BlueCat Linux Board Support Guide

BlueCat Linux 4.0

DOC-0551-00

for Motorola Sandpoint III Boards

Product names mentioned in *BlueCat Linux Board Support Guide for Motorola Sandpoint III Boards* are trademarks of their respective manufacturers and are used here for identification purposes only.

Printed in the United States of America.

Contents

CHAPTER 1 *Overview*

The *BlueCat Linux Board Support Guide for the Motorola Sandpoint III Board* provides information about the BlueCat Linux Board Support Package (BSP) for the Motorola Sandpoint III board hosting the Motorola PrPMC750 card.

Throughout this Board Support Guide (BSG), the BSP is referred to as the "sandpoint" and the board as the "Sandpoint", or simply as the "target board".

The chapters of this BSG provides the following information:

- *Chapter 1* is an overview of this BSG's individual chapters.
- *Chapter 2* describes BlueCat Linux downloading and booting procedures for the Sandpoint board using the BlueCat Linux `showcase` demo system as an example.
- *Chapter 3* provides configuration option information about the sandpoint BSP's default BlueCat Linux kernel.
- *Chapter 4* summarizes BlueCat Linux demo systems supported by the sandpoint BSP.
- *Chapter 5* lists the device drivers in the sandpoint BSP.
- *Chapter 6* provides known limitations and workarounds, as well as the defects fixed in this release.

CHAPTER 2 *Downloading and Booting BlueCat Linux on the Target*

This chapter provides instructions for downloading a BlueCat Linux demo system from a cross development host onto the target, and then booting the demo system on the target board.

Prerequisites

This document provides instructions on downloading and booting BlueCat Linux systems onto the Sandpoint target board. Scenarios detailing usage of the BlueCat Linux demo systems are also presented. A basic familiarity with the target board hardware and operation is required before using this guide. Users must also have an understanding of system administration for the particular cross development host on which BlueCat Linux and the Board Support Package (BSP) is installed. The manufacturer's documentation for the target board, as well as system administration reference material for the cross development host, should be readily available.

Before downloading and booting BlueCat Linux on the target board, it is assumed that the default BlueCat Linux PowerPC configuration and the sandpoint BSP have been installed on the cross development host. This means that the user must:

1. Install the BlueCat Linux PowerPC Core onto the cross development host, as described in the "Installing the Default Configuration" section in Chapter 1, "Installation" of the *BlueCat Linux User's Guide*.
2. Install the sandpoint BSP onto the cross development host as detailed in the "Installing Support for Target Boards" section of Chapter 1, "Installation" in the *BlueCat Linux User's Guide*.
3. Activate support for the sandpoint BSP as detailed in the "Activating Support for a Target Board" section of Chapter 1, "Installation" in the *BlueCat Linux User's Guide*.

Downloading and Booting Overview

The procedure for downloading and booting a BlueCat Linux system on the Sandpoint target consists of the following main steps:

- Setting up hardware
- Downloading and booting a BlueCat Linux system from a target Flash device or a network.

Downloading and booting a BlueCat Linux system can be performed using the LynuxWorks Boot Loader.The LynuxWorks Boot Loader is firmware that can boot BlueCat Linux from a Flash device or over a network, and to program them to a Flash device.

Setting up Hardware

Connecting Target Board Serial Ports to Host

The target board has two serial ports. The first port is used both by the LynuxWorks Boot Loader firmware and the BlueCat Linux system console.

Before using the board, at least the first serial port needs to be connected to the development host. It is recommended to connect the target serial connector to COM1 on the host.

The serial port settings on the host must be as follows:

- The serial port connected to the first target serial port has a baud rate of 9600
- The serial port connected to the second target serial port has a baud rate of 19200

Throughout this chapter, the terminal window connected to the first serial connector is referred to as the *Boot Loader* console or the *BlueCat Linux* console, depending on the context.

Connecting Target Board Ethernet Card to Host

The Ethernet port on the target board is used to provide a standard network connection for the board, and, in particular, to load BlueCat Linux embedded systems onto the board over a network.

The Ethernet port on the target board is used to connect to a LAN.

It is also required to set up networking on the host system. In particular, the user must choose a unique IP address for the development host, as well as for the target board. These addresses are referred to as `<host_IP>` and `<target_IP>`, respectively. For more information on how to set up networking on the host, please refer to the host operating system documentation.

TFTP must be enabled on the host. Refer to "Setting Up a TFTP Server" in Chapter 3 of the *BlueCat Linux User's Guide* for more information.

Programming the Boot Loader onto the Sandpoint Board

To burn the LynuxWorks Boot Loader firmware to the Sandpoint board, a 3rd party flash burning utility must be used. For example, a JTAG downloading tool or Flash device programmer can be used.

The Boot Loader includes the file `package.rom`. This file contains the Boot Loader image that should be burned onto the target board.

Setting up the LynuxWorks Boot Loader Firmware

Use the following procedure to set up the LynuxWorks Boot Loader firmware options for BlueCat Linux:

1. Reset the target board.
2. At the Boot Loader console, enter

   ```
   sandpoint> set autoboot 0
   sandpoint> set boot_tftp_host_ip <host_IP>
   sandpoint> set boot_tftp_client_ip <target_IP>
   sandpoint> save
   ```

   Where `<target_IP>` is the IP address of the target, `<host_IP>` is the IP address of the development host.
3. Reset the target board.

Target Board Switches

Before starting the download process, the target board configuration switches SW1 and SW2 must be set as shown in Table 2-1 and Table 2-2. Consult the manufacturer's documentation to determine the location of these switches on the target board.

Table 2-1: SW1 Settings

Switch	Definition	Value
1	ROM selection	ON
2	ROM1 write protect	OFF
3	reserved	ON
4	Force PCI to 33MHz	OFF
5	Use external clock	ON
6	Spread-Spectrum Clock	ON
7	Spread-Spectrum Range	ON
8	Force Power Supply ON	OFF

Table 2-2: SW2 Settings

Switch	Definition	Value
1 & 2	PCI Arbiter Architecture	ON
3	Interrupt Legacy Modes	OFF
4 & 5	Interrupt Architecture	OFF
6	ROM mode	ON
7	User Options	OFF
8	User Options	ON

Downloading a BlueCat Linux System to a Flash Device

This section provides instructions on how a BlueCat Linux embedded system can be downloaded into the target Flash device using the Boot Loader firmware or the BlueCat Linux OS loader.

NOTE: These instructions assume the presence of a secondary Flash device on the target board.

Specifically, these instructions are applicable to any of the available demo systems. This chapter uses the `osloader` demo system as an example.

Downloading a BlueCat Linux System to a Flash Device Using the Boot Loader

The following procedure is used to download `showcase` into the target board using the Boot Loader:

1. Copy the `osloader.kdi` file from the `$BLUECAT_PREFIX/demo/osloader` directory to the `/tftpboot` directory on the development host.
2. Reset the target board.
3. At the Boot Loader console, enter the following commands:

```
sandpoint> set flash_device tftp
sandpoint> set flash_tftp_file osloader.kdi
sandpoint> set flash_target flash1
sandpoint> set flash_offset 0
sandpoint> flash
```

   The `osloader` demo system is now programmed into the secondary Flash and can be booted as described in “Booting a Demo System from a Flash Device” below.

Booting a Demo System from a Flash Device

The following procedure is used to boot the `osloader` demo installed into the Flash device. For a detailed information on how to install the demo system to a Flash device, refer to “Downloading a BlueCat Linux System to a Flash Device” on page 7.

The following examples are for the LynuxWorks Boot Loader. For information on using and configuring the Boot Loader, see the *LynuxWorks Boot Loader User's Guide*.

1. Reset the target board.
2. At the Boot Loader console, type the following:

```
sandpoint> set boot_device flash1
sandpoint> set boot_flash_offset 0
sandpoint> set boot_os BlueCat
sandpoint> boot
```

   These commands start the `osloader` demo system programmed into the secondary Flash deviced.

The Sandpoint board can be configured to start a demo system programmed into a Flash device automatically when the board powers-up. Use the following commands to prepare the Sandpoint board to boot BlueCat Linux from a Flash device automatically:

```
sandpoint> set boot_device flash1
sandpoint> set boot_flash_offset 0
sandpoint> set boot_os BlueCat
sandpoint> set autoboot 1
sandpoint> save
```

As a result, the demo system programmed into Flash will be started by the Boot Loader monitor automatically on board power-up.

Booting a Demo System from a Network

A BlueCat Linux demo system can be booted from a network using the Boot Loader firmware.

Booting a Demo from the Network Using Boot Loader

The following demonstrates booting the `showcase` demo system over a network using the Boot Loader firmware. For information on using and configuring the Boot Loader, see the *LynuxWorks Boot Loader User's Guide*.

1. Copy the `showcase.kdi` file from the `$BLUECAT_PREFIX/demo/showcase` directory to the `/tftpboot` directory on the cross development host.
2. Reset the target board.
3. At the Boot Loader console, enter the following commands:

```
sandpoint> set boot_device tftp
sandpoint> set boot_tftp_file showcase.kdi
sandpoint> boot
```

These commands load the `showcase` demo system from a network onto the target board and then automatically start it.

