

# BlueCat Linux Board Support Guide

---

BlueCat Linux Release 4.1

DOC-0580-00

*for PowerPC 405GP Boards*

Product names mentioned in the *BlueCat Linux Board Support Guide for PowerPC 405GP Boards* are trademarks of their respective manufacturers and are used here only for identification purposes.

Copyright ©1987 - 2004, LynuxWorks, Inc. All rights reserved.  
U.S. Patents 5,469,571; 5,594,903

Printed in the United States of America.

All rights reserved. No part of the *BlueCat Linux Board Support Guide for PowerPC 405GP Boards* may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photographic, magnetic, or otherwise, without the prior written permission of LynuxWorks, Inc.

LynuxWorks, Inc. makes no representations, express or implied, with respect to this documentation or the software it describes, including (with no limitation) any implied warranties of utility or fitness for any particular purpose; all such warranties are expressly disclaimed. Neither LynuxWorks, Inc., nor its distributors, nor its dealers shall be liable for any indirect, incidental, or consequential damages under any circumstances.

(The exclusion of implied warranties may not apply in all cases under some statutes, and thus the above exclusion may not apply. This warranty provides the purchaser with specific legal rights. There may be other purchaser rights which vary from state to state within the United States of America.)

---

---

# Contents

	Typographical Conventions .....	v
	Special Notes .....	vi
	Technical Support .....	vi
<b>CHAPTER 1</b>	<b>OVERVIEW .....</b>	<b>1</b>
<b>CHAPTER 2</b>	<b>DOWNLOADING AND BOOTING BLUECAT LINUX ON THE TARGET .....</b>	<b>3</b>
	Prerequisites .....	3
	Downloading and Booting Overview .....	4
	Setting up Hardware .....	4
	Connecting the Target Board Serial Ports to the Host .....	4
	Connecting the Target Board Ethernet Card to the Host .....	4
	Programming IBM ROM Monitor Firmware onto the Walnut Board .....	5
	Setting up the IBM ROM Monitor Firmware .....	5
	Target Board Switches .....	9
	Booting a Demo System from a Network .....	11
<b>CHAPTER 3</b>	<b>KERNEL CONFIGURATION OPTIONS .....</b>	<b>15</b>
<b>CHAPTER 4</b>	<b>SUPPORTED DEMO SYSTEMS.....</b>	<b>33</b>
	Demo Systems .....	33
	developer Demo System .....	33
	osloader Demo System .....	34
	showcase Demo System .....	34
	Using Selected RPM Packages .....	34
	Using the BusyBox RPM Package .....	34
	Using the TinyLogin RPM Package .....	40
	Using the Zebra RPM Package .....	45

---

<b>CHAPTER 5</b>	<b>SUPPORTED DEVICE DRIVERS .....</b>	<b>53</b>
<b>CHAPTER 6</b>	<b>NEW FEATURES .....</b>	<b>55</b>
	Supported Cross-Development Hosts .....	55
	JFFS2 Support .....	55
	Support of Multithreaded Applications in GDB .....	56
	Cygwin Execution Environment Version 1.3.6 .....	56
<b>CHAPTER 7</b>	<b>KNOWN PROBLEMS AND LIMITATIONS.....</b>	<b>57</b>
	Walnut Target Board Problems and Limitations .....	57

---

# Preface

---

## Typographical Conventions

The typefaces used in this manual, summarized below, emphasize important concepts. All references to file names and commands are case-sensitive and should be typed accurately.

### Kind of Text

### Examples

Body text; *italicized* for emphasis, new terms, and book titles

Refer to the *BlueCat Linux User's Guide*.

Environment variables, file names, functions, methods, options, parameter names, path names, commands, and computer data

```
ls
-l
myprog.c
/dev/null
```

Commands that need to be highlighted within body text, or commands that must be typed as is by the user are **bolded**.

```
login: myname
# cd /usr/home
```

Text that represents a variable, such as a file name or a value that must be entered by the user

```
cat <filename>
mv <file1> <file2>
```

Blocks of text that appear on the display screen after entering instructions or commands

```
Loading file /tftpboot/shell.kdi
into 0x4000
.....
File loaded. Size is 1314816
Copyright 2002 LynuxWorks, Inc.
All rights reserved.

LynxOS (ppc) created Mon Jan 17
17:50:22 GMT 2002
user name:
```

## Special Notes

The following notations highlight any key points and cautionary notes that may appear in this manual.

---

**NOTE:** These callouts note important or useful points in the text.

---



**CAUTION!** Used for situations that present minor hazards that may interfere with or threaten equipment/performance.

---

## Technical Support

LynuxWorks Technical Support is available Monday through Friday (holidays excluded) between 8:00 AM and 5:00 PM Pacific Time (U.S. Headquarters) or between 9:00 AM and 6:00 PM Central European Time (Europe).

### LynuxWorks U.S. Headquarters

email [support@lnxw.com](mailto:support@lnxw.com)

phone (408) 979-3940  
(800) 327-5969

fax (408) 979-3945

### LynuxWorks Europe

email [tech\\_europe@lnxw.com](mailto:tech_europe@lnxw.com)

phone (+33) 1 30 85 06 00

fax (+33) 1 30 85 06 06

### World Wide Web

website <http://www.lynuxworks.com>

customer support <http://www.lynuxworks.com/support/custhelp.php3>

The *BlueCat Linux Board Support Guide for PowerPC 405GP Boards* provides information about the BlueCat Linux Board Support Package (BSP) for the IBM PowerPC 405GP (Walnut) reference board.

Throughout this Board Support Guide (BSG), the BSP is referred to as the “walnut” and the board as the “Walnut” or simply as the “target board.”

The chapters of this BSG provide the information listed below:

- *Chapter 1* is an overview of this BSG’s individual chapters.
- *Chapter 2* describes BlueCat Linux downloading and booting procedures for the Walnut board using the ROM Monitor firmware.
- *Chapter 3* provides configuration information about the walnut BSP’s default BlueCat Linux kernel.
- *Chapter 4* describes BlueCat Linux demo systems supported by the walnut BSP.
- *Chapter 5* provides a list of walnut BSP-supported device drivers, with important information about each of them.
- *Chapter 6* describes new features of this release.
- *Chapter 7* describes known limitations and workarounds for this release.



