

BlueCat Linux Board Support Guide

Release 4.0

DOC-0522-02

for ARM Integrator Boards

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

Copyright ©1987 - 2003, 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 ARM Integrator 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
CHAPTER 1	OVERVIEW	1
CHAPTER 2	DOWNLOADING AND BOOTING BLUECAT LINUX ON THE TARGET	3
	Prerequisites	3
	Downloading and Booting Overview	4
	Setting up Hardware	5
	Connecting Target Board Serial Ports to Host	5
	Connecting Target Board Ethernet Card to Host	5
	Installing Boot Loader into Flash on Integrator/AP	6
	Setting up the LynuxWorks Boot Loader Firmware	7
	Downloading a BlueCat Linux System to Flash	8
	Downloading a BlueCat Linux System to Flash Using Boot Loader	8
	Downloading a BlueCat System to Flash Using the OS Loader	9
	Booting a Demo System from Flash	10
	Downloading a BlueCat Linux System via the ARM JTAG Interface	10
	Downloading a BlueCat Linux System to Flash	11
	Booting a Demo System from a Network	13
	Booting a Demo System from a Network Using the Boot Loader Firmware	14
	Booting a Demo System from a Network Using the OS Loader	14

CHAPTER 3	KERNEL CONFIGURATION OPTIONS	17
CHAPTER 4	SUPPORTED DEMO SYSTEMS.....	41
	Demo Systems	41
	developer Demo System	41
	osloader Demo System	42
	showcase Demo System	42
CHAPTER 5	SUPPORTED DEVICE DRIVERS	49
CHAPTER 6	KNOWN LIMITATIONS AND WORKAROUNDS	51
	Integrator/AP Target Board Problems and Limitations	51
	VisualLinux for Integrator/AP Boards	52

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 *LynxOS 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

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

fax (408) 979-3945

LynuxWorks Europe

email 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>

This *BlueCat Linux Board Support Guide for ARM Integrator Boards* provides information about the BlueCat Linux Board Support Package (BSP) supporting ARM Integrator/AP boards with ARM 920T CPU cores.

Throughout this Board Support Guide (BSG), the BSP is referred to as the “integrator.” The target board is referred to as the “Integrator/AP.”

The chapters of this BSG provide the information listed below:

- *Chapter 1* is an overview of the individual chapters.
- *Chapter 2* describes BlueCat Linux downloading and booting procedures for the Integrator/AP target board, using the BlueCat Linux developer demo system as an example.
- *Chapter 3* provides configuration information about the default BlueCat Linux kernel for the integrator BSP.
- *Chapter 4* summarizes BlueCat Linux demo systems supported by the integrator BSP.
- *Chapter 5* provides a list of integrator BSP-supported device drivers, with important information about each of them.
- *Chapter 6* describes the 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 for 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. As such, a basic familiarity with the target board hardware and operation is required before using this guide. 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 ARM configuration and the integrator BSP have been installed on the cross-development host. This means that the user must:

1. Install the BlueCat Linux ARM 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 integrator 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 integrator 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 Integrator/AP target consists of the following main steps:

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

Downloading and booting a BlueCat Linux system can be performed using one of the following:

- The LynuxWorks Boot Loader
- The BlueCat Linux OS Loader demo system
- The ARM JTAG interface

NOTE: The JTAG Interface can be run only from a Windows host.

The LynuxWorks Boot Loader is a firmware intended to act as a boot loader for BlueCat Linux and LynxOS. Boot Loader is able to load BlueCat Linux embedded systems from Flash or over a network and to program them to Flash.

The OS Loader demo system includes the `i_osloader` and `osloader` downloadable images. `osloader` is the image with the base functionality of the BlueCat OS Loader configured in. This includes the ability to download BlueCat images from a TFTP host, execute them in RAM, and other important features.

`i_osloader` is extended with support for the Journalling Flash File System (JFFS) and can thus be used to download a BlueCat Linux custom or demo system into the target board's Flash memory. Please refer to Chapter 3, “Downloading and Booting BlueCat Linux” in the *BlueCat Linux User's Guide* for a discussion of the OS Loader.

The JTAG Interface, also known as the Standard Test Access Port and Boundary-Scan Architecture, is a protocol that can be used for cross-debugging and testing purposes. The JTAG Interface connects to a parallel port on the host and to a JTAG connector on the ARM Integrator board. JTAG allows for more user control in testing and debugging than a standard PROM. JTAG communicates directly with target hardware by establishing control over the CPU.

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 38400.
- The serial port connected to the second target serial port has a baud rate of 57600.

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.

There is an Ethernet port on the target board. The user must use this port to connect the Integrator/AP board to a LAN.

It 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 *development_host_IP* and *target_board_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.

Installing Boot Loader into Flash on Integrator/AP

To install Boot Loader on the Integrator/AP board, the on-board Flash chip must be programmed with the Boot Loader image (\$BLUECAT_PREFIX/boot/integrator.bin).

This can be done using the Boot Monitor software located in the Integrator/AP Boot ROM. Perform the following steps to install Boot Loader using Boot Monitor:

1. Create an S-Record (SREC) image of Boot Loader using the following command:

```
BlueCat:~# arm-lynx-linux-bluecat-objcopy -I \
binary -O srec $BLUECAT_PREFIX/boot/\
integrator.bin integrator.srec \
--change-address=0x2400000
```

2. Set up the Integrator/AP board to boot from the Boot ROM by setting the S1[1] and S1[4] DIP switches to the “ON” position.
3. Connect to the first Integrator/AP console using a terminal application with the following settings:

- Baud rate—38400
- Flow control—None

4. Power up the Integrator/AP board. The Boot Monitor prompt will appear at the first serial console:

```
ARM bootPROM [Version 1.3] Rebuilt on Feb 28 2001 at 11:18:48
Running on a Integrator Evaluation Board
Board Revision V1.0, ARM920T Processor
Memory Size is 64MBytes, Flash Size is 32MBytes
Copyright (c) ARM Limited 1999 - 2001. All rights reserved.
Board designed by ARM Limited
Hardware support provided at http://www.arm.com/
For help on the available commands type ? or h
boot Monitor >
```

5. Type the **L** command at the Boot Loader prompt to load the Boot Loader image:

```
boot Monitor > L
Load Motorola S-Records into flash
Type Ctrl/C to exit loader.
```

- Exit the terminal program connected to the first Integrator/AP serial port and type the following commands at the bash prompt:

```
bash$ stty -F /dev/ttyS0 ospeed 38400
bash$ cat integrator.srec > /dev/ttyS0
```

These commands assume that the host serial port to which the first Integrator/AP serial port is connected is `/dev/ttyS0`. For other ports, replace `/dev/ttyS0` with the proper port name.

