro quickly for another blade. Only the Macro name and the contents of the `SbootURI` and `BootFile` options need to be changed for each blade.

10.11.3 Using the DHCP Manager’s Command-line Interface Instead of the GUI

This section describes how to use the DHCP command line tools to configure the required DHCP Manager macros instead of using the GUI.

- **Create the global PXE macro by using the following DCHP table management command:**

```
# dhtadm -A -m PXEClient:Arch:00000:UNDI:002001 -d ’:BootSrvA=ip-address:’
```

where *ip-address* is the IP address of the Network Install Server. (This command is the equivalent of performing the steps described in Section 10.3.2, “Adding the Global PXE Macro for Solaris x86 to the DHCP Server” on page 10-8.)

- **Create the client-specific macro by using the DHCP table management commands appropriate to your blade. The commands below assume a blade with the properties described in FIGURE 10-7:**

```
# dhtadm -A -m 010003BA29F0DE -d’:SinstNM=cerberus:’
# dhtadm -M -m 010003BA29F0DE -e’SinstIP4=123.123.123.163’
# dhtadm -M -m 010003BA29F0DE -e’SinstPTH=/export/s9x’
# dhtadm -M -m 010003BA29F0DE -e’SrootNM=cerberus’
# dhtadm -M -m 010003BA29F0DE -e’SrootIP4=123.123.123.163’
# dhtadm -M -m 010003BA29F0DE -e’SrootPTH=/export/s9x/Solaris_9/Tools/Boot’
# dhtadm -M -m 010003BA29F0DE -e’BootFile=nbp.010003BA29F0DE’
# dhtadm -M -m 010003BA29F0DE -e’SbootURI=tftp://123.123.123.163/010003BA29F0DE’
```

These commands are the equivalent of performing Step 6 in Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10.

If you are performing a Jumpstart installation, you need to add the following two commands:

```
# dhtadm -M -m 010003BA29F0DE -e ’SjumpsCF=123.123.123.163:/export/jumpstart’
# dhtadm -M -m 010003BA29F0DE -e ’SsysidCF=123.123.123.163:/export/jumpstart’
```

- **Assign an IP address to the blade:**

```
# dhtadm -A ip-address -h blade-hostname -i010003BA29F0DE -m010003BA29F0DE network-address
```

where *ip-address* is the IP address of the blade, *blade-hostname* is the hostname of the blade, and *network-address* is the base address for the blade’s subnet. This command is the equivalent of performing Step 7 in Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10.

10.12 Installing Solaris x86 Onto a Blade by Using the Second, Third, or Fourth Network Interface

This section is for users who want to boot a blade by using a network interface other than the first interface. It provides information that you will need when you follow the instructions in Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10

The B100x blades have two interfaces. The B200x blades have four. You need to give the DHCP and Network Install Servers different information about the MAC address and bootpath if you are not using the first network interface on the blade. Also you need to use a different argument to the System Controller’s `bootmode` command, when you configure the blade temporarily to boot from the network.

10.12.1 Different Properties You Must Specify for the B100x Interfaces

The B100x has one dual-port BCM5704s Gigabit Ethernet device. Each port on this device is connected to one of the Ethernet switches in the B1600 chassis. The BIOS takes responsibility for assigning the MAC addresses to the Ethernet ports as shown in FIGURE 10-20.

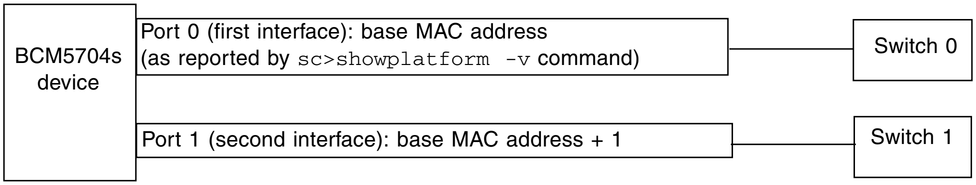

BCM5704s device
Port 0 (first interface): base MAC address (as reported by sc>showplatform -v command)
Port 1 (second interface): base MAC address + 1
Switch 0
Switch 1

FIGURE 10-20 The Network Interfaces on a B100x Blade

TABLE 10-1 Properties for the Two Interfaces on a B100x Server Blade

Variable	First Network Interface	Second Network Interface
MAC address	MAC address + 0	MAC address + 1
bootpath	`bootpath=/pci@0,0/pci108e,16a8@8`	`bootpath=/pci@0,0/pci108e,16a8@8,1`
bootmode command	`bootmode bootscript="boot net" s`*n*[*] or: `bootmode bootscript="boot snet0" s`*n*	`bootmode bootscript="boot snet1" s`*n*

[*] where *n* is the blade's slot number in the chassis

10.12.2 Different Properties You Must Specify for the B200x Interfaces

The B200x has two dual-port BCM5704s Gigabit Ethernet devices. Each port is connected to one of the Ethernet switches in the B1600 chassis. The BIOS takes responsibility for assigning the MAC addresses to the Ethernet ports as shown in FIGURE 10-21.

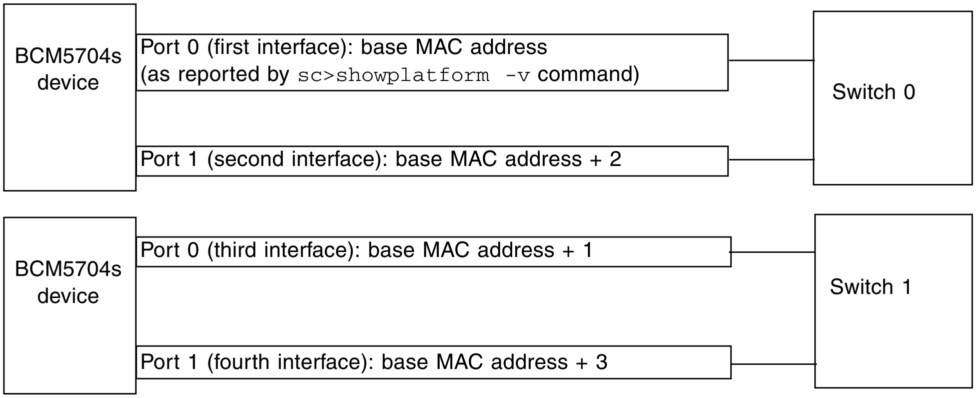

BCM5704s device
Port 0 (first interface): base MAC address (as reported by sc>showplatform -v command)
Port 1 (second interface): base MAC address + 2
Switch 0
BCM5704s device
Port 0 (third interface): base MAC address + 1
Port 1 (fourth interface): base MAC address + 3
Switch 1

Diagram showing a B200x blade's network interfaces and their connection to the switches in the chassis

FIGURE 10-21 The Network Interfaces on a B200x Blade

TABLE 10-2 Properties for the First Interface on a B200x Server Blade

Variable	First Network Interface
MAC address	MAC address + 0
bootpath	`bootpath=/pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci108e,16a8@3`

bootmode command	`bootmode bootscript="boot net" sn`*
	 or:
	 `bootmode bootscript="boot snet0" sn`

* where *n* is the blade's slot number in the chassis

TABLE 10-3 Properties for the Second Interface on a B200x Server Blade

Variable	Second Network Interface
MAC address	MAC address + 1
bootpath	`bootpath=/pci@0,0/pci8086,2545@3/pci8086,1460@1f/pci108e,16a8@3`
bootmode command	`bootmode bootscript="boot snet1" sn`*

* where *n* is the blade's slot number in the chassis

TABLE 10-4 Properties for the Third Interface on a B200x Server Blade

Variable	Third Network Interface
MAC address	MAC address + 2
bootpath	`bootpath=/pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci108e,16a8@3,1`
bootmode command	`bootmode bootscript="boot snet2"` *sn*[*]

[*] where *n* is the blade's slot number in the chassis

TABLE 10-5 Properties for the Fourth Interface on a B200x Server Blade

Variable	Fourth Network Interface (3)
MAC address	MAC address + 3
bootpath	`bootpath=/pci@0,0/pci8086,2545@3/pci8086,1460@1f/pci108e,16a8@3,1`
bootmode command	`bootmode bootscript="boot snet3"` *sn*[*]

[*] where *n* is the blade's slot number in the chassis

10.13 The New `add_install_client -b` Option

The `add_install_client` command in FIGURE 10-7 (see Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10) uses a new `-b` option to set certain boot property values that need to be specified during the network PXE boot process for blade platforms.

These values are `input-device`, `output-device`, `bootpath`, and `boot-args`. This section describes their purpose:

- `-b "input-device=ttya"`
  `-b "output-device=ttya"`

  Because the blades do not have a VGA screen or keyboard, the input-device and output-device must both be set to the serial console '`ttya`'. This ensures that the system console is re-directed to the blade's serial port, enabling you to interact with the blade through the console.

- `-b "bootpath=/pci@0,0/pci108e,16a8@8"`

  This property specifies the boot device for the blade. It removes the need for the Device Configuration Assistant to pause the system during booting to request you to select a boot device. Note that the `bootpath` value is platform-specific. For the correct values, see TABLE 10-1, TABLE 10-2, TABLE 10-3, TABLE 10-4, and TABLE 10-5.

- `-b "boot-args=' - install dhcp'"`

  This property holds a string of arguments that will be passed to the boot subsystem. In FIGURE 10-10 we use the property to esnure that a Jumpstart installation is performed when the blade PXE boots from the network. For more information, refer to `boot(1M)`, `kadb(1M)`, and `kernel(1M)`.

CHAPTER 11

Configuring IPMP for Network Resiliency on Solaris x86 Blades

This chapter contains the following sections:

11.1 Taking Advantage of Having Two Switches in the System Chassis

This chapter modifies and supplements the information available in Chapter 5 of the *Sun Fire B1600 Blade System Chassis Software Setup Guide*. Please read that chapter before following the instructions in this one.

The instructions in this chapter enable you to deploy a chassis containing Solaris x86 blades in a configuration that:

- Takes advantage of the redundant switch (you need to have dual SSCs installed in the chassis) to give the Solaris x86 blades two connections (B100x blades) or four connections (B200x blades) each to the network.
- Observes the separation of your data and management networks.

The next section (Section 11.2, "How IPMP Works on B100x and B200x Blades" on page 11-3) tells you how IPMP works on an x86 blade in the B1600 chassis. It states (and explains) the number of IP addresses each blade (B100x or B200x) needs for the type of configuration you require.

Note – The IPMP instructions provided in this chapter assume that you have two SSCs installed, that each is connected on all its ports to an external switch on the data network (the connections on each port of one SSC being duplicated on each port of the other, but connected to a different external switch on the data network), and that the NETMGT port on each SSC is connected to the management subnet. For information about configuring the switches and System Controllers in the chassis, refer to the *Sun Fire B1600 Blade System Chassis Software Setup Guide*.

Before you can set up IPMP on a blade, you need to reconfigure the blade to make it stop using DHCP. The DHCP configuration was required to enable you to install the operating system; instructions for migrating the blade to a static IP configuration (in preparation for IPMP) are provided in Section 11.3, "Migrating From DHCP to Static IP Addresses" on page 11-4.

Finally, instructions for configuring IPMP on a blade are provided in:

- Section 11.4, "Configuring IPMP on a B100x Blade" on page 11-7
- Section 11.5, "Configuring IPMP on a B200x Blade" on page 11-10

11.2 How IPMP Works on B100x and B200x Blades

The instructions in this chapter tell you how to use the Solaris IP Network Multipathing (IPMP) facility to take advantage of the redundant connections from each server blade to the switches in the chassis. A B100x blade's two 1000Mbps Ethernet interfaces are labeled respectively `bge0` and `bge1` (`bge0` is connected to the switch in SSC0, and `bge1` is connected to the switch in SSC1). A B200x blade's four 1000Mbps Ethernet interfaces are labeled respectively `bge0`, `bge1`, `bge2`, `bge3` (`bge0` and `bge1` are connected to the switch in SSC0, and `bge2` and `bge3` are connected to the switch in SSC1). When the Sun Fire B1600 blade system chassis is fully operational, both switches are constantly active.

The IPMP driver on a server blade works by periodically pinging the default gateway from each Ethernet interface using a test IP address. The test addresses are used privately by the IPMP driver for the ping process. If for any reason one of the pings fails (indicating that the path to the network is no longer available on the interface that was used to perform the ping) the IPMP driver ensures that network traffic uses only the interface or interfaces that remains valid. Both interfaces on a B100x blade, or all interfaces on a B200x blade, can be active. This is referred to as an active/active configuration.

Alternatively the interfaces can be configured in an active/standby configuration in which one interface on the blade is active and the other one (on a B100x blade) is a standby interface, or (on a B200x blade) the other three are standby interfaces. In this type of configuration, if the failed interface is the active one, the driver assigns the IP address to the standby interface (or one of the standby interfaces), and that interface becomes the active one.

Because both switches inside the chassis are active (when the chassis is working normally), the instructions in this chapter tell you how to perform an active/active configuration. This maximises the performance of the chassis by ensuring that no interfaces are idle. For information about performing an active/standby configuration, refer to the *IP Network Multipathing Administration Guide* (816-0850).

The IP addresses you require for each blade to support the active/active configuration are:

- Two active IP addresses (B100x blade).
  Four active IP addresses (B200x blade).

  The active IP addresses can be registered on a Name Server. They are the addresses by which other devices on the network communicate with the blade.

- Two test IP addresses (B100x blade).
  Four test IP addresses (B200x blade).

  Test addresses are required (one per interface) for the ping process. These addresses are private to the IPMP driver (they are not registered on the Name Server).

In the next chapter, instructions are provided for setting up multiple pairs of virtual IPMP interfaces, each pair providing redundant virtual connections to separate VLANs.

11.3 Migrating From DHCP to Static IP Addresses

To install Solaris x86 onto a blade, you need to use DHCP as described in Chapter 10 (the PXE installation process depends upon it). However, if you want to use IPMP, you must stop using DHCP because it is not possible to configure a DHCP Server to support IPMP data and test addresses and grouping.

This section tells you how to make the blades use static IP addresses instead of addresses assigned by DHCP.