CHAPTER 3 *Kernel Configuration Options*

The sandpoint BSP includes a default BlueCat Linux kernel with a number of configuration options. The following tables describe these options.

Table 3-1: BlueCat Linux Default Configuration for the sandpoint BSP

Table Number and Configuration Parameter
Table 3-2: "Code Maturity Level Options"
Table 3-3: "Loadable Module Support"
Table 3-4: "Platform Support"
Table 3-5: "General Setup"
Table 3-6: "Parallel Port Support"
Table 3-7: "Memory Technology Devices (MTD)"
Table 3-8: "RAM/ROM/Flash device Drivers"
Table 3-9: "Mapping Drivers for Chip Access"
Table 3-10: "Self-contained MTD Device Drivers"
Table 3-11: "NAND Flash Device Drivers"
Table 3-12: "Plug and Play Configuration"
Table 3-13: "Block Devices"
Table 3-14: "Multi-Device Support (RAID and LVM)"
Table 3-15: "Networking Options"
Table 3-16: "IP: Netfilter Configuration"
Table 3-17: "QoS and/or Fair Queueing"
Table 3-18: "ATA/IDE/MFM/RLL Support"
Table 3-19: "IDE, ATA And ATAPI Block Devices"

Table 3-1: BlueCat Linux Default Configuration for the sandpoint BSP

Table Number and Configuration Parameter
Table 3-20: "SCSI Support"
Table 3-21: "IEEE 1394 (FireWire) Support (Experimental)"
Table 3-22: "Network Device Support"
Table 3-23: "ARCnet Devices"
Table 3-24: "Appletalk Devices"
Table 3-25: "Ethernet (10 or 100Mbit)"
Table 3-26: "Ethernet (1000 Mbit)"
Table 3-27: "Wireless LAN (non-hamradio)"
Table 3-28: "Token Ring Devices"
Table 3-29: "Wan interfaces"
Table 3-30: "Amateur Radio Support"
Table 3-31: "IrDA (infrared) Support"
Table 3-32: "ISDN Subsystem"
Table 3-33: "Old CD-ROM drivers (not SCSI, not IDE)"
Table 3-34: "Console Drivers"
Table 3-35: "Frame-Buffer Support"
Table 3-36: "Input Core Support"
Table 3-37: "Character Devices"
Table 3-38: "Serial Drivers"
Table 3-39: "I2C Support"
Table 3-40: "L3 Serial Bus Support"
Table 3-41: "Mice"
Table 3-42: "Watchdog Cards"
Table 3-43: "Ftape, the Floppy Tape Device Driver"
Table 3-44: "Multimedia Devices"
Table 3-45: "File Systems"
Table 3-46: "Network File Systems"

Table 3-1: BlueCat Linux Default Configuration for the sandpoint BSP

Table Number and Configuration Parameter
Table 3-47: "Partition Types"
Table 3-48: "Sound"
Table 3-49: "USB Support"
Table 3-50: "USB Serial Converter Support"
Table 3-51: "Bluetooth Support"
Table 3-52: "Kernel Hacking"
Table 3-53: "Modular Advanced Power Management"

Table 3-2: Code Maturity Level Options

Option	Value	Description
CONFIG_EXPERIMENTAL	Y	Prompt for development and/or incomplete code/drivers

Table 3-3: Loadable Module Support

Option	Value	Description
CONFIG_MODULES	Y	Enable loadable module support
CONFIG_MODVERSIONS	Y	Set version information on all module symbols
CONFIG_KMOD	Y	Kernel module loader

Table 3-4: Platform Support

Option	Value	Description
CONFIG_6xx	Y	Processor Type
CONFIG_8260	N	MPC8260 CPM Support

Table 3-4: Platform Support

Option	Value	Description
CONFIG_SANDPOINT	Y	Machine Type
CONFIG_MPC10X_STORE_GATHERING	Y	Enable MPC10x store gathering
CONFIG_SMP	N	Symmetric multi-processing support
CONFIG_ALTIVEC	N	AltiVec Support
CONFIG_TAU	N	Thermal Management Support

Table 3-5: General Setup

Option	Value	Description
CONFIG_HIGHMEM	N	High memory support (Experimental)
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat Ignore `printk`
CONFIG_BLUECAT_LOADER	N	BlueCat OS Loader
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat small memory footprint
CONFIG_NET	Y	Networking support
CONFIG_BLUECAT_MEMSIZE	N	Memory sizing benchmarks
CONFIG_SYSCTL	Y	Sysctl support
CONFIG_SYSVIPC	Y	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_BINFMT_MISC	N	Kernel support for MISC binaries
CONFIG_PCI_NAMES	Y	PCI device name database
CONFIG_HOTPLUG	N	Support for hot-pluggable devices
CONFIG_PPC_RTC	Y	Support for `/dev/rtc`
CONFIG_CMDLINE_BOOL	N	Default bootloader kernel arguments

Table 3-6: Parallel Port Support

Option	Value	Description
CONFIG_PARPORT	Y	Parallel port support
CONFIG_PARPORT_PC	Y	PC-style hardware
CONFIG_PARPORT_SERIAL	N	Multi-IO cards (parallel and serial)
CONFIG_PARPORT_PC_FIFO	N	Use FIFO/DMA if available (Experimental)
CONFIG_PARPORT_PC_SUPERIO	N	SuperIO chipset support (Experimental)
CONFIG_PARPORT_OTHER	N	Support foreign hardware
CONFIG_PARPORT_1284	Y	IEEE 1284 transfer modes

Table 3-7: Memory Technology Devices (MTD)

Option	Value	Description
CONFIG_MTD	Y	Memory Technology Device (MTD) support
CONFIG_MTD_DEBUG	N	Debugging
CONFIG_MTD_PARTITIONS	Y	MTD partitioning support
CONFIG_MTD_REDBOOT_PARTS	N	RedBoot partition table parsing
CONFIG_MTD_CHAR	Y	Direct char device access to MTD devices
CONFIG_MTD_BLOCK	Y	Caching block device access to MTD devices
CONFIG_FTL	N	FTL (Flash Translation Layer) support
CONFIG_NFTL	N	NFTL (NAND Flash Translation Layer) support

Table 3-8: RAM/ROM/Flash device Drivers

Option	Value	Description
CONFIG_MTD_CFI	N	Detect Flash devices by Common Flash Interface (CFI) probe
CONFIG_MTD_JEDECPROBE	Y	Detect non-CFI AMD/JEDEC-compatible Flash devices
CONFIG_MTD_CFI_ADV_OPTIONS	N	Flash device driver advanced configuration options
CONFIG_MTD_CFI_INTELEXT	N	Support for Intel/Sharp Flash devices
CONFIG_MTD_CFI_AMDSTD	Y	Support for AMD/Fujitsu Flash devices
CONFIG_MTD_RAM	N	Support for RAM chips in bus mapping
CONFIG_MTD_ROM	N	Support for ROM chips in bus mapping
CONFIG_MTD_ABSENT	N	Support for absent chips in bus mapping
CONFIG_MTD_OBSOLETE_CHIPS	N	Older (theoretically obsoleted now) drivers for non-CFI chips

Table 3-9: Mapping Drivers for Chip Access

Option	Value	Description
CONFIG_MTD_SANDPOINT	Y	Flash device mapping on Sandpoint board
CONFIG_MTD_SANDPOINT_PART	:	Partitions layout
CONFIG_MTD_PCI	N	PCI MTD driver

Table 3-10: Self-contained MTD Device Drivers

Option	Value	Description
CONFIG_MTD_PMC551	N	Ramix PMC551 PCI Mezzanine RAM card support
CONFIG_MTD_SLRAM	N	Uncached system RAM
CONFIG_MTD_MTDRAM	N	Test driver using RAM
CONFIG_MTD_BLKMTD	N	MTD emulation using block device
CONFIG_MTD_DOC1000	N	M-Systems Disk-On-Chip 1000
CONFIG_MTD_DOC2000	N	M-Systems Disk-On-Chip 2000 and Millennium
CONFIG_MTD_DOC2001	N	M-Systems Disk-On-Chip Millennium-only alternative driver (see help)

Table 3-11: NAND Flash Device Drivers

Option	Value	Description
CONFIG_MTD_NAND	N	NAND Device Support

Table 3-12: Plug and Play Configuration

Option	Value	Description
CONFIG_PNP	N	Plug and Play support

Table 3-13: Block Devices

Option	Value	Description
CONFIG_BLK_DEV_FD	N	Normal PC floppy disk support
CONFIG_PARIDE	N	Parallel port IDE device support

Table 3-13: Block Devices (Continued)