- When the `cat` command completes, run the terminal application again and hit **Ctrl-C** in the terminal window:

```
Downloaded 16,386 records in 210 seconds.
Overwritten block/s
0 1 2
boot Monitor >
```

- Boot Loader is programmed to Flash now. To start it, power off the Integrator/AP board and set the S1[1] and S1[3] DIP switches to the “OFF” position (the S1[4] switch must be left “ON”). Power on the board.
- The Boot Loader prompt will now appear at the first Integrator/AP serial console:

```
BootLoader v2.0. Copyright 2002 LynuxWorks, Inc. All rights reserved.
Secondary Bootloader module started from 0x24020000
RAM: 63Mb (0x00100000 - 0x04000000)
SBM: 82Kb (0x00100000 - 0x00114a58)
Malloc pool: 429Kb (0x00114a58 - 0x00180000)
Free memory: 62Mb (0x00180000 - 0x04000000)
integrator>
```

Setting up the LynuxWorks Boot Loader Firmware

To set up the LynuxWorks Boot Loader firmware options for BlueCat Linux operations, perform the following steps:

- Reset the target board.
- At the Boot Loader console, enter the following commands:

```
integrator> set autoboot 0
integrator> set boot_tftp_host_ip \
<development_host_IP>
integrator> set boot_tftp_client_ip \
<target_board_IP>
integrator> set flash_tftp_host_ip \
<development_host_IP>
```

```
integrator> set flash_tftp_client_ip \  
<target_board_IP>  
integrator> save
```

where *<target_board_IP>* is the IP address of the target and
<development_host_IP> is the IP address of the development host.

3. Reset the target board.

Downloading a BlueCat Linux System to Flash

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

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

Downloading a BlueCat Linux System to Flash Using Boot Loader

To download `developer` into the target board using Boot Loader, perform the following steps:

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

```
integrator> set flash_device tftp  
integrator> set flash_tftp_file developer.kdi  
integrator> set flash_target flash0  
integrator> set flash_offset 0x80000  
integrator> flash
```

After these commands have been performed, the `developer` demo system is programmed into the Flash and can be booted as described in “Booting a Demo System from Flash” below.

Downloading a BlueCat System to Flash Using the OS Loader

To download `developer` into the target board using the OS Loader, perform the following steps

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

```
integrator> set boot_device tftp
integrator> set boot_tftp_file i_osloader.kdi
integrator> set boot_os BlueCat
integrator> boot
```

These commands load the `i_osloader` system to RAM and start it. As a result, the BlueCat OS Loader prompt (`>`) will appear in the BlueCat Linux console.

5. At the BlueCat OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_board_IP>
> set HOST <development_host_IP>
> set FILE tftp developer.kdi
> exec flash_fdisk /dev/mtdchar0 4-35
> flash /dev/mtdchar1 erase
> reset
```

where `<target_board_IP>` is the IP address of the target and `<development_host_IP>` is the IP address of the development host.

After these commands have been performed, the `developer` demo system is programmed into the Flash and can be booted as described below.

Booting a Demo System from Flash

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

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

```
integrator> set boot_device flash0
integrator> set boot_flash_offset 0x80000
integrator> set boot_os BlueCat
integrator> boot
```

These commands will start the demo system programmed into the Flash at offset 0x80000.

The Integrator/AP board can be configured to start a demo system programmed into Flash automatically at the board power-up. Use the following commands to prepare the Integrator/AP board to boot BlueCat Linux from Flash automatically:

```
integrator> set boot_device flash0
integrator> set boot_flash_offset 0x80000
integrator> set boot_os BlueCat
integrator> set autoboot 1
integrator> save
```

Downloading a BlueCat Linux System via the ARM JTAG Interface

This section provides instructions for downloading BlueCat Linux demo systems from a cross-development host into a target board’s RAM via the ARM JTAG parallel port interface. The scenario described below assumes the target and host are connected via the JTAG interface. Refer to your JTAG manufacturer documentation.

Downloading a BlueCat Linux System to Flash

To download `i_osloader` on the target board, perform the following steps:

1. Procure the ARM AXD debugger. The ARM Flash utility (AFU), `afu.axf`, is run from the ARM AXD debugger. Copy this utility onto the Windows host.

This utility permits copying a BlueCat Linux image in ELF format into the target board's Flash memory.

2. Prepare an `i_osloader` Kernel Downloadable Image (KDI) suitable for booting from the target board Flash. Type the following command at the BlueCat Linux prompt on the host to create an ELF-formatted version of `i_osloader`:

```
BlueCat:~# ld -o i_osloader -Ttext 0x800000 \
-e 0x800000 -b binary i_osloader.kdi
```

3. Change both the S1[1] DIP and S1[4] DIP switches to the "ON" position.
4. Using the JTAG debugger interface running on the Windows host, download `afu.axf` (the ARM Flash utility) onto the target board RAM.
5. The AFU prompt (`AFU>`) comes up on the JTAG debugger console. There are several commands that can be run from the AFU prompt. For a list of AFU commands, type `help` at the `AFU>` prompt. For a list of all downloaded files, type `list` at the `AFU>` prompt.

```
Copyright ARM Ltd. 1999 - 2001. All rights reserved.
ARM Flash Utility
Program Version 1.3
Date: 18 Jan 2001
Searching for flash devices
Flash device 1 found at 0x20000000 (4 blocks of size 0x20000)
Flash device 2 found at 0x24000000 (256 blocks of size 0x20000)
Device 1 found as Boot device - defaulting to Device 2
Scanning Flash blocks for usage
AFU> list
Listing images in Flash
Scanning Flash blocks for usage
Image 600 Block 0 End Block 28 address 0x24000000 exec 0x00800000 -
name Developer
SIB at Block 255 End Block 255 address 0x25fe0000
AFU>
```

Note the format of the screen output. The list of files downloaded is displayed with the following information: Starting and ending block, name, physical address, and image number.

Of all the fields, the name and the image number are important to note for subsequent steps.

6. From the JTAG debugger console AFU> prompt, program Integrator/AP Flash with the `i_osloader` image. The AFU program command is used for this purpose. The syntax of the command is as follows:

```
AFU> program <image_number> <name> <absolute_path>
```

where `<image_number>` is a user-specified image number chosen from the list of AFU image numbers, `<name>` is a name given by the user to the image to be downloaded, and `<absolute_path>` is the absolute address of the image file.

The following command programs an image named `i_osloader` with an image number of 601:

```
AFU> program 601 i_osloader c:\bc\i_osloader
```

7. Restart the target system from the MultiIce Server software (ARM JTAG-dependent).
8. Use the `boot Monitor>` prompt at the serial console to specify the number of the image to be booted. The following command boots image number 601 programmed in Step 5 above.