# *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 is a guide to downloading and booting BlueCat Linux systems onto the user's target board. Scenarios that use demo systems included in the BlueCat Linux distribution are presented. A basic familiarity with the target board hardware and operation is required. The user must also have an understanding of system administration for the particular cross-development host on which the BlueCat Linux Core and the BSP are installed. It is assumed that the user has the manufacturer's documentation for the target board as well as system administration reference material for the cross-development host.

Before downloading and booting BlueCat Linux on the target board, it is assumed that the default BlueCat Linux PowerPC configuration and the walnut 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" in the *BlueCat Linux User's Guide*.
2. Install the walnut BSP onto the cross-development host as detailed in the "Installing Target Board Support" section of Chapter 1, "Installation" in the *BlueCat Linux User's Guide*.
3. Activate support for the walnut 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 Walnut target consists of the following main steps:

- Setting up hardware
- Downloading and booting a BlueCat Linux system from a network using the factory-default ROM Monitor 1.15 firmware

---

## Setting up Hardware

### Connecting the Target Board Serial Ports to the Host

The target board has two serial ports. The first serial port is used both by the ROM Monitor 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 that the user 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 can have any baud rate.

Throughout this chapter, the terminal window connected to the first serial connector is referred to as the “ROM Monitor console” or the “BlueCat Linux console,” depending on the context.

### Connecting the Target Board Ethernet Card to the Host

The IBM Walnut board provides 10BaseT and 100BaseTX Ethernet connections via Category 5 Unshielded Twisted Pair cable using an RJ-45 connector, which is referenced to as J22 in the manufacture’s documentation.

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 from a network.

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

It is also required that the user 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 system administration reference material.

TFTP must be enabled on the host. For more information, refer to “Setting Up a TFTP Server” in Chapter 3, “Downloading and Booting BlueCat Linux” in the *BlueCat Linux User's Guide*.

---

## Programming IBM ROM Monitor Firmware onto the Walnut Board

To program the IBM ROM Monitor firmware, refer to the procedure described in the firmware documentation *PowerPC 405GP Reference Design Kit User's Manual* provided with the IBM PowerPC 405GP board.

---

## Setting up the IBM ROM Monitor Firmware

To set up the ROM Monitor firmware options for BlueCat Linux, perform the following steps:

1. Reset the target board. The ROM Monitor firmware main menu appears on the ROM Monitor console.

```
1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for s1 boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
0 - Exit menu and continue
```

2. Enable the Ethernet controller for booting images by selecting option **2 (Enable/disable boot devices)** in the main menu:

```
1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for sl boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
0 - Exit menu and continue
->2
```

The following submenu is displayed after the selection of option 2:

```
--- ENABLE AND DISABLE BOOT DEVICES ---
Boot Sources:
  001 Disabled Ethernet [ENET]
      local=0.0.0.0 remote=255.255.255.255
hwaddr=0004ace315c0
  004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255
hwaddr=ffffffffffff
  005 Enabled Serial Port 1 [S1]
-----
select device to change ->
```

Selecting a device enables a boot status of the device and disables the other boot devices. By selecting **001** you enable Ethernet as a boot device and disable Serial Port 1 and Serial Port 2.

```
select device to change ->001
[ENET] boot is enabled
```

The new setting is displayed below the main menu:

```
-----
select device to change ->5
[S1] boot is disabled

--- Device Configuration ---
Power-On Test Devices:
  000 Disabled System Memory [RAM]
  001 Enabled Ethernet [ENET]
  004 Disabled Serial Port 2 [S2]
-----
Boot Sources:
  001 Enabled Ethernet [ENET]
      local=0.0.0.0 remote=255.255.255.255
hwaddr=0004ace315c0
  004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255
hwaddr=ffffffffffff
  005 Disabled Serial Port 1 [S1]
-----
```

```

Debugger: Disabled
-----
 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for s1 boot
 A - Enable/disable I cache (Enabled)
 B - Enable/disable D cache (Disabled)
 0 - Exit menu and continue
->

```

3. Set up the IP addresses of the target and host boards. The IP addresses are used for BOOTP processing, debugger communications, and the host connectivity ping test. Do the following to set up IP addresses:

- a. Select option **3 (Change IP addresses)** in the main menu. The **Change IP addresses** option allows users to change IP addresses for the target and host boards.

```

 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for s1 boot
 A - Enable/disable I cache (Enabled )
 B - Enable/disable D cache (Disabled)
 0 - Exit menu and continue
->3

```

The following submenu is displayed after the selection of option **3**:

```

---  CHANGE IP ADDRESS  ---
Device List:
 001  Enabled   Ethernet      [ENET]
      local=0.0.0.0  remote=255.255.255.255
hwaddr=0004ace315c0
 004  Disabled  Serial Port 2 [S2]
      local=8.1.1.2  remote=255.255.255.255
hwaddr=ffffffffffff
-----
select device to change ->

```

- b. Type **001** to choose the Ethernet controller:

```

select device to change ->001

 1 - Change local address
 2 - Change remote address
 0 - Return to main menu
->

```

- c. To change the IP address of the target board, select option **1 (Change local address)** and enter `<target_IP>`, the address of the target in dotted decimal notation:

```

1 - Change local address
2 - Change remote address
0 - Return to main menu
->1
Current IP address = (0.0.0.0)
Enter new IP address ->Enter IP address in dot notation, (eg.
8.1.1.2)

```

`<target_IP>` **Enter**

```

--- Device Configuration ---
Power-On Test Devices:
 000 Disabled System Memory [RAM]
 001 Enabled Ethernet [ENET]
 004 Disabled Serial Port 2 [S2]
-----
Boot Sources:
 001 Enabled Ethernet [ENET]
      local=<target_IP> remote=255.255.255.255
hwaddr=0004ace315c0
 004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255
hwaddr=ffffffffffff
 005 Disabled Serial Port 1 [S1]
-----

