Using the Bay Command Console (BCC)

BayRS Version 13.00
Site Manager Software Version 7.00

BCC Version 4.05

Part No. 303562-A Rev. 00
October 1998

Bay Networks *Where Information Flows.™*

Bay Networks Where Information Flows.

4401 Great America Parkway
Santa Clara, CA 95054

8 Federal Street
Billerica, MA 01821

Copyright © 1998 Bay Networks, Inc.

All rights reserved. Printed in the USA. October 1998.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Bay Networks, Inc.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license. A summary of the Software License is included in this document.

Trademarks

ACE, AN, BCN, BLN, BN, FRE, Optivity, PPX, Quick2Config, and Bay Networks are registered trademarks and Advanced Remote Node, ANH, ARN, ASN, BayRS, BaySecure, BayStack, BCC, SPEX, System 5000, and the Bay Networks logo are trademarks of Bay Networks, Inc.

Microsoft, MS, MS-DOS, Win32, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

All other trademarks and registered trademarks are the property of their respective owners.

Restricted Rights Legend

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Notwithstanding any other license agreement that may pertain to, or accompany the delivery of, this computer software, the rights of the United States Government regarding its use, reproduction, and disclosure are as set forth in the Commercial Computer Software-Restricted Rights clause at FAR 52.227-19.

Statement of Conditions

In the interest of improving internal design, operational function, and/or reliability, Bay Networks, Inc. reserves the right to make changes to the products described in this document without notice.

Bay Networks, Inc. does not assume any liability that may occur due to the use or application of the product(s) or circuit layout(s) described herein.

Portions of the code in this software product may be Copyright © 1988, Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms of such portions are permitted, provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that such portions of the software were developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from such portions of the software without specific prior written permission.

SUCH PORTIONS OF THE SOFTWARE ARE PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

In addition, the program and information contained herein are licensed only pursuant to a license agreement that contains restrictions on use and disclosure (that may incorporate by reference certain limitations and notices imposed by third parties).

Bay Networks, Inc. Software License Agreement

NOTICE: Please carefully read this license agreement before copying or using the accompanying software or installing the hardware unit with pre-enabled software (each of which is referred to as "Software" in this Agreement). BY COPYING OR USING THE SOFTWARE, YOU ACCEPT ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. THE TERMS EXPRESSED IN THIS AGREEMENT ARE THE ONLY TERMS UNDER WHICH BAY NETWORKS WILL PERMIT YOU TO USE THE SOFTWARE. If you do not accept these terms and conditions, return the product, unused and in the original shipping container, within 30 days of purchase to obtain a credit for the full purchase price.

1. License Grant. Bay Networks, Inc. ("Bay Networks") grants the end user of the Software ("Licensee") a personal, nonexclusive, nontransferable license: a) to use the Software either on a single computer or, if applicable, on a single authorized device identified by host ID, for which it was originally acquired; b) to copy the Software solely for backup purposes in support of authorized use of the Software; and c) to use and copy the associated user manual solely in support of authorized use of the Software by Licensee. This license applies to the Software only and does not extend to Bay Networks Agent software or other Bay Networks software products. Bay Networks Agent software or other Bay Networks software products are licensed for use under the terms of the applicable Bay Networks, Inc. Software License Agreement that accompanies such software and upon payment by the end user of the applicable license fees for such software.

2. Restrictions on use; reservation of rights. The Software and user manuals are protected under copyright laws. Bay Networks and/or its licensors retain all title and ownership in both the Software and user manuals, including any revisions made by Bay Networks or its licensors. The copyright notice must be reproduced and included with any copy of any portion of the Software or user manuals. Licensee may not modify, translate, decompile, disassemble, use for any competitive analysis, reverse engineer, distribute, or create derivative works from the Software or user manuals or any copy, in whole or in part. Except as expressly provided in this Agreement, Licensee may not copy or transfer the Software or user manuals, in whole or in part. The Software and user manuals embody Bay Networks' and its licensors' confidential and proprietary intellectual property. Licensee shall not sublicense, assign, or otherwise disclose to any third party the Software, or any information about the operation, design, performance, or implementation of the Software and user manuals that is confidential to Bay Networks and its licensors; however, Licensee may grant permission to its consultants, subcontractors, and agents to use the Software at Licensee's facility, provided they have agreed to use the Software only in accordance with the terms of this license.

3. Limited warranty. Bay Networks warrants each item of Software, as delivered by Bay Networks and properly installed and operated on Bay Networks hardware or other equipment it is originally licensed for, to function substantially as described in its accompanying user manual during its warranty period, which begins on the date Software is first shipped to Licensee. If any item of Software fails to so function during its warranty period, as the sole remedy Bay Networks will at its discretion provide a suitable fix, patch, or workaround for the problem that may be included in a future Software release. Bay Networks further warrants to Licensee that the media on which the Software is provided will be free from defects in materials and workmanship under normal use for a period of 90 days from the date Software is first shipped to Licensee. Bay Networks will replace defective media at no charge if it is returned to Bay Networks during the warranty period along with proof of the date of shipment. This warranty does not apply if the media has been damaged as a result of accident, misuse, or abuse. The Licensee assumes all responsibility for selection of the Software to achieve Licensee's intended results and for the installation, use, and results obtained from the Software. Bay Networks does not warrant a) that the functions contained in the software will meet the Licensee's requirements, b) that the Software will operate in the hardware or software combinations that the Licensee may select, c) that the operation of the Software will be uninterrupted or error free, or d) that all defects in the operation of the Software will be corrected. Bay Networks is not obligated to remedy any Software defect that cannot be reproduced with the latest Software release. These warranties do not apply to the Software if it has been (i) altered, except by Bay Networks or in accordance with its instructions; (ii) used in conjunction with another vendor's product, resulting in the defect; or (iii) damaged by improper environment, abuse, misuse, accident, or negligence. THE FOREGOING WARRANTIES AND LIMITATIONS ARE EXCLUSIVE REMEDIES AND ARE IN LIEU OF ALL OTHER WARRANTIES EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Licensee is responsible for the security of

its own data and information and for maintaining adequate procedures apart from the Software to reconstruct lost or altered files, data, or programs.

4. Limitation of liability. IN NO EVENT WILL BAY NETWORKS OR ITS LICENSORS BE LIABLE FOR ANY COST OF SUBSTITUTE PROCUREMENT; SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES; OR ANY DAMAGES RESULTING FROM INACCURATE OR LOST DATA OR LOSS OF USE OR PROFITS ARISING OUT OF OR IN CONNECTION WITH THE PERFORMANCE OF THE SOFTWARE, EVEN IF BAY NETWORKS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL THE LIABILITY OF BAY NETWORKS RELATING TO THE SOFTWARE OR THIS AGREEMENT EXCEED THE PRICE PAID TO BAY NETWORKS FOR THE SOFTWARE LICENSE.

5. Government Licensees. This provision applies to all Software and documentation acquired directly or indirectly by or on behalf of the United States Government. The Software and documentation are commercial products, licensed on the open market at market prices, and were developed entirely at private expense and without the use of any U.S. Government funds. The license to the U.S. Government is granted only with restricted rights, and use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1) of the Commercial Computer Software—Restricted Rights clause of FAR 52.227-19 and the limitations set out in this license for civilian agencies, and subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013, for agencies of the Department of Defense or their successors, whichever is applicable.

6. Use of Software in the European Community. This provision applies to all Software acquired for use within the European Community. If Licensee uses the Software within a country in the European Community, the Software Directive enacted by the Council of European Communities Directive dated 14 May, 1991, will apply to the examination of the Software to facilitate interoperability. Licensee agrees to notify Bay Networks of any such intended examination of the Software and may procure support and assistance from Bay Networks.

7. Term and termination. This license is effective until terminated; however, all of the restrictions with respect to Bay Networks' copyright in the Software and user manuals will cease being effective at the date of expiration of the Bay Networks copyright; those restrictions relating to use and disclosure of Bay Networks' confidential information shall continue in effect. Licensee may terminate this license at any time. The license will automatically terminate if Licensee fails to comply with any of the terms and conditions of the license. Upon termination for any reason, Licensee will immediately destroy or return to Bay Networks the Software, user manuals, and all copies. Bay Networks is not liable to Licensee for damages in any form solely by reason of the termination of this license.

8. Export and Re-export. Licensee agrees not to export, directly or indirectly, the Software or related technical data or information without first obtaining any required export licenses or other governmental approvals. Without limiting the foregoing, Licensee, on behalf of itself and its subsidiaries and affiliates, agrees that it will not, without first obtaining all export licenses and approvals required by the U.S. Government: (i) export, re-export, transfer, or divert any such Software or technical data, or any direct product thereof, to any country to which such exports or re-exports are restricted or embargoed under United States export control laws and regulations, or to any national or resident of such restricted or embargoed countries; or (ii) provide the Software or related technical data or information to any military end user or for any military end use, including the design, development, or production of any chemical, nuclear, or biological weapons.

9. General. If any provision of this Agreement is held to be invalid or unenforceable by a court of competent jurisdiction, the remainder of the provisions of this Agreement shall remain in full force and effect. This Agreement will be governed by the laws of the state of California.

Should you have any questions concerning this Agreement, contact Bay Networks, Inc., 4401 Great America Parkway, P.O. Box 58185, Santa Clara, California 95054-8185.

LICENSEE ACKNOWLEDGES THAT LICENSEE HAS READ THIS AGREEMENT, UNDERSTANDS IT, AND AGREES TO BE BOUND BY ITS TERMS AND CONDITIONS. LICENSEE FURTHER AGREES THAT THIS AGREEMENT IS THE ENTIRE AND EXCLUSIVE AGREEMENT BETWEEN BAY NETWORKS AND LICENSEE, WHICH SUPERSEDES ALL PRIOR ORAL AND WRITTEN AGREEMENTS AND COMMUNICATIONS BETWEEN THE PARTIES PERTAINING TO THE SUBJECT MATTER OF THIS AGREEMENT. NO DIFFERENT OR ADDITIONAL TERMS WILL BE ENFORCEABLE AGAINST BAY NETWORKS UNLESS BAY NETWORKS GIVES ITS EXPRESS WRITTEN CONSENT, INCLUDING AN EXPRESS WAIVER OF THE TERMS OF THIS AGREEMENT.

Contents

Preface

Before You Begin xiii
Text Conventions xiv
Acronyms xvi
Bay Networks Technical Publications xvi
How to Get Help xvii

Chapter 1
Overview of the BCC

Platform Requirements 1-2
Number of BCC Sessions 1-2
Terminology and Concepts 1-3
Configuration Hierarchy 1-3
Context 1-5
Objects, Classes, and Instances 1-6
BCC Instance Identifier 1-7
Global (Box-Wide) Objects 1-7
Physical Device Objects 1-8
Parameters 1-10
Required Parameters 1-10
Derived Parameters 1-10
Optional Parameters 1-10

Chapter 2
Getting Started with the BCC

Entering and Exiting the BCC Interface 2-1
Displaying Context 2-3
Navigating in Configuration Mode 2-3
Navigating with the back Command 2-3
Navigating with Configuration Commands 2-4

Moving Back One or More Levels 2-4
Moving Back to Root Level 2-5
Moving Forward One or More Levels 2-5
Moving to Any Context in the Device Configuration 2-6
Displaying Configuration Data 2-8
Displaying the Total Device Configuration 2-8
Displaying the Configuration of One Object 2-10
Using Online Help 2-11
Help Commands 2-11
Help Command Examples 2-13

Chapter 3
Entering Commands and Using Command Files

Entering Commands 3-1
Using Command Abbreviations 3-2
Recalling Commands 3-2
Editing Command Lines 3-3
Entering Multiple Commands on a Line 3-4
Continuing a Command Line 3-4
System Commands 3-5
Configuration Command Syntax 3-5
Command Syntax Requirements 3-5
Using Basic (Full) Syntax 3-6
Using Default Syntax 3-7
Using Abbreviated Syntax 3-7
Specifying Parameter Values 3-8
Required, Derived, and Other Parameters 3-8
Specifying Multiple Parameter-Value Pairs 3-8
Disabling, Reenabling, and Deleting a Configured Object 3-10
Creating and Using BCC Command Files 3-10
Saving Commands to a File 3-11
Adding Comments to a Command File 3-11
Entering Configuration Commands from a File 3-12
Saving the Active Configuration as a Bootable Binary File 3-12

Chapter 4
Tutorial: Configuring a Network Device

Creating and Modifying a Device Configuration 4-1
Disabling a Configured Object 4-15
Enabling a Configured Object 4-16
Deleting a Configured Object 4-17

Appendix A
System Commands

Appendix B
TCL Support

Appendix C
System show Commands

show console C-2
show hardware C-5
show process C-7
show system C-10

Appendix D
Syntax for Module Location

Index

Figures

Figure 1-1. The Technician Interface and the BCC Interface 1-1
Figure 1-2. Sample BCC Configuration 1-4
Figure 1-3. Configuring IP and RIP on an Ethernet Interface 1-5
Figure 1-4. Configuring IP and RIP on an Ethernet Interface 1-6
Figure 2-1. Moving Away from Root Level 2-6
Figure 2-2. Navigating to an Object in the Configuration 2-7
Figure 2-3. Navigating with the BCC Recursive Search Feature 2-8
Figure 2-4. Objects You Can Configure at the Next Level 2-15
Figure 4-1. Sample BCC Configuration (BCN Router) 4-3
Figure 4-2. Typical BCC Configuration Cycle 4-4

Tables

Table 2-1. BCC Help Commands 2-11
Table 3-1. Keystrokes for Editing BCC Command Lines 3-3
Table 3-2. BCC Commands 3-10
Table A-1. System Commands A-2
Table D-1. Syntax for Specifying Module Location per Device D-1

Preface

The BCC is a command-line interface for configuring Bay Networks devices. If you are responsible for configuring and managing Bay Networks® AN®, ANH™, ARN™, ASN™, BN® (BCN® and BLN®), or System 5000™ routers, read this guide to learn how to use the Bay Command Console (BCC™).