1. **Make sure the addresses you intend to use for the blade or blades are not already in use by another device.**

   The addresses you use must be ones that cannot be assigned to another device by a DHCP server on the same subnet as the blade you are configuring. Either reserve the addresses in your DHCP configuration or use addresses for the blade or blades that are outside the range of addresses managed by the DHCP server.

   For a:

   - B100x blade, you will need two IP addresses, or four if you intend to use IPMP.
   - B200x blade you will need four IP addresses., or eight if you intend to use IPMP.

   For information about reserving addresses on the DHCP server, refer to the *Solaris DHCP Administration Guide*.

2. **On each blade for which you are configuring one or more static addresses, remove or rename the files `/etc/dhcp.`*interface*, where *interface* is `bge0` and `bge1` (plus `bge2` and `bge3` for a B200x blade).**

3. **Edit the `/etc/hosts` file on the blade to define the IP addresses for the interfaces on the blade.**

   For purposes of illustration, the instructions in this chapter assume a base hostname of "`medusa`" for the chassis being configured. Various suffixes are then added to this base hostname to indicate an individual component or a network interface on a particular blade.

   For example, for a B100x blade you will need entries in the `/etc/hosts` file that are similar to those in CODE EXAMPLE 11-1:

   CODE EXAMPLE 11-1 Sample `/etc/hosts` File Entries for a B100x Blade

```
127.0.0.1       local host
192.168.1.151   medusa-s1   loghost    # first interface
192.168.1.152   medusa-s1-1            # second interface
```

   For a B200x blade you will need entries in the `/etc/hosts` file that are similar to those in CODE EXAMPLE 11-2:

   CODE EXAMPLE 11-2 Sample `/etc/hosts` File Entries for a B200x Blade

```
127.0.0.1       local host
192.168.1.151   medusa-s1   loghost    # first interface
192.168.1.152   medusa-s1-1            # second interface
192.168.1.167   medusa-s1-2            # third interface
192.168.1.168   medusa-s1-3            # fourth interface
```

4. **On the blade, create an `/etc/nodename` file that contains the blade's hostname.**

   This will normally be the name used by the first network interface as specified in the `/etc/hosts` file (see Step 3). For example, if the blade's hostname is `medusa-s1`, the `/etc/nodename` file needs to contain the following information:

```
medusa-s1
```

5. **On the blade, create a `hostname.`*interface* file for each interface, where *interface* is `bge0` and `bge1` (plus `bge2` and `bge3` for a B200x blade).**

   CODE EXAMPLE 11-3 Sample File for `hostname.bge0`

```
medusa-s1
```

   CODE EXAMPLE 11-4 Sample File for `hostname.bge1`

```
medusa-s1-1
```

   For a B200x blade you will need `hostname.bge2` and `hostname.bge3` files as well.

CODE EXAMPLE 11-5 Sample File for `hostname.bge2`

```
medusa-s1-2
```

CODE EXAMPLE 11-6 Sample File for `hostname.bge3`

```
medusa-s1-3
```

6. **Disable routing, because the server blade is not being used to perform routing:**

```
# touch /etc/notrouter
# ndd -set /dev/ip ip_forwarding 0
```

7. **If your network routers do not advertise their presence to network devices, create an** `/etc/defaultrouter` **by typing the following command:**

```
# echo ip-address > /etc/defaultrouter
```

8. where *ip-address* is the IP address of the router on the same subnet as the blade. For example, if the IP address of the default router were 123.123.123.8, you would type:

```
# echo 123.123.123.8 > /etc/defaultrouter
```

9. **Reboot the blade to make it boot with its new static IP configuration:**

```
# reboot
```

11.4 Configuring IPMP on a B100x Blade

This section tells you how to configure IPMP on a B100x server blade with two interfaces so that *both* interfaces actively transmit and receive data.

Note – Before following the instructions in this section, make sure you have performed the steps required in Section 11.3, "Migrating From DHCP to Static IP Addresses" on page 11-4.

Note – You need to perform the instructions in this section on each B100x server blade that requires a redundant connection to the network.

---

1. **Log in as root to the console of the server blade whose interfaces you want to configure.**

   Type the following at the System Controller's `sc>` prompt:

   ```
   sc> console sn
   ```

   where *n* is the number of the slot containing the server blade you want to log into.

2. **Edit the `/etc/hosts` file on the server blade to add the blade's two test IP addresses.**

   For example:

   ```
   #
   # /etc/hosts on the server blade in system chassis Medusa, slot 0
   #
   127.0.0.1      localhost

   192.168.1.151  medusa-s1   loghost # First active data address
   192.168.1.152  medusa-s1-1         # Second active data address
   192.168.1.101  medusa-s1-test0     # Test address for bge0
   192.168.1.102  medusa-s1-test1     # Test address for bge1
   ```

3. **Set the netmask in the server blade's** `/etc/netmasks` **file for the IP addresses of the interfaces on the blade.**

   For example:

```
192.168.1.0     255.255.255.0
```

4. **If you have not already done so, disable routing because the server blade is not being used to perform routing:**

```
# touch /etc/notrouter
# ndd -set /dev/ip ip_forwarding 0
```

5. **In the** `/etc` **directory, create a** `hostname.bge0` **and a** `hostname.bge1` **file.**

CODE EXAMPLE 11-7 Sample `hostname.bge0` File

```
medusa-s1 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test0 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-8 Sample `hostname.bge1` File

```
medusa-s1-1 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test1 deprecated -failover netmask + broadcast + up
```

6. **Reboot the blade so that it boots with its new IPMP configuration:**

```
# reboot
```

7. **Inspect the configuration of the four network adapters:**

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.1.151 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:f0:de
bge0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 2
        inet 192.168.1.101 netmask ffffff00 broadcast 192.168.1.255
bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.1.152 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:f0:df
bge1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 3
        inet 192.168.1.102 netmask ffffff00 broadcast 192.168.1.255
```

The output above shows that four addresses have been defined. The two IPMP test addresses (associated with `bge0:1` and `bge1:1` respectively) are marked NOFAILOVER. This means that they will not be transferred to the surviving interface in the event of a failure.

8. **Test that the IPMP configuration works by temporarily removing one SSC from the chassis.**

   This will cause error messages similar to the following to appear on the console:

```
Nov 19 13:20:47 medusa-s1 bge: NOTICE: bge1: link down
Nov 19 13:20:47 medusa-s1 in.mpathd[107]: The link has gone down on bge1
Nov 19 13:20:47 medusa-s1 in.mpathd[107]: NIC failure detected on bge1 of group medusa_grp0
Nov 19 13:20:47 medusa-s1 in.mpathd[107]: Successfully failed over from NIC bge1 to NIC bge0
```

Note – It takes approximately 10 seconds for the IPMP daemon to detect and recover from a network failure with the default configuration. The configuration of the IPMP daemon is defined in the `/etc/default/mpathd` file.

11.5 Configuring IPMP on a B200x Blade

This section tells you how to configure IPMP on a B200x server blade with four interfaces so that all interfaces *actively* transmit and receive data. The section provides two different methods of achieving network resiliency using an active/active configuration.

- One method uses a single group of IPMP interfaces (FIGURE 11-1). In this method a failure on one interface will result in any of the other interfaces on the blade being used.

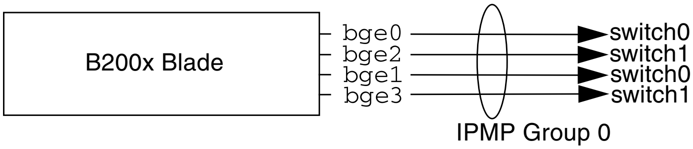


FIGURE 11-1 Diagram Showing a Single IPMP Group Containing All Four Blade Interfaces

- The other method uses two groups of IPMP interfaces, each containing one interface to one switch in the chassis and one interface to the other (see FIGURE 11-2). The advantage of this method is that it enables you to reserve a particular pair of interfaces for a particular service. In this configuration each seperate IPMP group can be used to provide a network resilient connection for a different set of services running on the server blade.

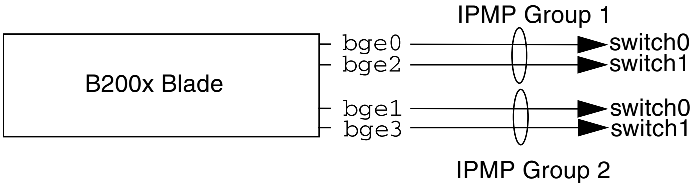


FIGURE 11-2 Diagram Showing Two IPMP Groups, Each Containing Two Interfaces

---

Note – Note that the achievement of network resilience (enabling a blade to recover from different hardware and network failures) depends upon each IPMP group containing one connection to each switch. A configuration in which both interfaces in a group of two were connected to the same switch would not continue to transport network traffic if that switch failed. In Section 11.2, "How IPMP Works on B100x and B200x Blades" on page 11-3, we saw that bge0 and `bge1` are connected to switch 0, and `bge2` and `bge3` to switch 1. This is also shown in FIGURE 11-2.

Note – Before following the instructions in this section, make sure you have performed the steps required in Section 11.3, "Migrating From DHCP to Static IP Addresses" on page 11-4.

Note – You need to perform the instructions in this section on each B200x server blade that requires a redundant connection to the network.

11.5.1 Configuring IPMP on a B200x Blade Using a Single IPMP Group for All Interfaces

1. **Log in as root to the console of the server blade whose interfaces you want to configure.**

   Type the following at the System Controller's `sc>` prompt:

   ```
   sc> console sn
   ```

   where *n* is the number of the first slot (of the two) containing the double-width blade you want to log into.

2. **Edit the `/etc/hosts` file on the server blade to add the blade's two test IP addresses.**

   For example:

   ```
   #
   # /etc/hosts on the server blade in system chassis Medusa, slot 0
   #
   127.0.0.1       localhost
   192.168.1.151   medusa-s1  loghost   # first data address
   192.168.1.152   medusa-s1-1          # second data address
   192.168.1.153   medusa-s1-2          # third data address
   192.168.1.154   medusa-s1-3          # fourth data address

   192.168.1.101   medusa-s1-test0      # test address for bge0
   192.168.1.102   medusa-s1-test1      # test address for bge1
   192.168.1.103   medusa-s1-test2      # test address for bge2
   192.168.1.104   medusa-s1-test3      # test address for bge3
   ```

3. **Set the netmask in the server blade's `/etc/netmasks` file for the IP addresses of the interfaces on the blade.**

   For example:

```
192.168.1.0     255.255.255.0
```

4. **If you have not already done so, disable routing because the server blade is not being used to perform routing:**

```
# touch /etc/notrouter
# ndd -set /dev/ip ip_forwarding 0
```

5. **In the `/etc` directory, create a `hostname.bge0` and a `hostname.bge1` file.**

CODE EXAMPLE 11-9 Sample `hostname.bge0` File

```
medusa-s1 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test0 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-10 Sample `hostname.bge1` File

```
medusa-s1-1 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test1 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-11 Sample `hostname.bge2` File

```
medusa-s1-2 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test2 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-12 Sample `hostname.bge3` File

```
medusa-s1-3 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test3 deprecated -failover netmask + broadcast + up
```

6. **Reboot the blade so that it boots with its new IPMP configuration:**

```
# reboot
```

7. **Inspect the configuration of the four network adapters:**

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.1.151 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:2d:d4:a0
bge0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 2
        inet 192.168.1.101 netmask ffffff00 broadcast 192.168.1.255
bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.1.152 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:2d:d4:a2
bge1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 3
        inet 192.168.1.102 netmask ffffff00 broadcast 192.168.1.255
bge2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
        inet 192.168.1.153 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:2d:d4:a1
bge2:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 4
        inet 192.168.1.103 netmask ffffff00 broadcast 192.168.1.255
bge3: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 5
        inet 192.168.1.154 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:2d:d4:a3
bge3:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 5
        inet 192.168.1.104 netmask ffffff00 broadcast 192.168.1.255
#
```

The output above shows that eight addresses have been defined. The four IPMP test addresses (associated with `bge0:1`, `bge1:1`, `bge2:1`, and `bge3:1`, respectively) are marked NOFAILOVER. This means that they will not be transferred to the surviving interface in the event of a failure.

8. **Test that the IPMP configuration works by temporarily removing one SSC from the chassis.**

   This will cause error messages similar to the following to appear on the console:

```
Nov 19 12:39:37 medusa-s1 bge: NOTICE: bge3: link down
Nov 19 12:39:37 medusa-s1 in.mpathd[108]: The link has gone down on bge3
Nov 19 12:39:37 medusa-s1 in.mpathd[108]: NIC failure detected on bge3 of group medusa_grp0
Nov 19 12:39:37 medusa-s1 bge: NOTICE: bge2: link down
Nov 19 12:39:37 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge3 to NIC bge2
Nov 19 12:39:37 medusa-s1 in.mpathd[108]: The link has gone down on bge2
Nov 19 12:39:37 medusa-s1 in.mpathd[108]: NIC failure detected on bge2 of group medusa_grp0
Nov 19 12:39:37 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge2 to NIC bge1
```

Note – It takes approximately 10 seconds for the IPMP daemon to detect and recover from a network failure with the default configuration. The configuration of the IPMP daemon is defined in the `/etc/default/mpathd` file.

11.5.2 Configuring IPMP on a B200x Blade Using Two IPMP Groups

1. **Log in as root to the console of the server blade whose interfaces you want to configure.**

   Type the following at the System Controller's `sc>` prompt:

   ```
   sc> console sn
   ```

   where *n* is the number of the first slot (of the two) containing the double-width blade you want to log into.