```

- d. To set up the IP address of the host board, select option **2 (Change remote address)** and enter `<host_IP>`, the address of the host in dotted decimal notation:

```

-----
select device to change ->001
 1 - Change local address
 2 - Change remote address
 0 - Return to main menu
->2
Current IP address = (255.255.255.255)
Enter new IP address ->Enter IP address in dot notation, (eg.
8.1.1.2)

```

`<host_IP>` **Enter**

```

Power-On Test Devices:
 000 Disabled System Memory [RAM]
 001 Enabled Ethernet [ENET]
 004 Disabled Serial Port 2 [S2]
-----

```

```

Boot Sources:
  001 Enabled Ethernet [ENET]
      local=<target_IP> remote=<host_IP>
hwaddr=0004ace315c0
  004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255
hwaddr=ffffffffffff
  005 Disabled Serial Port 1 [S1]
-----

```

4. Save the firmware configuration by selecting option **8 (Save changes to configuration)**:

```

1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for s1 boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
0 - Exit menu and continue
->8

```

5. Reset the target board.

---

## Target Board Switches

Before starting the download process, the target board configuration switches U54 and U79 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: Board Strapping Switches—U54**

Switch	Position	Description
1	ON	ROM width
2	ON	ROM location
3	OFF	Reserved
4	ON	PCI asynchronous mode enable
5	ON	Reserved
6	OFF	PCI internal arbiter enable

**Table 2-1: Board Strapping Switches—U54 (Continued)**

Switch	Position	Description
7	ON	Clock source 0
8	OFF	Clock source 1

**Table 2-2: Board Strapping Switches—U79**

Switch	Position	Description
1, 2	ON,ON	PCI clock frequency
3, 4	ON,OFF	SDRAM frequency
5	OFF	Boot Flash on-board
6	OFF	Flash/SRAM mapping
7	OFF	405GP SDRAM clock
8	OFF	Unused

The PHY Address Selection Jumpers J6 and J8 must be set as shown in Table 2-3.

**Table 2-3: PHY Address Selection Jumpers—J6, J8**

Jumper	Position
J6	2-3 closed
J8	1-2 closed

## Booting a Demo System from a Network

The IBM ROM Monitor firmware uses BOOTP and TFTP network protocols to load images over the network.

---

**NOTE:** The name of the boot image must be specified in the BOOTP server configuration file `/etc/bootptab`. The following is a template of the BOOTP configuration file:

```
enetc:ht=ethernet:bf=ibm405gp.kdi:bs:ip=192.168.4.12:sm=255.255.255.0:ha=0004ace315c0:
```

which has the following fields:

- `bf`—Specifies the name of the Kernel Downloadable Image (KDI) file that is downloaded from the TFTP server.
- `ip`—Specifies the IP address assigned to the target board.
- `sm`—Specifies the subnet mask in which the target board resides.
- `ha`—Specifies the hardware address of the Ethernet interface on the target board.

---

**NOTE:** The user needs to install the BOOTP daemon on the server and enable the BOOTP service.

---

To boot the `showcase` demo system from a network using the ROM Monitor firmware, perform the following steps:

1. Copy the `showcase.kdi` file from the `BLUECAT_PREFIX/demo/showcase` directory to the `/tftpboot` directory on the cross-development host and rename it according to the BOOTP server configuration. In this example, because the `ibm405.kdi` name has been specified in the BOOTP server configuration file for the `showcase` demo system, the `showcase.kdi` file must be copied to the `/tftpboot` directory under the name `ibm405gp.kdi`.

```
BlueCat:$ cp $BLUECAT_PREFIX/demo/showcase/\  
showcase.kdi /tftpboot/ibm405gp.kdi
```

2. Reset the target board.

3. Select option **0 (Exit menu and continue)** to run the demo.

```
1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for sl boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
0 - Exit menu and continue
->0
```

The showcase demo system is loaded from a network and then automatically started.

```
EMACO Ethernet Test OK.
ENET Speed is 100 Mbs...
HALF duplex connection
Booting from [ENET] Ethernet    ...
Sending bootp request ...

Loading file "/ibm405gp.kdi" ...
Sending tftp boot request ...
Transfer Complete ...
Loaded successfully ...
Entry point at 0x107000 ...

loaded at:      00400000 0067D600
board data at: 004754F4 0047553C
relocated to:  004785CC 00478614
zimage at:     00405800 004740E7
initrd at:     00475000 00475000
rootfs at:     00479000 0067D600
rootfs cksum:  3F50E8DA
avail ram:     0067E000 0077E000
Linux/PPC load: ramdisk_size=32000 hda=bswap hdb=bswap hdc=bswap
hdd=bswap rootl
Uncompressing Linux...done.
Now booting the kernel
Linux version 2.4.18-1 (root@winbuild3) (gcc version 2.95.3 20010315
(release))3
pciauto_bus_scan: current_bus = 0
pciauto_bus_scan: first_busno = 0
IBM Walnut port (C) 2000-2002 MontaVista Software, Inc.
(source@mvista.com)
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: ramdisk_size=32000 hda=bswap hdb=bswap hdc=bswap
hdd=bswap1
Calibrating delay loop... 199.88 BogoMIPS
Memory: 28904k available (852k kernel code, 312k data, 60k init, 0k
highmem)
Dentry-cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode-cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
```

```
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ
SERIAL_PCI ed
ttyS00 at 0xef600300x (irq = 0) is a 16550A
ttyS01 at 0xef600400x (irq = 1) is a 16550A
block: 64 slots per queue, batch=16
RAMDISK driver initialized: 16 RAM disks of 32000K size 1024 blocksize
emac: IBM OCP EMAC Ethernet driver, version 2.0
Maintained by Benjamin Herrenschmidt <benh@kernel.crashing.org>
mal0: Initialized, 2 tx channels, 1 rx channels
eth0: IBM emac, MAC 00:04:ac:e3:15:c0
eth0: Found Generic MII PHY (0x01)
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 4580
Freeing BlueCat RFS memory: 2065k freed
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 60k init
eth0: Link is Up
eth0: Speed: 100, Half duplex.
INIT: version 2.78 booting
INIT: Entering runlevel: 1
Network is configured as followseth0: Speed: 100, Half duplex.
:

Target IP address: 192.168.4.12
Gateway IP address:

Starting Apache server...
bash-2.04#
```



# *Kernel Configuration Options*

The walnut BSP comes with a default BlueCat Linux kernel. This kernel has a number of configuration options. This chapter details these options in the tables listed in Table 3-1: “BlueCat Linux Default Configuration for the walnut BSP Distribution” below.