Before You Begin

This guide is intended for users who have some experience supporting a multivendor internetworking system. You should be able to perform network device configuration, maintenance, and troubleshooting.

Because the BCC makes real-time changes to device configuration, Bay Networks recommends that you first learn about BCC behavior on a device not connected to your production network.

Text Conventions

This guide uses the following text conventions:

angle brackets (< >)	Indicate that you choose the text to enter based on the description inside the brackets. Do not type the brackets when entering the command. Example: If the command syntax is: **ping** *<ip_address>*, you enter: **ping 192.32.10.12**
bold text	Indicates text that you need to enter and command names and options. Example: Enter **show ip** {**alerts** | **routes**} Example: Use the **dinfo** command.
braces ({})	Indicate required elements in syntax descriptions where there is more than one option. You must choose only one of the options. Do not type the braces when entering the command. Example: If the command syntax is: **show ip** {**alerts** | **routes**}, you must enter either: **show ip alerts** or **show ip routes**.
brackets ([ ])	Indicate optional elements in syntax descriptions. Do not type the brackets when entering the command. Example: If the command syntax is: **show ip interfaces** [**-alerts**], you can enter either: **show ip interfaces** or **show ip interfaces -alerts**.
ellipsis points (. . .)	Indicate that you repeat the last element of the command as needed. Example: If the command syntax is: **ethernet/2/1** [<*parameter*> <*value*>] . . ., you enter **ethernet/2/1** and as many parameter-value pairs as needed.

italic text	Indicates file and directory names, new terms, book titles, and variables in command syntax descriptions. Where a variable is two or more words, the words are connected by an underscore. Example: If the command syntax is: **show at** <*valid_route*> *valid_route* is one variable and you substitute one value for it.
`screen text`	Indicates system output, for example, prompts and system messages. Example: `Set Bay Networks Trap Monitor Filters`
vertical line (|)	Separates choices for command keywords and arguments. Enter only one of the choices. Do not type the vertical line when entering the command. Example: If the command syntax is: **show ip** {**alerts** | **routes**}, you enter either: **show ip alerts** or **show ip routes**, but not both.

Acronyms

ARP	Address Resolution Protocol
ATM	asynchronous transfer mode
DCM	data collection module
DRAM	dynamic random access memory
IP	Internet Protocol
IPX	Internetwork Packet Exchange
LAN	local area network
MAC	media access control
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
SNMP	Simple Network Management Protocol
SRM-L	system resource module-link
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
WAN	wide area network

Bay Networks Technical Publications

You can now print Bay Networks technical manuals and release notes free, directly from the Internet. Go to *support.baynetworks.com/library/tpubs/*. Find the Bay Networks product for which you need documentation. Then locate the specific category and model or version for your hardware or software product. Using Adobe Acrobat Reader, you can open the manuals and release notes, search for the sections you need, and print them on most standard printers. You can download Acrobat Reader free from the Adobe Systems Web site, *www.adobe.com*.

You can purchase Bay Networks documentation sets, CDs, and selected technical publications through the Bay Networks Collateral Catalog. The catalog is located on the World Wide Web at *support.baynetworks.com/catalog.html* and is divided into sections arranged alphabetically:

- The “CD ROMs” section lists available CDs.
- The “Guides/Books” section lists books on technical topics.
- The “Technical Manuals” section lists available printed documentation sets.

Make a note of the part numbers and prices of the items that you want to order. Use the “Marketing Collateral Catalog description” link to place an order and to print the order form.

How to Get Help

For product assistance, support contracts, or information about educational services, go to the following URL:

http://www.baynetworks.com/corporate/contacts/

Or telephone the Bay Networks Technical Solutions Center at:

800-2LANWAN

Chapter 1
Overview of the BCC

The BCC is a command-line interface for configuring Bay Networks devices. After logging on to a device, you access the BCC by entering the **bcc** command at the Technician Interface prompt (Figure 1-1).

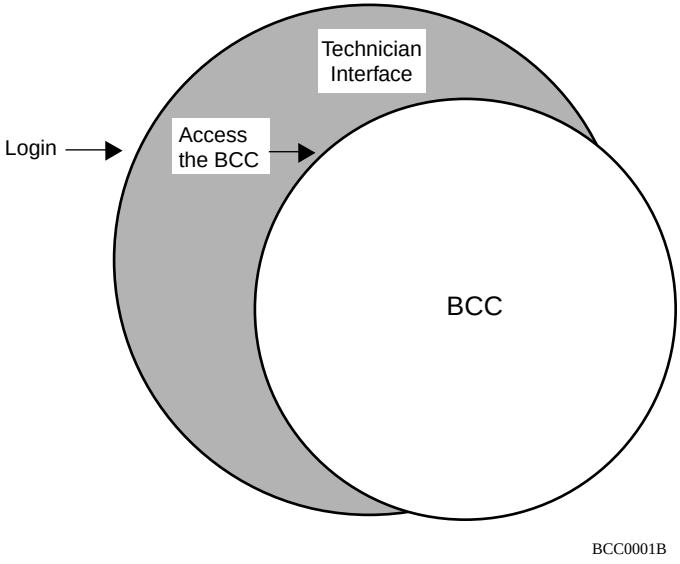

Figure 1-1. The Technician Interface and the BCC Interface

From the bcc> prompt, you can:

- Execute any system command not classified as “Technician Interface only” (see Appendix A, “System Commands”).
- Execute configuration commands to perform tasks such as creating or deleting IP interfaces on the router. Enter BCC configuration mode by entering the **config** command at the bcc> prompt.

Note: For a list of services you can configure using the BCC, see the *Release Notes*. You can obtain a complete hierarchical listing of all objects configurable on a device by entering the **help tree -all** command at any BCC prompt.

Platform Requirements

The BCC runs on AN, ANH, ARN, ASN , System 5000, and BN platforms including ARE, FRE®, and FRE-2 processor modules. Each slot must have:

- 16 MB of dynamic RAM (DRAM)
- 2 MB of free memory space available when you start the BCC

If you try to start the BCC with insufficient DRAM or free memory on a slot, the BCC returns an error message. In that case, use Site Manager instead of the BCC.

Number of BCC Sessions

You can open one BCC session per slot in read-write (configuration) mode. Other users can open additional BCC sessions in read-only (nonconfiguration) mode on the same slot, depending on available memory. Each BCC session is mutually exclusive. If you make a change during a BCC session in read-write mode, this change does not appear in other BCC sessions.

Terminology and Concepts

This section describes key terms and concepts of the BCC interface.

Configuration Hierarchy

The BCC configuration hierarchy begins at a root-level object, called *box* for AN/ANH, ARN, and BN platforms, and *stack* for ASN and System 5000 platforms. Under the root-level object are branch objects such as interfaces and protocols that fan out from the root level in a tree hierarchy.

You use the **help tree -all** and **show config -all** commands to display the configuration hierarchy of a Bay Networks router.

- The **help tree -all** command displays the hierarchy of every object you can configure.
- The **show config -all** command displays the hierarchy of objects you have configured.

Figure 1-2 illustrates a sample BCC configuration for an AN, BN, or ARN router.

- Box contains the global objects IP, SNMP, Telnet, and TFTP, plus the physical interface objects ethernet/2/1, ethernet/2/2, and serial/3/1.
- Global IP contains the global objects ARP, RIP, and OSPF.

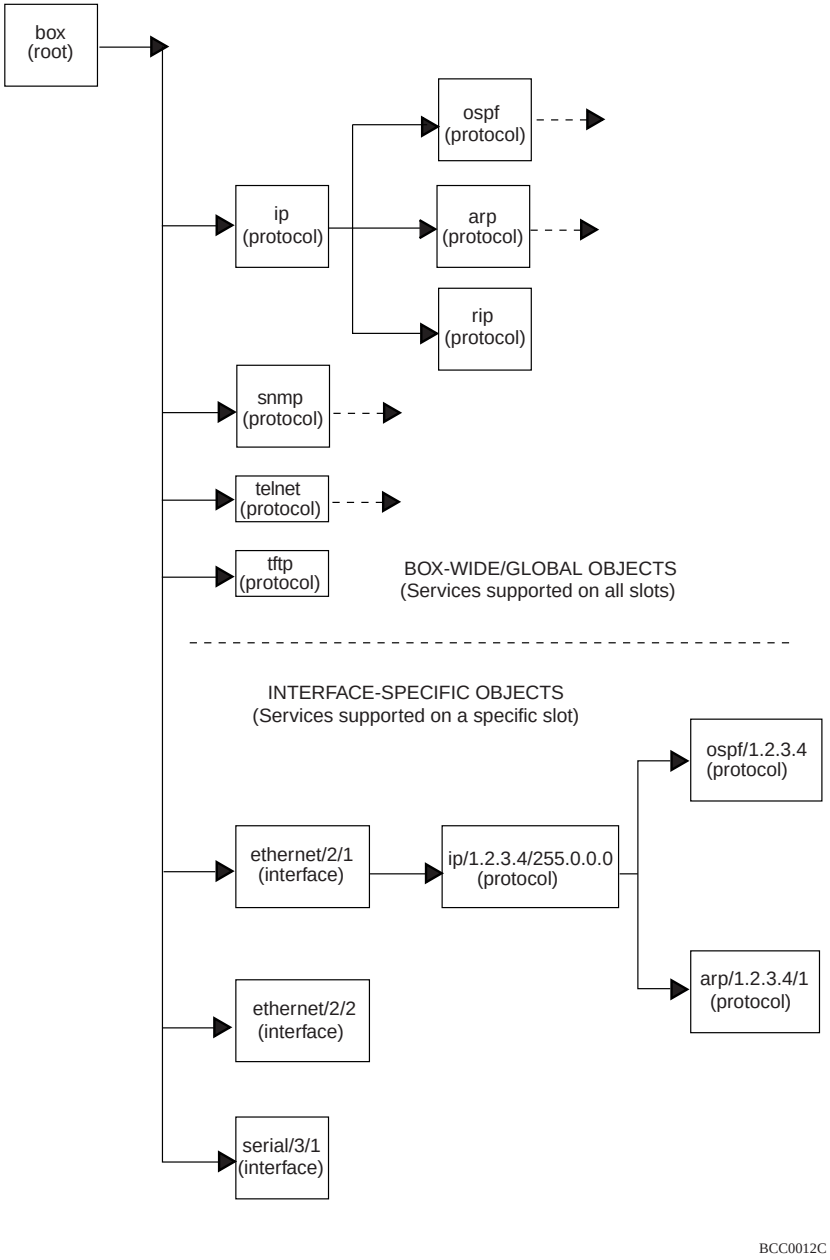

Figure 1-2. Sample BCC Configuration

You use BCC commands to create new objects and to modify or delete objects in an existing configuration hierarchy. You begin at root level in BCC configuration mode and navigate to objects in the device configuration tree.

For example, on a BLN router, you can use BCC commands to add a new physical interface (such as Ethernet) on *box*, add IP to the Ethernet interface, and then add RIP to IP on that interface. Figure 1-3 shows the sequence of commands necessary to build this configuration.

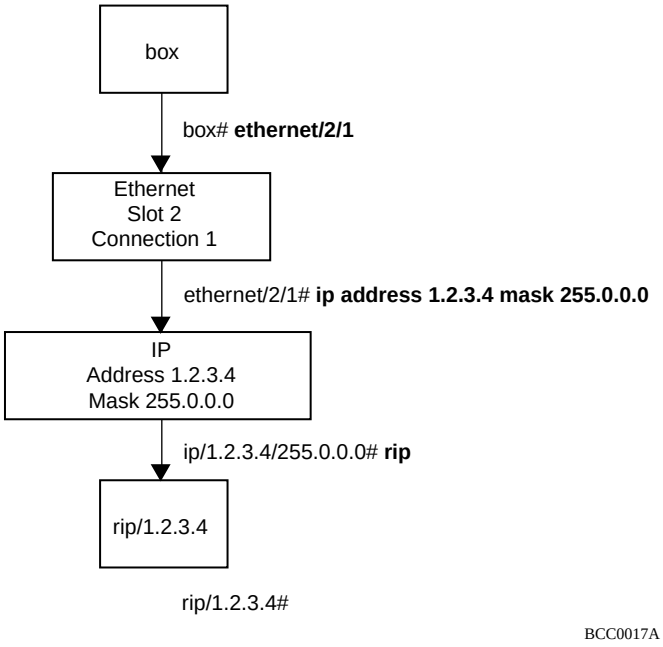

Figure 1-3. Configuring IP and RIP on an Ethernet Interface

Context

Your working location within the BCC configuration tree is referred to as the *context*. Just as a UNIX file system has a current working directory within which you can add, modify, or delete files, the BCC configuration tree has a current working context, within which you can add, modify, disable, reenable, or delete objects.