2. **Edit the `/etc/hosts` file on the server blade to add the blade's two test IP addresses.**

   For example:

   ```
   #
   # /etc/hosts on the server blade in system chassis Medusa, slot 0
   #
   127.0.0.1       localhost
   192.168.1.151   medusa-s1  loghost  # first data address
   192.168.1.152   medusa-s1-1         # second data address
   192.168.1.153   medusa-s1-2         # third data address
   192.168.1.154   medusa-s1-3         # fourth data address

   192.168.1.101   medusa-s1-test0     # test address for bge0
   192.168.1.102   medusa-s1-test1     # test address for bge1
   192.168.1.103   medusa-s1-test2     # test address for bge2
   192.168.1.104   medusa-s1-test3     # test address for bge3
   ```

3. **Set the netmask in the server blade's `/etc/netmasks` file for the IP addresses of the interfaces on the blade.**

   For example:

   ```
   192.168.1.0     255.255.255.0
   ```

4. **If you have not already done so, disable routing because the server blade is not being used to perform routing:**

```
# touch /etc/notrouter
# ndd -set /dev/ip ip_forwarding 0
```

5. **In the** `/etc` **directory, create a** `hostname.bge0` **and a** `hostname.bge1` **file.**

CODE EXAMPLE 11-13 Sample `hostname.bge0` File

```
medusa-s1 netmask + broadcast + group medusa_grp1 up \
addif medusa-s1-test0 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-14 Sample `hostname.bge1` File

```
medusa-s1-1 netmask + broadcast + group medusa_grp2 up \
addif medusa-s1-test1 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-15 Sample `hostname.bge2` File

```
medusa-s1-2 netmask + broadcast + group medusa_grp1 up \
addif medusa-s1-test2 deprecated -failover netmask + broadcast + up
```

CODE EXAMPLE 11-16 Sample `hostname.bge3` File

```
medusa-s1-3 netmask + broadcast + group medusa_grp2 up \
addif medusa-s1-test3 deprecated -failover netmask + broadcast + up
```

6. **Reboot the blade so that it boots with its new IPMP configuration:**

```
# reboot
```

7. **Inspect the configuration of the four network adapters:**

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.1.151 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp1
        ether 0:3:ba:2d:d4:a0
bge0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 2
        inet 192.168.1.101 netmask ffffff00 broadcast 192.168.1.255
bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.1.152 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp2
        ether 0:3:ba:2d:d4:a2
bge1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 3
        inet 192.168.1.102 netmask ffffff00 broadcast 192.168.1.255
bge2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
        inet 192.168.1.153 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp1
        ether 0:3:ba:2d:d4:a1
bge2:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 4
        inet 192.168.1.103 netmask ffffff00 broadcast 192.168.1.255
bge3: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 5
        inet 192.168.1.154 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp2
        ether 0:3:ba:2d:d4:a3
bge3:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500 index 5
        inet 192.168.1.104 netmask ffffff00 broadcast 192.168.1.255
#
```

The sample output above shows that eight addresses have been defined. Notice that bge0 and bge2 are reported as members of the IPMP group medusa_grp1, and that bge1 and bge3 are reported as members of the IPMP group medusa_grp2.

The four IPMP test addresses (associated with `bge0:1`, `bge1:1`, `bge2:1`, and `bge3:1`, respectively) are marked NOFAILOVER. This means that they will not be transferred to a surviving interface in the event of a failure.

8. **Test that the IPMP configuration works by temporarily removing one SSC from the chassis.**

   This will cause error messages similar to the following to appear on the console:

```
Nov 19 13:55:47 medusa-s1 bge: NOTICE: bge3: link down
Nov 19 13:55:47 medusa-s1 in.mpathd[108]: The link has gone down on bge3
Nov 19 13:55:47 medusa-s1 bge: NOTICE: bge2: link down
Nov 19 13:55:47 medusa-s1 in.mpathd[108]: NIC failure detected on bge3 of group medusa_grp2
Nov 19 13:55:47 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge3 to NIC bge1
Nov 19 13:55:47 medusa-s1 in.mpathd[108]: The link has gone down on bge2
Nov 19 13:55:47 medusa-s1 in.mpathd[108]: NIC failure detected on bge2 of group medusa_grp1
Nov 19 13:55:47 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge2 to NIC bge0
```

It takes approximately 10 seconds for the IPMP daemon to detect and recover from a network failure with the default configuration. The configuration of the IPMP daemon is defined in the `/etc/default/mpathd` file.

CHAPTER 12

Adding Blade Management and VLAN Tagging in Solaris x86

This chapter tells you how to configure the system chassis to permit secure management of server blades from the management network.

This chapter contains the following sections:

12.1 Introduction

This chapter tells you how to refine the configuration in Chapter 11 to enable you (as network administrator) to perform management tasks on the server blades from the management network (that is, by telnet connections direct to the server blades) without compromising the security of the management network.

Note – This chapter modifies and supplements Chapter 6 of the *Sun Fire B1600 Blade System Chassis Software Setup Guide*. In particular, the sample network described in that chapter (including the sample switch configuration) is taken as the starting point for the configuration examples in this one. Please read Chapter 6 of the *Sun Fire B1600 Blade System Chassis Software Setup Guide* before following the instructions below.

12.2 Setting up the Server Blades Using IPMP for Network Resiliency (VLAN Tagging)

The switch configuration described in Chapter 6 of the *Sun Fire B1600 Blade System Chassis Software Setup Guide* uses tagged VLANs to separate the data and management networks. For IPMP to work with this switch configuration, you need four IP addresses for each VLAN that the server blade is a member of. In other words, for a:

- B100x blade (two physical network interfaces) you need eight IP addressess, four for the management VLAN and four for the data VLAN.
- B200x blade (four physical network interfaces) you need 16 IP addressess, eight for the management VLAN and eight for the data VLAN.

This is because the IPMP driver supports tagged VLANs by using a separate pair of logical Ethernet interfaces for each VLAN. These logical interfaces each have to be named manually according to a simple formula:

`bge`(*VLAN id* x 1000) + *instance*

where *VLAN id* is the number of the VLAN (as configured on the switch ports that the server blade is connected to inside the chassis), and *instance* is:

- 0 or 1 (on a B100x blade), depending on whether the logical interface is associated with the physical interface `bge0` or `bge1`.

- 0, 1, 2, or 3 (on a B200x blade), depending on whether the logical interfaces is associated with the physical interface `bge0`, `bge1`, `bge2` or `bge3`.

The effect of creating these pairs of logical Ethernet interfaces is to ensure that frames for one network go to that network and not to any other. Whenever the IPMP driver has a frame to send to the switch, it tags it for whichever VLAN is destined to receive it, and then transmits it using one of the logical interfaces available for that VLAN. One of the switches then receives the frame. And, assuming that the switch has been configured to accept frames for the VLAN indicated by the tag, it forwards the frame onto that VLAN.

The important point is that the server blade's IPMP driver has transmitted the frame onto a particular VLAN, and has used a redundant virtual connection to that VLAN to do so. Any other VLANs that the server blade is a member of have been prevented from receiving the frame.

12.3 Configuring IPMP With Tagged VLAN Support on a B100x Blade

This section tells you how to configure IPMP on a server blade so that the two Ethernet interfaces both provide two active logical interfaces (one each to the data VLAN and the management VLAN).

For purposes of illustration the instructions below use sample configuration input from the network scenario described in Chapter 6 of the *Sun Fire B1600 Blade System Chassis Software Setup Guide*.

Note – You need to perform the instructions in this section on each B100x blade that requires a redundant connection to the data network and the management network.

---

1. **If you have not already done so, migrate the blade from its DHCP configuration to a configuration that uses static IP addresses.**

   To do this, follow the instructions in Section 11.3, "Migrating From DHCP to Static IP Addresses" on page 11-4.

2. **If you havenot already configured your switches by following the instructions in Chapter 6 of the** ***Sun Fire B1600 Blade System Chassis Software Setup Guide*****, do so now.**

3. **Log into the console of the server blade whose interfaces you want to configure.**

   Type the following at the `sc>` prompt:

```
sc> console sn
```

   where *n* is the number of the slot containing the server blade you want to log into.

4. **Edit the** `/etc/hosts` **file on the server blade to add the IP addresses for the management interfaces.**

   For example:

```
#
# Internet host table
#
127.0.0.1       localhost

192.168.1.150 medusa-s1  loghost
192.168.1.166 medusa-s1-1
192.168.1.100 medusa-s1-test0
192.168.1.116 medusa-s1-test1

192.168.2.150 medusa-s1-mgt
192.168.2.166 medusa-s1-1-mgt
192.168.2.100 medusa-s1-mgt-test0
192.168.2.116 medusa-s1-mgt-test1
```

5. **Remove the** `/etc/hostname.`*interface* **files, where** *interface* **is** `beg0` **or** `bge1`**:**

```
# rm /etc/hostname.bge0
# rm /etc/hostname.bge1
```

6. **Set the netmasks for the management and data networks in the server blade's** `/etc/netmasks` **file.**

   For example:

```
192.168.1.0     255.255.255.0
192.168.2.0     255.255.255.0
```

7. **Disable routing, because the server blade is not being used to perform routing.**

   Type:

```
# touch /etc/notrouter
# ndd -set /dev/ip ip_forwarding 0
```

8. **In the blade's `/etc` directory, create files called:**

   `hostname.bge2000`, `hostname.bge2001`,
   `hostname.bge3000`, `hostname.bge3001`

CODE EXAMPLE 12-1 Sample File for `hostname.bge2000`

```
medusa-s1-mgt netmask + broadcast + group medusa_grp0-mgt up \
addif medusa-s1-mgt-test0 netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-2 Sample File for `hostname.bge2001`

```
medusa-s1-1-mgt netmask + broadcast + group medusa_grp0-mgt up \
addif medusa-s1-mgt-test1 netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-3 A sample file for `hostname.bge3000` is as follows:

```
medusa-s1 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test0 netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-4 Sample File for `hostname.bge3001`:

```
medusa-s1-1 netmask + broadcast + group medusa_grp0 up \
addif medusa-s1-test1 netmask + broadcast + -failover deprecated up
```

9. **Inspect the configuration of the two network adapters by typing:**

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
bge2000: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 2
        inet 192.168.2.150 netmask ffffff00 broadcast 192.168.2.255
        groupname medusa_grp0-mgt
        ether 0:3:ba:29:e6:28
bge2000:1: flags=209040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 2
        inet 192.168.2.100 netmask ffffff00 broadcast 192.168.2.255
bge2001: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 3
        inet 192.168.2.166 netmask ffffff00 broadcast 192.168.2.255
        groupname medusa_grp0-mgt
        ether 0:3:ba:29:e6:29
bge2001:1: flags=209040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 3
        inet 192.168.2.116 netmask ffffff00 broadcast 192.168.2.255
bge3000: flags=211000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,FAILED,CoS> mtu 1500 index 4
        inet 192.168.1.150 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:e6:28
bge3000:1: flags=219040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 4
        inet 192.168.1.100 netmask ffffff00 broadcast 192.168.1.255
bge3001: flags=211000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,FAILED,CoS> mtu 1500 index 5
        inet 192.168.1.166 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:e6:29
bge3001:1: flags=219040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 5
        inet 192.168.1.116 netmask ffffff00 broadcast 192.168.1.255
```

The output above shows that eight addresses have been defined. The four IPMP test addresses are marked NOFAILOVER. This means that they will not be transferred to the surviving interface in the event of a failure.

10. **Test IPMP by temporarily removing one SSC from the chassis.**

    This will cause the following error messages to be displayed on the console:

```
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: The link has gone down on bge3001
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: NIC failure detected on bge3001 of group medusa_grp0
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge3001 to NIC bge3000
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: The link has gone down on bge2001
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: NIC failure detected on bge2001 of group medusa_grp0-mgt
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge2001 to NIC bge2000
```

Note – It takes approximately 10 seconds for the IPMP daemon to detect and recover from a network failure with the default configuration. The configuration of the IPMP daemon is defined in the `/etc/default/mpathd` file.

12.4 Configuring IPMP With Tagged VLAN Support on a B200x Blade

This section tells you how to configure IPMP on a B200x blade so that the four Ethernet interfaces all provide two active logical interfaces (one each to the data VLAN and the management VLAN).

For purposes of illustration the instructions below use sample configuration input from the network scenario described in Chapter 6 of the *Sun Fire B1600 Blade System Chassis Software Setup Guide*. They also assume that the server blade configuration for IPMP described in Chapter 11 has already been performed.

Note – You need to perform the instructions in this section on each B200x blade that requires a redundant connection to the data network and the management network.

---

1. **If you have not already done so, migrate the blade from its DHCP configuration to a configuration that uses static IP addresses.**

   To do this, follow the instructions in Section 11.3, "Migrating From DHCP to Static IP Addresses" on page 11-4.

2. **If you have not already configured your switches by following the instructions in Chapter 6 of the** *Sun Fire B1600 Blade System Chassis Software Setup Guide*, **do so now.**

3. **Log into the console of the server blade whose interfaces you want to configure.**

   Type the following at the `sc>` prompt:

```
sc> console sn
```

   where *n* is the number of the slot containing the server blade you want to log into.

4. **Edit the `/etc/hosts` file on the server blade to add the IP addresses for the management interfaces.**

   For example:

```
# Internet host table
#
127.0.0.1       localhost

192.168.1.150 medusa-s1  loghost
192.168.1.166 medusa-s1-1
192.168.1.182 medusa-s1-2
192.168.1.198 medusa-s1-3

192.168.1.100 medusa-s1-test0
192.168.1.116 medusa-s1-test1
192.168.1.132 medusa-s1-test2
192.168.1.148 medusa-s1-test3

192.168.2.150 medusa-s1-mgt
192.168.2.166 medusa-s1-1-mgt
192.168.2.182 medusa-s1-2-mgt
192.168.2.198 medusa-s1-3-mgt

192.168.2.100 medusa-s1-mgt-test0
192.168.2.116 medusa-s1-mgt-test1
192.168.2.132 medusa-s1-mgt-test2
192.168.2.148 medusa-s1-mgt-test3
```

5. **Remove the `/etc/hostname.`*interface* files, where *interface* is `beg0`, `bge1`, `beg2` or `bge3`:**

```
# rm /etc/hostname.bge0
# rm /etc/hostname.bge1
# rm /etc/hostname.bge2
# rm /etc/hostname.bge3
```

6. **Set the netmasks for the management and data networks in the server blade's `/etc/netmasks` file.**

   For example:

```
192.168.1.0     255.255.255.0
192.168.2.0     255.255.255.0
```

7. **Disable routing, because the server blade is not being used to perform routing.**

   Type:

```
# touch /etc/notrouter
# ndd -set /dev/ip ip_forwarding 0
```

8. **In the blade's** `/etc` **directory, create files called:**

   `hostname.bge2000`, `hostname.bge2001`,
   `hostname.bge2002`,`hostname.bge2003`,
   `hostname.bge3000`, `hostname.bge3001`,
   `hostname.bge3002`,`hostname.bge3003`

CODE EXAMPLE 12-5 Sample File for `hostname.bge2000`

```
medusa-s0-mgt group medusa_grp0-mgt netmask + broadcast + failover up
addif medusa-s0-test0-mgt netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-6 Sample File for `hostname.bge2001`

```
medusa-s0-1-mgt group medusa_grp0-mgt netmask + broadcast + failover up
addif medusa-s0-test1-mgt netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-7 Sample File for `hostname.bge2002`

```
medusa-s0-2-mgt group medusa_grp0-mgt netmask + broadcast + failover up
addif medusa-s0-test2-mgt netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-8 Sample File for `hostname.bge2003`

```
medusa-s0-3-mgt group medusa_grp0-mgt netmask + broadcast + failover up
addif medusa-s0-test3-mgt netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-9 Sample File for `hostname.bge3000`

```
medusa-s0 group medusa_grp0 netmask + broadcast + failover up
addif medusa-s0-test0 netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-10 Sample File for `hostname.bge3001`

```
medusa-s0-1 group medusa_grp0 netmask + broadcast + failover up
addif medusa-s0-test1 netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-11 Sample File for `hostname.bge3002`

```
medusa-s0-2 group medusa_grp0 netmask + broadcast + failover up
addif medusa-s0-test2 netmask + broadcast + -failover deprecated up
```

CODE EXAMPLE 12-12 Sample File for `hostname.bge3003`

```
medusa-s0-3 group medusa_grp0 netmask + broadcast + failover up addif
medusa-s0-test3 netmask + broadcast + -failover deprecated up
```

9. **Inspect the configuration of the two network adapters by typing:**

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
bge2000: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 2
        inet 192.168.2.150 netmask ffffff00 broadcast 192.168.2.255
        groupname medusa_grp0-mgt
        ether 0:3:ba:29:e6:28
bge2000:1: flags=209040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 2
        inet 192.168.2.100 netmask ffffff00 broadcast 192.168.2.255
bge2001: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 3
        inet 192.168.2.166 netmask ffffff00 broadcast 192.168.2.255
        groupname medusa_grp0-mgt
        ether 0:3:ba:29:e6:29
bge2001:1: flags=209040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 3
        inet 192.168.2.116 netmask ffffff00 broadcast 192.168.2.255
bge2002: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 4
        inet 192.168.2.182 netmask ffffff00 broadcast 192.168.2.255
        groupname medusa_grp0-mgt
        ether 0:3:ba:29:e6:2a
bge2002:1: flags=209040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 4
        inet 192.168.2.132 netmask ffffff00 broadcast 192.168.2.255
bge2003: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 5
        inet 192.168.2.198 netmask ffffff00 broadcast 192.168.2.255
        groupname medusa_grp0-mgt
        ether 0:3:ba:29:e6:2b
bge2003:1: flags=209040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 5
        inet 192.168.2.148 netmask ffffff00 broadcast 192.168.2.255
bge3000: flags=211000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,FAILED,CoS> mtu 1500 index 6
        inet 192.168.1.150 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:e6:28
bge3000:1: flags=219040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 6
        inet 192.168.1.100 netmask ffffff00 broadcast 192.168.1.255
bge3001: flags=211000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,FAILED,CoS> mtu 1500 index 7
        inet 192.168.1.166 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:e6:29
bge3001:1: flags=219040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 7
        inet 192.168.1.116 netmask ffffff00 broadcast 192.168.1.255
bge3002: flags=211000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,FAILED,CoS> mtu 1500 index 8
        inet 192.168.1.182 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:e6:2a
bge3002:1: flags=219040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 8
        inet 192.168.1.132 netmask ffffff00 broadcast 192.168.1.255
bge3003: flags=211000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,FAILED,CoS> mtu 1500 index 9
        inet 192.168.1.198 netmask ffffff00 broadcast 192.168.1.255
        groupname medusa_grp0
        ether 0:3:ba:29:e6:2b
bge3003:1: flags=219040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,CoS> mtu 1500 index 9
        inet 192.168.1.148 netmask ffffff00 broadcast 192.168.1.255
#
```

The output above shows that 16 addresses have been defined. The eight IPMP test addresses are marked NOFAILOVER. This means that they will not be transferred to the surviving interface in the event of a failure.

10. **Test IPMP by temporarily removing one SSC from the chassis.**

    This will cause the following error messages to be displayed on the console:

```
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: The link has gone down on bge3001
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: NIC failure detected on bge3001 of group medusa_grp0
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge3001 to NIC bge3000
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: The link has gone down on bge3003
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: NIC failure detected on bge3003 of group medusa_grp0
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge3003 to NIC bge3002
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: The link has gone down on bge2001
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: NIC failure detected on bge2001 of group medusa_grp0-mgt
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge2001 to NIC bge2000
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: The link has gone down on bge2003
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: NIC failure detected on bge2003 of group medusa_grp0-mgt
Nov 24 16:43:15 medusa-s1 in.mpathd[108]: Successfully failed over from NIC bge2003 to NIC bge2002
```

    It takes approximately 10 seconds for the IPMP daemon to detect and recover from a network failure with the default configuration. The configuration of the IPMP daemon is defined in the `/etc/default/mpathd` file.

CHAPTER 13

Testing the Solaris x86 Blade Memory (DIMMs)

This chapter tells you how to run memory diagnostic tests on a B100x or B200x blade.

This chapter contains the following sections:

13.1 Running the Memory Diagnostics Utility

This chapter tells you how to run memory diagnostic tests on a blade. The utility for testing blade memory is provided on the *Sun Fire B1600 Blade Platform Documentation, Drivers, and Installation* CD and on the following website:

`http://www.sun.com/servers/entry/b100x/`

If the test suite finds memory errors, then swap out the defective DIMMs by following the instructions in the *Sun Fire B1600 Blade System Chassis Administration Guide*.

1. **On a workstation connected to the network, either:**
   - Mount the *Sun Fire B1600 Blade Platform Documentation, Drivers, and Installation* CD:

     ```
     # cd /cdrom/cdrom0/solaris_x86
     ```

   - Or, go to `http://www.sun.com/servers/entry/b100x/` and download the memory diagnostic utility (`memdiag-02.tar`) to a known location on the network. (The `-01` in this file name indicates the version number; later versions will have a different number.)
2. **Use FTP to transfer the** `memdiag-02.tar` **to the** `/tftpboot` **directory on the system you are using as the DHCP server for your network.**
3. **Become root on the DHCP server, and extract the contents of the** `memdiag-02.tar` **file.**

Caution – If your /tftpboot directory contains either a pxelinux.bin file or a pxeconf.cfg directory and you want to preserve these, then rename them before extracting the memdiag.tar archive. Otherwise the tar xvf command will overwrite them.

To extract the contents of the memdiag-02.tar file, type:

```
# cd /tftpboot
# tar xvf memdiag-02.tar
x ., 0 bytes, 0 tape blocks
x ./pxelinux.bin, 10820 bytes, 22 tape blocks
x ./pxelinux.cfg, 0 bytes, 0 tape blocks
x ./pxelinux.cfg/memtestz, 48234 bytes, 95 tape blocks
x ./pxelinux.cfg/default, 503 bytes, 1 tape blocks
x ./pxelinux.cfg/bootinfo.txt, 28 bytes, 1 tape blocks
x ./pxelinux.cfg/README, 1739 bytes, 4 tape blocks
x ./pxelinux.cfg/THIRDPARTYLICENSEREADME, 17926 bytes, 36 tape
blocks
```

4. **Start the DHCP Manager GUI by typing:**

```
# DISPLAY=mydisplay:0.0
# export DISPLAY
# /usr/sadm/admin/bin/dhcpmgr &
```

where *mydisplay* is the name of the system (for example, a desktop workstation) that you are using to display the DHCP Manager's GUI (Graphical User Interface).

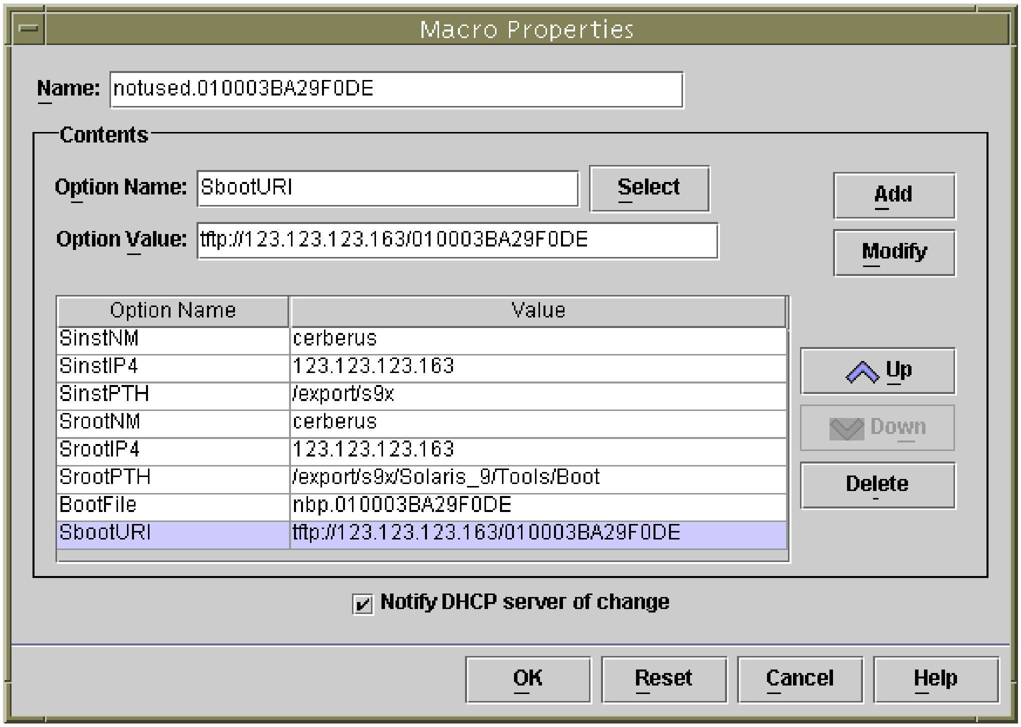
5. **Use the DHCP Manager to prevent the blade (temporarily) from booting with the Solaris network install image:**

   a. **In the DHCP manager main window click on the Macros tab and select the blade's configuration macro by selecting the entry that matches the blade's Client Id.**

   b. **Select Properties from the Edit menu.**

   c. **Make a note of the macro name (so that you can restore it when you have finished testing the memory DIMMs).**

   d. **In the Macro Properties window, rename the macro by changing the contents of the name field (see FIGURE 13-1).**



Macro Properties

Name: notused.010003BA29F0DE

Contents

Option Name: SbootURI

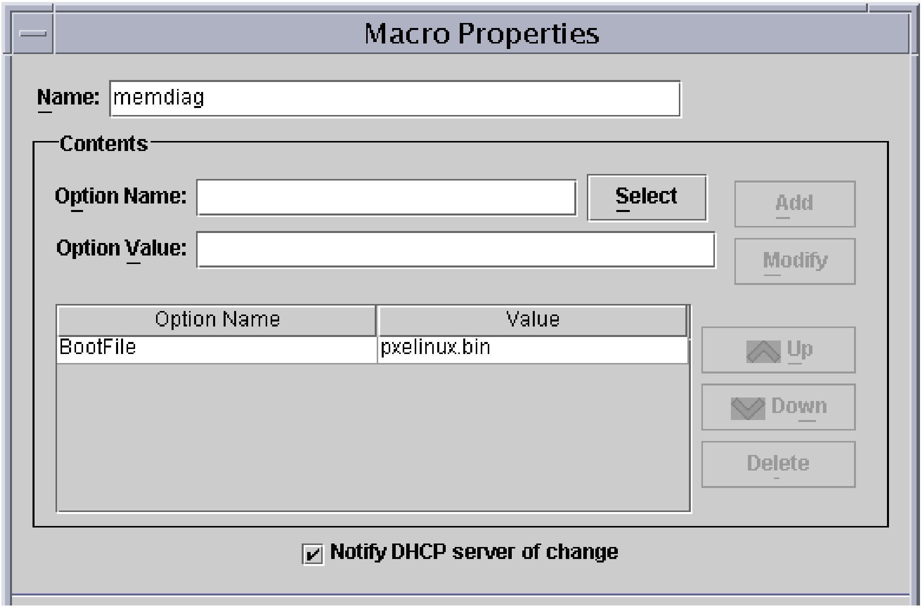
Option Value: tftp://123.123.123.163/010003BA29F0DE

Option Name	Value
SinstNM	cerberus
SinstIP4	123.123.123.163
SinstPTH	/export/s9x
SrootNM	cerberus
SrootIP4	123.123.123.163
SrootPTH	/export/s9x/Solaris_9/Tools/Boot
BootFile	nbp.010003BA29F0DE
SbootURI	tftp://123.123.123.163/010003BA29F0DE

☑ Notify DHCP server of change

FIGURE 13-1 Changing the Name of the Blade's Macro to Stop it From Booting Solaris x86

6. **Create a new macro called** `memdiag` **containing an option called** `BootFile` **that has the value** `pxelinux.bin` **(see FIGURE 13-2).**



Name: memdiag

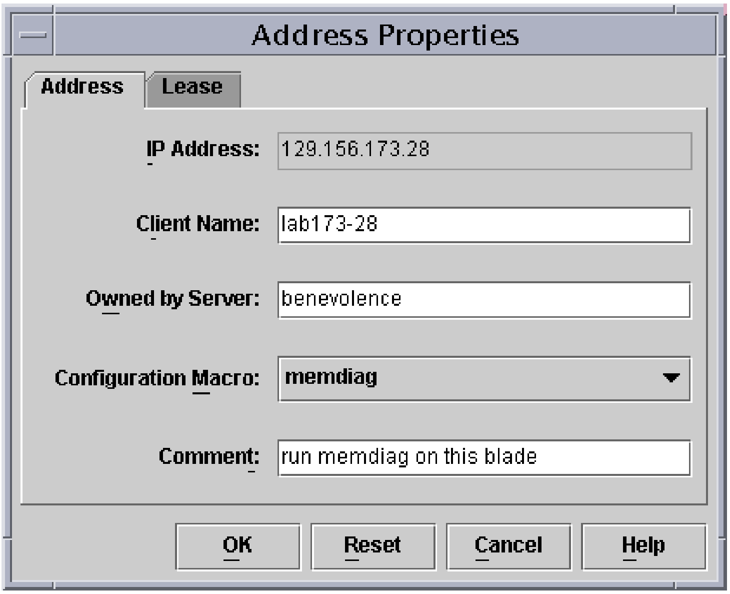
Option Name	Value
BootFile	pxelinux.bin

☑ Notify DHCP server of change

FIGURE 13-2 Macro Properties Window Showing the `memdiag` Macro

7. **In the DHCP manager window, click the Addresses tab, and select the entry for the blade you want to test.**

8. **From the Configuration Macro drop-down menu, select the** `memdiag` **macro.**



IP Address: 129.156.173.28

Client Name: lab173-28

Owned by Server: benevolence

Configuration Macro: memdiag

Comment: run memdiag on this blade

FIGURE 13-3 Selecting the `memdiag` Macro

9. **Log into the active System Controller by following the instructions in Chapter 2 of the** *Sun Fire B1600 Blade System Chassis Software Setup Guide*, **if you are logging into a brand new chassis in its factory default state.**

   Otherwise log in using the user name and password assigned to you by your system administrator.