Option	Value	Description
CONFIG_BLK_CPQ_DA	N	Compaq SMART2 support
CONFIG_BLK_CPQ_CISS_DA	N	Compaq Smart Array 5xxx support
CONFIG_BLK_DEV_DAC960	N	Mylex DAC960/DAC1100 PCI RAID Controller support
CONFIG_BLK_DEV_LOOP	N	Loopback device support
CONFIG_BLK_DEV_NBD	N	Network block device support
CONFIG_BLK_DEV_RAM	Y	RAM disk support
CONFIG_BLK_DEV_RAM_SIZE	4096	Default RAM disk size
CONFIG_BLK_DEV_INITRD	N	Initial RAM disk (`initrd`) support
CONFIG_BLUECAT_RFS	Y	BlueCat RFS support

Table 3-14: Multi-Device Support (RAID and LVM)

Option	Value	Description
CONFIG_MD	N	Multiple devices driver support (RAID and LVM)
CONFIG_MD_LINEAR	N	Linear (append) mode
CONFIG_MD_RAID0	N	RAID-0 (striping) mode
CONFIG_MD_RAID1	N	RAID-1 (mirroring) mode
CONFIG_MD_RAID5	N	RAID-4/RAID-5 mode
CONFIG_MD_MULTIPATH	N	Multipath I/O support

Table 3-15: Networking Options

Option	Value	Description
CONFIG_PACKET	Y	Packet socket
CONFIG_PACKET_MMAP	N	Packet socket: mmapped IO

Table 3-15: Networking Options (Continued)

Option	Value	Description
CONFIG_NETLINK	Y	Kernel/User `netlink` socket
CONFIG_RTNETLINK	N	Routing messages
CONFIG_NETLINK_DEV	N	Netlink device emulation
CONFIG_NETFILTER	Y	Network packet filtering (replaces `ipchains`)
CONFIG_NETFILTER_DEBUG	N	Network packet filtering debugging
CONFIG_FILTER	N	Socket Filtering
CONFIG_UNIX	Y	Unix domain sockets
CONFIG_INET	Y	TCP/IP networking
CONFIG_IP_MULTICAST	Y	IP: multicasting
CONFIG_IP_ADVANCED_ROUTER	N	IP: advanced router
CONFIG_IP_PNP	N	IP: kernel level autoconfiguration
CONFIG_NET_IPIP	N	IP: tunneling
CONFIG_NET_IPGRE	N	IP: GRE tunnels over IP
CONFIG_IP_MROUTE	N	IP: multicast routing
CONFIG_INET_ECN	N	IP: TCP Explicit Congestion Notification support
CONFIG_SYN_COOKIES	Y	IP: TCP syncookie support (disabled per default)
CONFIG_IPV6	N	The IPv6 protocol (Experimental)
CONFIG_KHTTPD	N	Kernel httpd acceleration (Experimental)
CONFIG_ATM	N	Asynchronous Transfer Mode (ATM) (Experimental)
CONFIG_IPX	N	The IPX protocol
CONFIG_ATALK	N	Appletalk protocol support
CONFIG_DECNET	N	DECnet Support
CONFIG_BRIDGE	N	802.1d Ethernet Bridging
CONFIG_X25	N	CCITT X.25 Packet Layer (Experimental)

Table 3-15: Networking Options (Continued)

Option	Value	Description
CONFIG_LAPB	N	LAPB Data Link Driver (Experimental)
CONFIG_LLC	N	802.2 LLC (Experimental)
CONFIG_NET_DIVERT	N	Frame Diverter (Experimental)
CONFIG_ECONET	N	Acorn Econet/AUN protocols (Experimental)
CONFIG_WAN_ROUTER	N	WAN router
CONFIG_NET_FASTROUTE	N	Fast switching (read help!)
CONFIG_NET_HW_FLOWCONTROL	N	Forwarding between high speed interfaces

Table 3-16: IP: Netfilter Configuration

Option	Value	Description
CONFIG_IP_NF_CONNTRACK	M	Connection tracking (required for masq/NAT)
CONFIG_IP_NF_FTP	M	FTP protocol support
CONFIG_IP_NF_IRC	N	IRC protocol support
CONFIG_IP_NF_QUEUE	N	Userspace queueing via NETLINK (Experimental)
CONFIG_IP_NF_IPTABLES	M	IP tables support (required for filtering/masq/NAT)
CONFIG_IP_NF_MATCH_LIMIT	M	limit match support
CONFIG_IP_NF_MATCH_MAC	M	MAC address match support
CONFIG_IP_NF_MATCH_MARK	M	netfilter MARK match support
CONFIG_IP_NF_MATCH_MULTIPORT	M	Multiple port match support
CONFIG_IP_NF_MATCH_TOS	M	TOS match support
CONFIG_IP_NF_MATCH_TCPMSS	M	tcpmss match support
CONFIG_IP_NF_MATCH_STATE	M	Connection state match support
CONFIG_IP_NF_MATCH_UNCLEAN	M	Unclean match support (Experimental)

Table 3-16: IP: Netfilter Configuration (Continued)

Option	Value	Description
CONFIG_IP_NF_MATCH_OWNER	M	Owner match support (Experimental)
CONFIG_IP_NF_FILTER	M	Packet filtering
CONFIG_IP_NF_TARGET_REJECT	M	REJECT target support
CONFIG_IP_NF_TARGET_MIRROR	M	MIRROR target support (Experimental)
CONFIG_IP_NF_NAT	M	Full NAT
CONFIG_IP_NF_TARGET_MASQUERADE	M	MASQUERADE target support
CONFIG_IP_NF_TARGET_REDIRECT	M	REDIRECT target support
CONFIG_IP_NF_MANGLE	N	Packet mangling
CONFIG_IP_NF_TARGET_LOG	N	LOG target support
CONFIG_IP_NF_TARGET_TCPMSS	M	TCPMSS target support
CONFIG_IP_NF_COMPAT_IPCHAINS	M	ipchains (2.2-style) support
CONFIG_IP_NF_COMPAT_IPFWADM	N	ipfwadm (2.0-style) support

Table 3-17: QoS and/or Fair Queueing

Option	Value	Description
CONFIG_NET_SCHED	N	QoS and/or fair queueing

Table 3-18: ATA/IDE/MFM/RLL Support

Option	Value	Description
CONFIG_IDE	Y	ATA/IDE/MFM/RLL support

Table 3-19: IDE, ATA And ATAPI Block Devices

Option	Value	Description
CONFIG_BLK_DEV_IDE	Y	Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
CONFIG_BLK_DEV_HD_IDE	N	Use old disk-only driver on primary interface
CONFIG_BLK_DEV_IDEDISK	Y	Include IDE/ATA-2 DISK support
CONFIG_IDEDISK_MULTI_MODE	N	Use multi-mode by default
CONFIG_BLK_DEV_IDECD	Y	Include IDE/ATAPI CDROM support
CONFIG_BLK_DEV_IDETAPE	Y	Include IDE/ATAPI TAPE support
CONFIG_BLK_DEV_IDEFLOPPY	Y	Include IDE/ATAPI FLOPPY support
CONFIG_BLK_DEV_CMD640	N	CMD640 chipset bugfix/support
CONFIG_BLK_DEV_ISAPNP	N	ISA-PNP EIDE support
CONFIG_BLK_DEV_RZ1000	N	RZ1000 chipset bugfix/support
CONFIG_BLK_DEV_IDEPCI	Y	Generic PCI IDE chipset support
CONFIG_IDEPCI_SHARE_IRQ	Y	Sharing PCI IDE interrupts support
CONFIG_BLK_DEV_IDEDMA_PCI	Y	Generic PCI bus-master DMA support
CONFIG_BLK_DEV_OFFBOARD	N	Boot off-board chipsets first support
CONFIG_IDEDMA_PCI_AUTO	Y	Use PCI DMA by default when available
CONFIG_IDEDMA_PCI_WIP	N	ATA Work(s) In Progress (Experimental)
CONFIG_BLK_DEV_AEC62XX	N	AEC62XX chipset support
CONFIG_BLK_DEV_ALI15X3	N	ALI M15x3 chipset support
CONFIG_BLK_DEV_AMD74XX	N	AMD Viper support
CONFIG_BLK_DEV_CMD64X	N	CMD64X chipset support
CONFIG_BLK_DEV_CY82C693	N	CY82C693 chipset support
CONFIG_BLK_DEV_CS5530	N	Cyrix CS5530 MediaGX chipset support
CONFIG_BLK_DEV_HPT34X	N	HPT34X chipset support
CONFIG_BLK_DEV_HPT366	N	HPT366 chipset support

Table 3-19: IDE, ATA And ATAPI Block Devices (Continued)

Option	Value	Description
CONFIG_BLK_DEV_NS87415	N	NS87415 chipset support (Experimental)
CONFIG_BLK_DEV_OPTI621	N	OPTi 82C621 chipset enhanced support (Experimental)
CONFIG_BLK_DEV_PDC202XX	N	PROMISE PDC202{46|62|65|67|68} support
CONFIG_BLK_DEV_SVWKS	N	ServerWorks OSB4/CSB5 chipsets support
CONFIG_BLK_DEV_SIS5513	N	SiS5513 chipset support
CONFIG_BLK_DEV_SLC90E66	N	SLC90E66 chipset support
CONFIG_BLK_DEV_TRM290	N	Tekram TRM290 chipset support (Experimental)
CONFIG_BLK_DEV_VIA82CXXX	N	VIA82CXXX chipset support
CONFIG_BLK_DEV_SL82C105	N	Winbond SL82c105 support
CONFIG_IDE_CHIPSETS	N	Other IDE chipset support
CONFIG_IDEDMA_IVB	N	IGNORE word93 Validation BITS
CONFIG_BLK_DEV_ATARAID	N	Support for IDE Raid controllers