The BCC understands the context of an object in terms of its location along a path that begins at the root level of the device configuration tree. Each semicolon in the path marks a transition from one level to the next branch level in the device configuration tree. The semicolon is also equivalent to a Return key entered at the end of a command, effectively starting a new command line.

For example, if you configure an IP interface (address 1.2.3.4, mask 255.0.0.0) on ethernet/2/1 of a BLN router, the BCC understands its location as *box; ethernet/2/1; ip/1.2.3.4/255.0.0.0* (Figure 1-4.).

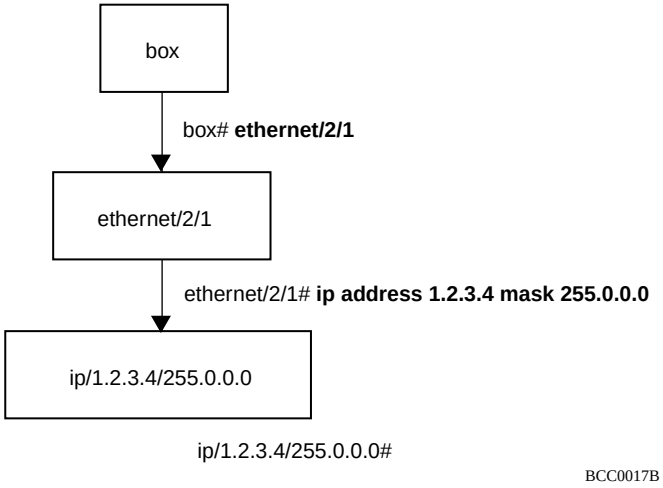

Figure 1-4. Configuring IP and RIP on an Ethernet Interface

Objects, Classes, and Instances

In BCC terminology, configurable entities are referred to as *objects* of a particular *class*, each of which constitutes an *instance.*

- An *object* is a configurable physical or logical entity such as a physical interface or a protocol on an interface. Every configurable object belongs to a specific *class* that defines its characteristics.
- A *class* is a template for a configurable object (such as Ethernet or the protocol IP). When you add a new object to the configuration of a device, the BCC creates a copy (an *instance*) of the appropriate template.

- An *instance* is an object uniquely identifiable within the total device configuration. Each instance is identified by its BCC instance identifier.

BCC Instance Identifier

A BCC instance identifier uniquely identifies a single instance of an object configured on a device. The BCC instance ID consists typically of the name of the object, combined with the values you specify for its required parameters. For example, the BCC instance ID for an Ethernet interface on a BN platform consists of **ethernet/**<*slot*>**/**<*connector*>, as in **ethernet/2/1**.

For some objects, the BCC automatically appends an internally generated number to ensure the uniqueness of the BCC instance ID. For example, the BCC creates an instance ID for ARP on IP based on the object name (arp), plus the address of the underlying IP interface (1.2.3.4), plus an internally generated integer, resulting in an ID such as **arp/1.2.3.4/1**.

In other rare cases, a configurable object may also have required parameters that do not become part of its BCC instance ID. For example, the global OSPF object has a required *router-id* parameter that does not become part of the instance ID. (Many such required parameters end in "-id," such as *global-id* and *router-id*.)

Each object has its own requirements for unique instance identification within the total device configuration.

Global (Box-Wide) Objects

Global (or *box-wide*) objects provide services uniformly to all slots of a network device. Examples include global IP, BGP, TCP, SNMP, FTP, TFTP, and Telnet. Some protocols, such as IP, RIP, and OSPF, have global and interface-level objects.

Physical Device Objects

The following sections provide BCC terms for the physical device.

Box

The BCC uses the term *box* to identify the chassis for a Bay Networks nonstackable device. Every box has a type parameter; the value assigned to the type parameter identifies the type of Bay Networks device. For example, the type parameter has the value “an” for an AN router and the value “frecn” for a BCN router with a FRE controller module.

Board

The BCC uses the term *board* or *module* to identify any logic or circuit board in a Bay Networks device. Each board typically occupies a *slot* in a network device. On some Bay Networks products, one board may contain another board such as an RMON data collection module (DCM). All board objects have a *type* parameter that identifies the module type. For example, “qenf” is the value of the type parameter for a Quad Ethernet with hardware filters module.

Note: For module descriptions based on the literal value of the *type* parameter for any board object, see the *Release Notes*.

Slot

The BCC uses the term *slot* to identify the location, as well as a physical and electrical means, for attaching modules to logic and power connections available on the device chassis.

- Multislot devices such as the BLN or BCN router accommodate one system (SRM-L) or link (interface) module per slot.
- Single-slot devices such as the AN, ASN, and ARN routers accommodate one base module (in slot 1), one or two adapter modules, and one expansion module.

Connector

The BCC uses the term *connector* to identify the physical and electrical means to interconnect a network device directly or indirectly to a physical layer transmission medium.

Line

The BCC uses the term *line* to identify the physical (and in some cases, logical) circuit identified typically by means of a slot, connector, interface type (ethernet, sync, fddi, and so on), and, where applicable, a channel number (such as with T1/E1 interface types).

Port

The BCC uses the term *port* to identify an interface object defined by its type (for example, an Ethernet port) and location (slot and connector) within a network device. On a network device, a port is also a logical point of termination for data sent or received by a specific protocol or application.

Interface

The BCC uses the term *interface* to identify circuitry and digital logic associated with the interconnection between a physical network medium (such as Ethernet) and a higher-layer protocol entity (such as IP).

Note: A *logical interface* is an addressable entity for originating and terminating connections across an IP network.

Parameters

A *parameter* is an attribute (or property) of a configurable object. Parameters can be classified as one of the following:

- Required
- Derived
- Optional

Required Parameters

For any BCC object, required parameters are a minimum set of parameters for which the BCC requires you to supply values. For example, the required parameters of a physical port are **slot** and **connector**.

Derived Parameters

Derived parameters are parameters for which the BCC supplies a value. For example, a derived parameter of the global OSPF object is **router-id**. In this case, the BCC derives a value for router-id from the address of the first IP interface configured on the device.

Optional Parameters

Optional parameters are parameters for which you can specify customized values, replacing any default values set by the system. For example, an optional parameter of an Ethernet interface is **bofl-retries**. This parameter normally has a default value of **5** (5 retries), but you can change this to another numeric value.

Chapter 2
Getting Started with the BCC

This chapter provides information about the following topics:

Topic	Page
Entering and Exiting the BCC Interface	2-1
Displaying Context	2-3
Navigating in Configuration Mode	2-3
Displaying Configuration Data	2-8
Using Online Help	2-11

Entering and Exiting the BCC Interface

To access the BCC interface on a Bay Networks router:

1. **Open a Technician Interface session with the target router.**

   For detailed information about opening a Technician Interface session, see *Using Technician Interface Software*.

2. **Enter the Manager or User command at the `Login` prompt that appears on your Telnet or console display.**

   The Manager login allows you to enter any system command and allows read-write access to the device configuration. The User login allows you to enter only user-level system command and allows read-only access to the device configuration.

3. **Enter bcc at the Technician Interface prompt.**

```
Router1> bcc

Welcome to Bay Command Console!

 * To enter configuration mode, type config

 * To list all system commands, type ?

 * To exit the BCC, type exit
bcc>
```

4. **Start BCC configuration mode by entering config at the** `bcc>` **prompt.**

```
bcc>  config
box#
```

You enter configuration mode at the root (box) level of the BCC configuration tree. The prompt ends with a pound symbol (#) if you have read-write privileges (Manager only), or with a greater than symbol (>) if you have read-only privileges (Manager or User).

If you enter BCC configuration mode as Manager and want to change your privilege level for the current session from read-write to read-only, enter **config -read-only**. To change Manager privileges back to read-write, enter **config -read-write**. You cannot change your privilege level from read-only to read-write if you logged in as User.

Caution: When you enter BCC configuration commands with read-write privileges, you immediately modify the device configuration.

5. **When you finish using BCC configuration mode, enter the exit command at any prompt.**

```
box#  exit
bcc>
```

6. **When you finish using the BCC, enter the exit command at the** `bcc>` **prompt.**

```
bcc>  exit
Router1>
```

Exiting the BCC returns you to the Technician Interface prompt.

For more detailed information about Technician Interface access, login, or logout procedures, see *Using Technician Interface Software.*

Displaying Context

In configuration mode, the BCC displays a context-sensitive prompt. The prompt identifies the configured object at your current working location within the configuration hierarchy. For example, after logging in to a BLN router as Manager, then configuring or navigating to the Ethernet interface on slot 2 connector 1, the BCC displays the following prompt:

```
ethernet/2/1#
```

To display the complete path from root level to your current level in the device configuration tree, enter the **pwc** (print working context) command.

Example:

```
rip/192.168.125.34# pwc
box; ethernet/2/1; ip/192.168.125.34/255.255.255.224; rip/192.168.125.34;
```

The **pwc** command displays the BCC instance identifier of each configured object in the path.

Navigating in Configuration Mode

You can navigate from one object to another in BCC configuration mode by using:

- The **back** command
- Configuration commands

Navigating with the back Command

In BCC configuration mode, use the **back** command to move a specific number of levels back toward root level. The syntax for the **back** command is as follows:

back [<*n*>]

Entering the **back** command with no argument moves you back one level closer to root level.

Example:

```
rip/192.168.125.34# back
ip/192.168.125.34/255.255.255.224# back
ethernet/2/1#
```

Entering the **back** command with an integer moves you from your current working location, back toward root, the number of levels you specify.

Example:

```
rip/192.168.125.34# back 2
ethernet/2/1#
```

In this example, the **back 2** command moves you from the current working location (rip/192.168.125.34), back two levels to ethernet/2/1 (with ip/192.168.125.34/255.255.255.224 as the intervening level).

Note: If you enter an integer value that exceeds the actual number of levels back to root (`box` or `stack`) level, the BCC returns to root level.

Navigating with Configuration Commands

Using BCC configuration commands, you can:

- Move back to a previous level.
- Move back to root level.
- Move forward to the next level.
- Move from your current level to any other level in the device configuration tree.

Moving Back One or More Levels

To move from your current working level back one or more levels closer to root level of the device configuration tree, you can enter the full BCC instance ID of the desired object, as follows:

Example (go back one level):

```
rip/192.168.125.34# ip/192.168.125.34/255.255.255.224
ip/192.168.125.34/255.255.255.224#
```

Example (go back two levels):

```
rip/192.168.155.151#  ethernet/2/1
ethernet/2/1#
```

In the second example, the BCC searches back toward root until it finds a context or level where the object you specified (in this case, ethernet/2/1) exists in the router configuration tree. The BCC enters the context of this object, and the prompt displays your new location.

Moving Back to Root Level

You can move back to root level in configuration mode by entering the name of the object at that level.

For an AN, ANH, ARN, or BN router, enter:

```
ip/1.2.3.4#  box
box#
```

For an ASN or System 5000 router, enter:

```
ip/1.2.3.4#  stack
stack#
```

Moving Forward One or More Levels

To move from your current working level to the next configured level (Figure 2-1), enter the BCC instance ID of the desired object, as follows:

Example:

```
box#  ethernet/2/1
ethernet/2/1#  ip/1.2.3.4/255.0.0.0
ip/1.2.3.4/255.0.0.0#  rip
rip/1.2.3.4#
```

Notice that a slash character (/) joins the name and any required parameter values to make a BCC instance ID for any configured object.

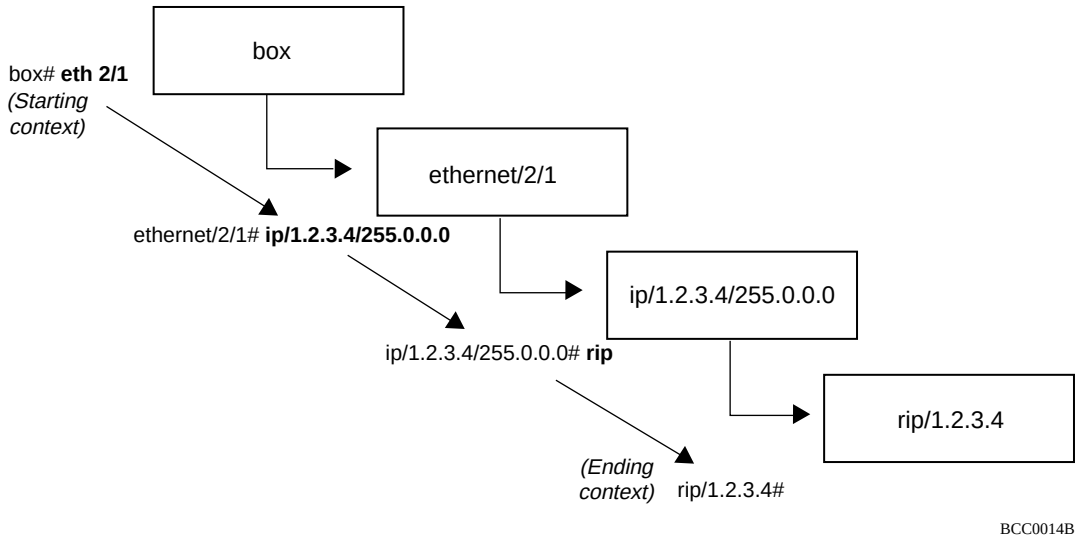

Figure 2-1. Moving Away from Root Level

Moving to Any Context in the Device Configuration

To navigate to any configured object, you can specify a full, or absolute, path from root (`box` or `stack`) level at any prompt. When you enter a path, specify the BCC instance identifier of each object.