```
boot Monitor> bi 601  
  
Current Boot Image = 0  
New Boot Image = 601  
  
boot Monitor>
```

9. Change the S1[1] DIP switch to the “ON” position and the S1[4] DIP switch to the “OFF” position.
10. Restart the system. The BlueCat Linux OS Loader prompt (`>`) should appear after system reset. The screen output below may take up to 20 seconds to appear.

```
Uncompressing  
Linux..... done,  
booting the kernel.  
Linux version 2.4.10-1 (rmelvin@dopey.lynx.com) (gcc version 2.95.3  
20010315 (release)) #10 Mon Jan 28 11:58:25 PST 2002  
Processor: ARM/VLSI Arm920sid(wb) revision 0  
Architecture: ARM-Integrator  
On node 0 totalpages: 16384  
zone(0): 16384 pages.  
zone(1): 0 pages.  
zone(2): 0 pages.  
Kernel command line: root=101  
Calibrating delay loop... 24.88 BogoMIPS  
Memory: 64MB = 64MB total  
Memory: 62392KB available (1304K code, 256K data, 184K init)  
Dentry-cache hash table entries: 8192 (order: 4, 65536 bytes)  
Inode-cache hash table entries: 4096 (order: 3, 32768 bytes)
```

```

Mount-cache hash table entries: 1024 (order: 1, 8192 bytes)
Buffer-cache hash table entries: 4096 (order: 2, 16384 bytes)
Page-cache hash table entries: 16384 (order: 4, 65536 bytes)
POSIX conformance testing by UNIFIX
FIFO_CFG: 0000 FIFO_PRIO: 0000
PCI: bus0: Fast back to back transfers enabled
PCI: bus1: Fast back to back transfers enabled
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Memory clock = 20.000 MHz
Module id: 0
Starting kswapd vl.8
pty: 256 Unix98 ptys configured
Software Watchdog Timer: 0.05, timer margin: 60 sec
block: queued sectors max/low 41378kB/13792kB, 128 slots per queue
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024
blocksize
eepro100.c:v1.09jt-t 9/29/99 Donald Becker
http://cesdis.gsfc.nasa.gov/linux/drivers/eepro100.html
eepro100.c: $Revision: 1.36 $ 2000/11/17 Modified by Andrey V.
Savochkin
<saw@saw.sw.com.sg> and others STANDARD VERSION
eth0: Intel Corporation 82557 [Ethernet Pro 100],
00:02:B3:28:32:7C, IRQ 16.
Board assembly 751767-003, Physical connectors present: RJ45
Primary interface chip i82555 PHY #1.
Secondary interface chip i82555.
General self-test: passed.
Serial sub-system self-test: passed.
Internal registers self-test: passed.
ROM checksum self-test: passed (0x3258698e).
es1371: version v0.30 time 19:59:14 Jan 28 2002
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 4096 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.95 (c) 1998-1999 Rebel.com
RAMDISK: Compressed image found at block 8940
Freeing BlueCat RFS memory: 203K
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 629 blocks [1 disk] into ram disk... done.
VFS: Mounted root (ext2 filesystem).
BlueCat Loader Shell
>

```

The benefit of using `i_osloader` is that from its location in target Flash memory, it enables demos and custom systems to be run from target RAM without having to burn them into Flash each time.

Booting a Demo System from a Network

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

Bootling a Demo System from a Network Using the Boot Loader Firmware

To boot the `developer` demo system over a network using the Boot Loader firmware, perform the following steps:

1. Copy the `developer.kdi` file from the `$BLUECAT_PREFIX/demo/developer` 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:

```
integrator> set boot_device tftp
integrator> set boot_tftp_file developer.kdi
integrator> boot
```

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

Bootling a Demo System from a Network Using the OS Loader

To boot the `developer` demo system over a network using the OS Loader, perform the following steps:

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

```
integrator> set boot_device tftp
integrator> set boot_tftp_file osloader.kdi
integrator> set boot_os BlueCat
integrator> boot
```

These commands start the `osloader` demo system from RAM. As a result, the BlueCat OS Loader prompt (`>`) will appear in the BlueCat Linux console.

5. At the BlueCat OS Loader prompt, enter the following commands:

```
> set IF eth0
> set IP <target_board_IP>
> set HOST <development_host_IP>
> set KERNEL tftp developer.kernel
> set RFS tftp developer.rfs
> set CMD ramdisk_size=28472
> boot
```

where *<target_board_IP>* is the IP address of the target and *<development_host_IP>* is the IP address of the development host. These commands load the *developer* demo system from a network onto the target board and then automatically start it.

Kernel Configuration Options

The integrator BSP comes with a default BlueCat Linux kernel. This kernel has a number of configuration parameters. This chapter details these parameters and their user-configurable options in the tables listed below.

Table 3-1: integrator BSP Kernel Configuration Parameters

Table Number and Configuration Parameter
Table 3-2: Code Maturity Level Options
Table 3-3: Loadable Module Support
Table 3-4: System Type (Processor Type)
Table 3-5: General Setup
Table 3-6: Parallel Port Support
Table 3-7: Memory Technology Devices (MTD)
Table 3-8: RAM/ROM/Flash Chip 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: Multidevice Support (RAID and LVM)
Table 3-15: Networking Options
Table 3-16: QoS and/or Fair Queueing
Table 3-17: Network Device Support
Table 3-18: ARCnet Devices

Table 3-1: integrator BSP Kernel Configuration Parameters (Continued)

Table Number and Configuration Parameter
Table 3-19: Ethernet (10 or 100Mbit)
Table 3-20: Ethernet (1000 Mbit)
Table 3-21: Wireless LAN (non-hamradio)
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: ATA/IDE/MFM/RLL Support
Table 3-27: SCSI Support
Table 3-28: IEEE 1394 (FireWire) Support (Experimental)
Table 3-29: I2O Device Support
Table 3-30: ISDN Subsystem
Table 3-31: Input Core Support
Table 3-32: Character Devices
Table 3-33: Serial Drivers
Table 3-34: I2C Support
Table 3-35: L3 Serial Bus Support
Table 3-36: Mice
Table 3-37: Joysticks
Table 3-38: Watchdog Cards
Table 3-39: Ftape, the Floppy Tape Device Driver
Table 3-40: Multimedia Devices
Table 3-41: File Systems
Table 3-42: Network File Systems
Table 3-43: Partition Types
Table 3-44: Sound
Table 3-45: Multimedia Capabilities Port Drivers

Table 3-1: integrator BSP Kernel Configuration Parameters (Continued)

Table Number and Configuration Parameter
Table 3-46: USB Support
Table 3-47: Bluetooth Support
Table 3-48: Kernel Hacking
Table 3-49: Modular Advanced Power Management
Table 3-50: Messenger Support

Table 3-2: Code Maturity Level Options

Option	Value	Description
CONFIG_EXPERIMENTAL	Y	Prompt for development and/or incomplete code/drivers
CONFIG_OBSOLETE	N	Prompt for obsolete 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: System Type (Processor Type)

Option	Value	Description
CONFIG_ARCH_ANAKIN	Y	ARM system type
CONFIG_CPU_ARM720T	Y	Support ARM720T processor
CONFIG_CPU_ARM920T	Y	Support ARM920T processor

Table 3-4: System Type (Processor Type) (Continued)

Option	Value	Description
CONFIG_CPU_ARM920_CPU_IDLE	Y	ARM920T CPU idle
CONFIG_CPU_ARM920_I_CACHE_ON	Y	ARM920T I-Cache on
CONFIG_CPU_ARM920_D_CACHE_ON	Y	ARM920T D-Cache on
CONFIG_CPU_ARM920_WRITETHROUGH	N	Force write through caches on ARM920T

Table 3-5: General Setup

Option	Value	Description
CONFIG_ANGELBOOT	N	Load kernel using Angel Debug Monitor
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat Ignore printk
CONFIG_BLUECAT_THUMB	N	BlueCat Kernel support for THUMB binaries
CONFIG_BLUECAT_LOADER	N	BlueCat OS Loader
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat small memory footprint
CONFIG_PCI_INTEGRATOR	Y	PCI support
CONFIG_CPU_FREQ	N	Support CPU clock change (Experimental)
CONFIG_PCI_NAMES	Y	PCI device name database
CONFIG_HOTPLUG	N	Support hot-pluggable devices
CONFIG_NET	Y	Networking support
CONFIG_BLUECAT_MEMSIZE	N	Memory sizing benchmarks
CONFIG_SYSVIPC	Y	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_SYSCTL	Y	Sysctl support
CONFIG_FPE_NWFPE	Y	NWFPE math emulation
CONFIG_FPE_FASTFPE	N	FastFPE math emulation (Experimental)
CONFIG_KCORE_ELF	Y	Kernel core (/proc/kcore) format
CONFIG_BINFORMAT_AOUT	M	Kernel support for a.out binaries

Table 3-5: General Setup (Continued)

Option	Value	Description
CONFIG_BINFORMAT_ELF	Y	Kernel support for ELF binaries
CONFIG_BINFORMAT_MISC	M	Kernel support for MISC binaries
CONFIG_PM	N	Power Management support (Experimental)
CONFIG_ARTHUR	N	RISC OS personality
CONFIG_LEDS	Y	Timer and CPU usage LEDs
CONFIG_LEDS_TIMER	Y	Timer LED
CONFIG_LEDS_CPU	Y	CPU usage LED
CONFIG_ALIGNMENT_TRAP	Y	Kernel-mode alignment trap handler

Table 3-6: Parallel Port Support

Option	Value	Description
CONFIG_PARPORT	N	Parallel port support

Table 3-7: Memory Technology Devices (MTD)

Options	Value	Description
CONFIG_MTD	Y	Memory Technology Device (MTD) support
CONFIG_MTD_DEBUG	N	Debugging
CONFIG_MTD_PARTITIONS	N	MTD partitioning support
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 Chip Drivers

Options	Value	Description
CONFIG_MTD_CFI	Y	Detect Flash chips by Common Flash Interface (CFI) probe
CONFIG_MTD_JEDECPROBE	N	Detect non-CFI AMD/JEDEC-compatible Flash chips
CONFIG_MTD_CFI_ADV_OPTIONS	N	Flash chip driver advanced configuration options
CONFIG_MTD_CFI_INTELEXT	Y	Support for Intel/Sharp Flash chips
CONFIG_MTD_CFI_AMDSTD	N	Support for AMD/Fujitsu Flash chips
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_PHYSMAP	N	CFI Flash device in physical memory map
CONFIG_MTD_NORA	N	CFI Flash device mapped on Nora
CONFIG_MTD_ARM_INTEGRATOR	N	CFI Flash device mapped on ARM Integrator/P720T
CONFIG_MTD_CDB89712	N	Cirrus CDB89712 evaluation board mappings
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_BLK_CPQ_DA	N	Compaq SMART2 support

Table 3-13: Block Devices (Continued)

Option	Value	Description
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	Y	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: Multidevice Support (RAID and LVM)

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

Table 3-15: Networking Options

Option	Value	Description
CONFIG_PACKET	Y	Packet socket
CONFIG_PACKET_MMAP	N	Packet socket: mmaped IO
CONFIG_NETLINK	Y	Kernel/User netlink socket
CONFIG_RTNETLINK	N	Routing messages
CONFIG_NETLINK_DEV	Y	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

Table 3-15: Networking Options (Continued)

Option	Value	Description
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	N	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)
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: QoS and/or Fair Queueing

Option	Value	Description
CONFIG_NET_SCHED	N	QoS and/or fair queueing

Table 3-17: Network Device Support

Option	Value	Description
CONFIG_NETDEVICES	Y	Network device 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)
CONFIG_FDDI	N	FDDI driver support
CONFIG_HIPPI	N	HIPPI driver support (Experimental)
CONFIG_PPP	M	PPP (point-to-point protocol) support
CONFIG_PPP_MULTILINK	N	PPP multilink support (Experimental)
CONFIG_PPP_ASYNC	M	PPP support for async serial ports
CONFIG_PPP_SYNC_TTY	M	PPP support for sync tty ports
CONFIG_PPP_DEFLATE	M	PPP Deflate compression
CONFIG_PPP_BSDCOMP	M	PPP BSD-Compress compression
CONFIG_PPPOE	N	PPP over Ethernet (Experimental)
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 3-18: ARCnet Devices

Option	Value	Description
CONFIG_ARCNET	N	ARCnet support

Table 3-19: Ethernet (10 or 100Mbit)

Option	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	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	Y	DECchip Tulip (dc21x4x) PCI support
CONFIG_TULIP_MWI	N	New bus configuration (Experimental)
CONFIG_TULIP_MMIO	N	Use PCI shared mem for NIC registers
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

Table 3-19: Ethernet (10 or 100Mbit) (Continued)

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

Table 3-20: Ethernet (1000 Mbit)

Option	Value	Description
CONFIG_ACENIC	N	Alteon AceNIC/3Com GA620 Gigabit support 3C985/NetGear
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

Table 3-21: Wireless LAN (non-hamradio)

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

Table 3-22: Token Ring Devices

Option	Value	Description
CONFIG_TR	N	Token Ring driver support

Table 3-23: Wan Interfaces

Option	Value	Description
CONFIG_WAN	N	Wan interfaces support

Table 3-24: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur Radio support

Table 3-25: IrDA (Infrared) Support

Option	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

Table 3-26: ATA/IDE/MFM/RLL Support

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

Table 3-27: SCSI Support

Option	Value	Description
CONFIG_SCSI	N	SCSI support?

Table 3-28: IEEE 1394 (FireWire) Support (Experimental)

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

Table 3-29: I2O Device Support

Option	Value	Description
CONFIG_I2O	N	I2O support

Table 3-30: ISDN Subsystem

Option	Value	Description
CONFIG_ISDN	N	ISDN support

Table 3-31: Input Core Support

Option	Value	Description
CONFIG_INPUT	Y	Input core support
CONFIG_INPUT_KEYBDEV	N	Keyboard support
CONFIG_INPUT_MOUSEDEV	N	Mouse support
CONFIG_INPUT_JOYDEV	N	Joystick support
CONFIG_INPUT_EVDEV	N	Event interface support

Table 3-32: Character Devices

Option	Value	Description
CONFIG_VT	N	Virtual terminal
CONFIG_SERIAL	N	Standard/generic (8250/16550 and compatible UARTs) serial support
CONFIG_SERIAL_NONSTANDARD	Y	Nonstandard serial port support
CONFIG_COMPUTONE	N	Computone IntelliPort Plus serial support
CONFIG_ROCKETPORT	N	Control Rocketport support