**Table 3-1: BlueCat Linux Default Configuration for the walnut BSP Distribution**

<b>Table Number and Configuration Parameter</b>
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: “Plug and Play Configuration”
Table 3-9: “Block Devices”
Table 3-10: “Multidevice Support (RAID and LVM)”
Table 3-11: “Networking Options”
Table 3-12: “QoS and/or Fair Queueing”
Table 3-13: “ATA/IDE/MFM/RLL Support”
Table 3-14: “SCSI Support”
Table 3-15: “IEEE 1394 (FireWire) Support (Experimental)”
Table 3-16: “Network Device Support”
Table 3-17: “ARCnet Devices”

**Table 3-1: BlueCat Linux Default Configuration for the walnut BSP Distribution (Continued)**

Table Number and Configuration Parameter
Table 3-18: "Ethernet (10 or 100Mbit)"
Table 3-19: "Ethernet (1000 Mbit)"
Table 3-20: "On-chip Net Devices"
Table 3-21: "Wireless LAN (Non-Ham Radio)"
Table 3-22: "Token Ring Devices"
Table 3-23: "WAN Interfaces"
Table 3-24: "Amateur Radio Support"
Table 3-25: "IrDA (Infrared) Support"
Table 3-26: "ISDN Subsystem"
Table 3-27: "Old CD-ROM Drivers (not SCSI, not IDE)"
Table 3-28: "Frame Buffer Support"
Table 3-29: "Input Core Support"
Table 3-30: "Character Devices"
Table 3-31: "Serial Drivers"
Table 3-32: "I2C Support"
Table 3-33: "L3 Serial Bus Support"
Table 3-34: "Mice"
Table 3-35: "Joysticks"
Table 3-36: "Watchdog Cards"
Table 3-37: "Ftape, the Floppy Tape Device Driver"
Table 3-38: "Multimedia Devices"
Table 3-39: "File Systems"
Table 3-40: "Network File Systems"
Table 3-41: "Partition Types"
Table 3-42: "Sound"
Table 3-43: "USB Support"

**Table 3-1: BlueCat Linux Default Configuration for the walnut BSP Distribution (Continued)**

Table Number and Configuration Parameter
Table 3-44: “Bluetooth Support”
Table 3-45: “Kernel Hacking”
Table 3-46: “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**

Options	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**

Options	Value	Description
CONFIG_40x	Y	Processor Type
CONFIG_WALNUT	Y	Machine Type
CONFIG_SMP	N	Symmetric multiprocessing support
CONFIG_MATH_EMULATION	Y	Math emulation
CONFIG_PM	N	Power Management support (Experimental)

**Table 3-4: Platform Support (Continued)**

Options	Value	Description
CONFIG_UART0_TTYS0	Y	TTYS0 device and default console
CONFIG_PPC4xx_DMA	N	PPC4xx DMA controller support

**Table 3-5: General Setup**

Options	Value	Description
CONFIG_HIGHMEM	N	High memory support (Experimental)
CONFIG_PCI	Y	Enable PCI
CONFIG_PC_KEYBOARD	Y	PC PS/2-style Keyboard
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_BINFORMAT_MISC	N	Kernel support for MISC binaries
CONFIG_PCI_NAMES	N	PCI device name database
CONFIG_HOTPLUG	N	Support for hot-pluggable devices

**Table 3-6: Parallel Port Support**

Options	Value	Description
CONFIG_PARPORT	N	Parallel port support
CONFIG_GEN_RTC	Y	Generic <code>/dev/rtc</code> emulation
CONFIG_CMDLINE_BOOL	N	Default boot loader kernel arguments

---

**Table 3-7: Memory Technology Devices (MTD)**

Options	Value	Description
CONFIG_MTD	N	Memory Technology Device (MTD) support

**Table 3-8: Plug and Play Configuration**

Options	Value	Description
CONFIG_PNP	N	Plug and Play support

**Table 3-9: Block Devices**

Options	Value	Description
CONFIG_BLK_DEV_FD	N	Normal PC floppy disk support
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	Y	Initial RAM disk ( <code>initrd</code> ) support

**Table 3-10: Multidevice Support (RAID and LVM)**

Options	Value	Description
CONFIG_MD	N	Multiple devices driver support (RAID and LVM)

**Table 3-11: Networking Options**

Options	Value	Description
CONFIG_PACKET	N	Packet socket
CONFIG_NETLINK_DEV	N	Netlink device emulation
CONFIG_NETFILTER	N	Network packet filtering (replaces ipchains)
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	Y	IP: kernel level autoconfiguration
CONFIG_IP_PNP_DHCP	N	IP: DHCP support
CONFIG_IP_PNP_BOOTP	Y	IP: BOOTP support
CONFIG_IP_PNP_RARP	Y	IP: RARP support
CONFIG_NET_IPIP	N	IP: tunneling
CONFIG_NET_IPGRE	N	IP: GRE tunnels over IP
CONFIG_IP_MROUTE	N	IP: multicast routing
CONFIG_ARPD	N	IP: ARP daemon support (Experimental)
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)

**Table 3-11: Networking Options (Continued)**

Options	Value	Description
CONFIG_KHTTPD	N	Kernel httpd acceleration (Experimental)
CONFIG_ATM	N	Asynchronous Transfer Mode (ATM) (Experimental)
CONFIG_VLAN_8021Q	N	802.1Q VLAN support (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)
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 3-12: QoS and/or Fair Queuing**