Example:

To move from ip/192.168.33.66/255.255.255.0 (on ethernet/2/1) to rip/1.2.3.4 on ethernet/2/2 (Figure 2-2), enter the following command:

```
ip/192.168.33.66/255.255.255.0# box;ethernet/2/2;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

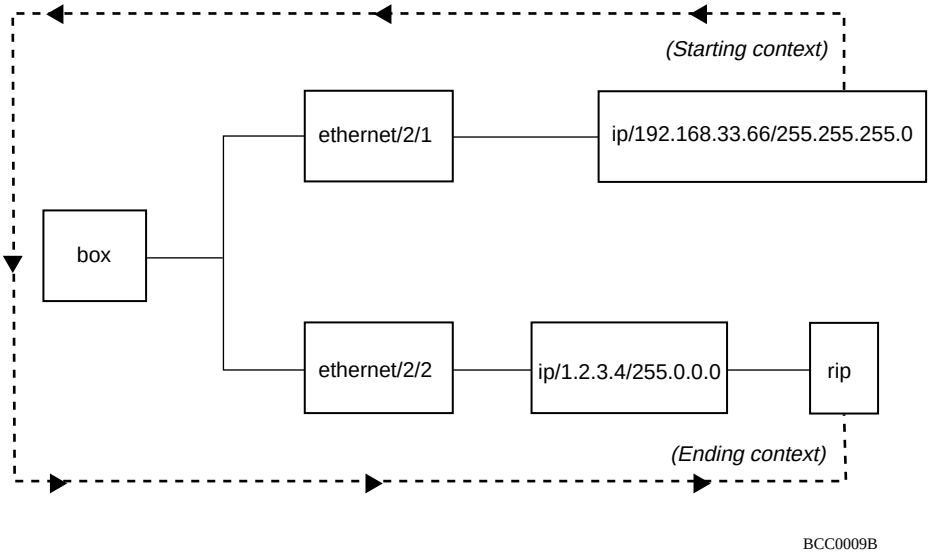

Figure 2-2. Navigating to an Object in the Configuration

The BCC can automatically search backward (recursively) toward root level until it finds a level where the object you specify first in the command line exists in the device configuration tree.

Example:

To move from ip/192.168.33.66/255.255.255.0 on ethernet/2/1 to rip/1.2.3.4 on ethernet/2/2, enter the following command:

```
ip/192.168.33.66/255.255.255.0#  ethernet/2/2;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

In this example, the BCC searches backward to find ethernet/2/2 (specified first in the command line), and then moves sequentially to the other locations (ip/1.2.3.4/255.0.0.0 and rip) specified next in the command line.

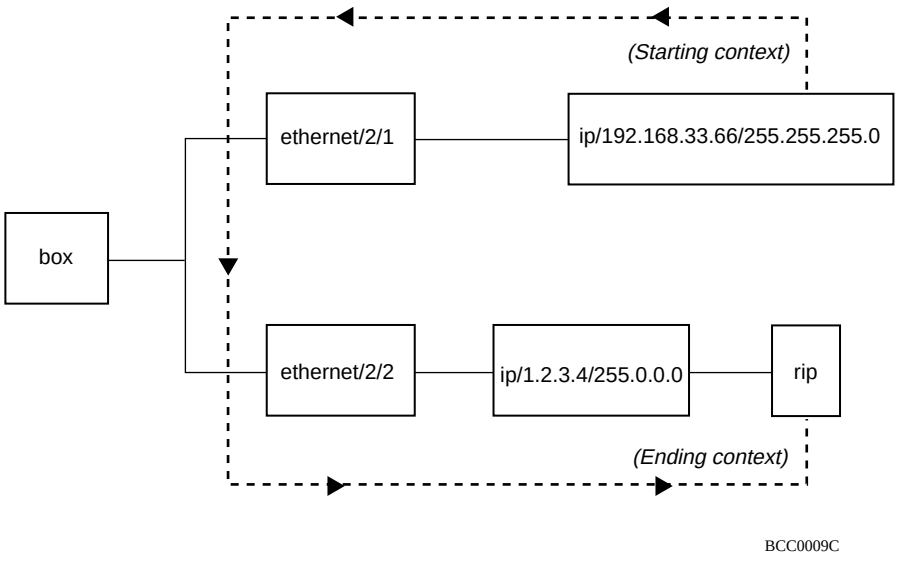

Figure 2-3. Navigating with the BCC Recursive Search Feature

Displaying Configuration Data

To display Bay Networks device configuration commands and data, use the **show config** command.

Displaying the Total Device Configuration

The **show config -all** command displays the entire device configuration from any BCC prompt. The output of this command describes:

- Existing objects
- Objects that the BCC automatically added to the active device configuration
- Navigation (**back**) commands necessary to move to the context of the previous object configured, or to return to the root level of the active device configuration

Example:

```
bcc> show config -all
box type freln
  board slot 1
      type srml
  back
  board slot 2
      type qenf
  back
  board slot 3
      type wffddi2m
     .          .          .
     .          .          .
     .          .          .
  ethernet slot 2 connector 1
      circuit-name E21-alpha
    ip address 192.168.3.4 mask 255.255.255.224
        broadcast 192.168.3.5
      rip
      back
      arp
      back
    back
  back
     .          .          .
     .          .          .
     .          .          .
  ftp
      default-volume 2
  back
  snmp
    community label public
        access read-write
      manager address 0.0.0.0
      back
      manager address 192.168.9.9
.
```

Note: For descriptions of all board type values (such as “qenf”), see the *Release Notes*.

Displaying the Configuration of One Object

To view objects configured within a specific part of the BCC configuration tree, enter the **show config** command or the **lso** command.

Example:

To view the configuration of a specific configured object, navigate to the object and enter the **show config** command. The following example shows all objects configured on ethernet/2/1 of a BLN router:

```
box# eth 2/1
ethernet/2/1# show config
ethernet slot 2 connector 1
    circuit-name E21-alpha
  ip address 192.168.125.34 mask 255.255.255.224
      broadcast 192.168.125.32
    arp
    back
    rip
    back
  back
back
```

Example:

You can also enter the **lso** command to view any instances of objects configured at your current location in the active device configuration, as follows:

```
ip/192.168.155.151/255.255.255.0#  lso
  arp/192.168.155.151/1   rip/192.168.155.151
```

Use the **lso** command if you have no need to display the contents of the current object in full BCC configuration syntax.

Using Online Help

This section describes how to use BCC online Help.

Help Commands

Table 2-1 lists the commands you use to access BCC online Help.

Table 2-1. BCC Help Commands

Command	Help Feature
?	List the names of all objects and parameters you can configure, and the system commands you can enter.

<*BCC_instance_ID*> <*parameter*> **?**	Get legal, current, and default values for this parameter of a configured object that you can access from your current working context.
	
	Example:
	`box# ethernet/2/1 bofl ?`
	`Current Value: enable`
	`Legal Value: {enable disable}`
	`Default Value: enable`

<*command*> **?**	Display usage Help or next-level options for a command.
<*command*> <*option*> **?**	Display next-level options for this command or any arguments available to limit the output from this command.
help	Get an overview of Help-oriented command features.
help <*command*>	Get full details of a specific command.
help commands	Display the syntax of all BCC commands in alphabetical order.
help commands -more	Display the syntax and brief command descriptions for all BCC commands in alphabetical order.
help editing	Get Help on how to edit BCC commands and command lines.
help learning-bcc	Get Help on performing common BCC operations.
help <*object_name*>	Show usage and parameter Help for this object.
help <*parameter_name*>	Show usage Help for this parameter.

(continued)

Table 2-1. BCC Help Commands *(continued)*

Command	Help Feature
help <*path*> <*object_name*>	Describe parameters of an object outside the current context. (<*path*> is the sequence of objects between root level and the desired context.) Example: **`help ip ospf`** Hint: Use the **help tree [-all]** command to determine the path to the desired object.
help syntax	Get Help on symbols used to express BCC command syntax.
help tree	List, in hierarchical format, all objects configurable on the current object.
help tree -all	List, in hierarchical format, all objects configurable on this device.
info	List values currently assigned to parameters of this object.
<*object*> **?**	Get command usage Help and list parameters of an object you can add or modify from your current location.
<*parameter*>	Get the current value for this parameter of the current object.
<*parameter*> **?**	Get legal, current, and default values for this parameter of the current object.

Help Command Examples

This section provides examples of how you can use the Help commands to get BCC information.

Listing Objects You Can Configure

To list the objects you can configure from the current location, enter the **help tree** command.

Example:

```
ethernet/2/1# help tree
The configuration tree below this context is:
  ip
    arp
    rip
    ospf
      neighbor
    rdisc
    igmp
      relay
  ipx
    rip
    sap
    static-route
    adjacent-host
    static-service
    route-filter
    server-network-filter
    server-name-filter
  auto-neg
```

You can also use the output of the **help tree** [**-all**] command to find the configuration or navigation path to a specific object.

Listing Available Objects, Parameters, and System Commands

To list the names of all objects, parameters, and system commands you can enter from the current context, enter the **?** command.

Example:

```
ip/192.168.125.34/255.255.255.224# ?
Sub-Contexts:
  arp     igmp    ospf    rdisc   rip

Parameters in Current Context:
  address                 cost                    on
  address-resolution      end-station-support     proxy
  all-subnet-broadcast    has                     redirects
  assocaddr               host-cache-aging        state
  broadcast               mask                    udp-checksum
  cache-size              mask-reply
  configured-mac-address  mtu-discovery

System Commands:
  ?                       display                 ping
  back                    enable                  pktdump
  bccSource               exit                    pop
  bconfig                 format                  prom
  boot                    getcfg                  pwc
  cd                      help                    readexe
  clear                   help-file-version       record
  clearlog                history                 reset
  commit                  ifconfig                restart
  compact                 info                    rm
  config                  loadmap                 save
  cp                      log                     securelogin
  cwc                     logout                  show
  date                    lso                     stamp
  debug                   mget                    stop
  delete                  mlist                   system
  diags                   more                    tic
  dinfo                   mset                    type
  dir                     partition               unmount
  disable                 password                xmodem
```

The “Sub-Contexts” section lists the objects that you can add from your current location in the device configuration tree (Figure 2-4).

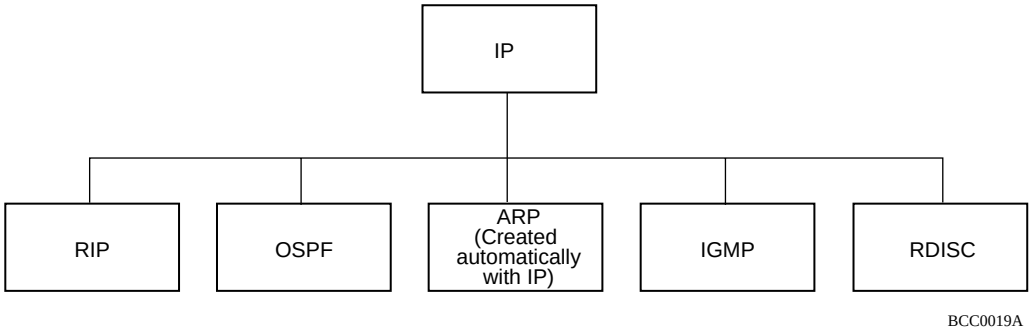

Figure 2-4. Objects You Can Configure at the Next Level

Displaying Values Assigned to Parameters

To display the values currently assigned to all parameters of the current object, enter the **info** command.

Example:

```
ip/1.2.3.4/255.0.0.0# info
  on ethernet/2/1
  state enabled
  address 1.2.3.4
  mask 255.0.0.0
  assocaddr 0.0.0.0
  cost 1
  broadcast 1.2.3.6
  configured-mac-address 0x
  mtu-discovery off
  mask-reply off
  all-subnet-broadcast off
  address-resolution arp
  proxy off
  host-cache-aging cache-off
  udp-checksum on
  end-station-support off
  redirects on
  cache-size 128
```

To display the value assigned to a specific parameter, enter the parameter name, as follows:

```
ip/1.2.3.4/255.0.0.0# proxy
  proxy off
```

Chapter 3
Entering Commands and Using Command Files

This chapter provides information about the following topics:

Topic	Page
Entering Commands	3-1
System Commands	3-5
Configuration Command Syntax	3-5
Creating and Using BCC Command Files	3-10

Entering Commands

This section contains information about:

- Using Command Abbreviations
- Recalling Commands
- Editing Command Lines
- Entering Multiple Commands on a Line
- Continuing a Command Line

Using Command Abbreviations

When you enter BCC commands in configuration mode, you can shorten object and parameter names (for example, *eth* = *ethernet*). You must enter a sufficient number of characters for the BCC to recognize that name uniquely.