10. **Connect to the blade's console and shutdown the blades operating system.**

    a. **Type:**

```
sc> console -f Sn
```

where *n* is the slot number of the blade.

b. **At the blade's operating system prompt, type:**

```
# shutdown -i5 -g0
```

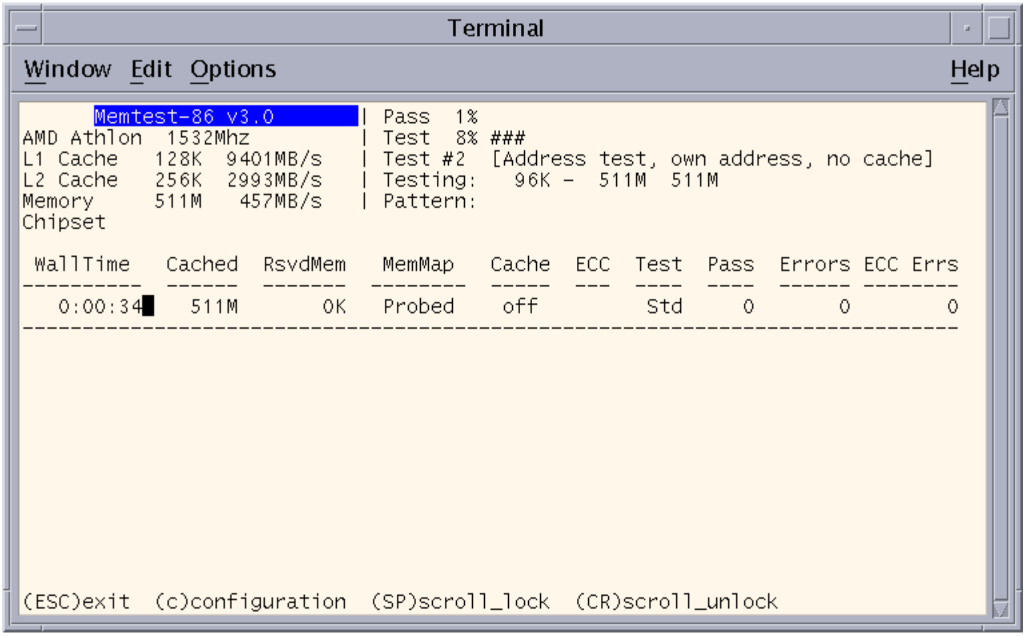
11. **Type the following command at the System Controller's** `sc>` **prompt to cause the blade to boot from the network:**

```
sc> bootmode bootscript="boot net" sn
sc> reset -y Sn
```

where *n* is the number of the slot containing the blade you are testing.

12. **To monitor the test output, access the console of the blade you are testing:**

```
sc> console -f Sn
```



```
      Memtest-86 v3.0       | Pass   1%
AMD Athlon  1532Mhz         | Test   8% ###
L1 Cache    128K  9401MB/s  | Test #2  [Address test, own address, no cache]
L2 Cache    256K  2993MB/s  | Testing:   96K -  511M  511M
Memory      511M   457MB/s  | Pattern:
Chipset

 WallTime    Cached  RsvdMem   MemMap   Cache  ECC  Test  Pass  Errors ECC Errs
----------   ------  -------  --------  -----  ---  ----  ----  ------ --------
   0:00:34     511M       0K   Probed    off         Std     0       0        0
-------------------------------------------------------------------------------

(ESC)exit  (c)configuration  (SP)scroll_lock  (CR)scroll_unlock
```

FIGURE 13-4 Sample Output from the Memory Test Utility

13. **To interrupt the memory tests, press the [Escape] key or reset the blade.**

14. **When you have finished testing the memory, restore the blade's DHCP configuration by following the instructions in Section 13.4, "Restoring the Blade's DHCP Configuration" on page 13-10.**

13.2 Duration of the Memory Tests

The time it takes to perform a memory test depends on the hardware characteristics of the blade; specifically, it is determined by the processor speed, memory size, memory controller, and memory speed.

The number of errors detected by the test suite is provided in the Errors column (see FIGURE 13-4). Each time the suite completes a test cycle it increments the Pass counter.

TABLE 13-1 Typical Duration of One Test Cycle

Blade	Typical Duration of One Test Cycle	Duration per Gigabyte of RAM
B100x	Approx 31 minutes for a 512MB blade	Approx 62 minutes/GB
B200x	Approx 40 minutes for a 2GB blade	Approx 20 minutes/GB

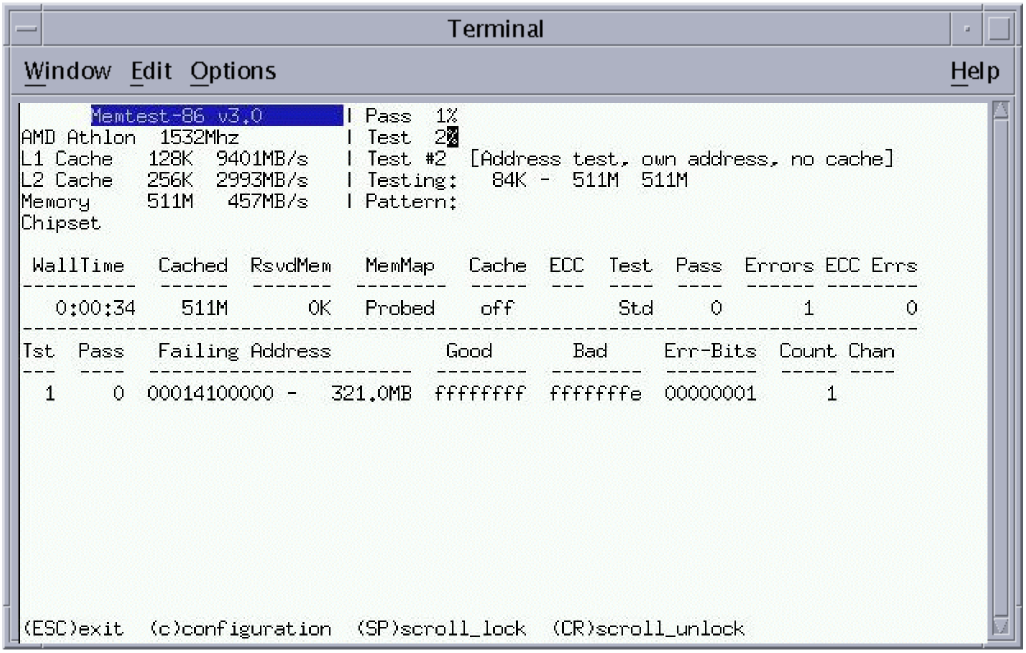
The memory tests will continue to run until you interrupt them by pressing the escape key or by resetting the blade.

Normally two complete test cycles will be enough to detect the problem with a faulty DIMM. However, you might want to perform the tests for a longer period, for example, overnight.

13.3 Error Reporting and Diagnosis

The `memtest86` utility detects whether the memory on the blade is corrupted. The example in FIGURE 13-5 shows an error that has occurred at address `0x14100000` (321MB). The screen output in FIGURE 13-5 differs from the output in FIGURE 13-4, because in FIGURE 13-5 an error is reported. The following information is provided:

`Tst`: the number of the test that detected the error
`Pass`: the number of the test cycle during which the error was detected
`Failing Address`: the physical address at which the error occurred
Good: the expected content of the memory location being tested
`Bad`: the actual content of the tested memory location
`Err-Bits`: the bit position of the error within the double-word being tested
`Count`: the number of times this error has been detected during all passes of the test



```
      Memtest-86 v3.0        | Pass   1%
AMD Athlon  1532Mhz          | Test   2%
L1 Cache    128K  9401MB/s   | Test #2  [Address test, own address, no cache]
L2 Cache    256K  2993MB/s   | Testing:   84K -  511M  511M
Memory      511M   457MB/s   | Pattern:
Chipset

 WallTime   Cached  RsvdMem   MemMap   Cache  ECC  Test  Pass  Errors ECC Errs
 ---------  ------  -------  --------  -----  ---  ----  ----  ------ --------
   0:00:34    511M       0K   Probed    off         Std     0       1        0
-------------------------------------------------------------------------------
Tst  Pass   Failing Address          Good       Bad     Err-Bits  Count Chan
---  ----  -----------------------  --------  --------  --------  ----- ----
  1     0  00014100000 -   321.0MB  ffffffff  fffffffe  00000001      1




(ESC)exit  (c)configuration  (SP)scroll_lock  (CR)scroll_unlock
```

FIGURE 13-5 Example of `memtest86` Detecting a Memory Error

When you have noted the physical address at which an error occurred, you can derive the number of the DIMM that needs replacing.

On a B100x blade, the memory controller maps the lowest address range to the lowest numbered DIMM, the next address range to the next DIMM, and so on (see TABLE 13-2).

TABLE 13-2 Mapping of Address Ranges to DIMMs on a B100x Blade

Total RAM	Banks	DIMM 0	DIMM 1	DIMM 2	DIMM 3
512MB	1	0-511MB			
1GB	2	0-511MB	512MB-1023MB		
3GB	2	0-1023M	1024MB-2047MB	2048MB-3071MB	
4GB	4	0-1023MB	1024MB-2047MB	2048MB-3071MB	3072MB-4095MB

On a B200x blade the memory controller maps the lowest address range to the lowest numbered DIMM pair. On a B200x blade you can only isolate a memory error to a pair of DIMMs.

TABLE 13-3 Mapping of Address Ranges to DIMMs on a B200x Blade

Total RAM	Banks	DIMM 0 or 1	DIMM 2 or 3
1GB	2	0-1023MB	
2GB	4	0-1023MB	1GB-2047MB
2GB	2	0-2047MB	
4GB	4	0-2047MB	2048MB-4095MB

Note – Memory errors can have several causes. They do not always indicate a defective DIMM but can be caused by noise, cross-talk, or signal integrity issues. If you repeatedly detect a memory error at a particular physical address even after you have changed the affected DIMM or DIMM pair, it is likely that the corruption has not been caused by a defective DIMM. Another source of memory errors is a defective cache. If you think this might be the problem, run the `memtest86` tests with the Cache Mode set to "Always on" in the Configuration menu.

13.4 Restoring the Blade's DHCP Configuration

When you have finished running the memory test utility you can restore the blade's DHCP settings to enable it to boot once again using the Solaris x86 network install image. This is not necessary if the operating system is already installed on the blade's hard disk. However, if you want the blade to boot again from the network to re-install Solaris x86, do the following:

1. **In the DHCP manger window click on the Macros tab and select the blade's configuration macro.**

   This is the macro that you renamed in Step 5 (see Section 13.1, "Running the Memory Diagnostics Utility" on page 13-2).

2. **Select Properties from the Edit menu.**

3. **Restore the macro name to the blade's Client Id.**

   You noted the orginal macro name in Step 5 (see Section 13.1, "Running the Memory Diagnostics Utility" on page 13-2).

   When you have restored the macro name, the blade is able to boot from the Solaris x86 network install image.

4. **In the DHCP manager's main window, click the Addresses tab, and select the entry for the blade.**

5. **From the Configuration drop-down menu, select the Client Id for the blade.**

   The blade is now ready to be booted from the network.

13.5 Further Information

This utility is a version of the `memtest86` tool that has been configured by Sun for use on the B100x and B200x blades.

For full information about the range of tests you can perform and the different algorithms used by the memory diagnostic test suite, contact your Sun Solutions Center.

CHAPTER 14

Troubleshooting the Solaris x86 PXE Boot Installation

This chapter provides information on problems that can occur during or after a PXE boot installation of the Solaris x86 operating system. It covers the following problems:

Synopsis: `prom_panic: Could not mount filesystem`

The following error can appear at startup when the blade is attempting to perform a PXE boot:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000 000000000000
CLIENT IP: 123.123.123.172  MASK: 255.255.255.0  DHCP IP: 123.123.123.163
SunOS Secondary Boot version 3.00

prom_panic: Could not mount filesystem.
Entering boot debugger:.
[136039]:
```

Cause:

The secondary bootstrap program was unable to mount the file system for the Solaris x86 install image.

Solution:

Check that the `SrootPTH` macro has been entered correctly as displayed by the `add_install_client` output (see FIGURE 10-7 in Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10).

Synopsis: Cannot Read `SUNW.i86pc` File for Blade

The following error can appear at startup when the blade is attempting to perform a PXE boot and Jumpstart installation:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000 000000000000
CLIENT IP: 123.123.123.172  MASK: 255.255.255.0  DHCP IP: 123.123.123.163
GATEWAY IP: 123.123.123.8


Solaris network boot ...

Cannot read file 123.123.123.163:/tftpboot/SUNW.i86pc.
Type <ENTER> to retry network boot or <control-C> to try next boot device
```

where `123.123.123.163` is the IP address of the Network Install Server containing the Solaris x86 image for the blade.