Options	Value	Description
CONFIG_NET_SCHED	N	QoS and/or fair queuing

**Table 3-13: ATA/IDE/MFM/RLL Support**

Options	Value	Description
CONFIG_IDE	N	ATA/IDE/MFM/RLL support

**Table 3-14: SCSI Support**

Options	Value	Description
CONFIG_SCSI	N	SCSI support

**Table 3-15: IEEE 1394 (FireWire) Support (Experimental)**

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

**Table 3-16: Network Device Support**

Options	Value	Description
CONFIG_NETDEVICES	Y	Network device support

**Table 3-17: ARCnet Devices**

Options	Value	Description
CONFIG_ARCNET	N	ARCnet 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 3-18: Ethernet (10 or 100Mbit)**

Options	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100Mbit)
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	N	EISA, VLB, PCI, and on-board controllers
CONFIG_NET_POCKET	N	Pocket and portable adapters

**Table 3-19: Ethernet (1000 Mbit)**

Options	Value	Description
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 Semiconductor 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_E1000	N	Intel PRO/1000 support

**Table 3-20: On-chip Net Devices**

Options	Value	Description
CONFIG_IBM_OCP_ENET	Y	IBM on-chip ethernet
CONFIG_IBM_OCP_ENET_ERROR_MSG	N	Verbose error messages
CONFIG_IBM_OCP_ENET_RX_BUFF	64	Number of receive buffers
CONFIG_IBM_OCP_ENET_TX_BUFF	8	Number of transmit buffers
CONFIG_IBM_OCP_ENET_GAP	8	Frame gap
CONFIG_IBM_OCP_ENET_SKB_RES	0	Skb reserve amount
CONFIG_FDDI	N	Fiber Distributed Data Interface (FDDI) driver support
CONFIG_HIPPI	N	High Performance Parallel Interface (HIPPI) driver support (Experimental)
CONFIG_PPP	N	Point-to-Point Protocol (PPP) support
CONFIG_SLIP	N	Serial Line Internet Protocol (SLIP) support

**Table 3-21: Wireless LAN (Non-Ham Radio)**

Options	Value	Description
CONFIG_NET_RADIO	N	Wireless LAN (non-ham radio)

**Table 3-22: Token Ring Devices**

Options	Value	Description
CONFIG_TR	N	Token Ring driver support
CONFIG_NET_FC	N	Fibre Channel driver support
CONFIG_RCPCI	N	Red Creek Hardware VPN (Experimental)
CONFIG_SHAPER	N	Traffic Shaper (Experimental)

---

**Table 3-23: WAN Interfaces**

Options	Value	Description
CONFIG_WAN	N	WAN interfaces support

**Table 3-24: Amateur Radio Support**

Options	Value	Description
CONFIG_HAMRADIO	N	Amateur radio support

**Table 3-25: IrDA (Infrared) Support**

Options	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

**Table 3-26: ISDN Subsystem**

Options	Value	Description
CONFIG_ISDN	N	ISDN support

**Table 3-27: Old CD-ROM Drivers (not SCSI, not IDE)**

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

**Table 3-28: Frame Buffer Support**

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

**Table 3-29: Input Core Support**

Options	Value	Description
CONFIG_INPUT	N	Input core support

**Table 3-30: Character Devices**

Options	Value	Description
CONFIG_VT	Y	Virtual terminal
CONFIG_VT_CONSOLE	Y	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	Nonstandard serial port support

**Table 3-31: Serial Drivers**

Options	Value	Description
CONFIG_SERIAL_8250	N	8250/16550 and compatible serial support (Experimental)
CONFIG_UNIX98_PTYS	N	Unix98 PTY support

---

**Table 3-32: I2C Support**

Options	Value	Description
CONFIG_I2C	N	I2C support

**Table 3-33: L3 Serial Bus Support**

Options	Value	Description
CONFIG_L3	N	L3 support

**Table 3-34: Mice**

Options	Value	Description
CONFIG_BUSMOUSE	N	Bus mouse 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 3-35: Joysticks**

Options	Value	Description
CONFIG_QIC02_TAPE	N	QIC-02 tape support

**Table 3-36: Watchdog Cards**

Options	Value	Description
CONFIG_WATCHDOG	N	Watchdog Timer support
CONFIG_INTEL_RNG	N	Intel i8x0 Random Number Generator support
CONFIG_NVRAM	N	/dev/nvram support
CONFIG_RTC	N	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

**Table 3-37: Ftape, the Floppy Tape Device Driver**

Options	Value	Description
CONFIG_FTAPE	N	Ftape (QIC-80/Travan) support
CONFIG_AGP	N	/dev/agpgart (AGP support)
CONFIG_DRM	N	Direct Rendering Manager (XFree86 DRI support)

**Table 3-38: Multimedia Devices**

Options	Value	Description
CONFIG_VIDEO_DEV	N	Video for Linux

**Table 3-39: File Systems**

Options	Value	Description
CONFIG_QUOTA	N	Quota support
CONFIG_AUTOFS_FS	N	Kernel automounter support

**Table 3-39: File Systems (Continued)**

<b>Options</b>	<b>Value</b>	<b>Description</b>
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_EXT3_FS	N	Ext3 Journalling file system support (Experimental)
CONFIG_FAT_FS	N	DOS FAT file system support
CONFIG_EFS_FS	N	EFS file system support (read-only) (Experimental)
CONFIG_CRAMFS	N	Compressed ROM file system support
CONFIG_TMPFS	Y	Virtual memory file system support (former shm file system)
CONFIG_RAMFS	N	Simple RAM-based file system support
CONFIG_ISO9660_FS	N	ISO 9660 CDROM file system support
CONFIG_MINIX_FS	N	Minix file system support
CONFIG_VXFS_FS	N	FreeVxFS file system support (VERITAS VxFS™-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	N	/dev file system support (Experimental)
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 file system support