Example:

```
box#  eth 5/1
ethernet/5/1#  back
box#  tf
tftp#
```

You can abbreviate system commands; for example, the BCC recognizes **sh** as **show** in contexts where there are no other commands, configurable objects, or parameter names that also start with **sh**.

Recalling Commands

The BCC supports a configurable command history buffer, from which you can recall commands recently entered. The command history buffer contains up to 20 commands by default. You can increase the number of commands in the history buffer to a maximum of 40 by setting new values for the history parameter of the console and telnet objects.

Example:

```
box# telnet
telnet# server
server# history 30
server# history
   history 30
server# box
box# console portnum 1
console/1# history 30
console/1#
```

Recall commands from the history buffer as follows:

- To recall the previous command, press the up arrow key, or press [Control]+p.
- To recall the next command, press the down arrow key, or press [Control]+n.

Editing Command Lines

Table 3-1 describes the keystrokes you can use to edit BCC command lines.

Table 3-1. Keystrokes for Editing BCC Command Lines

Editing Function	Keystrokes
Move the cursor left	CONTROL + b or left arrow key
Move the cursor right	CONTROL + f or right arrow key
Delete the current line	CONTROL + u
Delete the word at the cursor location	CONTROL + w
Delete the character at the cursor location	CONTROL + d
Move the cursor to the beginning of the line	CONTROL + a
Move the cursor to the end of the line	CONTROL + e
Toggle insert mode	CONTROL + o
Delete previous character	BKSP or DEL, or CONTROL + h
Interrupt	CONTROL + c
Start echo to the screen	CONTROL + q
Stop echo to the screen	CONTROL + s
Recall previous command	CONTROL + p or up arrow key
Recall next command	CONTROL + n or down arrow key

For example, use the up arrow key (or [Control] + p) to retrieve your last input, then use other control-key combinations to edit the command line as needed.

Entering Multiple Commands on a Line

To enter multiple commands on the same line, type a semicolon (;) wherever you would press Return to terminate a command.

Example:

Configure ethernet/2/1 from root, then configure ip/1.2.3.4/255.0.0.0 on ethernet/2/1 and RIP on ip/1.2.3.4/255.0.0.0, as follows:

```
box#  ethernet/2/1;ip 1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

Continuing a Command Line

You can continue a command line by entering a backslash (\ ) character at the end of the current text line. The BCC treats characters on the next physical line as part of the same BCC logical command line.

You must immediately follow the backslash (\ ) with a newline (Return) character. The BCC treats these two characters and any trailing spaces as if they were exactly one space. Until you press Return without a preceding backslash (\ ) character, the BCC replaces the pound symbol (#) in the context-sensitive prompt with an underscore (_) character.

Example:

```
ip/1.2.3.4/255.0.0.0# cost 2 \
ip/1.2.3.4/255.0.0.0_ mask-reply on \
ip/1.2.3.4/255.0.0.0_ proxy on \
ip/1.2.3.4/255.0.0.0_ aging cache-on
ip/1.2.3.4/255.0.0.0#
```

Some command symbols normally used in pairs to denote the beginning and the end of a set of data also produce the continuation (underscore) prompt, including braces ({ }), brackets ([ ]), and quotation marks (“ ”).

Example:

```
box#  { ...
box_  ...}
box#
```

Note: If you inadvertently type one of the opening symbols and see an underscore prompt, just type the corresponding closing symbol to restore the normal (#) prompt in BCC configuration mode.

System Commands

The BCC supports all system commands described in Appendix A. For Help on a specific command, enter **help** <*command*> (for example, **help save**) at any BCC prompt.

Configuration Command Syntax

This section describes BCC configuration commands and the syntax requirements for those commands. This section also describes how to enter BCC configuration commands using the following formats:

- Basic (full) syntax
- Default syntax
- Abbreviated syntax

Caution: Configuration commands make real-time changes to the device configuration.

Command Syntax Requirements

BCC syntax consists of object names, parameter names and values, and various types of punctuation.

- All object and parameter names appear as one word (hyphenated where necessary) in the BCC command line.
- Parameters have either a single value or multiple values enclosed in braces {x y z} in the command line. You can accept the default value or supply a value for each parameter associated with a configurable object.
- Parameters and their values must appear as a pair in the same command line.

- Syntax for specifying the object you want to configure may vary according to the number of slots in the Bay Networks device to which you are connected. Appendix D lists the syntax for specifying the physical location of a module for each Bay Networks device that the BCC supports.
- If you enter the name of an object without values for its required parameters, or with values inappropriate for its required parameters, the BCC returns usage Help, as shown in the following example:

```
box# ethernet
Required parameter "slot" was not specified for ethernet.
Usage: "ethernet slot <value> connector <value>"
Or:    "ethernet <slot>/<connector>
```

Using Basic (Full) Syntax

The basic, or full, syntax for BCC commands consists of the following required and optional elements:

{*<object-name>*} {*<required_parameter> <value>* ... } ...
<parameter> <value> ... *<parameter> <value>*

The BCC requires input for any elements enclosed by braces ({ }).

{<*object-name*>} is the name of an object you want to configure (for example, **ip**).

The BCC assumes that an object you specify is new (and will create it) if it is not in the current configuration. If an object you specify already exists in the current configuration, the BCC assumes that you want to modify that object.

{<*required_parameter*> <*value*>} is any parameter-value pair required to uniquely identify an object you specify in a BCC command line. An object may have one or more required parameters.

For example, the full syntax for configuring an Ethernet interface on an AN/ANH, ARN, or BN router is:

ethernet slot *<slot_no.>* **connector** *<connector_no.>*

On an ASN or System 5000 router, the full syntax is:

ethernet slot *<slot_no.>* **module** *<module_no.>* **connector** *<connector_no.>*

Note: You cannot change the value of a parameter used by the BCC to create an instance identifier. For example, you cannot modify the address value assigned to an IP interface.To change the value of any required parameter, you must delete the associated object, and then add it back into the device configuration with new required values.

[<*parameter*> <*value*>] is any parameter-value pair you can optionally customize for an object you specify in a BCC command line.

Using Default Syntax

Using default syntax, you do not need to enter the name of a required parameter; you enter only its value at the proper location in the command line.

For example, the default syntax for configuring an Ethernet interface on an AN/ANH, ARN, or BN router is:

ethernet *<slot>*/*<connector>*

The following commands are equivalent.

Using full syntax:

```
box#  ethernet slot 2 connector 1
ethernet/2/1#
```

Using default syntax:

```
box#  ethernet 2/1
ethernet/2/1#
```

Using Abbreviated Syntax

You can abbreviate BCC configuration commands as follows:

Example:

```
box# eth 2/1
```

This command is the same as the following two commands:

```
box# ethernet slot 2 connector 1
```

```
box# ethernet 2/1
```

If you press Return before entering a sufficient number of characters for the BCC to recognize the name of the object or parameter you want to configure, the BCC returns an error message.

Example:

```
box# e

ambiguous command name "e": enable eof error ethernet eval exit expr
```

The BCC returns a list of all the commands available in the current context that start with the letter "e." Choose one command from the list, and enter enough characters for the BCC to recognize that command when you press Return.

You cannot abbreviate BCC instance identifiers.

Specifying Parameter Values

You must specify each parameter value in the form of a parameter-value pair. Each pair is a command argument pertaining to the object named first in the command line.

For example, the following command changes the BOFL timeout interval to 4 seconds on ethernet/1/1:

```
box#  ethernet/1/1 bofl-timeout 4
```

bofl-timeout 4 is the parameter-value pair.

Required, Derived, and Other Parameters

The BCC indicates when parameter values are required (you must supply a value) or derived (the BCC supplies a value). For all other parameters, the BCC supplies a default value that you can change.

Specifying Multiple Parameter-Value Pairs

You can specify parameter values as follows:

- Enter an object name and one parameter-value pair per command line.
- Enter an object name and multiple parameter-value pairs (each pair separated by a space) on the same command line.

Example:

In the following example, you specify one parameter-value pair on each command line.

```
box#  ethernet 2/1
ethernet 2/1#  bofl-retries 6
ethernet 2/1#  bofl-timeout 7
ethernet 2/1#  hardware-filter enabled
ethernet 2/1#
```

Example:

In the following example, you specify multiple parameter-value pairs on each command line.

```
ethernet 2/1#  ip address 1.2.3.4 mask 255.255.255.0 redirects off
ip.1.2.3.4/255.255.255.0#  ospf area 2.3.4.54 hello-interval 5
ospf/1.2.3.4#
```

Disabling, Reenabling, and Deleting a Configured Object

Table 3-2 lists the commands you can use to disable, reenable, and delete any configured object. To use these commands, you must be in configuration mode with read-write privileges.

Table 3-2. BCC Commands

Command	Function
disable	Change the state of a configured object from enabled to disabled, as follows: `ip/1.2.3.4/255.0.0.0# disable` You can accomplish the same change by assigning the value “disabled” to the state parameter of an object that you want to disable.
enable	Change the state of a configured object from disabled to enabled, as follows: `ip/1.2.3.4/255.0.0.0# enable` You can accomplish the same change by assigning the value “enabled” to the state parameter of an object that you want to reenable.
delete	Delete the object identified in the BCC context-sensitive prompt. For example, the following command deletes an IP interface (address 1.2.3.4): `ip/1.2.3.4/255.0.0.0# delete` **CAUTION:** Deleting an object at one level of the configuration tree causes the BCC to automatically delete any branches configured on that object. For example, if you delete an IP interface, the BCC deletes any protocols (such as RIP, ARP, or OSPF) configured on that interface.

Creating and Using BCC Command Files

You can save BCC commands to an ASCII file, edit the file, add comments, and then use the **source** command to read the file into the device’s active configuration. The following sections describe how to complete these tasks.

Saving Commands to a File

If you log in to a Bay Networks router from a PC or workstation using Telnet or terminal emulation, you can use the native capabilities of the PC or workstation to:

- Save the output of a **show config** command to an ASCII file.
- Save a sequence of manually entered BCC commands to an ASCII file.

You can also use an ASCII text editor on a workstation to create a file from which the BCC can read configuration and system commands.

Note: You can also save the output of the **show config** command to an ASCII text file on the router by entering **show config** [**-all**] **-file** *<volume>:<filename>* at any BCC prompt.

Adding Comments to a Command File

You can use a text editor (such as *vi* on a UNIX workstation) to add descriptive comments to a BCC command file. Enter comments in the following format:

```
box# <command> ;# comment
```

or

```
box# #comment
box# <command>
```

Example:

```
box#  board slot 1 type andse;# 192.168.47.129  192.168.47.21
```

When you finish editing the file, save it on your workstation or PC. The comments are for reference only. Comments do not appear in the output of any **show config** command.

Entering Configuration Commands from a File

When you are logged in to the BCC as Manager, you can use the **source** command in configuration mode to read BCC configuration and navigation commands from a designated ASCII source file into the active device configuration.

Caution: The **source** command makes immediate changes to the active device configuration.

The **source** command *merges* new configuration data from a file with existing data in device memory. If the file you specify contains configuration commands pertaining to objects already defined on the device, those commands overwrite the current configuration.

Note: If the BCC detects an error in the source file, it stops reading commands into the device configuration.

Syntax for the **source** command is as follows:

source *<volume>*:*<filename>*

Saving the Active Configuration as a Bootable Binary File

When you finish using BCC commands to modify an existing configuration, save the new configuration to a file on an NVFS (flash) volume. (At boot time, the router loses any configuration changes not previously saved to an NVFS volume.) The following command saves *config* as a bootable binary file on a volume you specify:

```
bcc>  save config <volume>:<filename>
```

Chapter 4
Tutorial: Configuring a Network Device

This chapter provides a tutorial that guides you through the initial configuration of a Bay Networks device using the BCC. It includes the following sections:

Topic	Page
Creating and Modifying a Device Configuration	4-1
Disabling a Configured Object	4-15
Enabling a Configured Object	4-16
Deleting a Configured Object	4-17

Creating and Modifying a Device Configuration

You configure a Bay Networks device by defining a set of objects, starting at the root level of the device's configuration hierarchy. Each object has a set of parameters with values set either by you or by the device software.

Following is a typical BCC configuration sequence.

1. Open a Technician Interface session and start the BCC interface.
2. Start BCC configuration mode.
3. Use BCC configuration commands to create new objects in the device configuration and modify default values for parameters of each object to meet the requirements of your network.
4. Enable any box-wide protocols not enabled automatically by the BCC, for example, TFTP and Telnet Server.
5. Use the **save** command to save your configuration as a bootable (binary) file on the device.
6. Exit BCC configuration mode, exit the BCC to the Technician Interface, and log out of the device.

You may find it helpful to first diagram what you want to configure in terms of the BCC configuration tree or hierarchy for the device. Refer to the following sample router configuration.

Sample Router Configuration

The following example shows a sequence of commands you can use to configure a BCN router on a network. You first complete the physical installation of the router, then boot the router using the image (*bn.exe*) and the minimum configuration file (*ti.cfg)*.

This example creates the following objects in the total router configuration (Figure 4-1):

- IP (global)
  - -- ARP (global) on IP
  - -- RIP (global) on IP
- SNMP (global)
  - -- Community “public” on SNMP
    - Manager (address 0.0.0.0) on community “public”
- FTP (global)
- TFTP (global)
- Telnet (global)
  - -- Server (global) on Telnet
- Quad Ethernet interface in slot 13
  - -- IP interface (address 192.168.133.114) on Ethernet connector 1
    - ARP on IP interface 192.168.133.114
    - RIP on IP interface 192.168.133.114
- Serial interface in slot 5
- Dual token ring interface in slot 9
- FDDI interface in slot 11

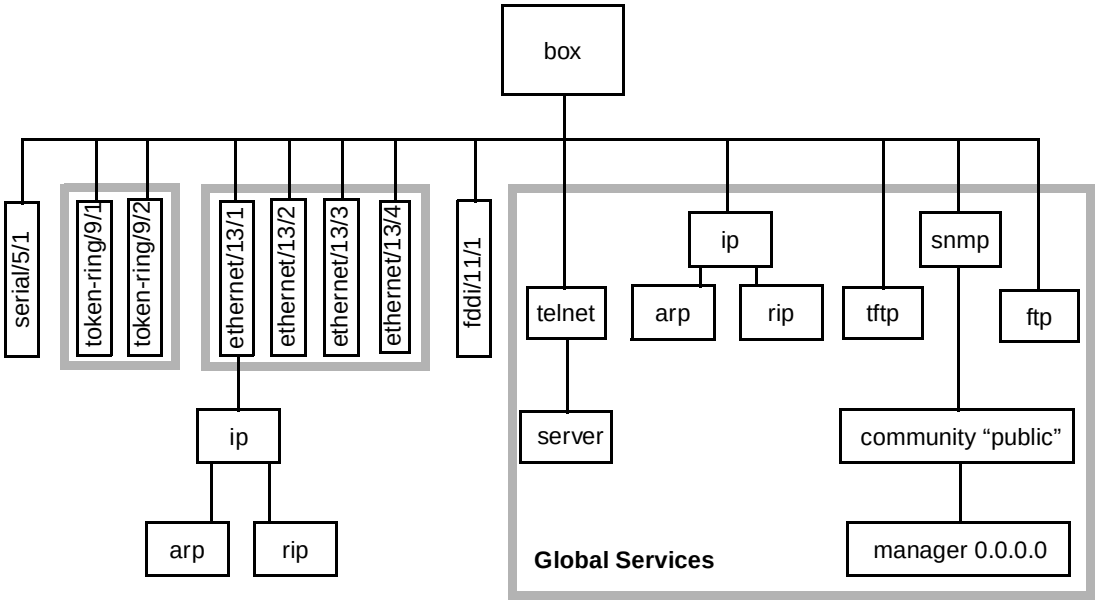

Figure 4-1. Sample BCC Configuration (BCN Router)

After you create a diagram of the device configuration tree, configure the device using a cycle of BCC configuration commands similar to those shown in Figure 4-2.

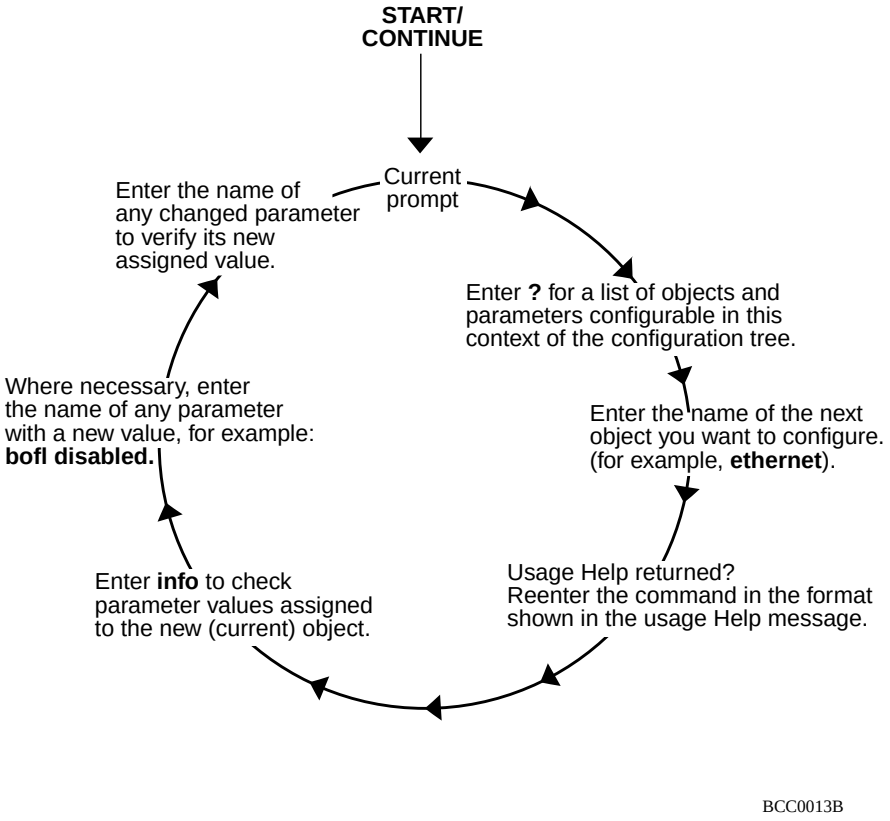

Figure 4-2. Typical BCC Configuration Cycle

To create the sample configuration shown in Figure 4-1 using BCC commands, proceed as follows:

1. **Log on to the router as Manager, so that you can modify the device configuration.**

```
Bay Networks, Inc. and its Licensors.
Copyright 1992, 1993, 1994, 1995, 1996, 1997, 1998. All rights reserved.

Login: Manager
Mounting new volume...
Device label:
Directory: 2:
New Present Working Directory: 2:

Welcome to the Backbone Technician Interface

    Router1>
```

2. **Start the BCC by entering bcc at the Technician Interface prompt.**

```
Router1> bcc

    Welcome to Bay Command Console!

    * To enter configuration mode, type config

    * To list all system commands, type ?

    * To exit the BCC, type exit

bcc>
```

3. **Enter BCC configuration mode by entering config.**

```
bcc>  config
box#
```

4. Check the hardware configuration of the router.

```
box# show config

box type frecn (BCN router with a FRE-2 controller)
    board slot 5
        type sync (Serial link module in slot 5)
    back
    board slot 7
        type srml (System resource module in slot 7)
     back
    board slot 9
        type dtok (Dual token ring link module in slot 9)
    back
    board slot 11
        type wffddi2m (Multimode FDDI link module in slot 11)
    back
    board slot 13
        type qenf (Quad Ethernet with filters in slot 13)
    back
    console portnum 1 (console device on port 1)
        prompt {"%slot%:"}
        auto-manager-script automgr.bat
        auto-user-script autouser.bat
     back
    back
    box#
```

Note how output of the **show config** command automatically includes navigation (**back**) commands. If you save this output to a file, you can reenter the commands automatically using the BCC **source** command. (See "Entering Configuration Commands from a File" on page 3-12.)

For descriptions of the values of the board type parameter, see the *Release Notes*.

5. Choose a port (interface type, slot, and connector) for the initial IP interface to the router.

```
box# ethernet slot 13 connector 1
ethernet/13/1#
```

6. Check what you can configure (sub-contexts and parameters) at this level.

```
ethernet/13/1# ?
Sub-Contexts:
  ip    ipx

Parameters in Current Context:
  bofl                    circuit-name            on
  bofl-number             connector               receive-queue-length
  bofl-retries            hardware-filter         slot
  bofl-timeout            has                     state
  bofl-tmo-divisor        name                    transmit-queue-length

System Commands:
  ?               delete          mkdir           reset
  attribute       diags           mlist           restart
  back            dinfo           more            rm
  boot            dir             mount           rmdir
  cd              disable         mset            save
  clear           enable          partition       securelogin
  clearlog        exit            password        show
  commit          format          ping            source
  compact         help            pktdump         stamp
  config          history         pop             stop
  context         loadmap         prom            system
  cp              log             pwc             tic
  cwc             logout          readexe         unmount
  date            lso             reconfig        xmodem
  debug           mget            record
ethernet/13/1#
```

Based on the “Sub-Contexts” list, you can add **ip** (interface IP) or **ipx** (interface IPX) to this Ethernet port. The list of configurable objects depends on the *board* type described in the current prompt. For example, 100BASE-T modules additionally allow you to configure an “autoneg” (autonegotiation) object at this level.

You can also modify the values currently assigned to “Parameters in Current Context” (parameters of ethernet/13/1), shown above.

The “System Commands” list always appears along with information about configurable objects and parameters.

7. Add IP (address 192.168.133.114) to ethernet/13/1.

```
ethernet/13/1# ip 192.168.133.114
Usage: "ip address <value> mask <value>"
Or:    "ip <address>/<mask>"
Required parameter "mask" was not specified for ip.
```

The error message appears because the BCC requires you to enter a mask value whenever you create an IP interface. Because the first octet of the address is 192 (0xC0), this is a Class C address, requiring the first 3 octets to be the network portion of the interface address. You can express the corresponding mask value as either 255.255.255.0 (using dotted-decimal notation) or 24 (the number of bits making up the network portion of the IP interface address), as follows:

```
ethernet/13/1# ip 192.168.133.114/24
ip/192.168.133.114/255.255.255.0#
```

The BCC converts the integer representing the number of mask bits into a dotted-decimal mask value, as shown in the last prompt.

8. Check the values currently assigned to parameters of IP on this interface.

```
ip/192.168.133.114/255.255.255.0# info
  state enabled
  address 192.168.133.114
  mask 255.255.255.0
  assocaddr 0.0.0.0
  cost 1
  broadcast 0.0.0.0
  configured-mac-address 0x
  mtu-discovery off
  mask-reply off
  all-subnet-broadcast off
  address-resolution arp
  proxy off
  host-cache-aging cache-off
  udp-checksum on
  end-station-support off
  redirects on
  cache-size 128
```

9. Change the subnet mask to 255.255.255.224.

```
ip/192.168.133.114/255.255.255.0# mask 255.255.255.224
"mask" parameter may not be modified
ip/192.168.133.114/255.255.255.0#
```

You cannot modify any parameter values included in the BCC instance ID of a configured object, in this case, ip/192.168.133.114/255.255.255.0. To change the value of any parameter that is part of a BCC instance ID, you must first delete the object, then re-create it with the desired parameter values. In this case, you must delete the IP interface and then re-create it on ethernet/13/1, using the mask value of 255.255.255.224, as follows:

```
ip/192.168.133.114/255.255.255.0# delete
ethernet/13/1# ip 192.168.133.114/255.255.255.224
ip/192.168.133.114/255.255.255.224#
```

10. Check to see what you can configure at this level.

```
ip/192.168.133.114/255.255.255.224# ?
Sub-Contexts:
  arp     igmp    ospf    rdisc   rip

Parameters in Current Context:
  address                 cost                    name
  address-resolution      end-station-support     on
  all-subnet-broadcast    has                     proxy
  assocaddr               host-cache-aging        redirects
  broadcast               mask                    state
  cache-size              mask-reply              udp-checksum
  configured-mac-address  mtu-discovery

System Commands:
  .             .           .
  .             .           .
  .             .           .
```

You can modify values currently assigned to parameters of ip/192.168.133.114/255.255.255.224, or you can add RIP, OSPF, Router Discovery, or IGMP to this interface.

11. Add RIP as the routing protocol (by default, RIP1) on this interface.

```
ip/192.168.133.114/255.255.255.224# rip
rip/192.168.133.114#
```

12. Return to root (box) level to configure global system services.

```
rip/192.168.133.114# box
box#
```

13. Check which global services and interfaces you can configure at this level.

```
box# ?
Sub-Contexts:
  atm             fddi            ipx             serial          virtual
  backup-pool     ftp             isdn-switch     snmp
  board           hssi            mce1            telnet
  console         http            mct1            tftp
  ethernet        ip              ntp             token-ring

Parameters in Current Context:
  build-date      description     location        system-name
  build-version   has             mib-counters    type
  contact         help-file-name  on              uptime

System Commands:
 .              .             .
 .              .             .
 .                .             .
```

You can add any of the following global services (affecting all slots): HTTP, IP, IPX, FTP, NTP, SNMP, Telnet, and TFTP.

You can add any of the following interfaces: ATM, Ethernet, FDDI, HSSI, serial, token ring, or virtual.

You can view the configuration of a board in any slot, but you cannot modify the parameters of any board object.

14. List the objects already configured at box level.

```
box# lso
board/11        board/5         board/9         ethernet/13/1
board/13        board/7         console/1       ip
box#
```

When you added the first instance of IP to the box (ip/192.168.133.114/255.255.255.224), the BCC automatically created the global IP object at box level.

15. Add SNMP to the device.

```
box# snmp
snmp#
```

16. Check what you can configure next at this level.