Cause:

The data structures used by DHCP to transfer the DHCP option strings currently impose a limit of 255 characters on the length of these strings. If this limit is exceeded one of the option strings will be truncated. If this happens to be the value of the `Bootfile` option, then the PXE boot protocol will attempt to perform a non-client-specific PXE boot by reading the file `SUNW.i86pc`. This file is not suitable for booting B100x and B200x blades and in any case it will not normally exist in the `/tftpboot` directory on the Network Install Server.

Solution:

When configuring the DHCP options strings (see Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10), you need to take into account that long names for the Install server path and the root server path will quickly use up the available option string space of 255 characters. For a screen shot of the window in the DHCP Manager's GUI where the path for the option string is specified, see FIGURE 10-8.

If you have encountered this problem, reduce the length of the SrootPTH and SinstPTH option strings. You can achieve this by creating a link to the full path stored in the Network Install Server's file system. For example, supposing the paths for SrootPTH and SinstPTH are:

```
SrootPTH=/export/install/media/b100xb200x/solaris9install/Solaris_9/Tools/Boot
SinstPTH=/export/install/media/b100xb200x/solaris9-install
```

You can reduce the length of these specified paths by creating a link to the solaris9-install image on the Network Install Server. To do this:

1. **Log in as root to the Network Install Server and type the following command:**

```
# ln -s /export/install/media/b100xb200x/solaris9-install /export/s9-install
```

2. **Adjust the macros in the DHCP server as follows:**

```
SrootPTH=/export/s9-install/Solaris_9/Tools/Boot
SinstPTH=/export/s9-install
```

In this example, this has reduced the total length of these two DHCP option strings by 62 characters.

Synopsis: PXE Access Violation Before Primary Bootstrap Has Loaded

The following error can appear at startup when the blade is attempting to perform a PXE boot:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000 000000000000
CLIENT IP: 123.123.123.172  MASK: 255.255.255.0  DHCP IP: 123.123.123.163
GATEWAY IP: 123.123.123.8
TFTP.
PXE-T02: Access violation
PXE-E3C: TFTP Error - Access Violation

PXE-M0F: Exiting Broadcom PXE ROM.
```

Cause:

This error message indicates that, during the PXE boot process, the blade was unable to download the primary bootstrap program from the install server's `/tftpboot` area. There are a number of possible reasons for this:

- You did not execute the `add_install_client` command.
- You did not execute the `add_install_client` command for a Solaris x86 install image that supports client-specific booting.
- You ran the `add_install_client` on the wrong Network Install Server.
- You ran the `add_install_client` correctly but the DHCP macros are pointing at the wrong Network Install Server.
- The primary bootstrap program has been deleted from the Network Install Server's `/tftpboot` directory.

Solution:

If you think you did not execute the `add_install_client` command, then execute it now (see Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10). When you have done so, check that the files for the primary bootstrap, the secondary bootstrap, and the client-specific boot settings exist in the `/tftpboot` area on the Network Install Server.

If any of them do not exist there (or do not have read permissions), you will encounter access violation errors during the PXE boot process.

To check you have the correct client-specific files in the /tftpboot area, do the following:

1. **Search for all the files that contain the blade's MAC address in their filename.**

   Assuming a blade MAC address of 00:03:BA:29:F0:DE, you would type the following command (remembering that in these filenames the MAC address is preceded by 01 and has its colon characters removed):

```
# cd /tftpboot
# ls -l *010003BA29F0DE*
lrwxrwxrwx   1 root   other         26 Oct 29 12:35 010003BA29F0DE -> inetboot.I86PC.Solaris_9-1
-rw-r--r--  1 root   other        639 Oct 29 12:35 010003BA29F0DE.bootenv.rc
lrwxrwxrwx  1 root   other         21 Oct 29 12:35 nbp.010003BA29F0DE -> nbp.I86PC.Solaris_9-1
-rw-r--r--  1 root   other        568 Oct 29 12:35 rm.010003BA29F0DE
```

   The output from this command shows the:

   - Primary bootstrap files
     In our example, the client-specific primary bootstrap file is called nbp.010003BA29F0DE. This file is a symbolic link to a copy (in the /tftpboot area) of the primary bootstrap program belonging to the Solaris x86 image you are using for the blade or blades. In our example, this copy of the install image's primary bootstrap file is called nbp.I86PC.Solaris_9-1.
   - Secondary bootstrap files
     In our example, the client-specific secondary bootstrap file is called 010003BA29F0DE. This file is a symbolic link to a copy (in the /tftpboot area) of the secondary bootstrap program belonging to the Solaris x86 image you are using for the blade or blades. In our example, this copy of the install image's secondary bootstrap file is called inetboot.I86PC.Solaris_9-1.
   - Client-specific boot settings file
     In our example, this file is called 010003BA29F0DE.bootenv.rc.

   The files listed in the above output with an arrow (->) after them are links. The filename after the arrow is the file that they link to.

2. **Use the ls command to check that the copies required of the install image's original bootstrap files do in fact exist in the /tftpboot area:**

```
# ls -l nbp.I86PC.Solaris_9-1
-rwxr-xr-x   1 root     other      14596 Oct 29 12:35 nbp.I86PC.Solaris_9-1
#
# ls -l inetboot.I86PC.Solaris_9-1
-rwxr-xr-x   1 root     other     401408 Oct 29 12:35 inetboot.I86PC.Solaris_9-1
```

   The copies of the install image's bootstrap files in /tftpboot are created by the add_install_client utility (which you ran in Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10).

If they do not exist in /tftpboot, then either you have not run the add_install_client utility, or you have run it for a network install image that does not support client-specific PXE booting.

In either case run the add_install_client utility for the correct install image, following the instructions in Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10.

3. **If the bootstrap files pointed to by the links do exist in** /tftpboot **(in other words, if they are listed by the** ls **command that you ran in Step 2), then check they are the same size as the original bootstrap programs belonging to the Solaris x86 install image that you intend to use for the blade or blades.**

   To do this, run the ls commands for the original bootstrap files belonging to the install image you intend to use, and compare their file sizes with the file sizes reported in Step 2 for the client-specific files in /tftpboot.

   In the sample commands provided in Chapter 10, the Solaris x86 install image was located in the directory /export/s9x on the Network Install Server. The sample commands below assume the same path:

```
# cd /export/s9x/Solaris_9/Tools/Boot

# ls -l usr/platform/i86pc/lib/fs/nfs/inetboot
-rw-r--r--   1 root     sys       401408 Oct  7 23:55 usr/platform/i86pc/lib/fs/nfs/inetboot

# ls -l boot/solaris/nbp
-rw-r--r--   1 root     sys        14596 Sep 23 15:45 boot/solaris/nbp
```

4. **If the necessary files did not exist in the** /tftpboot **directory on the Network Install Server, or if they were not identical to the bootstrap files belonging to the install image you have been intending to use for the blade or blades, then run the** add_install_client **utility again for the correct image (see Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10).**

   If the files did appear to exist and to be the correct files, a final check is to compare the checksums for the different files using the sum(1) command. If the checksum for the client-specific copy matches the checksum for the original file belonging to the install image, then the files are identical. If not, run the add_install_client utility again, making sure you run it for the correct Solaris x86 install image.

Synopsis: Cannot Read Secondary Bootstrap

The following error can appear at startup when the blade is attempting to perform a PXE boot:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000 000000000000
CLIENT IP: 123.123.123.172  MASK: 255.255.255.0  DHCP IP: 123.123.123.163
GATEWAY IP: 123.123.123.8



Solaris network boot ...

Cannot read file 123.123.123.163:/tftpboot/010003BA29F0DE.
Type <ENTER> to retry network boot or <control-C> to try next boot device ...
```

Cause:

- The primary bootstap loaded, but for some reason the secondary bootstrap program could not be loaded.

Solution:

Carry out the same checks as were recommended in the solution to the following problem: "Synopsis: PXE Access Violation Before Primary Bootstrap Has Loaded" on page 14-5

Synopsis: Blade Appears to Hang After Primary Bootstrap is Loaded

The following error can appear at startup when the blade is attempting to perform a PXE boot:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000 000000000000
CLIENT IP: 123.123.123.172  MASK: 255.255.255.0  DHCP IP: 123.123.123.163
GATEWAY IP: 123.123.123.8



Solaris network boot ...
```

Cause:

Possible causes include:

- The client-specific boot-settings file has been corrupted or is missing.
- When you executed the `add_install_client` command you did not use the `-b "input-device=ttya"` and `-b "output-device=ttya"` parameters.
- You executed the he `add_install_client` command with incorrect data in the `-b` arguments. For example `-b "input-device=ttyb"`, or `-b "output-device=tty"`.
- The blade booted using a non-client specific PXE boot image.

Solution:

The first thing to check is that you have run the `add_install_client` command correctly (see Section 10.4, "Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade" on page 10-10). If you are not sure, you can simply run the command again. Then carry out the same checks as were recommended in the solution to the problem: "Synopsis: PXE Access Violation Before Primary Bootstrap Has Loaded" on page 14-5

Synopsis: Secondary Boot Program Aborts to > Prompt

The following error can appear at startup when the blade is attempting to perform a PXE boot:

```
Broadcom UNDI PXE-2.1 (build 082) v6.2.11
Copyright (C) 2000-2003 Broadcom Corporation
Copyright (C) 1997-2000 Intel Corporation
All rights reserved.

CLIENT MAC ADDR: 00 03 BA 29 F0 DE  GUID: 00000000 0000 0000 0000 000000000000
SunOS Secondary Boot version 3.00 255.255.255.0  DHCP IP: 123.123.123.163
GATEWAY IP: 123.123.123.8
/dev/diskette0: device not installed, unknown device type 0


            Solaris Intel Platform Edition Booting System


>
```

Cause:

Possible causes include:

- The client-specific boot-settings file has been corrupted and the secondary boot program was unable to interpret its contents.
- You executed the `add_install_client` command with incorrect data for the `-b` arguments. For example, you might have missed a quote mark when setting the `boot-args` property (see Section 10.10, “Configuring a Jumpstart Installation” on page 10-39).

Solution:

The first thing to check is that you have run the `add_install_client` command correctly (see Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10). If you are not sure, you can simply run the command again. Then carry out the same checks as were recommended in the solution to the problem: “Synopsis: PXE Access Violation Before Primary Bootstrap Has Loaded” on page 14-5.

Synopsis: Malformed Bootpath

The following error can appear at startup when the blade is attempting to perform a PXE boot:

```
Error: Malformed bootpath

Property The bootpath property:

/pci@0,0/pci78887,7

is badly formed, and will be ignored.

Press Enter to  Continue.






Enter_Continue
```

Cause:

Possible causes include:

- The client-specific boot-settings file has been corrupted and the Device Configuration Assistant was unable to unable to interpret its contents.
- You executed the `add_install_client` command with an incorrect `bootpath` value.

Solution:

The first thing to check is that you have run the `add_install_client` command correctly (see Section 10.4, “Configuring the Install Server and the DHCP Server to Install Solaris x86 Onto Each Blade” on page 10-10). If you are not sure, you can simply run the command again. Then carry out the same checks as were recommended in the solution to the problem: “Synopsis: PXE Access Violation Before Primary Bootstrap Has Loaded” on page 14-5.

Synopsis: Installation Stops at Screen Called ’Solaris Device Configuration Assistant’

The following screen can appear at startup when the blade is attempting to perform a PXE boot:

```
Solaris Device Configuration Assistant

 The Solaris(TM) (Intel Platform Edition) Device Configuration Assistant
 scans to identify system hardware, lists identified devices, and can
 boot the Solaris software from a specified device. This program must be
 used to install the Solaris operating environment, add a driver, or
 change the hardware on the system.



 > To perform a full scan to identify all system hardware, choose Continue.

 > To diagnose possible full scan failures, choose Specific Scan.

 > To add new or updated device drivers, choose Add Driver.

 About navigation...
     - The mouse cannot be used.
     - If the keyboard does not have function keys or they do not respond,
       press ESC. The legend at the bottom of the screen will change to show
       the ESC keys to use for navigation.
     - The F2 key performs the default action.