Table 3-20: SCSI Support

Option	Value	Description
CONFIG_SCSI	N	SCSI support

Table 3-21: IEEE 1394 (FireWire) Support (Experimental)

Option	Value	Description
CONFIG_IEEE1394	N	IEEE 1394 (FireWire) support (Experimental)

Table 3-22: Network Device Support

Option	Value	Description
CONFIG_NETDEVICES	Y	Network device support

Table 3-23: ARCnet Devices

Option	Value	Description
CONFIG_ARCNET	N	ARCnet support

Table 3-24: Appletalk Devices

Option	Value	Description
CONFIG_APPLETALK	N	Appletalk interfaces support
CONFIG_DUMMY	N	Dummy net driver support
CONFIG_BONDING	N	Bonding driver support
CONFIG_EQUALIZER	N	EQL (serial line load balancing) support
CONFIG_TUN	N	Universal TUN/TAP device driver support
CONFIG_ETHERTAP	N	Ethertap network tap (OBSOLETE)

Table 3-25: Ethernet (10 or 100Mbit)

Option	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100Mbit)
CONFIG_MACE	N	MACE (Power Mac ethernet) support
CONFIG_BMAC	N	BMAC (G3 ethernet) support
CONFIG_GMAC	N	GMAC (G4/iBook ethernet) support

Table 3-25: Ethernet (10 or 100Mbit) (Continued)

Option	Value	Description
CONFIG_OAKNET	N	National DP83902AV (Oak ethernet) support
CONFIG_HAPPYMEAL	N	Sun Happy Meal 10/100baseT support
CONFIG_SUNGEM	N	Sun GEM support
CONFIG_NET_VENDOR_3COM	N	3COM cards
CONFIG_NET_VENDOR_SMC	N	Western Digital/SMC cards
CONFIG_NET_VENDOR_RACAL	N	Racal-Interlan (Micom) NI cards
CONFIG_HP100	N	HP 10/100VG PCLAN (ISA, EISA, PCI) support
CONFIG_NET_PCI	Y	EISA, VLB, PCI and on board controllers
CONFIG_PCNET32	N	AMD PCnet32 PCI support
CONFIG_ADAPTEC_STARFIRE	N	Adaptec Starfire support (Experimental)
CONFIG_TULIP	N	DECchip Tulip (dc21x4x) PCI support
CONFIG_DE4X5	N	Generic DECchip & DIGITAL EtherWORKS PCI/EISA
CONFIG_DGRS	N	Digi Intl. RightSwitch SE-X support
CONFIG_DM9102	N	Davicom DM910x/DM980x support
CONFIG_EEPRO100	Y	EtherExpressPro/100 support
CONFIG_FEALNX	N	Myson MTD-8xx PCI Ethernet support
CONFIG_NATSEMI	N	National Semiconductor DP8381x series PCI Ethernet support
CONFIG_NE2K_PCI	N	PCI NE2000 and clones support (see help)
CONFIG_8139TOO	N	RealTek RTL-8139 PCI Fast Ethernet Adapter support
CONFIG_SIS900	N	SiS 900/7016 PCI Fast Ethernet Adapter support
CONFIG_EPIC100	N	SMC EtherPower II
CONFIG_SUNDANCE	N	Sundance Alta support
CONFIG_TLAN	N	TI ThunderLAN support

Table 3-25: Ethernet (10 or 100Mbit) (Continued)

Option	Value	Description
CONFIG_VIA_RHINE	N	VIA Rhine support
CONFIG_WINBOND_840	N	Winbond W89c840 Ethernet support
CONFIG_NET_POCKET	N	Pocket and portable adapters

Table 3-26: Ethernet (1000 Mbit)

Option	Value	
CONFIG_ACENIC	N	Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support
CONFIG_DL2K	N	D-Link DL2000-based Gigabit Ethernet support
CONFIG_NS83820	N	National Semiconduct DP83820 support
CONFIG_HAMACHI	N	Packet Engines Hamachi GNIC-II support
CONFIG_YELLOWFIN	N	Packet Engines Yellowfin Gigabit-NIC support (Experimental)
CONFIG_SK98LIN	N	SysKonnect SK-98xx support
CONFIG_FDDI	N	FDDI driver support
CONFIG_HIPPI	N	HIPPI driver support (Experimental)
CONFIG_PLIP	N	PLIP (parallel port support)
CONFIG_PPP	N	PPP (point-to-point protocol) support
CONFIG_SLIP	N	SLIP (serial line) support
CONFIG_NET_FC	N	Fibre Channel driver support
CONFIG_RCPCI	N	Red Creek Hardware VPN (Experimental)
CONFIG_SHAPER	N	Traffic Shaper (Experimental)

Table 3-27: Wireless LAN (non-hamradio)

Option	Value	Description
CONFIG_NET_RADIO	N	Wireless LAN (non-hamradio)

Table 3-28: Token Ring Devices

Option	Value	Description
CONFIG_TR	N	Token Ring driver support

Table 3-29: Wan interfaces

Option	Value	Description
CONFIG_WAN	N	Wan interfaces support

Table 3-30: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur Radio support

Table 3-31: IrDA (infrared) Support

Option	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

Table 3-32: ISDN Subsystem

Option	Value	Description
CONFIG_ISDN	N	ISDN support

Table 3-33: Old CD-ROM drivers (not SCSI, not IDE)

Option	Value	Description
CONFIG_CD_NO_IDESCSI	N	Support non-SCSI/IDE/ATAPI CDROM drives

Table 3-34: Console Drivers

Option	Value	Description
CONFIG_VGA_CONSOLE	N	Support for VGA Console

Table 3-35: Frame-Buffer Support

Option	Value	Description
CONFIG_FB	N	Support for frame buffer devices (Experimental)

Table 3-36: Input Core Support

Option	Value	Description
CONFIG_INPUT	N	Input core support
CONFIG_INPUT_MOUSEDEV_SCREEN_X	1024	Horizontal screen resolution
CONFIG_INPUT_MOUSEDEV_SCREEN_Y	768	Vertical screen resolution

Table 3-37: Character Devices

Option	Value	Description
CONFIG_VT	Y	Virtual terminal
CONFIG_VT_CONSOLE	N	Support for console on virtual terminal
CONFIG_SERIAL	Y	Standard/generic (8250/16550 and compatible UARTs) serial support
CONFIG_SERIAL_CONSOLE	Y	Support for console on serial port
CONFIG_SERIAL_EXTENDED	N	Extended dumb serial driver options
CONFIG_SERIAL_NONSTANDARD	N	Non-standard serial port support
CONFIG_QIC02_TAPE	N	QIC-02 tape support
CONFIG_INTEL_RNG	N	Intel i8x0 Random Number Generator support
CONFIG_NVRAM	N	/dev/nvram support
CONFIG_RTC	Y	Enhanced Real Time Clock Support
CONFIG_DTLK	N	Double Talk PC internal speech card support
CONFIG_R3964	N	Siemens R3964 line discipline
CONFIG_APPLICOM	N	Applicom intelligent fieldbus card support
CONFIG_AGP	N	/dev/agpgart (AGP Support)
CONFIG_DRM	N	Direct Rendering Manager (XFree86 DRI support)
CONFIG_MWAVE	N	ACP Modem (Mwave) support

Table 3-38: Serial Drivers

Option	Value	Description
CONFIG_SERIAL_8250	N	8250/16550 and compatible serial support (Experimental)
CONFIG_SERIAL_8250_MANY_PORTS	N	Support more than 4 serial ports

Table 3-38: Serial Drivers (Continued)

Option	Value	Description
CONFIG_SERIAL_8250_SHARE_IRQ	N	Support for sharing serial interrupts
CONFIG_SERIAL_8250_DETECT_IRQ	N	Autodetect IRQ on standard ports (unsafe)
CONFIG_SERIAL_8250_MULTIPORT	N	Support special multiport boards
CONFIG_SERIAL_8250_HUB6	N	Support Bell Technologies HUB6 card
CONFIG_UNIX98_PTYS	Y	Unix98 PTY support
CONFIG_UNIX98_PTY_COUNT	256	Maximum number of Unix98 PTYs in use (0-2048)
CONFIG_PRINTER	N	Parallel printer support
CONFIG_PPDEV	N	Support for user-space parallel port device drivers
CONFIG_BLUECAT_BPAR	N	BlueCat bidirectional parallel port transfer driver

Table 3-39: I2C Support

Option	Value	Description
CONFIG_I2C	N	I2C support

Table 3-40: L3 Serial Bus Support

Option	Value	Description
CONFIG_L3	N	L3 support
CONFIG_L3_BIT_SA1100_GPIO	N	SA11x0 GPIO adapter