CONFIG_CYCLADES	N	Cyclades async mux support
CONFIG_DIGIEPCA	N	Digiboard Intelligent Async support
CONFIG_DIGI	N	Digiboard PC/Xx Support
CONFIG_ESP SERIAL	N	Hayes ESP serial port support
CONFIG_MOXA_INTELLIO	N	Moxa Intellio support
CONFIG_MOXA_SMARTIO	N	Moxa SmartIO support
CONFIG_ISI	N	Multi-Tech multiport card support (Experimental)
CONFIG_SYNCLINK	N	Microgate SyncLink card support
CONFIG_N_HDL C	N	HDLC line discipline support
CONFIG_RISCOM8	N	SDL RiSCom/8 card support
CONFIG_SPECIALIX	N	Specialix IO8+ card support
CONFIG_SX	N	Specialix SX (and SI) card support
CONFIG_RIO	N	Specialix RIO system support
CONFIG_STALDRV	N	Stallion multiport serial support
CONFIG_UNIX98_PTYS	Y	Unix98 PTY support
CONFIG_UNIX98_PTY_COUNT	256	Maximum number of Unix98 PTYs in use (0-2048)
CONFIG_QIC02_TAPE	N	QIC-02 tape support
CONFIG_INTEL_RNG	N	Intel i8x0 Random Number Generator support

Table 3-32: Character Devices (Continued)

Option	Value	Description
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
CONFIG_AGP	N	/dev/agpgart (AGP Support)
CONFIG_DRM	N	Direct Rendering Manager (XFree86 DRI support)
CONFIG_MWAVE	N	ACP Modem (Mwave) support

Table 3-33: Serial Drivers

Option	Value	Description
CONFIG_SERIAL_AMBA	Y	ARM AMBA serial port support
CONFIG_SERIAL_AMBA_CONSOLE	Y	Support for console on AMBA serial port
CONFIG_SERIAL_8250	N	8250/16550 and compatible serial support (Experimental)

Table 3-34: I2C Support

Option	Value	Description
CONFIG_I2C	N	I2C support

Table 3-35: L3 Serial Bus Support

Option	Value	Description
CONFIG_L3	N	L3 support

Table 3-36: Mice

Option	Value	Description
CONFIG_BUSMOUSE	N	Bus Mouse Support
CONFIG_MOUSE	N	Mouse Support (not serial and bus mice)

Table 3-37: Joysticks

Option	Value	Description
CONFIG_INPUT_GAMEPORT	N	Game port support
CONFIG_INPUT_SERIO	N	Serial port device support

Table 3-38: Watchdog Cards

Option	Value	Description
CONFIG_WATCHDOG	Y	Watchdog Timer Support
CONFIG_WATCHDOG_NOWAYOUT	N	Disable watchdog shutdown on close
CONFIG_SOFT_WATCHDOG	Y	Software Watchdog
CONFIG_WDT	N	WDT Watchdog timer
CONFIG_WDTPCI	N	WDT PCI Watchdog timer
CONFIG_PCWATCHDOG	N	Berkshire Products PC Watchdog
CONFIG_ACQUIRE_WDT	N	Acquire SBC Watchdog Timer
CONFIG_ADVANTECH_WDT	N	Advantech SBC Watchdog Timer
CONFIG_IB700_WDT	N	IB700 SBC Watchdog Timer
CONFIG_60XX_WDT	N	SBC-60XX Watchdog Timer
CONFIG_W83877F_WDT	N	W83877F (EMACS) Watchdog Timer
CONFIG_MIXCOMWD	N	Mixcom Watchdog
CONFIG_I810_TCO	N	Intel i810 TCO timer / Watchdog
CONFIG_MACHZ_WDT	N	ZF MachZ Watchdog

Table 3-39: Ftape, the Floppy Tape Device Driver

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

Table 3-40: Multimedia Devices

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video For Linux

Table 3-41: 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_EFS_FS	N	EFS file system support (read only) (Experimental)

Table 3-41: File Systems (Continued)

Option	Value	Description
CONFIG_JFFS_FS	N	Journalling Flash File System (JFFS) support
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)
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	N	/dev file system support (Experimental)
CONFIG_DEVPTS_FS	N	/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 3-42: 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
CONFIG_NFS_V3	Y	Provide NFSv3 client support
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 3-43: Partition Types

Option	Value	Description
CONFIG_PARTITION_ADVANCED	N	Advanced partition selection

Table 3-44: Sound

Option	Value	Description
CONFIG_SOUND	Y	Sound support
CONFIG_SOUND_BT878	N	BT878 audio dma
CONFIG_SOUND_CMPCI	N	C-Media PCI (CMI8338/8738)
CONFIG_SOUND_EMU10K1	N	Creative SBLive! (EMU10K1)
CONFIG_SOUND_FUSION	N	Crystal SoundFusion (CS4280/461x)
CONFIG_SOUND_CS4281	N	Crystal Sound CS4281
CONFIG_SOUND_ES1370	N	Ensoniq AudioPCI (ES1370)

Table 3-44: Sound (Continued)

Option	Value	Description
CONFIG_SOUND_ES1371	Y	Creative Ensoniq AudioPCI 97 (ES1371)
CONFIG_SOUND_ESSSOLO1	N	ESS Technology Solo1
CONFIG_SOUND_MAESTRO	N	ESS Maestro, Maestro2, Maestro2E driver
CONFIG_SOUND_MAESTRO3	N	ESS Maestro3/Allegro driver (Experimental)
CONFIG_SOUND_ICH	N	Intel ICH (i8xx) audio support
CONFIG_SOUND_RME96XX	N	RME Hammerfall (RME96XX) support
CONFIG_SOUND_SONICVIBES	N	S3 SonicVibes
CONFIG_SOUND_TRIDENT	N	Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core
CONFIG_SOUND_MSNDCLAS	N	Support for Turtle Beach MultiSound Classic, Tahiti, Monterey
CONFIG_SOUND_MSNDPIN	N	Support for Turtle Beach MultiSound Pinnacle, Fiji
CONFIG_SOUND_VIA82CXXX	N	VIA 82C686 Audio Codec
CONFIG_SOUND_OSS	Y	OSS sound modules
CONFIG_SOUND_TRACEINIT	Y	Verbose initialization
CONFIG_SOUND_DMAP	N	Persistent DMA buffers
CONFIG_SOUND_AD1816	N	AD1816(A) based cards (Experimental)
CONFIG_SOUND_SGALAXY	N	Aztech Sound Galaxy (non-PnP) cards
CONFIG_SOUND_ADLIB	N	Adlib Cards
CONFIG_SOUND_ACI_MIXER	N	ACI mixer (miroSOUND PCM1-pro/PCM12/PCM20)
CONFIG_SOUND_CS4232	N	Crystal CS4232 based (PnP) cards
CONFIG_SOUND_SSCAPE	N	Ensoniq SoundScape support
CONFIG_SOUND_GUS	N	Gravis Ultrasound support
CONFIG_SOUND_VMIDI	N	Loopback MIDI device support
CONFIG_SOUND_TRIX	N	MediaTrix AudioTrix Pro support
CONFIG_SOUND_MSS	N	Microsoft Sound System support

Table 3-44: Sound (Continued)

Option	Value	Description
CONFIG_SOUND_MPU401	N	MPU-401 support (NOT for SB16)
CONFIG_SOUND_NM256	N	NM256AV/NM256ZX audio support
CONFIG_SOUND_MAD16	N	OPTi MAD16 and/or Mozart based cards
CONFIG_SOUND_PAS	N	ProAudioSpectrum 16 support
CONFIG_SOUND_PSS	N	PSS (AD1848, ADSP-2115, ESC614) support
CONFIG_SOUND_SB	N	100% Sound Blaster compatibles (SB16/32/64, ESS, Jazz16) support
CONFIG_SOUND_AWE32_SYNTH	N	AWE32 synth
CONFIG_SOUND_WAVEFRONT	N	Full support for Turtle Beach WaveFront (Trobez Plus, Tropez, Maui) synth/soundcards
CONFIG_SOUND_MAUI	N	Limited support for Turtle Beach Wave Front (Maui, Tropez) synthesizers
CONFIG_SOUND_YM3812	N	Yamaha FM synthesizer (YM3812/OPL-3) support
CONFIG_SOUND_OPL3SA1	N	Yamaha OPL3-SA1 audio controller
CONFIG_SOUND_OPL3SA2	N	Yamaha OPL3-SA2 and SA3 based PnP cards
CONFIG_SOUND_YMFPCI	N	Yamaha YMF7xx PCI audio (native mode)
CONFIG_SOUND_UART6850	N	6850 UART support
CONFIG_SOUND_AEDSP16	N	Gallant Audio Cards (SC-6000 and SC-6600 based)

Table 3-45: Multimedia Capabilities Port Drivers

Option	Value	Description
CONFIG_MCP	N	Multimedia drivers

Table 3-46: USB Support

Option	Value	Description
CONFIG_USB	N	Support for USB
CONFIG_USB_STORAGE_SDDR09	N	SanDisk SDDR-09 (and other SmartMedia) support

Table 3-47: Bluetooth Support

Option	Value	Description
CONFIG_BLUEZ	N	Bluetooth subsystem support

Table 3-48: Kernel Hacking

Option	Value	Description
CONFIG_NO_FRAME_POINTER	N	Compile kernel without frame pointer
CONFIG_DEBUG_ERRORS	N	Verbose kernel error messages
CONFIG_DEBUG_USER	N	Verbose user fault messages
CONFIG_DEBUG_INFO	N	Include debugging information in kernel binary
CONFIG_DEBUG_SLAB	N	Debug memory allocations
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_BLUECAT_KDBG	N	Include kdbg kernel debugger
CONFIG_DEBUG_SPINLOCK	N	Spinlock debugging
CONFIG_DEBUG_LL	N	Kernel low-level debugging functions

Table 3-49: Modular Advanced Power Management

Option	Value	Description
CONFIG_BLUECAT_APM	N	MAPM support

Table 3-50: Messenger Support

Option	Value	Description
CONFIG_BLUECAT_IOPMAN	N	Enable IOP Manager support
CONFIG_BLUECAT_MSNG	N	Enable Messenger support

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

Demo Systems

Table 4-1 lists demo systems supported by the integrator BSP, their default boot devices, and their RAM and ROM requirements.

Table 4-1: Demo Systems Supported by the integrator BSP

Demo System	Default Supported Boot Device(s)	ROM Requirements	RAM Requirements
developer	Ethernet (using the OS loader) Ethernet (using the Boot Loader) Flash	3.6 MB	16384 KB
osloader	Ethernet (using the OS loader) Ethernet (using the Boot Loader) Flash	952 KB	5632 KB
showcase	Ethernet (using the OS loader) Ethernet (using the Boot Loader) Flash	2.9 MB	14336 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. For a description of `developer`, 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 target boards. Refer to Chapter 4, “BlueCat Linux Demo Systems” in the *BlueCat Linux User’s Guide* for details.

i_osloader

This is a slightly larger version of the BlueCat Linux OS Loader. It can be found in the `osloader` demo directory. `i_osloader` is used to download and burn BlueCat Linux images into target Flash memory.

showcase Demo System

This section and its subsections provide information about the `showcase` demo system. The `showcase` demo system showcases certain features specific to the target board and configures an Apache web server. A generic description of `showcase` can be found in Chapter 4, “BlueCat Linux Demo Systems” in the *BlueCat Linux User’s Guide*.

Board-specific features are described below:

DEMO

An Apache web server and Micro Windows demonstration system

SYNOPSIS

This demo system starts and configures the Apache HTTP daemon turning the target board into a web server.

REQUIREMENTS

Storage:	Medium
RAM:	Medium
Network:	Yes
Disk:	None
Special:	None

Kernel Option: When not mounting a Flash File System (FFS) as the root file system, the `ramdisk_size=6000` option must be specified. If an FFS image is installed into target Flash memory to be mounted as a root file system (see

“Downloading and Booting showcase with an FFS as the RFS”), then the