**Table 3-39: File Systems (Continued)**

Options	Value	Description
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 3-40: Network File Systems**

Options	Value	Description
CONFIG_CODA_FS	N	Coda file system support (advanced network file system)
CONFIG_INTERMEZZO_FS	N	InterMezzo file system support (Experimental, replicating file system)
CONFIG_NFS_FS	Y	NFS file system support
CONFIG_NFS_V3	N	Provide NFSv3 client support
CONFIG_ROOT_NFS	Y	Root file system on NFS
CONFIG_NFSD	N	NFS 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 3-41: Partition Types**

Options	Value	Description
CONFIG_PARTITION_ADVANCED	N	Advanced partition selection

---

**Table 3-42: Sound**

Options	Value	Description
CONFIG_SOUND	N	Sound card support

**Table 3-43: USB Support**

Options	Value	Description
CONFIG_USB	N	Support for USB

**Table 3-44: Bluetooth Support**

Options	Value	Description
CONFIG_BLUEZ	N	Bluetooth subsystem support

**Table 3-45: Kernel Hacking**

Options	Value	Description
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_XMON	N	Include xmon kernel debugger
CONFIG_BLUECAT_KDBG	N	Include kdbg kernel debugger
CONFIG_SERIAL_TEXT_DEBUG	N	Support for early boot texts over serial port

**Table 3-46: Modular Advanced Power Management**

Options	Value	Description
CONFIG_BLUECAT_APM	N	Modular Advanced Power Management (MAPM) support



This chapter provides information about BlueCat Linux demo systems supported by the walnut BSP.

---

## Demo Systems

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

**Table 4-1: Demo Systems Supported by walnut BSP**

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
developer	Network (using ROM Monitor 1.15 firmware)	3.5 MB	14880 KB
osloader	Network (using ROM Monitor 1.15 firmware)	676 KB	3584 KB
showcase	Network (using ROM Monitor 1.15 firmware)	2.5 MB	12019 KB

### developer Demo System

The `developer` demo system is a package consisting of the functionalities of the `shell`, `ftp`, `ping`, `gdb`, and `v1_demo` systems. For descriptions of `developer` and its component demo systems, refer to Chapter 4, “BlueCat Linux Demo Systems” in the *BlueCat Linux User’s Guide*.

## 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, “BlueCat Linux Demo Systems” in 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, “BlueCat Linux Demo Systems” in the *BlueCat Linux User’s Guide* for details.

---

## Using Selected RPM Packages

The following sections describe how to use selected RPM packages that are frequently deployed in the embedded systems environment.

### Using the 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 UNIX utilities, such as `fileutils`, `shellutils`, `findutils`, `textutils`, `grep`, `gzip`, and `tar`. 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 counterparts. The options that are included, however, provide the expected functionality and behave much like their GNU correlates.

The following sections describe the steps necessary for creating and booting a BlueCat Linux system containing BusyBox and demonstrate the use of the BusyBox utilities.

## Creating a BlueCat Linux System for BusyBox

To create a BlueCat Linux image for BusyBox, perform the following steps:

1. Create a new directory by typing:

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

2. Set up the BlueCat Linux kernel configuration using the standard kernel configuration tools and copy the kernel configuration file to the `$BLUECAT_PREFIX/demo/busybox` directory. For instance, type the following commands:

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

Select **Save and Exit** to update the `.config` file, then type the following command:

```
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 a BlueCat Linux 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`) with 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
```

```
ln -s ${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/rc.sysinit
chmod 755 /bin /sbin
# 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
```

9. Create a composite BlueCat Linux image:

```
BlueCat:$ mkboot -m -k busybox.disk -f \
busybox.rfs busybox.kdi
```

Refer to the *BlueCat Linux User's Guide* for more information about a BlueCat Linux composite image.

---

**NOTE:** The Makefile for the developer demo system can be used to produce the BusyBox kernel and RFS images. To do so, modify the Makefile as follows:

- i. Change the line `KDI_NAME = developer` to `KDI_NAME = busybox`.
- ii. Comment out the following lines:

```
# IS_NEED_REBUILD_SRC=$(shell if ! make -q -C src; then echo this ; fi)
# cd src; make all
```

- iii. Change the following lines:

```
clean      :
            rm -f *.rfs *.tar *.kernel *.disk *.kdi *.srec; cd src; make clean
```

to read as follows:

```
clean      :
            rm -f *.rfs *.tar *.kernel *.disk *.kdi *.srec
```

- iv. Run the `make all` command.
- v. If the following message appears:

```
make: circular busybox <- busybox dependency dropped.
make: *** No rule to make target '.rfs', needed by 'busybox'. stop.
```

edit the Makefile to delete the following line:

```
KDI_NAME = busybox
```

Then retype the line in. There may be trailing characters on this line that cause errors in the Makefile.

---

## Booting BusyBox Images from a Network

To boot BlueCat Linux with the BusyBox utility from a network using the ROM Monitor firmware, perform the following steps:

1. Copy the `busybox.kdi` file from the `$BLUECAT_PREFIX/demo/busybox/` directory to the `/tftpboot` directory on the cross-development host and rename it according to the BOOTP server configuration:

---

**NOTE:** The name of the boot image must be specified in the BOOTP server configuration file `/etc/bootptab`.

---

```
BlueCat:$ cp $BLUECAT_PREFIX/demo/busybox/\
busybox.kdi /tftpboot/ibm405.kdi
```

2. Select the **Exit menu and continue** option in the main menu by pressing the **0** key.

```
-----
Debugger: Disabled
-----
 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for s1 boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
 0 - Exit menu and continue
->0
```

The BusyBox utility is loaded onto the target board and then automatically started.