 F2_Continue    F3_Specific Scan    F4_Add Driver    F6_Help
```

Cause:

Possible causes include:

- The client-specific boot-settings file has been corrupted and the Device Configuration Assistant was unable to unable to interpret its contents.
- You executed the `add_install_client` command without specifying a `bootpath` value.
- There are missing or invalid key words in the configuration files that form your Jumpstart configuration. For example:
  - The `x86-class` file does not contain a valid `install_type` key word and value.

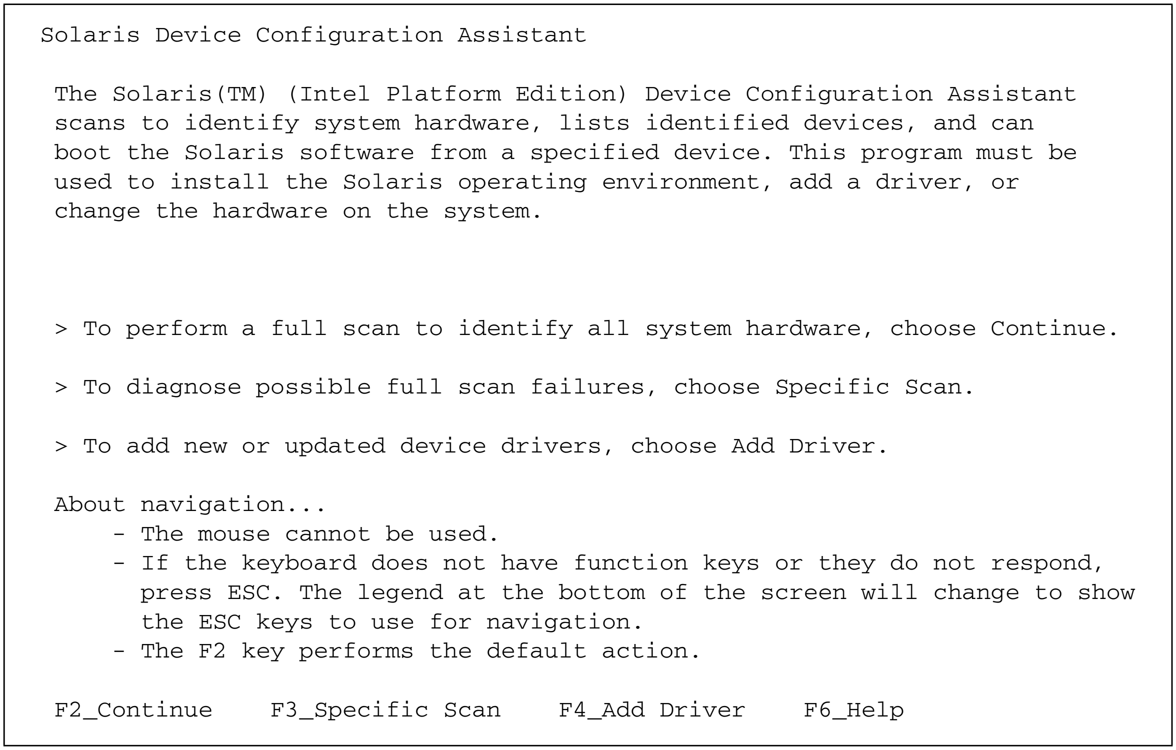
    - The sysidcfg file does not contain a valid system_locale key word and value.
    - The sysidcfg file does not contain valid NIS parameters for your site.
- You executed the add_install_client command with an incorrect bootpath specified. For example, this problem will occur if you specify the bootpath for the B100x when the blade is a B200x. For the correct bootpath values for the blades and their different interfaces, see Section 10.12, “Installing Solaris x86 Onto a Blade by Using the Second, Third, or Fourth Network Interface” on page 10-47.

Solution:

For information about setting up Jumpstart correctly for your requirements, refer to the *Solaris 9 Installation Guide*, and see Section 10.9, “Preparatory Steps for Setting up a Jumpstart Installation for a Blade” on page 10-34, and Section 10.10, “Configuring a Jumpstart Installation” on page 10-39.

Synopsis: Blade Boots to Device Configuration Assistant on Every Reboot After an Interactive Network Installation

The following screen can also appear when you are performing an interactive network installation of Solaris x86 on a blade that has previously had Solaris x86 or Linux running on it but that has a disk partition table that does not contain separate Boot and Solaris partitions.



```
Solaris Device Configuration Assistant

 The Solaris(TM) (Intel Platform Edition) Device Configuration Assistant
 scans to identify system hardware, lists identified devices, and can
 boot the Solaris software from a specified device. This program must be
 used to install the Solaris operating environment, add a driver, or
 change the hardware on the system.



 > To perform a full scan to identify all system hardware, choose Continue.

 > To diagnose possible full scan failures, choose Specific Scan.

 > To add new or updated device drivers, choose Add Driver.

 About navigation...
     - The mouse cannot be used.
     - If the keyboard does not have function keys or they do not respond,
       press ESC. The legend at the bottom of the screen will change to show
       the ESC keys to use for navigation.
     - The F2 key performs the default action.

 F2_Continue    F3_Specific Scan    F4_Add Driver    F6_Help
```

Cause

The blade's hard disk partition table does not define separate Boot and Solaris partitions. Because of this the bootpath property was not set at the end of the install process in the file `/a/boot/solaris/bootenv.rc`.

Solution

If you want to install the blade using a single Solaris disk partition, follow the instructions in Chapter 8 to perform a Jumpstart installation. In particular, make sure you use the `x86-finish` script as described in Section 10.9, “Preparatory Steps for Setting up a Jumpstart Installation for a Blade" on page 10-34. This will ensure that, before the blade is rebooted, the bootpath property is correctly set in the file /a/boot/solaris/bootenv.rc.

Alternatively you can simply step through the DCA screens by pressing [F2] and [ENTER], then selecting the hard disk as the boot device. When Solaris has booted you can then use an editor to add the correct bootpath property to the file /a/boot/solaris/bootenv.rc.

- For a B100x, use the following entry:

```
setprop bootpath /pci@0,0/pci-ide@11,1/ide@0/cmdk@0,0:a
```

- For a B200x, use the following entry:

```
setprop bootpath /pci@0,0/pci-ide@1f,1/ide@0/cmdk@0,0:a
```

To prevent this problem from occurring when you reboot after a future interactive network installation, perform the installation as described in Chapter 10, and follow the instructions in Section 10.8.6, "Removing the Entire Disk Partition Table Before Restarting the Solaris Install Program" on page 10-28.

PART 4 Appendixes

APPENDIX A

Upgrading Firmware

This chapter provides information on upgrading the System Controller firmware and Blade System Chip firmware. The chapter contains the following sections

A.1 Introduction

Note – To perform the update procedures in this chapter, you need to have a connection from the NETMGT port to the management network. This is because you need to transfer the new firmware from a location on your network.

This chapter tells you how to upgrade the firmware on:

- The System Controllers,
- One or more Blade Support Chips (each server blade contains a single one of these, called a BSC for short),

The BSC on each server blade is a management agent for the System Controller. It communicates information about the server blade it resides in to the System Controller. It also receives and processes any commands that you type into the System Controller's command-line interface.

Follow the instructions in this chapter if you have been advised by a Sun support engineer to download new firmware onto a System Controller, server blade, or integrated switch.

New firmware for System Controllers and server blades will be made available as patches on SunSolve. These patches are not operating system patches and are not installed using the standard Solaris `patchadd(1m)` utility. Once the patches have been unpacked they deliver the firmware images with the filename format shown in TABLE A-1.

TABLE A-1 The Filenames of the Firmware

Firmware Image	Filename
System Controller application	`SunFireB1600-sc-`*vxxxx*`.flash`[1]
Blade Support Chip firmware	`SunFireB100x-bsc-`*vxxxx*`.flash`[1] `SunFireB200x-bsc-`*vxxxx*`.flash`[1]

[1] Note that -*vxxxx* represents the version number of the firmware.

In addition to following the instructions in this chapter, please perform any special instructions that are provided in the patch README files.

A.2 Installing Firmware Images on a TFTP Server

The latest firmware patches are available from the following website:

`wwws.sun.com/software/download/network.html`

When you have downloaded the Sun Fire B1600 firmware patches (and unpacked the firmware images), you need to install them onto a TFTP server. This makes them available to the System Controller's `flashupdate` command.

You can install firmware images on the Linux TFTP server that you created when preparing to perform the PXE boot installation (see Section 4.2.2.2, "Configuring the TFTP Server" on page 4-9 for more information). Alternatively, if you are using a Solaris TFTP server, see the chapter on updating firmware in the *Sun fire B1600 Blade System Chassis Administration Guide*.

- **To install the firmware onto the TFTP server, at that system's # prompt, type the following:**

```
# cd /tftp-root-dir
# mkdir firmware
# cp SunFireB1600-sc-vxxxx.flash /tftp-root-dir/firmware
# chmod 444 /tftp-root-dir/firmware/SunFireB1600-sc-vxxxx.flash

# cd bsc-firmware-patch-dir
# cp SunFireB100x-bsc-vxxxx.flash /tftp-root-dir/firmware
# chmod 444 /tftp-root-dir/SunFireB100x-bsc-vxxxx.flash
```

where:

- *vxxxx* is the version of the firmware,
- *tftp-root-dir* is the TFTP root directory on the TFTP server. On Linux systems this directory is called `/tftp`, and on Solaris systems it is called `/tftpboot`.
- *sc-firmware-patch-dir* is the directory into which you unpacked the contents of the System Controller firmware packages.
- *switch-firmware-patch-dir* is the directory into which you unpacked the contents of the switch firmware packages.
- *bsc-firmware-patch-dir* is the directory into which you unpacked the contents of the BSC firmware packages. Note that this example shows the location of BSC firmware for a B100x server blade.

A.3 Upgrading the System Controller Firmware

Note – You must have a-level user privileges to perform an update of the System Controller firmware. For information about the levels of user permission that are available, see the *Sun Fire B1600 Blade System Chassis Administration Guide*.

Note – To make the standby System Controller take over as the active one so that you can upgrade the firmware on it, use the `setfailover` command. For information, see Step 7.

To perform the upgrade, do the following:

1. **Check the current version of the System Controller firmware.**

   Type:

```
sc>showsc

Sun Advanced Lights Out Manager for Blade Servers 1.1
Copyright 2003 Sun Microsystems, Inc. All Rights Reserved.
ALOM-B 1.1

Release: 1.1.8

Parameter                         Running Value      Stored Value
----------------------------------------------------------------
Bootable Image :                1.0.97 (Jan 06 03)
Current Running Image :         1.0.97 (Jan 06 03)
...
```

   The current version of the System Controller firmware appears in the line labeled "Current Running Image".

2. **Read the patch README file supplied with the System Controller firmware image and note the version of the firmware it describes.**

   Also note any special instructions and cautions.

3. **Establish that the upgrade is necessary.**

   If the current System Controller firmware revision matches the version numbers listed in the patch README file, the upgrade is not necessary for this System Controller.

   If the current System Controller firmware revision is lower than the latest firmware revision specified in the patch README file, proceed to Step 4.

4. **At the** sc> **prompt, type:**

```
sc> flashupdate -s ipaddress -f  path/filename [-v] [-y] sscn/sc
```

   where:

   path specifies the path of the new firmware you intend to download,

   filename specifies the filename of the new firmware you intend to download,

   ipaddress specifies the IP address of the computer on which the new firmware is stored (in other words, of the TFTP server),

   n is either 0 or 1 depending on whether you are downloading new firmware onto SSC0 or SSC1,

   and where the -v (verbose) option displays detailed screen output to enable you to observe the progress of the firmware update, and the -y option causes the update command to execute without prompting you for confirmation to proceed.

   For example:

```
sc> flashupdate -s 129.156.237.102 -f /firmware/SunFireB1600-
sc-vxxxx.flash -v -y sscn/sc
```

5. **When the update operation has completed, you must reset the System Controller for the new firmware to come into use.**

   Type:

```
sc> resetsc -y
```

   where the -y option causes the System Controller to reset without prompting you for confirmation to proceed.

6. **Confirm that the System Controller is now running the new firmware.**

   Type:

```
sc>showsc

Sun Advanced Lights Out Manager for Blade Servers 1.2
Copyright 2003 Sun Microsystems, Inc. All Rights Reserved.
ALOM-B 1.2

Release: 1.2.1

Parameter                          Running Value     Stored Value
----------------------------------------------------------------
Bootable Image :                       1.2.1 (May 29 03)
Current Running Image :                1.2.1 (May 29 03)
```

7. **To upgrade the firmware on the standby System Controller, you must first make the standby System Controller take over from the active System Controller:**

   - **At the** sc> **prompt, type**:

```
sc> setfailover
SSC0 is in Active Mode
SSC1 is in Standby Mode.
Are you sure you want to failover to SSC1?
All connections and user sessions will now be lost on SSC0 (y/n)? y


System Controller in SSC0 is now in Standby mode
```

   - **To check which System Controller is active, type**:

```
sc> setfailover
SSC0 is in Standby Mode
SSC1 is in Active Mode.
Are you sure you want to failover to SSC1?
All connections and user sessions will now be lost on SSC0 (y/n)? n
sc>
```

8. **Repeat Step 1 through Step 6 above.**

A.3.1 Example for Upgrading the System Controller Firmware

- To download a new image (called `SunFireB1600-sc-v1.1.8.flash`) onto the System Controller in SSC0 from a TFTP server whose IP address is 129.156.237.102, you would need to type the following at the SC's command line:

```
sc> flashupdate -s 129.156.237.102 -f /firmware/SunFireB1600-sc-
v1.1.8.flash ssc0/sc
Warning: Are you sure you want to update the flash image (y/n)? y
Erasing segment 2f Programming address ffaeffef
Update of SSC0/SC complete.
The system must be reset (using resetsc) for the new image to be
loaded
sc> resetsc -y
```

A.4 Upgrading the Blade Support Chip Firmware on One or More Blades

1. **Check the current version of the blades' BSC firmware.**

   The current version of the firmware running on each blade is listed at the end of the output from the `showsc -v` command. Type:

```
sc>showsc -v

Sun Advanced Lights Out Manager for Blade Servers 1.2
Copyright 2003 Sun Microsystems, Inc. All Rights Reserved.
ALOM-B 1.2

Release: 1.2.1
:
:
FRU   Software Version            Software Release Date
--------------------------------------------------------
S0    v5.1.0-SUNW,Sun-Fire-B100x  Jun  5 2003 10:27:31
S1    v5.1.0-SUNW,Sun-Fire-B100x  Jun  5 2003 10:27:31
S2    v5.1.0-SUNW,Sun-Fire-B200x  Jun  5 2003 10:27:31
S4    v5.1.0-SUNW,Sun-Fire-B200x  Jun  5 2003 10:27:31
S6    v4.1.1-SUNW,Sun-Fire-B200x  May 27 2003 10:36:23
S8    v4.1.1-SUNW,Sun-Fire-B200x  May 27 2003 10:36:23
:
:
S15   v5.1.0-SUNW,Sun-Fire-B100x  Jun  5 2003 10:27:31
```

   (Note that the : character indicates omitted information.)

2. **Read the patch README file supplied with the BSC firmware image and note the version of the firmware that it describes.**

   Also note any special instructions and cautions.

3. **Establish that the upgrade is necessary.**

   If the current BSC firmware revision for a blade matches the version numbers given in the patch README file, the upgrade is not necessary for that blade.

   If the current BSC firmware revision is lower than the latest firmware revision specified in the patch README file, proceed to Step 4.

4. **At the** `sc>` **prompt, type:**