```
integrator_part_conf="12-61" root=1f01
```

options must be specified.

DESCRIPTION

The `showcase` demo system automatically boots up in single-user mode.

Network interfaces are set up in the `.bashrc.integrator` file, located in the `$BLUECAT_PREFIX/demo.integrator/showcase/local/etc` directory. When `showcase` is booted up, the IP address values in this file are automatically used. If changing the default IP address, it is recommended that this be done before building the `showcase` demo.

Now the Apache server is accessible from any networked system using the IP address entered by the user, serving `http://target_IP_address`.

The `showcase` demo system can be downloaded into target Flash memory as either a BlueCat Linux image composed of a kernel image and a compressed root file system, or as a BlueCat Linux image containing a kernel image and root FFS image (as detailed in “Downloading and Booting showcase with an FFS as the RFS”).

Downloading and Booting showcase with an FFS as the RFS

This section describes how to configure and download the `showcase` demo system into target Flash memory with the FFS image mounted as a root file system (RFS).

1. Rebuild the `showcase.kdi` BlueCat Linux image on the cross-development host so it does not include a compressed RFS image. Additionally, the kernel command line in the BlueCat Linux image must have the `integrator_part_conf` and `root` options enabled, and the Flash memory partition configuration string must be in accordance with the argument supplied to the `flash_fdisk` command in Step 3 below.

Please note that the board-specific Flash mapping driver (`integrator-flash.c`) assigns `/dev/mtd{char,block}<n>` devices to the AFS partitions used by the ARM Firmware Suite and ARM Boot Monitor. The partitions created by the `integrator_part_conf` kernel configuration option (BlueCat partitions) have numbers that go beyond the AFS partition numbers.

So, by default, the BlueCat partitions created by the `integrator_part_conf` kernel configuration option at run time are numbered not from 1, but from 1 + (number of AFS partitions). This also applies to the `root=1f<nn>` kernel configuration option.

To solve this problem, support for the AFS partitions must be turned off in the board-specific Flash mapping driver. To do so, replace the line:

```
ret = parse_afs_partitions(mtd, &parts);
```

in the `$BLUECAT_PREFIX/usr/src/linux/drivers/mtd/maps/integrator-flash.c` file with:

```
ret = 0;
```

Also, make sure that support for the Flash device installed on the board as well as the JFFS feature are both enabled in the kernel. To do so, change to the `showcase` demo system directory and run the kernel configuration tool:

```
BlueCat:$ cd $BLUECAT_PREFIX/demo/showcase
BlueCat:$ make xconfig
```

Change the **Memory Technology Devices (MTD)/RAM/ROM/Flash chip drivers/Support for Intel/Sharp flash chips** option to **Y**. Set the **File systems/Journalling Flash File System (JFFS) support** option to **Y**. Exit the menu, saving all changes.

Finally, the following commands, when issued on the cross-development host, rebuilds the BlueCat Linux kernel, ensuring that all the changes above take effect and sets the appropriate kernel command line:

```
BlueCat:$ make clean; make all
BlueCat:$ echo "integrator_part_conf=\
\"12-61\" root=1f01" | mkboot -m -k \
showcase.kernel -c stdin showcase.kdi
```

2. While still on the cross-development host, copy `showcase.jffs` and `showcase.kdi` to the `tftpboot` directory, using the following command:

```
BlueCat:$ cp showcase.jffs showcase.kdi /tftpboot
```

3. Boot the BlueCat OS loader (`i_osloader`) onto the target. From the target system console, create two partitions in target Flash memory, one for the kernel image and the other for the root file system image by entering the following command at the `i_osloader` (BLOSH) prompt on the target board:

```
> exec flash_fdisk /dev/mtdchar0 4-11:12-61
```

This command creates a 1 MB first partition for the kernel at the beginning of target Flash memory and a 6.25 MB second partition for the root file system image immediately following the first partition.

4. Set the environment variables to properly configure network access by entering the following commands:
 - > **set IP** *target_IP_address*
 - > **set HOST** *host_IP_address*
 - > **set IF** **eth0**
5. Set the `FILE` environment variable so that it points to the `showcase` FFS image by entering the following command:
 - > **set FILE** **tftp /tftpboot/showcase.jffs**
6. Burn the FFS image into target Flash memory by entering the following command:
 - > **flash /dev/mtdchar2 erase**
7. Set the `FILE` environment variable so that it points to the `showcase` BlueCat Linux kernel image, for example, by entering the following command:
 - > **set FILE** **tftp /tftpboot/showcase.kdi**

NOTE: To download and boot other custom systems with FFS mounted as root, the kernel needs to be rebuilt with FFS and Memory Technology Device (MTD) enabled.

8. Burn the BlueCat Linux kernel image into target Flash memory by entering the following command:
 - > **flash /dev/mtdchar1 erase**
9. Reset the target board by entering the following command:
 - > **reset**
10. Follow the procedure in “Booting a Demo System from Flash” in Chapter 2 to start the `showcase` demo.

NOTE: Because the kernel must scan the entire Flash partition, mounting the `showcase` FFS image as a root file system takes about 40 seconds during the kernel bootstrap process.

Modifying Target Board IP and Gateway Addresses

The `showcase` demo system includes an Apache web server feature.

The Apache web server is a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (web) server. See <http://www.apache.org> for further information.

Users may wish to define a unique IP address for their target board rather than use the default address (172.17.1.218) defined in the `showcase` demo system. To change the default IP address for the `showcase` demo system, open the `showcase` demo system's `.bashrc.integrator` file with any text editor (such as `vi`) and perform the following steps:

1. Change directory to the location of the `.bashrc.integrator` file:

```
BlueCat:$ cd $BLUECAT_PREFIX/demo.integrator/\
showcase/local/etc
```

2. Change the `showcase` demo system's `.bashrc.integrator` default target board IP address (172.17.1.218) to a new user-selected one (for example, 216.100.252.140) by editing it as follows:

Find this line:

```
TARGET_IP=172.17.1.218
```

Edit it to read as follows:

```
TARGET_IP=216.100.252.140
```

3. Set a gateway IP address by entering a value at this line in the `.bashrc.integrator` file as well:

```
GATE_IP=
```

4. Rebuild the `showcase` demo system by entering the following commands in the `$BLUECAT_PREFIX/demo.integrator/showcase` directory:

```
Bluecat:$ touch showcase.spec
Bluecat:$ make rootfs
Bluecat:$ make kdi
```

5. Copy the updated `showcase.kdi` to the `tftpboot` directory to replace the old `showcase.kdi`.

Now when the showcase demo system is downloaded onto the target board, the Apache web server BlueCat Linux web pages can be accessed at the user's customized IP address, as shown in the figure below:

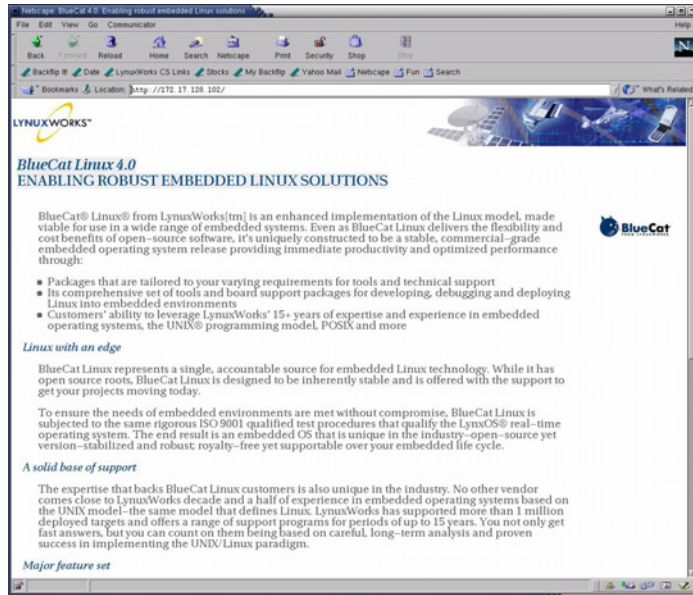


Figure 4-1: BlueCat Linux Apache Web Server Web Page

Table 5-1 lists the device drivers supported by the integrator BSP and details important information about them:

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

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options
UART	serial_amba.c	drivers/serial	CONFIG_SERIAL_AMBA CONFIG_SERIAL_AMBA_CONSOLE
Ethernet Intel PRO/100	eeepro100.c	drivers/net	CONFIG_EEPRO100
Flash 28F32053	integrator-flash.c	drivers/mtd/maps	CONFIG_MTD_ARM_INTEGRATOR
Keyboard PS/2	amba_kmi_keyb.c	drivers/char	CONFIG_KMI_KEYB
Mouse PS/2	amba_kmi_keyb.c	drivers/char	CONFIG_KMI_MOUSE
Video NEC PowerVR NEON 250	*.c	drivers/video/neon	CONFIG_FB_NEON

Known Limitations and Workarounds

Integrator/AP Target Board Problems and Limitations

- The developer demo has incorrect device nodes for the target serial ports. To resolve this problem, edit the `developer.spec` file by replacing the following lines:

```
mknod /dev/ttyS0 c 4 64
mknod /dev/ttyS1 c 4 65
```

with:

```
mknod /dev/ttyS0 c 204 16
mknod /dev/ttyS1 c 204 17
```

and then rebuild the `developer` demo system.

- The default rate of the target serial ports is 38400 bps. Make sure that an application that connects to the target over a serial line is configured to use this rate. For example, for the cross debugger do the following:

```
gdb -b 38400
```

- 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.

- The `showcase` demo requires a VGA card, PS/2 mouse, and PS/2 keyboard connected in order to run. The demo does not print any Linux boot-up messages to the serial console.

- 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.
- Debugging of multithreaded applications via GDB is not supported.
- 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.

VisualLynux for Integrator/AP Boards

This distribution of BlueCat Linux can be used with LynuxWorks' VisualLynux version 2.1.1 or higher.

VisualLynux 2.1.1 needs special configuration data for all installed BlueCat Linux Cross-Development Tools (CDTs). (For information on configuring BlueCat Linux CDTs for VisualLynux 4.0 and higher, see the *VisualLynux 4.0 Release Notes*.)

1. Invoke the VisualLynux Configuration Utility. It should automatically be invoked during VisualLynux installation. Otherwise, use the following:

Start->Programs->VisualLynux->VisualLynux Configurations

2. The VisualLynux Configuration Utility dialog box appears. Check the box for the appropriate BlueCat Linux CDT and select the **BlueCat CDT(s)** page.
3. From the drop-down list for **Current OS version**, select **4.0**.
4. Click on **BlueCat 4.0 for ARM (elf)** in the **BlueCat 4.0 Cross Development Tools** list. Check the box left of the entry.
5. Enter the name of the CDT installation directory. You may also use the **Browse...** button.
6. Click **OK** to write this data to the Registry.

NOTE: The location of the cross `gdb` tool has been changed in BlueCat Linux 4.0 version CDTs. You must manually change the following Registry entry *after* running the VisualLinux Configuration Utility:

```
HKEY_LOCAL_MACHINE\Software\LynuxWorks, Inc.\  
BlueCat\4.0\arm\elf\LynuxWorks Debugger
```

The key `Path` should specify the full pathname for the cross `gdb`.

```
installation_point/cdt/bin/gdb.exe
```

where *installation_point* is the full pathname of the BlueCat Linux CDT installation directory.