Table 3-41: Mice

Option	Value	Description
CONFIG_BUSMOUSE	Y	Bus Mouse Support
CONFIG_ATIXL_BUSMOUSE	N	ATIXL busmouse support
CONFIG_LOGIBUSMOUSE	N	Logitech busmouse support
CONFIG_MS_BUSMOUSE	N	Microsoft busmouse support
CONFIG_MOUSE	Y	Mouse Support (not serial and bus mice)
CONFIG_PSMOUSE	Y	PS/2 mouse (aka "auxiliary device") support
CONFIG_82C710_MOUSE	N	C&T 82C710 mouse port support (as on TI Travelmate)
CONFIG_PC110_PAD	N	PC110 digitizer pad support

Table 3-42: Watchdog Cards

Option	Value	Description
CONFIG_WATCHDOG	N	Watchdog Timer Support

Table 3-43: Ftape, the Floppy Tape Device Driver

Option	Value	Description
CONFIG_FTAPE	N	Ftape (QIC-80/Travan) support

Table 3-44: Multimedia Devices

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video For Linux

Table 3-45: File Systems

Option	Value	Description
CONFIG_QUOTA	N	Quota support
CONFIG_AUTOFS_FS	N	Kernel automounter support
CONFIG_AUTOFS4_FS	N	Kernel automounter version 4 support (also supports v3)
CONFIG_REISERFS_FS	N	Reiserfs support
CONFIG_ADFS_FS	N	ADFS file system support
CONFIG_AFFS_FS	N	Amiga FFS file system support (Experimental)
CONFIG_HFS_FS	N	Apple Macintosh file system support (Experimental)
CONFIG_BFS_FS	N	BFS file system support (Experimental)
CONFIG_CMS_FS	N	CMS file system support (Experimental)
CONFIG_EXT3_FS	N	Ext3 journalling file system support (Experimental)
CONFIG_FAT_FS	N	DOS FAT fs support
CONFIG_UMSDOS_FS	N	UMSDOS: Unix-like file system on top of standard MSDOS fs
CONFIG_EFS_FS	N	EFS file system support (read only) (Experimental)
CONFIG_JFFS_FS	Y	Journalling Flash File System (JFFS) support
CONFIG_JFFS_FS_VERBOSE	0	JFFS debugging verbosity (0 = quiet, 3 = noisy)
CONFIG_JFFS_PROC_FS	N	JFFS stats available in /proc filesystem
CONFIG_JFFS2_FS	N	Journalling Flash File System v2 (JFFS2) support
CONFIG_CRAMFS	N	Compressed ROM file system support
CONFIG_TMPFS	N	Virtual memory file system support (former shm fs)

Table 3-45: File Systems (Continued)

Option	Value	Description
CONFIG_RAMFS	N	Simple RAM-based file system support
CONFIG_ISO9660_FS	N	ISO-9660 CDROM file system support
CONFIG_MINIX_FS	N	Minix fs support
CONFIG_FREEVXFS_FS	N	FreeVxFS file system support (VERITAS VxFS(TM) compatible)
CONFIG_NTFS_FS	N	NTFS file system support (read only)
CONFIG_HPFS_FS	N	OS/2 HPFS file system support
CONFIG_PROC_FS	Y	/proc file system support
CONFIG_DEVFS_FS	Y	/dev file system support (Experimental)
CONFIG_DEVFS_MOUNT	N	Automatically mount at boot
CONFIG_DEVFS_DEBUG	N	Debug devfs
CONFIG_DEVPTS_FS	Y	/dev/pts file system for Unix98 PTYs
CONFIG_QNX4FS_FS	N	QNX4 file system support (read only) (Experimental)
CONFIG_ROMFS_FS	N	ROM file system support
CONFIG_EXT2_FS	Y	Second extended fs support
CONFIG_SYSV_FS	N	System V/Xenix/V7/Coherent file system support
CONFIG_UDF_FS	N	UDF file system support (read only)
CONFIG_UFS_FS	N	UFS file system support (read only)

Table 3-46: Network File Systems

Option	Value	Description
CONFIG_CODA_FS	N	Coda file system support (advanced network fs)
CONFIG_INTERMEZZO_FS	N	InterMezzo file system support (Experimental, replicating fs)
CONFIG_NFS_FS	Y	NFS file system support

Table 3-46: Network File Systems (Continued)

Option	Value	Description
CONFIG_NFS_V3	N	Provide NFSv3 client support
CONFIG_NFSD	Y	NFS server support
CONFIG_NFSD_V3	N	Provide NFSv3 server support
CONFIG_SMB_FS	N	SMB file system support (to mount Windows shares etc.)
CONFIG_NCP_FS	N	NCP file system support (to mount NetWare volumes)

Table 3-47: Partition Types

Option	Value	Description
CONFIG_PARTITION_ADVANCED	Y	Advanced partition selection
CONFIG_ACORN_PARTITION	N	Acorn partition support
CONFIG_OSF_PARTITION	N	Alpha OSF partition support
CONFIG_AMIGA_PARTITION	N	Amiga partition table support
CONFIG_ATARI_PARTITION	N	Atari partition table support
CONFIG_MAC_PARTITION	Y	Macintosh partition map support
CONFIG_MSDOS_PARTITION	Y	PC BIOS (MSDOS partition tables) support
CONFIG_BSD_DISKLABEL	N	BSD disklabel (FreeBSD partition tables) support
CONFIG_MINIX_SUBPARTITION	N	Minix subpartition support
CONFIG_SOLARIS_X86_PARTITION	N	Solaris (x86) partition table support
CONFIG_UNIXWARE_DISKLABEL	N	Unixware slices support
CONFIG_LDM_PARTITION	N	Windows Logical Disk Manager (Dynamic Disk) support
CONFIG_SGI_PARTITION	N	SGI partition support
CONFIG_ULTRIX_PARTITION	N	Ultrix partition table support
CONFIG_SUN_PARTITION	N	Sun partition tables support

Table 3-48: Sound

Option	Value	Description
CONFIG_SOUND	N	Sound card support

Table 3-49: USB Support

Option	Value	Description
CONFIG_USB	N	Support for USB
CONFIG_USB_STORAGE_DEBUG	N	USB Mass Storage verbose debug
CONFIG_USB_STORAGE_DATAFAB	N	Datafab MDCFE-B Compact Flash Reader
CONFIG_USB_STORAGE_FREECOM	N	Freecom USB/ATAPI Bridge support
CONFIG_USB_STORAGE_ISD200	N	ISD-200 USB/ATA Bridge support
CONFIG_USB_STORAGE_JUMPSHOT	N	Lexar Jumpshot Compact Flash Reader
CONFIG_USB_STORAGE_DPCM	N	Microtech CompactFlash/SmartMedia reader
CONFIG_USB_STORAGE_HP8200e	N	HP CD-Writer 82xx support
CONFIG_USB_STORAGE_SDDR09	N	SanDisk SDDR-09 (and other SmartMedia) support

Table 3-50: USB Serial Converter Support

Option	Value	Description
CONFIG_USB_SERIAL_GENERIC	N	USB Generic Serial Driver
CONFIG_USB_SERIAL_BELKIN	N	USB Belkin and Peracom Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_WHITEHEAT	N	USB ConnectTech WhiteHEAT Serial Driver (Experimental)
CONFIG_USB_SERIAL_DIGI_ACCELEPORT	N	USB Digi International AccelePort USB Serial Driver

Table 3-50: USB Serial Converter Support (Continued)

Option	Value	Description
CONFIG_USB_SERIAL_EMPEG	N	USB Empeg empeg-car Mark I/II Driver (Experimental)
CONFIG_USB_SERIAL_FTDI_SIO	N	USB FTDI Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_VISOR	N	USB Handspring Visor / Palm m50x / Sony Clie Driver
CONFIG_USB_SERIAL_IR	N	USB IR Dongle Serial Driver (Experimental)
CONFIG_USB_SERIAL_EDGEPORT	N	USB Inside Out Edgeport Serial Driver (Experimental)
CONFIG_USB_SERIAL_KEYSPAN_PDA	N	USB Keyspan PDA Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_KEYSPAN	N	USB Keyspan USA-xxx Serial Driver (Experimental)
CONFIG_USB_SERIAL_KEYSPAN_USA28	N	USB Keyspan USA-28 Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA28X	N	USB Keyspan USA-28X Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA28XA	N	USB Keyspan USA-28XA Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA28XB	N	USB Keyspan USA-28XB Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA19	N	USB Keyspan USA-19 Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA18X	N	USB Keyspan USA-18X Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA19W	N	USB Keyspan USA-19W Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA49W	N	USB Keyspan USA-49W Firmware
CONFIG_USB_SERIAL_MCT_U232	N	USB MCT Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_PL2303	N	USB Prolific 2303 Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_CYBERJACK	N	USB REINER SCT cyberJack pinpad/e-com chipcard reader (Experimental)

Table 3-50: USB Serial Converter Support (Continued)

Option	Value	Description
CONFIG_USB_SERIAL_XIRCOM	N	USB Xircom / Entregra Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_OMNINET	N	USB ZyXEL omni.net LCD Plus Driver (Experimental)

Table 3-51: Bluetooth Support

Option	Value	Description
CONFIG_BLUEZ	N	Bluetooth subsystem support

Table 3-52: Kernel Hacking

Option	Value	Description
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_KGDB	N	Include kgdb kernel debugger
CONFIG_XMON	N	Include xmon kernel debugger
CONFIG_BLUECAT_KDBG	N	Include kdbg kernel debugger

Table 3-53: Modular Advanced Power Management

Option	Value	Description
CONFIG_BLUECAT_APM	N	MAPM support

CHAPTER 4 *Supported Demo Systems*

This chapter provides information about BlueCat Linux demo systems supported by the sandpoint Board Support Package (BSP).