```
sc> flashupdate [-v] [-y] -s ipaddress -f  path sn [sn...]
```

where:

the `-v` (verbose) option displays detailed screen output to enable you to observe the progress of the firmware update, and the `-y` option causes the update command to execute without prompting you for confirmation to proceed.

ipaddress specifies the IP address of the computer on which the new firmware is stored (in other words, of the TFTP server),

path specifies the path and filename of the new firmware you intend to download,

n specifies the blade whose firmware you want to upgrade,

and where **[s*n*...]** indicates an optional space-separated list of blades to be updated.

5. **Check that the new firmware is running on the blades.**

To do this, repeat Step 1 to see an updated list of the firmware on the blades.

A.4.1 Example of Upgrading Firmware on a Single Blade

- To download a new image (called `SunFireB100x-bsc-v5.0.0.flash`) onto the blade in slot 3 from the firmware directory on a TFTP server whose IP address was 129.156.237.102, you would need to type:

```
sc> flashupdate -s 129.156.237.102 -f /firmware/SunFireB100x-bsc-
v5.0.0.flash s3
Warning: Are you sure you want to update S3 bsc image;
all console connections to the fru will be reset (y/n)? y
131072 bytes of 131072 completed on S3
Update of S3 complete
sc>
```

A.4.2 Examples for Upgrading Firmware on a Number of Blades

- To download a new image (called `SunFireB100x-bsc-v5.0.0.flash`) onto the blades in slots 5, 10, and 13 from a TFTP server whose IP address was 129.156.237.102, you would need to type:

```
sc> flashupdate -s 129.156.237.102 -f /firmware/SunFireB1600x-bsc-
v5.0.0.flash s5 s10 s13
Warning: Are you sure you want to update s5 bsc image;
all console connections to s5 will be reset (y/n)? y
131072 bytes of 131072 completed on s5
Update of s5 complete
Warning: Are you sure you want to update s10 bsc image;
all console connections to s10 will be reset. (y/n)? y
131072 bytes of 131072 completed on s10
Update of s10 complete
Warning: Are you sure you want to update s13 bsc image;
all console connections to s13 will be reset (y/n)? y
131072 bytes of 131072 completed on s13
Update of s13 complete
sc>
```

APPENDIX **B**

Monitoring Components

This chapter contains the following sections:

B.1 Introduction

The System Controller's command-line interface includes commands that provide global information about the chassis and its components. These are the `showsc`, `showplatform`, `showenvironment`, and `showfru` commands.

- `showsc` tells you the current state of the System Controller's configurable parameters.
- `showdate` shows you date and time settings for the System Controller.
- `showplatform` tells you the status (Ok, Faulty, Not Present) of each component (it can also tell you the MAC address of each component).
- `showenvironment` provides information about the operational state of the components in the chassis (for example, it tells you the internal temperatures, the speed of the fans, and the level of current on the supply rails).
- `showfru` provides information stored by each component about itself. This information includes static data (for example, hardware version information) and dynamic data (for example, recent events generated by the component).

This chapter tells you how to use these commands to monitor a blade in your chassis. For full information on monitoring the components in a chassis, see the *Sun Fire B1600 Blade System Chassis Administration Guide*.

B.2 Viewing the System Controller Details

When you run the showsc command, all of the configurable properties of the System Controller are listed. For example:

```
sc> showsc

Sun Advanced Lights Out Manager for Blade Servers 1.2
Copyright 2003 Sun Microsystems, Inc. All Rights Reserved.
ALOM-B 2.1

Release: 1.2.1

Parameter                              Running Value      Stored Value
-----------------------------------------------------------------------
Bootable Image :                       0.2.0 (Apr 04 03)
Current Running Image :                0.2.0 (Apr 04 03)
SC IP address:                         192.168.130.213    192.168.130.213
SC IP netmask address:                 255.255.255.0      255.255.255.0
SC IP gateway address:                 192.168.130.1      192.168.130.1
SSC0/SC (Active) IP private address:   192.168.130.212    192.168.130.212
SSC1/SC (Standby) IP private address:  192.168.130.152    192.168.130.152
SMS IP address:                        0.0.0.0            0.0.0.0
SC VLAN:                               Disabled           Disabled
SC DHCP:                               Disabled           Disabled
SC Network interface is:               Enabled            Enabled
SC Telnet interface is:                Enabled            Enabled
NTP:                                   Disabled           Disabled
Blade OS auto restart when hung:
S0                                     Disabled           Disabled
S1                                     Disabled           Disabled
S2                                     Disabled           Disabled
S3                                     Disabled           Disabled
Blade auto poweron:
S0                                     Disabled           Disabled
S1                                     Disabled           Disabled
S2                                     Disabled           Disabled
S3                                     Disabled           Disabled
The CLI prompt is set as:              sc>                sc>
Event Reporting via telnet interface:  Enabled            Enabled
The CLI event level is set as:         CRITICAL           CRITICAL
The CLI timeout (seconds) is set at:   0                  0
Mask password with *'s:                Disabled           Disabled
sc>
```

- **To view all of the above details plus the version number of the currently installed firmware on the server blades, use the** `-v` **option as follows:**

```
sc> showsc -v
:
FRU   Software Version             Software Release Date
--------------------------------------------------------
S0    v5.1.0-SUNW,Sun-Fire-B100x  Jun  5 2003 10:27:31
S1    Not Present
S2    v5.0.2-SUNW,Serverblade1    Jan 17 2003 11:03:37
S3    Not Present
S4    v5.0.2-SUNW,Serverblade1    Jan 17 2003 11:03:37
S5    v5.0.2-SUNW,Serverblade1    Jan 17 2003 11:03:37
S6    v5.0.2-SUNW,Serverblade1    Jan 17 2003 11:03:37
S7    Not Present
S8    v5.1.0-SUNW,Sun-Fire-B200x  Jun  5 2003 10:27:31
S10   v5.1.0-SUNW,Sun-Fire-B200x  Jun  5 2003 10:27:31
S12   Not Present
S13   v5.0.2-SUNW,Serverblade1    Jan 17 2003 11:03:37
S14   v5.1.0-SUNW,Sun-Fire-B100x  Jun  5 2003 10:27:31
S15   Not Present
S16   Not Present
sc>
```

where the : character indicates omitted data.

Note – B200x blades occupy two slots. The second of these two slots is not shown in the output.

B.3 Checking the Date and Time

Note – Users with any of the four levels of user permission on the System Controller can check the date and time on the System Controller by using the `showdate` command. For information about the levels of permission available, see the *Sun Fire B1600 Blade System Chassis Administration Guide*.

The server blades receive their time and date settings from the System Controller. The System Controller can receive its time settings from a time server (using Network Time Protocol), or you can set it yourself using the `setdate` command:

```
sc> setdate [mmdd]HHMM[.SS] | mmddHHMM[cc]yy[.SS]
```

where *mm* is the month (two digits), *dd* is the day (two digits), *HH* is the hour (two digits), *MM* is the minutes (two digits), *SS* is seconds (two digits), *cc* is the century (20), and *yy* is the year (two digits).

Note – When you set the date and time, you must use Co-ordinated Universal Time (UTC). The server blades work out the local time for your time-zone by using an offset from Co-ordinated Universal Time on the System Controller. They receive the time from the System Controller.

- **To check the date and time on the SC, type:**

```
sc> showdate
Wed Mar 27 11:42:40 UTC 2002
```

For information about setting the date and time, refer to the *Sun Fire B1600 Blade System Chassis Software Setup Guide*.

B.4 Checking the Status of the Hardware Components

Note – Users with any of the four levels of user permission on the System Controller can check the operational status of the hardware by using the `showplatform` command. For information about the levels of permission available, see the *Sun Fire B1600 Blade System Chassis Administration Guide*.

To check the operational status of the Switch and System Controllers, server blades, and Power Supply Units, type:

```
sc>showplatform -v

FRU       Status         Type          Part No.  Serial No.
--------  -------------  ------------  --------  ----------
S0        OK             SF B100x      5405548   000408
S1        OK             SF B100x      5405547   000261
S2        OK             SF B200x      5405526   000336
S4        OK             SF B200x      5405527   000122
S6        OK             SF B100x      5405078   000467
S7        Not Present    ***           ***       ***
S8        OK             SF B100x      5405547   000377
S9        Not Present    ***           ***       ***
S10       OK             SF B100x      5405526   240024
S12       Not Present    ***           ***       ***
S13       OK             SF B100x      5405078   000695
S14       OK             SF B100x      5405547   000455
S15       OK             SF B200x      5405537   000445
SSC0      OK              SF B1600 SSC   5405185   0004703-0309000
SSC0/SC
SSC0/SWT
SSC1      OK             SF B1600 SSC  5405185   00000000000000
SSC1/SC
SSC1/SW
PS0        OK             SF B1600 PSU  3001544   002555abcdef1234
PS1       OK              SF B1600 PSU  3001544   002555abcdef1234
CH        OK           SF B1600     5405082   000000

Domain    Status                MAC Address         Hostname
--------  -------------------  -----------------  ----------------
S0        OS Running            00:03:ba:29:ef:ce  local.locald>
S1        OS Running            00:03:ba:29:f1:be
S2        OS Running            00:03:ba:2d:d0:3c
S4        OS Running            00:03:ba:2e:19:40
:
SSC0/SWT  OS Running            00:03:ba:1b:71:ff
SSC1/SWT  OS Running            00:03:ba:1b:9c:3f
SSC0/SC   OS Running (Active)   00:03:ba:1b:72:18
SSC1/SC   OS Stopped            00:03:ba:1b:9c:58
sc>
```

where the : character indicates omitted data.

Note – B200x blades occupy two slots. The second of these two slots is not shown in the output.

Note – If you do not specify -v on the command line for this command, you will see only the operational status of each piece of hardware, not the MAC address.

B.5 Checking Operating Conditions Inside the Blades

You can use the showenvironment command to check the operating temperatures, the fans, and the voltage supply rails for each blade, switch, power supply unit, and SSC inside the chassis. The command also displays the warning and shutdown thresholds.

Note – Users with any of the four levels of user permission on the System Controller can check the health of the platform and its components by using the showenvironment command. For information about the levels of permission available, see the *Sun Fire B1600 Blade System Chassis Administration Guide*.

Checking a Server Blade or Server Blades

- **To check a single server blade type:**

```
sc> showenvironment sn
```

where *n* is the number of the slot containing the blade. For example:

```
sc> showenvironment s0

============ Environmental Status ============

System Temperatures (Celsius)   Current   Status
----------------------------------------------------------------
S0          /temp/enclosure     26        OK
S0          /temp/CPU die       48        OK

System Voltages (Volts)         Current   Status
----------------------------------------------------------------
S0          /VSensor/5V         100%      OK
S0          /VSensor/3V3        100%      OK
S0          /VSensor/2V5         99%      OK
S0          /VSensor/Vcore      100%      OK

System Fans (RPM)               Current   Status
----------------------------------------------------------------
S0          /fan/cpu_fan        100%      OK
sc>
```

- **To check a number of server blades, specify them in a space-separated list. For example:**

```
sc>showenvironment s0 s1 s2

============ Environmental Status ============

System Temperatures (Celsius)   Current   Status
----------------------------------------------------------------
S0         /temp/enclosure      26        OK
S0         /temp/CPU die        48        OK
S1         /temp/enclosure      26        OK
S1         /temp/CPU die        42        OK
S2         /temp/enclosure      27        OK
S2         /temp/CPU die        46        OK

System Voltages (Volts)         Current   Status
----------------------------------------------------------------
S0         /VSensor/5V          100%      OK
S0         /VSensor/3V3         100%      OK
S0         /VSensor/2V5          99%      OK
S0         /VSensor/Vcore       100%      OK
S1         /VSensor/5V          100%      OK
S1         /VSensor/3V3         100%      OK
S1         /VSensor/2V5          99%      OK
S1         /VSensor/Vcore       100%      OK
S2         /VSensor/5V           99%      OK
S2         /VSensor/3V3         100%      OK
S2         /VSensor/2V5          99%      OK
S2         /VSensor/Vcore        99%      OK

System Fans (RPM)               Current   Status
----------------------------------------------------------------
S0         /fan/cpu_fan         100%      OK
S1         /fan/cpu_fan         100%      OK
S2         /fan/cpu_fan         100%      OK
sc>
```

B.6 Checking the Information Stored by a Blade About Itself

You can use the showfru command to view a database of information stored by each component about itself.

Note – To use the showfru command, you need to have c-level user permission. For more information about permission levels, see the *Sun Fire B1600 Blade System Chassis Administration Guide*.

- **To view the information stored by a component about itself, do the following:**

```
sc> showfru FRU list
```

where *FRU list* is a single FRU or a space-separated list of FRUs. The FRUs can be ssc0, ssc1, ps0, ps1, or s*n* (where *n* is the number of the slot containing the blade).

For example, to see FRUID information about SSC0 and the blade in slot s0, you would type:

```
sc> showfru ssc0 s0
----------------------------------------------------------------
FRUID Records for FRU SSC0
----------------------------------------------------------------
/FRUID/ManR/UNIX_Timestamp32: Mon Oct 14 22:49:04 UTC 2002
/FRUID/ManR/Fru_Description: SUNW,Sun Fire B1600 SSC, 8x1GB NET,
1x10MB
NET MGT, 1 Serial MGT
/FRUID/ManR/Manufacture_Loc: Hsinchu, Taiwan
/FRUID/ManR/Sun_Part_No: 5405185
/FRUID/ManR/Sun_Serial_No:
:

----------------------------------------------------------------
FRUID Records for FRU S0
----------------------------------------------------------------
/FRUID/ManR/UNIX_Timestamp32: Sat Dec 21 06:24:58 UTC 2002
/FRUID/ManR/Fru_Description: SUNW,Sun Fire B100x, 1 CPU, 512MB,
30GB HDD
/FRUID/ManR/Manufacture_Loc: Hsinchu,Taiwan
/FRUID/ManR/Sun_Part_No: 5405547
/FRUID/ManR/Sun_Serial_No: 000075
:
sc>
```

where the : character on a line by itself indicates omitted data.

Index