## Using BusyBox Utilities

This section provides examples of using the BusyBox utilities. Entering a command from the following list results in the respective output:

- `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 Jul  9  2003 /sbin/reboot
/ # chmod 755 /sbin/reboot
/ # ls -la /sbin/reboot
-rwxr-xr-x  1 0      0          8068 Jul  9  2003 /sbin/reboot

```
- `echo`

```

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

```
- `date`

```

/ # date
Wed Jul 9 23:46:06 UTC 1970

```
- `uname`

```

/ # uname -a
Linux (none) 2.4.18-1 #2 Wed Jul 9 12:48:49 MSD 2003 ppc unknown

```
- `mount`

```

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

```
- `ifconfig`

```

/ # ifconfig eth0 192.168.4.12
/ # ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:04:AC:E3:15:C0
      inet addr:192.168.4.12 Bcast:192.168.4.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:30 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:3430 (3.3 kb) TX bytes:0 (0.0 b)
      Interrupt:15

/ # 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=1.3 ms
64 bytes from 192.168.4.121: icmp_seq=1 ttl=255 time=0.3 ms

```

```
64 bytes from 192.168.4.121: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 192.168.4.121: icmp_seq=3 ttl=255 time=0.2 ms

--- 192.168.4.121 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.5/1.3 ms
```

## Using the TinyLogin RPM Package

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

The following sections describe the steps necessary for creating and booting a BlueCat Linux system containing TinyLogin and demonstrate the use of the TinyLogin utility.

## Creating a BlueCat Linux System for TinyLogin

To create a BlueCat Linux image for TinyLogin, perform the following steps:

1. Create a new directory by typing:

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

2. Set up the BlueCat Linux kernel configuration by using 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
```

Select **Save and Exit** to update the `.config` file, then type the following command:

```
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 a BlueCat Linux 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 Filee
```

---

**NOTE:** In this .spec file, the /bin/login and /usr/bin/passwd symbolic links point to /usr/bin/tinylogin. This allows the user to change his/her password by simply 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

l: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:500:10:0:/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
```

## 12. Create a composite BlueCat Linux image:

```
BlueCat:$ mkboot -m -k tinylogin.disk -f \  
tinylogin.rfs tinylogin.kdi
```

---

**NOTE:** The Makefile for the developer demo system can be used to produce the TinyLogin kernel and RFS images. To do so, modify the Makefile as follows:

i. Change the line `KDI_NAME = developer` to `KDI_NAME = tinylogin`.

ii. Comment out the following lines:

```
# IS_NEED_REBUILD_SRC=$(shell if ! make -q -C src; then echo this ; fi)  
# cd src; make all
```

iii. Change the following lines:

```
clean      :  
    rm -f *.rfs *.tar *.kernel *.disk *.kdi *.srec; cd src; make clean
```

to read as follows:

```
clean      :  
    rm -f *.rfs *.tar *.kernel *.disk *.kdi *.srec
```

iv. Run the `make all` command.

v. If the following message appears:

```
make: circular tinylogin <- tinylogin dependency dropped.  
make: *** No rule to make target `rfs', needed by `tinylogin'. stop.
```

edit the Makefile to delete the following line:

```
KDI_NAME = tinylogin
```

Then retype the line in. There may be trailing characters on this line that cause errors in the Makefile.

---

## Booting the TinyLogin Images from a Network

To boot BlueCat Linux with the TinyLogin utility from a network using the ROM Monitor firmware, perform the following steps:

1. Copy the `tinylogin.kdi` file from the `BLUECAT_PREFIX/demo/tinylogin` directory to the `/tftpboot` directory on the cross-development host and rename it according to the BOOTP server configuration:

---

**NOTE:** The name of the boot image must be specified in the BOOTP server configuration file `/etc/bootptab`.

---

```
BlueCat:$ cp $BLUECAT_PREFIX/demo/tinylogin/\
tinylogin.kdi /tftpboot/ibm405.kdi
```

2. Select the **Exit menu and continue** option in the main menu by pressing the **0** key.

```
-----
Debugger: Disabled
-----
 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for s1 boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
 0 - Exit menu and continue
->0
```

The TinyLogin utility is loaded onto the target board and then automatically started.

## Using the 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

```

- Changing the root password:

```

myhostname login: root
login[16]: 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
logout
myhostname login: root
Password: new_root_password
login[17]: root login on `console'

bash-2.04# exit
logout

```

- Getting the root permissions:

```

myhostname login: guest
Password: guest_password
bash-2.04$ tinylogin su
Password:
login[19]: root login on `console'

bash-2.04#

```

## Using the Zebra RPM Package

GNU Zebra is free software that manages a TCP/IP-based routing protocol. It takes a multiserver and multithread 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.

The following sections describe the steps necessary for creating and booting a BlueCat Linux system containing Zebra and demonstrate the use of the Zebra utility.

## Creating a BlueCat Linux System for Zebra

To create a BlueCat Linux image for Zebra, perform the following steps:

1. Create a new directory by typing:

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

2. Set up the BlueCat Linux kernel configuration by using the standard kernel configuration tools and copy the kernel configuration file to the `$BLUECAT_PREFIX/demo/zebra` directory. For instance, type the following commands:

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

Select **Save and Exit** to update the `.config` file, then type the following command:

```
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 a BlueCat Linux 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
#
led ${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/vtysch           /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"

l: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 192.168.4.12/16

!
! Static default route.
!
ip route 80.240.0.0 255.255.0.0 192.168.4.121

log stdout

```

---

**NOTE:** This configuration file sets the password to `zebra`. The user has to enter this password when 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
```

10. Create a composite BlueCat Linux image:

```
BlueCat:$ mkboot -m -k zebra.disk -f \  
zebra.rfs zebra.kdi
```

---

**NOTE:** The Makefile for the developer demo system can be used to produce the Zebra kernel and RFS images. To do so, modify the Makefile as follows:

i. Change the line `KDI_NAME = developer` to `KDI_NAME = zebra`.

ii. Comment out the following lines:

```
# IS_NEED_REBUILD_SRC=$(shell if ! make -q -C src; then echo this ; fi)  
# cd src; make all
```

iii. Change the following lines:

```
clean      :  
    rm -f *.rfs *.tar *.kernel *.disk *.kdi *.srec; cd src; make clean
```

to read as follows:

```
clean      :  
    rm -f *.rfs *.tar *.kernel *.disk *.kdi *.srec
```

iv. Run the `make all` command.

v. If the following message appears:

```
make: circular zebra <- zebra dependency dropped.  
make: *** No rule to make target '.rfs', needed by 'zebra'. stop.
```

edit the Makefile to delete the following line:

```
KDI_NAME = zebra
```

Then retype the line in. There may be trailing characters on this line that cause errors in the Makefile.

---

## Booting the Zebra Images from a Network

To boot BlueCat Linux with the Zebra utility from a network using the ROM Monitor firmware, perform the following steps:

1. Copy the `zebra.kdi` file from the `$BLUECAT_PREFIX/demo/zebra` directory to the `/tftpboot` directory on the cross-development host and rename it according to the BOOTP server configuration:

---

**NOTE:** The name of the boot image must be specified in the BOOTP server configuration file `/etc/bootptab`.

---

```
BlueCat:$ cp $BLUECAT_PREFIX/demo/zebra/\
zebra.kdi /tftpboot/ibm405.kdi
```

2. Select the **Exit menu and continue** option in the main menu by pressing the **0** key.

```
-----
Debugger: Disabled
-----
 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for sl boot
A - Enable/disable I cache (Enabled )
B - Enable/disable D cache (Disabled)
0 - Exit menu and continue
->0
```

The Zebra utility is loaded onto the target board and then automatically started.

## 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 00:04:AC:E3:15:C0
          inet addr:192.168.4.12  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:15
```

```
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  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 2 192.168.4.121
PING 192.168.4.121 (192.168.4.121) from 192.168.4.12 : 56(84) bytes of
data.
64 bytes from 192.168.4.121: icmp_seq=0 ttl=255 time=1.242 msec
64 bytes from 192.168.4.121: icmp_seq=1 ttl=255 time=182 usec

--- 192.168.4.121 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.182/0.712/1.242/0.530 ms

bash-2.04# ping -c 3 192.168.4.104
PING 192.168.4.104 (192.168.4.104) from 192.168.4.12 : 56(84) bytes of
data.
64 bytes from 192.168.4.104: icmp_seq=0 ttl=255 time=739 usec
64 bytes from 192.168.4.104: icmp_seq=1 ttl=255 time=368 usec
64 bytes from 192.168.4.104: icmp_seq=2 ttl=255 time=376 usec

--- 192.168.4.104 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.368/0.494/0.739/0.173 ms

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

Linux 2.4.18-1 (myhostname) (0)

login: root
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>* 192.168.0.0/16 is directly connected, eth0
S>* 213.24.0.0/16 [1/0] via 192.168.4.121, eth0
Router#
```

Table 5-1 lists the device drivers supported by the walnut BSP and provides important information about them.

**Table 5-1: Device Drivers Supported by the walnut BSP**

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Dual UART Two 16650-compatible devices	<code>serial.c</code>	<code>drivers/char</code>	<code>CONFIG_SERIAL</code> <code>CONFIG_SERIAL_CONSOLE</code>	
Ethernet Controller On-chip EMAC	<code>ibm_ocp*</code>	<code>drivers/net/ibm_ocp</code>	<code>CONFIG_IBM_OCP_ENET</code>	
RTC	<code>genrtc.c</code>	<code>drivers/char</code>	<code>CONFIG_GEN_RTC</code>	
Keyboard PS/2 keyboard	<code>pc_keyb.c</code> <code>keyboard.c</code>	<code>drivers/char</code>	<code>CONFIG_PC_KEYBOARD</code> <code>CONFIG_VT</code>	Not tested explicitly
Mouse PS/2 mouse	<code>pc_keyb.c</code>	<code>drivers/char</code>	<code>CONFIG_MOUSE</code> <code>CONFIG_PSMOUSE</code> <code>CONFIG_VT</code>	Not tested explicitly



This chapter describes the new features of this release.

---

## **Supported Cross-Development Hosts**

The BlueCat Linux development environment requires an installed, functional cross-development host with an Intel 386 or higher CPU. This host needs to be running one of the following development environments:

- Windows 2000/Pro with SP1 or later
- Windows XP
- PC running Red Hat Linux 7.3
- PC running Red Hat Linux 8.0
- PC running Red Hat Linux 9.0

---

## **JFFS2 Support**

This release provides support for Journalling Flash File System version 2 (JFFS2).

The following BlueCat Linux components were updated to support this feature:

- The BlueCat Linux kernel

The BlueCat Linux kernel contains the latest version of JFFS2 available from public domain.

- The BlueCat Linux `mkrootfs` utility  
A new `-j` option is used for `mkrootfs` to create a JFFS2 partition. Refer to `mkrootfs` man page for more information.
- The BlueCat Linux `osloader` demo system  
`osloader` is extended with support for the JFFS2 and now can be used to download a desired BlueCat Linux custom or demo system into the target board's Flash memory using either JFFS or JFFS2. A JFFS2 partition is created in the same way as a JFFS partition. Refer to the *BlueCat Linux User's Guide* for details.

---

## Support of Multithreaded Applications in GDB

This release contains a new version of GDB that supports debugging of multithreaded applications.

---

## Cygwin Execution Environment Version 1.3.6

This release contains a new version of the Cygwin execution environment (1.3.6). It is recommended that users remove any previous version of Cygwin and perform the installation of Cygwin version 1.3.6 as described in the *BlueCat Linux User's Guide*.

This chapter describes known problems and limitations of this release.

---

## Walnut Target Board Problems and Limitations

The following are known problems and limitations of this release:

- Use the following command in order to use Ethernet in the `i_osloader` demo system:

```
make -f Makefile.i xconfig
```

Enable your network card, then type the following command:

```
make -f Makefile.i all
```

The demo now will have a correct Ethernet configuration and can be used to boot other BlueCat Linux demos over the network.

- If `mkrootfs` is terminated (either by an error or by a signal), it tries to clean all its temporary files before exiting. Due to certain features of the Cygwin execution environment, however, 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.
- The `tc1x` 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.