Demo Systems

Table 4-1 lists the demo systems supported in the sandpoint BSP distribution, the boot devices supported by each demo system, and their respective RAM and ROM requirements.

Table 4-1: Demo Systems Supported by sandpoint BSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
`developer`	**Flash**, **Network** using the Boot Loader or the OS Loader	3563 KB	16384 KB
`osloader`	**Flash**, **Network** using the Boot Loader or the OS Loader	711 KB	4608 KB
`showcase`	**Flash**, **Network** using the Boot Loader or the OS Loader	2570 KB	13824 KB

developer Demo System

The `developer` demo system is a package consisting of the functionalities of the `shell`, `ftp`, `ping`, `gdb`, and `vl_demo` systems. Refer to Chapter 4 of the *BlueCat*

Linux User's Guide for descriptions of `developer` and its component demo systems.

osloader Demo System

`osloader` is the BlueCat OS Loader system used to boot a BlueCat Linux system on the target board. Refer to Chapter 4 of the *BlueCat Linux User's Guide* for details.

showcase Demo System

The `showcase` demo system starts and configures the Apache HTTP daemon turning the target board into a Web server. Refer to Chapter 4 of the *BlueCat Linux User's Guide* for details.

Using Selected RPM Packages

This section provides a description on how to use selected RPM packages that are frequently deployed in the embedded systems environment.

Using BusyBox RPM Package

The BusyBox RPM package combines tiny versions of many common UNIX utilities into a single small executable. It provides minimalist replacements for most of the utilities that can be usually found in `fileutils`, `shellutils`, `findutils`, `textutils`, `grep`, `gzip`, `tar`, etc. BusyBox provides a fairly complete POSIX environment for any small or embedded system.

The utilities in BusyBox generally have fewer options than their full-featured GNU cousins. However, the options that are included provide the expected functionality and behave very much like their GNU counterparts.

This section describes the steps necessary for creating and booting a BlueCat Linux system containing BusyBox and demonstrates use of the BusyBox utilities.

Creating a BlueCat Linux System for BusyBox

The user must step through the following procedure to create a BlueCat Linux image for BusyBox.

1. Create a new directory:

   ```
   BlueCat:$ mkdir -p \
   $BLUECAT_PREFIX/demo/busybox/local
   ```

2. Setup the BlueCat Linux kernel configuration with the standard kernel configuration tools and copy the kernel configuration file to the `$BLUECAT_PREFIX/demo/busybox` directory. For example:

   ```
   BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux

   BlueCat:$ make xconfig

   BlueCat:$ cp .config \
   $BLUECAT_PREFIX/demo/busybox/busybox.config
   ```

   NOTE: The kernel config file for the `developer` demo (`$BLUECAT_PREFIX/demo/developer/developer.config`) is also recommended as a starting point.

3. Create the BlueCat kernel downloadable image (`busybox.kernel`):

   ```
   BlueCat:$ cd $BLUECAT_PREFIX/demo/busybox

   BlueCat:$ mkkernel ./busybox.config \
   ./busybox.kernel ./busybox.disk
   ```

4. Create a spec file (`busybox.spec`) that contains the following minimal directives:

   ```
   strip on

   mkdir /dev
   mknod /dev/console c 5 1

   mkdir    /lib
   mkdir -p /usr/lib
   mkdir    /bin
   mkdir    /sbin
   mkdir -p /etc/rc.d
   mkdir    /proc

   cp ./local/fstab ./local/inittab  /etc
   cp ./local/rc.sysinit             /etc/rc.d

   lcd ${BLUECAT_PREFIX}/sbin
   cp reboot busybox /sbin

   ln -s /sbin/busybox /sbin/init
   ```

```
ln -s /sbin/busybox /sbin/ifconfig
ln -s /sbin/busybox /sbin/route
ln -s /sbin/busybox /bin/mount
ln -s /sbin/busybox /bin/sh
ln -s /sbin/busybox /bin/ping

chmod 711 /etc/rc.d/rcsysinit
chmod 755 /bin /sbin
cp ${BLUECAT_PREFIX}/lib/libnss_files-*.so     /lib
# End of File
```

5. Create the `local/fstab` file with the following contents:

```
proc    /proc   proc    defaults        0       0
```

6. Create the `local/inittab` file with the following contents:

```
# System initialization.
::sysinit:/etc/rc.d/rc.sysinit

::respawn:/bin/sh
```

NOTE: The first two fields in every record of the `inittab` file are ignored by the BusyBox `init`, so they must be empty. For example, the line `1:12345:respawn:/bin/sh` is not valid.

7. Create the `local/rc.sysinit` file with the following contents:

```
#!/bin/sh

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
mount -a
```

8. Create a root file system image (`busybox.rfs`) by entering the following command:

```
BlueCat:$ mkrootfs -lv ./busybox.spec ./busybox.rfs
```

NOTE: `Makefile` for the `developer` demo system can be used to produce the BusyBox kernel and RFS images. The user must change the line `KDI_NAME = developer` to `KDI_NAME = busybox` and then run the `make all` command.

Booting the BusyBox Images from a Network

Use the following procedure to boot the BlueCat Linux with the BusyBox utility from a network using the BlueCat Linux OS Loader. Refer to Chapter 2, “Downloading and Booting BlueCat Linux on the Target” for additional details about the BlueCat Linux OS Loader.

1. At the OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_IP>
> set HOST <host_IP>
> set KERNEL tftp busybox.kernel
> set RFS tftp busybox.rfs
> set CMD console=ttyS0,57600 ramdisk_size=28472
> boot
```

   where <*target_IP*> is an IP address of the target, and <*host_IP*> is an IP address of the development host.

   As a result, the BusyBox utility is loaded onto the target board and then automatically started.

Using BusyBox Utilities

This section provides the examples of using the BusyBox utilities.

- ls

```
/ # ls /
bin         etc         lost+found  sbin
dev         lib         proc        usr
```

- cat

```
/ # cat /etc/inittab
# System initialization.
::sysinit:/etc/rc.d/rc.sysinit

::respawn:/bin/sh
```

- chmod

```
/ # chmod a-x /sbin/reboot
/ # ls -la /sbin/reboot
-rw-r--r--   1 0      0          8068 Jun 26  2002 /sbin/reboot
/ # chmod 755 /sbin/reboot
/ # ls -la /sbin/reboot
-rwxr-xr-x   1 0      0          8068 Jun 26  2002 /sbin/reboot
```

- echo

```
/ # echo !!!!!!!!!
!!!!!!!!!
```

- date

```
/ # date
Sun Jun  2 18:37:32 UTC 2002
```

- uname

```
/ # uname -a
Linux (none) 2.4.10-1 #7 Wed Jun 26 13:39:16 MSD 2002 ppc unknown
```

- mount

```
/ # mount
/dev/root on / type ext2 (rw)
proc on /proc type proc (rw)
```

- ifconfig

```
/ # ifconfig eth0 192.168.4.13
/ # ifconfig eth0
eth0    Link encap:Ethernet  HWaddr 00:90:27:0D:C3:3B
      inet addr:192.168.4.13 Bcast:192.168.4.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:100
         RX bytes:306 (306.0 b)  TX bytes:0 (0.0 b)
        Interrupt:19 Base address:0xf000
```

- ping

```
/ # ping 192.168.4.121
PING 192.168.4.121 (192.168.4.121): 56 data bytes
64 bytes from 192.168.4.121: icmp_seq=0 ttl=255 time=1.1 ms
64 bytes from 192.168.4.121: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 192.168.4.121: icmp_seq=2 ttl=255 time=0.1 ms

--- 192.168.4.121 ping statistics ---                                    5
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.3/1.1 ms
```

Using TinyLogin RPM Package

The TinyLogin RMP package is a suite of tiny Unix utilities for handling logging into, being authenticated by, changing one's password for, and maintaining users and groups on an embedded system. It also provides shadow password support to enhance system security.

This section describes the steps necessary for creating and booting a BlueCat Linux system containing TinyLogin and demonstrates use of the TinyLogin utility.

Creating a BlueCat Linux System for TinyLogin

Use the following procedure to create a BlueCat Linux image for TinyLogin:

1. Create a new directory by typing:

   ```
   BlueCat:$ mkdir -p \
   $BLUECAT_PREFIX/demo/tinylogin/local
   ```

2. Setup the BlueCat Linux kernel configuration with the standard kernel configuration tools and copy the kernel configuration file to the `$BLUECAT_PREFIX/demo/tinylogin` directory. For instance, type the following commands:

   ```
   BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux

   BlueCat:$ make xconfig

   BlueCat:$ cp .config \
   $BLUECAT_PREFIX/demo/tinylogin/tinylogin.config
   ```

   NOTE: The kernel config file for the `developer` demo (`$BLUECAT_PREFIX/demo/developer/developer.config`) is also recommended as a starting point.

3. Create the BlueCat kernel downloadable image (`tinylogin.kernel`):

   ```
   BlueCat:$ cd $BLUECAT_PREFIX/demo/tinylogin

   BlueCat:$ mkkernel ./tinylogin.config \
   ./tinylogin.kernel ./tinylogin.disk
   ```

4. Create a spec file (`tinylogin.spec`) that contains the following minimal directives:

   ```
   strip on

   mkdir /dev
   mknod /dev/console c 5 1
   ln -s /dev/console /dev/tty
   ln -s /dev/console /dev/tty1

   mkdir    /bin
   mkdir    /sbin
   mkdir -p /etc/rc.d
   mkdir    /proc
   mkdir    /tmp
   mkdir -p /usr/bin

   mkdir /root

   mkdir /dev/pts
   mknod /dev/ptmx c 5 2
   ```

```
chmod 0666 /dev/ptmx

cp ./local/fstab ./local/passwd ./local/inittab  /etc
cp ./local/securetty ./local/shadow             /etc
cp ./local/rc.sysinit                          /etc/rc.d
cp ${BLUECAT_PREFIX}/etc/shells                /etc
chmod 644 /etc/shells
cp ${BLUECAT_PREFIX}/etc/group                 /etc

lcd ${BLUECAT_PREFIX}/sbin
cp reboot init mingetty /sbin

cp ${BLUECAT_PREFIX}/usr/bin/tinylogin          /usr/bin
ln -s /usr/bin/tinylogin        /usr/bin/passwd
ln -s /usr/bin/tinylogin        /bin/login

lcd ${BLUECAT_PREFIX}/bin
cp mount bash ls cat hostname /bin
ln -s /bin/bash /bin/sh

chmod 711 /etc/rc.d/rc.sysinit

chmod 755 /bin /sbin /usr/bin

chmod 04755 /usr/bin/tinylogin
# End of File
```

NOTE: In this spec file the `/bin/login` and `/usr/bin/passwd` symlinks point to `/usr/bin/tinylogin`. This allows the user to change the user's password simply by typing `passwd`.

5. Create the `local/fstab` file with the following contents:

```
none /proc    proc
none /dev/pts devpts
```

6. Create the `local/inittab` file with the following contents:

```
id:1:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

1:12345:respawn:/sbin/mingetty tty1
```

7. Create the `local/securetty` file with the following contents:

```
console
tty1
```

8. Create the `local/passwd` file with the following contents:

```
root:x:0:0:/root:/:/bin/bash
guest:x:500:10::/:/bin/bash
```

9. Create the `local/shadow` file:

```
root::10942:0:99999:7:::
```

```
guest::500:10:99999:7:::
```

10. Create the `local/rc.sysinit` file with the following contents:

```
#!/bin/sh

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

mount -a
hostname myhostname
```

11. Create a root file system image (`tinylogin.rfs`) by entering the following command:

```
BlueCat:$ mkrootfs -lv ./tinylogin.spec \
./tinylogin.rfs
```

NOTE: `Makefile` for the `developer` demo system can be used to produce the TinyLogin kernel and RFS images. The user must change the line `KDI_NAME = developer` to `KDI_NAME = tinylogin` and then run the `make all` command.

Booting the TinyLogin Images from a Network

Use the following procedure to boot BlueCat Linux with the TinyLogin utility from a network using the BlueCat Linux OS Loader. Refer to Chapter 2, “Downloading and Booting BlueCat Linux on the Target” for additional details about the BlueCat Linux OS Loader.

1. At the OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_IP>
> set HOST <host_IP>
> set KERNEL tftp tinylogin.kernel
> set RFS tftp tinylogin.rfs
> boot
```

   Where <*target_IP*> is an IP address of the target, and <*host_IP*> is an IP address of the development host.

As a result, the TinyLogin utility is loaded onto the target board and automatically started.

Using TinyLogin Utility

This section provides examples of using the TinyLogin utility:

- Changing the guest password:

```
myhostname login: guest
bash-2.04$ passwd
Changing password for guest
Enter the new password (minimum of 5, maximum of 8 characters)

                        Please use a combination of upper and lower
case letters and numbers.
Enter new password: <new_guest_password>
Re-enter new password: <new_guest_password>
passwd[13]: password for `guest' changed by user `guest'
Password changed.
bash-2.04$ exit
myhostname login: guest
Password: <new_guest_password>
bash-2.04$ exit
```

- Changing the root password:

```
myhostname login: root
login[15]: root login  on `console'

bash-2.04# passwd
Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)

Please use a combination of upper and lower case letters and numbers.
Enter new password: <new_root_password>
Re-enter new password: <new_root_password>
passwd[16]: password for `root' changed by user `root'
Password changed.
bash-2.04# exit
myhostname login: root
Password: <new_root_password>
login[18]: root login  on `console'

bash-2.04# exit
```

- Getting the root permissions:

```
myhostname login: guest
Password: <guest_password>
bash-2.04$ tinylogin su
Password:
login[17]: root login  on `console'

bash-2.04#
```

Using Zebra RPM Package

GNU Zebra is a free software that manages a TCP/IP based routing protocol. It takes multi-server and multi-thread approach to resolve the current complexity of the Internet.

GNU Zebra supports BGP4, BGP4+, OSPFv2, OSPFv3, RIPv1, RIPv2, and RIPng.

GNU Zebra is intended to be used as a Route Server and a Route Reflector. It is not a toolkit, it provides full routing power under a new architecture. GNU Zebra is unique in design in that it has a process for each protocol.

This section describes the steps necessary for creating and booting a BlueCat Linux system containing Zebra and demonstrates use of the Zebra utility.

Creating a BlueCat Linux System for Zebra

Use the following procedure to create a BlueCat Linux image for Zebra:

1. Create a new directory by typing:

   ```
   BlueCat:$ mkdir -p $BLUECAT_PREFIX/demo/zebra/local
   ```

2. Setup the BlueCat Linux kernel configuration with the standard kernel configuration tools and copy the kernel configuration file to the `$BLUECAT_PREFIX/demo/zebra` directory. For example:

   ```
   BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux
   BlueCat:$ make xconfig
   BlueCat:$ cp .config \
   $BLUECAT_PREFIX/demo/zebra/zebra.config
   ```

   NOTE: In the kernel configuration the following options must be set to `Y`:

   ```
   CONFIG_NETLINK=Y
   CONFIG_RTNETLINK=Y
   ```

   By default Zebra is configured to communicate with the kernel via the netlink socket.

3. Create the BlueCat kernel downloadable image (`zebra.kernel`):

   ```
   BlueCat:$ cd $BLUECAT_PREFIX/demo/zebra
   ```

```
BlueCat:$ mkkernel ./zebra.config ./zebra.kernel \
./zebra.disk
```

4. Create a spec file (`zebra.spec`) that contains the following minimal directives:

```
strip on

mkdir /dev
mknod /dev/console c 5 1
ln -s /dev/console /dev/tty
ln -s /dev/console /dev/tty1
# Standard 16550 serial driver device
mknod /dev/ttyS0   c 4 64
mknod /dev/ttyS1   c 4 65

mkdir -p /lib/security
mkdir -p /usr/lib
mkdir    /bin
mkdir    /sbin
mkdir -p /etc/rc.d
mkdir -p /etc/pam.d
mkdir -p /etc/xinetd.d
mkdir -p /etc/zebra
mkdir    /proc
mkdir    /tmp
mkdir -p /usr/bin
mkdir -p /usr/sbin
mkdir -p /var/run
mkdir -p /usr/libexec

mkdir -p /var/log/zebra

mkdir /root

mkdir /dev/pts
mknod /dev/ptmx c 5 2

chmod 0666 /dev/ptmx

cp ./local/fstab ./local/passwd ./local/inittab ./local/mtab /etc
cp ./local/other                                /etc/pam.d
cp ./local/rc.sysinit                           /etc/rc.d
cp ./local/hosts                                /etc
cp ./local/protocols                            /etc
cp ./local/resolv.conf                          /etc
cp ${BLUECAT_PREFIX}/etc/pwdb.conf              /etc
cp ${BLUECAT_PREFIX}/etc/nsswitch.conf          /etc
cp ${BLUECAT_PREFIX}/etc/services         /etc

cp ${BLUECAT_PREFIX}/etc/security /etc

cp ./local/shadow                               /etc
cp ./local/pam.d                                /etc
cp ./local/xinetd.d/*                           /etc/xinetd.d
cp ./local/zebra.conf                           /etc/zebra/

cp ${BLUECAT_PREFIX}/lib/libnss_files-*.so      /lib
cp ${BLUECAT_PREFIX}/lib/libnss_dns-*.so        /lib
cp ${BLUECAT_PREFIX}/lib/libpwdb.so             /lib
cp ${BLUECAT_PREFIX}/lib/security               /lib
```

```
cp ./local/empty   /var/log/wtmp

lcd ${BLUECAT_PREFIX}/sbin
cp reboot init mingetty ifconfig /sbin

cp ${BLUECAT_PREFIX}/lib/security/pam_permit.so /lib/security

cp ${BLUECAT_PREFIX}/etc/xinetd.conf           /etc

cp ${BLUECAT_PREFIX}/usr/bin/telnet            /usr/bin

cp ${BLUECAT_PREFIX}/etc/shells                /etc
chmod 644 /etc/shells

cp ${BLUECAT_PREFIX}/etc/group                 /etc

#
# General Binaries
#
lcd ${BLUECAT_PREFIX}/bin
cp ping mount bash cat ls hostname ps /bin
cp login /bin
ln -s /bin/bash /bin/sh

cp ${BLUECAT_PREFIX}/usr/bin/vtysh               /usr/bin

# internet services utils
cp ${BLUECAT_PREFIX}/usr/sbin/xinetd           /usr/sbin
cp ${BLUECAT_PREFIX}/usr/sbin/in.telnetd       /usr/sbin
cp ${BLUECAT_PREFIX}/usr/sbin/zebra             /usr/sbin

chmod 711 /etc/rc.d/rc.sysinit

chmod 755 /bin /sbin /usr/bin /usr/sbin

# End of File
```

5. Create the `local/inittab` file with the following contents:

```
id:1:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/sbin/halt
l6:6:wait:/sbin/reboot

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting
Down"

pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown
Cancelled"

1:12345:respawn:/sbin/mingetty tty1
```

6. Create the `local/rc.sysinit` file with the following contents:

```
#!/bin/sh

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

mount -a
xinetd -stayalive -reuse

hostname myhostname

zebra -d
```

7. Create the `local/zebra.conf` file with the following contents:

```
!
! zebra configuration file
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
!
interface lo
ip address 127.0.0.1/8

interface eth0
ip address 172.17.3.11/16

!
! Static default route.
!
ip route 213.24.0.0 255.255.0.0 172.17.0.1

log stdout
```

NOTE: This configuration file sets the Zebra password to `zebra`. The user must enter this password whenever connecting to Zebra or changing the Zebra configuration mode by entering the `enable` command at the command prompt.

8. Copy the `fstab`, `passwd`, `mtab`, `other`, `hosts`, `protocols`, `resolv.conf`, `shadow`, `pam.d/*`, `xinetd.d/*`, and `empty` files from the `$BLUECAT_PREFIX/demo/developer/local` directory to the `$BLUECAT_PREFIX/demo/zebra/local` directory:

9. Create a root file system image (`zebra.rfs`) by entering the following command:

```
BlueCat:$ mkrootfs -lv ./zebra.spec ./zebra.rfs
```

NOTE: `Makefile` for the `developer` demo system can be used to produce the Zebra kernel and RFS images. The user must change the line `KDI_NAME = developer` to `KDI_NAME = zebra` and then run the `make all` command.

Booting the Zebra Images from a Network

The user must step through the following procedure to boot the BlueCat Linux with the Zebra utility from a network using the BlueCat Linux OS Loader. Refer to Chapter 2, “Downloading and Booting BlueCat Linux on the Target” for additional details about BlueCat Linux OS Loader.

1. At the OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_IP>
> set HOST <host_IP>
> set CMD ramdisk_size=8192
> set KERNEL tftp zebra.kernel
> set RFS tftp zebra.rfs
> boot
```

   where <*target_IP*> is an IP address of the target, and <*host_IP*> is an IP address of the development host.

As a result, the Zebra utility is loaded onto the target board and automatically starts.

Using the Zebra Utility

This section provides examples of using the Zebra utility:

```
myhostname login: root
bash-2.04# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:80:4D:46:22:B8
          inet addr:172.17.3.11  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3904  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

bash-2.04# ping -c 2 172.17.0.1
PING 172.17.0.1 (172.17.0.1) from 172.17.3.11 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
Warning: time of day goes back, taking countermeasures.
64 bytes from 172.17.0.1: icmp_seq=0 ttl=255 time=0 usec
64 bytes from 172.17.0.1: icmp_seq=1 ttl=255 time=650 usec

--- 172.17.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.000/0.325/0.650/0.325 ms

bash-2.04# ping -c 3 213.24.253.87
PING 213.24.253.87 (213.24.253.87) from 172.17.3.11 : 56(84) bytes of
data.
64 bytes from 213.24.253.87: icmp_seq=0 ttl=254 time=902 usec
Warning: time of day goes back, taking countermeasures.
64 bytes from 213.24.253.87: icmp_seq=1 ttl=254 time=1.821 msec
64 bytes from 213.24.253.87: icmp_seq=2 ttl=254 time=898 usec

--- 213.24.253.87 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.898/1.207/1.821/0.434 ms


bash-2.04# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is zebra (version 0.91a).
Copyright 1996-2001 Kunihiro Ishiguro.


User Access Verification

Password: zebra
Router> enable
Password: zebra
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 172.17.0.0/16 is directly connected, eth0
S>* 213.24.0.0/16 [1/0] via 172.17.0.1, eth0
Router#
```

CHAPTER 5 *Supported Device Drivers*

Table 5-1 lists the device drivers supported by the sandpoint BSP:

Table 5-1: Device Drivers Supported by the sandpoint BSP

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Dual UART, Two 16650-compatible devices	`serial.c`	`drivers/char`	`CONFIG_SERIAL` `CONFIG_SERIAL_CONSOLE`	
Ethernet Controller, Intel EEPRO/100-compatible PCI card	`eepro100.c`	`drivers/net`	`CONFIG_EEPRO100`	
Flash, 1 MB 28F800B	`sandpoint.c`	`drivers/mtd/maps`	`CONFIG_MTD_SANDPOINT` `CONFIG_MTD_SANDPOINT_PART`	Supported via JFFS
IDE, Two ATA33 bus-master IDE ports	`ide.c`	`drivers/ide`	`CONFIG_IDE` `CONFIG_BLK_DEV_IDE`	Not tested explicitly
Parallel Port, IEEE 1284 parallel port	`parport_pc.c`	`drivers/parport`	`CONFIG_PARPORT` `CONFIG_PARPORT_PC`	Not tested explicitly
Keyboard, PC keyboard	`pc_keyb.c` `keyboard.c`	`drivers/char`	`CONFIG_VT`	Not tested explicitly

Table 5-1: Device Drivers Supported by the sandpoint BSP (Continued)

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Mouse, PS/2 mouse	pc_keyb.c	drivers/char	CONFIG_MOUSE CONFIG_PSMOUSE CONFIG_VT	Not tested explicitly
RTC	rtc.c	drivers/char	CONFIG_RTC	

CHAPTER 6 *Defect Fixes and Known Problems*

Defect Fixes

Table 6-1 details the defects fixed in this release of BlueCat Linux:

Table 6-1: Defect Fixes in BlueCat Linux

Platform	Subcomponent	ID	Summary
All	BlueCat Misc	16057	**Ctrl-C**, **Ctrl-Z** etc. do not work from shell
Windows (cross)	BlueCat Installation procedure	16130	Windows install should support non-C drives
All	BlueCat OS Loader	16358	The BLOSH `ntar` command hangs the system
All	BlueCat Misc	17308	Certain sequence of file updates sometimes causes FFS to crash after reboot
All	BlueCat Debuggers	18404	Update BlueCat `gdbserver` to process signals to `gdb` on the host correctly
All	BlueCat Linux Misc	18696	Provide support for NFS in BlueCat Linux
All	BlueCat mkimage	19605	`mkrootfs` doesn't support paths > 128 characters in `lcd` command

Motorola Sandpoint III BSP Problems and Limitations

- Only the Motorola 750 CPUs are supported with this BSP. Currently, there is no support for the Motorola 7400 CPU.
- Due to unavailability of a secondary Flash device on the available Sandpoint hardware, support of installing and running BlueCat Linux in

secondary flash, as well as generic Flash features such as JFFS, are untested and not supported.

- Due to limitations of the Sandpoint hardware (Sandpoint errata, item #2), the software reset does not work in this release.
- If `mkrootfs` is terminated (either by an error or by a signal), it tries to clean all its temporary files before exiting. However, due to certain features of the Cygwin execution environment, such temporary files can remain uncleaned in the `/tmp` directory on a Windows host. It is recommended that the `/tmp` directory be regularly checked and cleaned.
- Debugging multithreaded applications via GDB is not supported.
- The `tclx` RPM package is not included in the Windows-hosted distribution.
- On Windows hosts, some file permissions (including `r` and `s`) always have default values. To set permissions different from the default values, the `chmod` command should be used in the `.spec` file.