```
snmp# ?
Sub-Contexts:
  community       trap-entity     trap-event

Parameters in Current Context:
  authentication-traps    lock-timeout            state
  has                     name                    type-of-service
  lock                    on
  lock-address            scope-delimiter

System Commands:
 .             .            .
 .             .            .
 .               .             .
```

You can modify values currently assigned to parameters of SNMP, and you can add a community, define a trap entity, or define a trap event.

17. View the parameter definitions using the BCC help command.

```
snmp# help snmp

                              snmp Parameters:
    ____________________________________________________________________

   state
          Indicates the administrative state of this object (enabled or
          disabled).

   authentication-traps
          Sends trap for sets from false Mgr or Community.

   lock
          Allows the locking mechanism to be disabled.

   lock-address
          Allows the lock address to be cleared.

   lock-timeout
                .                  .                  .
                .                  .                  .
                .                  .                  .
```

18. Define the SNMP community “public.”

```
snmp# community public
community/public#
```

19. **Check the values currently assigned to parameters of this SNMP community.**

```
community/public# info
  on snmp
  label public
  access read-only
  scope-type {}
```

20. **To allow network management applications (such as Site Manager) to modify the device configuration, change the value of the access parameter to read-write**.

```
community/public# access read-write
community/public#
```

21. **Define an SNMP manager for the router.**

```
community/public# manager
Usage: "manager address <value>"
Or:    "manager <address>"
Required parameter "address" was not specified for manager.
```

    The BCC error message indicates what you left out and automatically provides usage Help on how to configure an SNMP manager.

22. **Try again to add the manager, this time supplying a value for its required parameter, address. (You must enter a value for a required parameter, but you can omit the name of the parameter.)**

```
community/public# manager 0.0.0.0
manager/public/0.0.0.0#
```

23. **Enable the Telnet server entity on the router.**

```
manager/public/0.0.0.0# telnet
telnet#  server
server#
```

24. **Add TFTP services globally to the router.**

```
server# tftp
tftp#
```

    The BCC automatically searches back (toward root) to find the parent context suitable for Telnet and TFTP (in this case, box). The BCC then adds Telnet and TFTP to the device configuration. Note the new (`tftp#`) prompt.

25. Check the values currently assigned to parameters of TFTP.

```
tftp# info
  on box
  state enabled
  default-volume 2
  retry-timeout 5
  close-timeout 25
  retry-count 5
```

26. Change the default volume number for TFTP to 5.

```
tftp# def 5
tftp#
```

27. Verify the change to the default volume number.

```
tftp# def
  default-volume 5
```

28. Add FTP globally to the router.

```
tftp# ftp
ftp#
```

The BCC automatically searches back (toward root) to find the parent context suitable for FTP (in this case, box). The BCC then adds FTP to the device configuration. Note the new (`ftp#`) prompt.

29. Check the definitions for parameters of FTP.

```
ftp# help ftp
                                ftp Parameters:
     ____________________________________________________________________

   on
          Identifies the parent(s) of this object.

   state
          Indicates the administrative state of this object (enabled or
          disabled).

   default-volume
          Specifies the default volume where transferred files are
          written/retrieved.
     .       .       .
     .       .       .
     .       .       .
```

30. Check values currently assigned to parameters of FTP.

```
ftp# info
  on box
  state enabled
  default-volume 2
  login-retries 3
  idle-timeout 900
  max-sessions 3
  tcp-window-size 60000
```

31. Change the default volume number to 5.

```
ftp# def 5
ftp#
```

32. Verify the change to the default volume number.

```
ftp# def
  default-volume 5
```

33. Recheck the total device configuration.

```
ftp# show config -all
box type frecn
  board slot 5
      type sync
  back
  board slot 7
      type srml
  back
  board slot 9
      type dtok
  back
  board slot 11
      type wffddi2m
  back
  board slot 13
      type qenf
  back
  console portnum 1
      prompt {"%slot%:"}
      auto-manager-script automgr.bat
      auto-user-script autouser.bat
  back
  ethernet slot 13 connector 1
      circuit-name E131
    ip address 192.168.133.114 mask 255.255.255.224
     .       .       .
     .       .       .  (remaining configuration not shown here)
```

34. Return to root level.

```
ftp# box
box#
```

35. **Save the file using a name other than *config* until you can test the configuration.**

```
box#  save config startup.cfg
```

36. **Test the intial IP interface.**

```
box#  ping 192.168.133.114
IP ping: 192.168.133.114 is alive (size = 16 bytes)
```

37. **Ensure that the initial IP interface connects to another device on the network.**

```
box# ping 192.168.133.97
IP ping: 192.168.133.97 is alive (size = 16 bytes)
```

38. **When you finish configuring the router, exit configuration mode.**

```
box#  exit
bcc>
```

39. **Exit the BCC, which returns you to the Technician Interface prompt.**

```
bcc>  exit
Router1>
```

40. **Enter the logout command to close your console or Telnet session with the router.**

```
Router1> logout
```

Disabling a Configured Object

In most cases, the BCC automatically enables objects that you add to the device configuration. However, you can disable an object to manage or troubleshoot the device. Here is a BN router example of how to disable an object (rip) on ip/1.2.3.4/255.0.0.0:

1. **Specify the configuration context for the object you want to disable.**

```
box#  ethernet/2/1;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

2. **Disable RIP.**

```
rip/1.2.3.4#  disable
rip/1.2.3.4#
```

3. **Verify that you disabled RIP.**

```
rip/1.2.3.4#  state
  state disabled
rip/1.2.3.4#
```

You can also disable an object from its parent context, using the following syntax:

disable *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  disable rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after disabling the branch object.

Enabling a Configured Object

If you disable a configured object, you can use the BCC **enable** command to reenable that object. Here is a BN router example of how to enable an object (rip) previously disabled on ip/1.2.3.4/255.0.0.0:

1. **Specify the configuration context for RIP.**

   ```
   box#  ethernet/2/1;ip/1.2.3.4/255.0.0.0;rip
   rip/1.2.3.4#
   ```

2. **Reenable RIP.**

   ```
   rip/1.2.3.4#  enable
   rip/1.2.3.4#
   ```

3. **Verify that you reenabled RIP.**

   ```
   rip/1.2.3.4#  state
     state enabled
   rip/1.2.3.4#
   ```

You can also enable an object from its parent context, using the following syntax:

enable *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  enable rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after enabling the branch object.

Deleting a Configured Object

Because of the tree hierarchy, objects on higher branches of the tree depend on the state (and existence) of objects closer to the root of the tree. Deleting an object also deletes anything configured on that object.

Caution: Before using the BCC to delete an interface, make sure that you did not use Site Manager to configure it with a protocol that the BCC does not recognize. If you did, use Site Manager to delete the interface.

Here is a BN router example of how to delete an IP interface from the active device configuration:

1. **Navigate to the object you want to delete.**

```
box#  ethernet/13/1
ethernet/13/1#  ip/192.168.133.114/255.255.255.224
ip/192.168.133.114/255.255.255.224#
```

2. **List all objects configured on the current object.**

```
ip/192.168.133.114/255.255.255.224# lso
arp/192.168.133.114/1   rip/192.168.133.114
```

3. **Delete the object.**

```
ip/192.168.133.114/255.255.255.224#  delete
ethernet/13/1#
```

4. **Verify that you deleted the object.**

```
ethernet/13/1# lso
```
(no objects listed)

Note that ip/192.168.133.114/255.255.255.224 no longer appears in the list of objects configured on ethernet/13/1. With a single **delete** command, the BCC automatically deleted the branch objects (arp/192.168.133.114/1 and rip/192.168.133.114) configured on ip/192.168.133.114/255.255.255.224.

You can also delete an object from its parent context, using the following syntax:

delete *<BCC_instance_identifier>*

Example:

```
ip/1.2.3.4/255.0.0.0#  delete rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after deleting the branch object.

Appendix A
System Commands

The BCC supports the system commands listed in Table A-1.

For more detailed information about any of these commands:

- Enter **help** <*command*> at any BCC prompt (for example, **help save**).
- Refer to the guide *Using Technician Interface Software.*

Table A-1. System Commands

Command	Purpose	Login/Access Privileges	
		Manager	User
?	List the names of objects, parameters, and system commands you can enter next.	✔	✔
<command> **?**	Display syntax usage Help for *<command>.*		
<object> **?**	List parameters of an object you can access or configure from your current level in the device configuration tree.		
<parameter> **?**	List the current, default, and legal values for this parameter of the current object.		
<object> *<parameter>* **?**	List the current, default, and legal values for the specified parameter of this object.		
! [*<n>*] (Technician Interface only)	Repeat the last command, or repeat the last command *<n>* times.	✔	✔
back [*<n>*] (BCC configuration mode only)	Move your current working location back *<n>* levels closer to the root level of the BCC configuration tree.	✔	✔
bcc	Start the BCC from the Technician Interface prompt.	✔	✔
boot [{*<vol>*:*<image_name>* | - } { *<vol>*:*<config_name>* | - }]	Reboot the system.	✔	
cd [*<vol>*:][*<directory>*]	Set or display the current working volume and directory.	✔	✔
clear *<subcommands>* *<flags>*	Clear specific device information, such as IP and RIP data.	✔	
clearlog [*<slot_ID>*]	Clear all previous events from the system event log.	✔	
commit	Commit (make effective) new values that you assign to MIB attributes.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
compact *<volume>*:	Compact existing files into a contiguous address space on a volume, and compact all unused space into a single contiguous block of free space for new files on the same volume.	✔	
config [**-read-write**|**-read-only**] (BCC only)	Enter BCC configuration mode. If you logged on as Manager, you have read-write access to the device configuration; if you logged on as User, you have read-only access, which allows you to display the existing configuration, but prevents you from changing the configuration. If you enter BCC configuration mode as Manager and want to change your privilege level for the current session from read-write to read-only, enter **config -read-only**. To change Manager privileges back to read-write, enter **config -read-write**.	✔	✔
copy *<vol>*:*<filename1> <vol>*:*<filename2>* (Technician Interface only)	Copy the contents of the first file to the second file.	✔	
cp *<vol>*:*<filename1> <vol>*:*<filename2>* (BCC only)	Copy the contents of the first file to the second file.	✔	
date [*<mm/dd/yy>*] [*<hh:mm:ss>*] [{+|-}*<hh:mm>*]	Display or update the system time and time zone.	✔	✔
delete *<filename> <volume>* (Technician Interface only; see **rm** for BCC equivalent)	Delete the file from the specified volume.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
delete [*<BCC_instance_ID>*] (BCC configuration read-write mode only)	Delete the object described in the current prompt, or delete the object known by the specified *<ID>* and configured previously from the current level.	✔	
diags [*<slot _ID>*]	Perform CPU/memory, backbone, and link diagnostics before downloading the GAME image and rebooting on the specified slot.	✔	
dinfo	Display the status of each file system volume on this device.	✔	✔
dir [*<vol>***:**]	Display the contents of the file system volume specified.	✔	✔
disable (BCC configuration read-write mode only)	Disable the object at your current location in the BCC configuration tree.	✔	
enable (BCC configuration read-write mode only)	Enable the object at your current location in the BCC configuration tree.	✔	
format *<volume>*:	Erase the entire contents of the file system volume specified and reinitialize it to a usable state.	✔	
get {*<obj_name>*|*<obj_id>***}.**{*<attr_name>*| *<attr_id>*|*}[.{*<inst_id>*|*}] (Technician Interface only)	Retrieve the values of data objects in the MIB.	✔	✔
getcfg (Technician Interface, AN routers only)	Display network boot parameters. (Display the current parameter settings used to determine the source of image and configuration files.)	✔	✔
help	Display an overview of BCC Help-oriented features.	✔	✔
help *<command>*	Display detailed Help on a specific command.	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
help commands	Display an alphabetical list of all commands, with syntax and terse descriptions.	✔	✔
help commands -more	Display syntax and brief command descriptions for all BCC commands in alphabetical order.	✔	✔
help *<object_name>*	Display definitions for parameters of the currrent object. For example, **help ip ospf** defines parameters of the global ospf object, and **help ethernet ip** defines parameters of the IP interface on an Ethernet port.	✔	✔
help *<parameter_name>*	Display definitions of all parameters of the current object. For example, **help bofl** displays the definitions of all parameters of **ethernet**.	✔	✔
help syntax	Display Help on how to interpret BCC syntax symbols.	✔	✔
help [*<task>*]	Display Help on how to perform a specific task. The **help** commands are **help learning-bcc** and **help editing**.	✔	✔
help tree [**-all]**	Display a hierarchical list of all objects you can configure on this platform (use **-all**) or on the current object (omit **-all**).	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
help [<*command*> | **-all**] (Technician Interface only)	The **help** command, entered without arguments at the Technician Interface prompt, displays an alphabetical list of all commands with syntax only. The list excludes commands available only in BCC mode. The combination **help** <*command*> displays detailed Help on a specific command, excluding any commands available only in BCC mode. The combination **help -all** displays detailed Help on all system commands, excluding those available only in BCC mode.	✔	✔
history [<*n*>] (<*n*> option Technician Interface only)	Display the command history list or, for the Technician Interface only, recall command number <*n*> from the history list.	✔	✔
info (BCC configuration mode only)	List values currently assigned to all configurable parameters of this object.	✔	✔
ip <*subcommand*> <*flags*> (Technician Interface only)	Display data from IP; show a different view for each subcommand or flag you enter.	✔	✔
ip6 <*subcommand*> <*flags*> (Technician Interface only)	Display data from IPv6; show a different view for each subcommand or flag you enter.	✔	✔
list [<*instances*> [<*obj_name*>]] (Technician Interface only)	List objects in the MIB.	✔	✔
loadmap [<*slot_list*> | **all**] [<*filepath*>]	Display the load address and size of each dynamically loadable application (for example, a protocol).	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
log [*<vol>*:*<logfile>*] [**-d***<date>*] [**-t***<time>*] [**-e"***<entity>"*] [**-f***<severity>*] [**-s***<slot_ID>*] [**-p***<rate>*] [**-c***<code #>*]	Display the current system event log. Follow the optional **-e** flag immediately with the entity name in uppercase characters and enclosed in quotation marks (no intervening spaces). Specify *<severity>* using letters with no intervening spaces: f = fault w = warning i = info t = trace d = debug **Examples:** -fwid -ffitd -fwi -fwitd Use the optional **-p** flag to set an interval for polling the log and displaying the result.	✔	✔
log [**-x**|**-i**] [**-e"***<entity>"*] [**-f***<severity>*] [**-s***<slot_ID>*]	Excludes (**-x**) or includes (**-i**) event logging indicated by the command options.	✔	✔
log -z [**-s***<slot_ID>*]	Displays current filter setting.	✔	✔
logout	Exit the current login session.	✔	✔
lso [**-l**] (BCC configuration mode only)	List objects configured on the current object. The optional **-l** flag causes the BCC to list object IDs in wraparound screen format.	✔	✔
mget {*<obj_name>* | *<obj_id>*} . {*<attr_name>* | *<attr_id>* | * } [. {*<inst_id>* | *}] (BCC only)	Retrieve the values of data objects in the MIB.	✔	✔
mlist [**instances** [*<obj_name>*]] (BCC only)	List objects in the MIB.	✔	✔
more [**on** | **off**] [*<#_lines_per_screen>*]	Set or display the status of the **more** utility.	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
mset {*<obj_name>*|*<obj_id>*}**.**{*<attr_name>*| *<attr_id>*}.{*<inst_id>*} {*<value>*} (BCC only)	Modify (set) the values of data objects in the device MIB.	✔	
partition {create|delete} [*<vol>***:**]	Create or delete a partition on existing file system media.	✔	✔
password Manager	Change the password of the Manager account.	✔	
password User	Change the password of the User account.	✔	✔
ping -*<protocol> <address>* [**-t***<timeout>*]	Initiate an ECHO request/reply handshake.	✔	✔
pktdump *<line_number>* [**-s***<start>*] [**-c***<count>*]	Display packets that have been captured by an interface configured for Packet Capture.	✔	
prom [**-v**|**-w**] *<vol>*:*<ROM_Update_File><slot_ID>* [*<slot_ID>*...]	Update or verify the software located on a flash PROM device.	✔	
pwc (BCC configuration mode only)	Display the path to your current working location in the tree, starting from root.	✔	✔
readexe *<vol>*:*<filename>*	Validate the checksums of an executable image and print out all the file header information.	✔	✔
record {**open**|**close**} [**-fileonly**] [**-pause**] *<vol>*:*<filename>* (Technician Interface only)	Record to a file all messages written to the terminal. You can open, pause, and close a recording session.	✔	
reset [*<slot_ID>*]	Reboot the GAME image on the specified slot. If the slot ID argument is absent, reboot the entire device.	✔	
restart [*<slot_ID>*]	Restart the GAME image on the specified slot. If the slot ID argument is absent, the GAME image restarts on all slots.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
rm *<vol>***:***<filename>* (BCC only)	Remove (delete) the file from the specified volume.	✔	
save {**config** | **aliases** | **log**} *<vol>***:***<filename>*	Store the current configuration, alias list, or system event.	✔	
securelogin	Turn SecurID access to the device on and off via Telnet.	✔	
set { *<obj_name>* | *<obj_id>* } .{*<attr_name>*| *<attr_id>* } . {*<inst_id>* } {*<value>*} ... (Technician Interface only)	Modifies data objects in the MIB.	✔	
show config [**-all**] (BCC only)	Show the total configuration of the device (use **-all**) or the total configuration of the current object (omit -**all**).	✔	✔
source *<vol>***:***<filename>* (BCC configuration mode only)	Read BCC configuration commands from a text file, dynamically changing the active device configuration.	✔	
source {**aliases** | **env** | **perm** } *<vol>*:*<filename>* (Technician Interface only)	Read a list of aliases, environment variables, or dynamic permissions from a file.	✔	
stamp	Display the device image version name and timestamp.	✔	✔
system (Technician Interface only)	Start a new Technician Interface session that allows you to run system manager privileged commands.	✔	✔
telnet [**-d**] [**-e** *<escape_char>*] [*<host_ip>* [*<port>*]] (Technician Interface and BCC top level only; not available in BCC configuration mode)	Communicate with other hosts supporting the Telnet protocol.	✔	✔
tftp {**get** | **put**} *<address>* *<vol>***:***<filename>* [*<vol>***:***<filename>*] (Technician Interface and bcc> top level only)	Send files to, or retrieve files from, other hosts supporting TFTP.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
type [**-x**] *<vol>***:***<filename>*	Display the contents of the designated file in ASCII or HEX (**-x**) format.	✔	✔
xmodem {**rb** | **sb}** [**y** | **w** | **p** | **n**] *<filename>* **...**	Transfer files to or from this device over a dial (out-of-band) connection.	✔	

Appendix B
TCL Support

The BCC supports the following subset of Tool Command Language (TCL) scripting commands on the router platform:

- append
- break
- case
- catch
- close
- concat
- continue
- eof
- error
- eval
- exit
- expr
- flush
- for
- foreach
- gets
- global
- if
- incr
- interp
- join
- lappend
- lindex
- linsert
- list
- llength
- lrange
- lreplace
- lsearch
- lsort
- open
- proc
- puts
- rename
- return
- set
- source
- split
- subst
- switch
- unset
- uplevel
- upvar
- while

For more information about definitions, syntax, and applications for these TCL commands, refer to the following book:

Ousterhout, J. *Tcl and the Tk Toolkit*. Reading, Mass.: Addison-Wesley, ISBN 0-201-63337-X.

Appendix C
System show Commands

This appendix describes the following BCC system **show** commands.

Topic	Page
show console	C-2
show hardware	C-5
show process	C-7
show system	C-10

For information about the **show config** command, see "Displaying Configuration Data" on page 2-8.

show console

The **show console** *<option>* commands display console port configuration and statistics information.

The **show console** command supports the following options:

config	stats

config

Displays configuration and Technician Interface environment information for the serial ports configured on your system or for a specific port.

The table includes the following information:

Port Number	Port number for the information displayed. Valid ports are 1, 2, 3, and 4. Not all systems have four physical ports. A configured port that does not exist is in the Absent state.
State	Port's current state, as follows: • *Absent* - Not physically present • *Disabled* - Unavailable • *Down* - Unavailable • *Init* - Initializing • *Up* - Available
Port Name	Name that the system assigns to the port. You cannot specify a name. You can use the name to correlate a port number to a physical port. The name of the port should be printed next to the physical port connection, for example, Port 1 - CONSOLE. The names do not specify the port's use. All ports are serial ports used for Technician Interface sessions only. For example, port MODEM1 may be a modem connection or a dummy terminal connection depending on its configuration. Although port 4 is called PRINTER, it is exactly like ports 1, 2, and 3. Port 4 doesn't support a printer. Port 4 is called PRINTER only because that label is printed near the port connector on the link module.
Slot Number	Slot on which the login session for the serial port is running.
Baud Rate	Current baud rate setting for the serial port.
Data Bits	Number of data bits in the serial port's configuration.

Parity	Serial port's current parity setting.
Stop Bits	Number of stop bits in the serial port's configuration.
Modem Enable	Configuration of modem control, as follows: • *Disabled* - Port is directly connected to a device, such as a dummy terminal or a terminal server. • *Enabled* - Port is attached to a modem and modem leads are enabled.
Lines/Screen	Number of lines that the serial port displays before displaying the more prompt.
More Enable	Setting of the Technician Interface **more** feature: Enabled or Disabled (according to the MIB record). The Technician Interface **more** command affects only the current login session; it does not change the MIB, and so does not affect the setting of this field.
Port Prompt	Technician Interface prompt.
Login Retries	Maximum number of login retries; relevant only if modem control is enabled. This value determines the maximum number of failed login attempts that a system allows on the serial port. If the maximum occurs, the system hangs up on the line, causing a modem connection to lose carrier detect.
Login Timeout (min.)	Number of minutes allowed between when the system displays the login banner and a user enters a login ID; relevant only if modem control is enabled. If this timeout occurs, the system hangs up on the line.
Password Time Out	Number of minutes allowed to enter a password. If this timeout period expires, the system hangs up on the line.
Command Time Out	Command line timeout value; relevant only if modem control is enabled. If you do not enter a command in this number of minutes, the system hangs up on the serial port.
User Abort Logout	Switch to execute control (^C) to break out of the user autoscript. When a user autoscript is in effect and this parameter is enabled, you can break out of the script when logged in as User, but not as Manager. Also, if this parameter is enabled and the script terminates due to an error, the system automatically logs you out.
Initial Search Path	List of file system volumes to be searched when you run a script without a volume specifier or if an autoscript does not contain a volume specifier. The environment variable PATH is set to this string. The string format is as follows: *<vol>*:[;*<vol>*: ...] Example: 2:;3:;4:;5:

Manager's AutoScript	Name of the script to run when the Manager account logs in to the router. If the script name does not contain a volume specifier, the system searches the volumes listed in "Initial Search Path."
User's AutoScript	Name of the script to run when the User account logs in to the router. If the script name does not contain a volume specifier, the system searches the volumes listed in "Initial Search Path."
History Depth	Maximum number of Technician Interface commands stored in the local command history table. The table stores each command you enter at the prompt on a first in first out (FIFO) basis.
# files to autosave	Number of times that the system saves the events log to a new file automatically when the log is full.
Volume for autosave	The target volume where the system stores new log files saved through the log autosave feature.

stats

Displays login information and console port error statistics.

The table includes the following information:

Port Number	Port number for the information displayed.
Port Name	Corresponding port name.
Port State	Port's current state: Absent, Disabled, Down, Init, or Up.
Total Logins	Number of logins (failed and successful) on the console port.
User Login Errors	Number of failed login attempts that the User login account has made on the console port.
Manager Login Errors	Number of failed login attempts that the Manager login account has made on the console port.
Other Login Errors	Number of failed login attempts made by login accounts other than User and Manager on the console port.
TTY Frame Errors	Number of frame errors on the console port.
TTY Overrun Errors	Number of overrun errors on the console port.
TTY Parity Errors	Number of parity errors on the console port.
TTY FIFO Errors	Number of FIFO errors on the console port.

A high number of errors over a short period of time may indicate a problem with the line.

show hardware

The **show hardware** *<option>* commands display information about router hardware.

The **show hardware** command supports the following subcommand options:

backplane	memory [<slot>]
config_file	proms
image	slots

backplane

Displays information about the state of the backplane hardware. The table includes the backplane type, revision, and serial number. The revision and serial numbers are in decimal format.

config_file

Displays the configuration file used to boot the router or reset a slot. The table shows the volume and file name used as the source of the configuration. The table also shows the date and load time.

image

Displays the router's software image for each slot, including the integration that is the source of the image, the date and time of the image's creation, and the file name that contains the image.

memory [<*slot*>]

Displays memory configuration and capacity information about all slots or a specific slot. The table includes the following information:

Slot	Slot number.
Local Memory	Total memory capacity in megabytes of the processor on the slot.
Global Memory	Current memory configuration in megabytes of the processor on the slot.
Total Memory	Total local and global memory in megabytes.

proms

Displays PROM information for all slots. The table includes the revision and build date of the bootstrap PROM and the diagnostics PROM.

slots

Displays hardware information about all slots in the system. The table includes information about the processor module and link module for each slot, as well as the module type, revision, and serial number. The revision and serial numbers are in decimal format.

For the AN, the table indicates that the AN has an 802.3 repeater (HUB) by indicating that the link module is an ANSEDSH.

For the ASN, the table displays the revision and serial number of the chassis, processor module, and the network module type, revision, and serial number.

show process

The **show process** *<option>* commands display information about the use of resources (buffers, CPU, memory, lists, and so on) on the router.

The **show process** command supports the following subcommand options:

buffers [detail | total]	list [detail | total]
cpu [detail | total]	memory [detail | total]

buffers [detail | total]

Displays the number and percentage of buffers used by all processes on the router.

The table includes the following information:

Name	Name of the process (if you specify the detail option).
Slot	Slot number.
Used	Number of buffers used.
%Used	Percentage of buffers used.
Max	Maximum buffers (if you specify the total option).
Free	Free buffers (if you specify the total option).

cpu [detail | total]

Displays the CPU usage in hundredths of seconds and the percentage of total CPU time used by all processes on the router.

detail	Displays details about CPU usage for all processes.
total	Displays only CPU statistics totals.

Note: For information about how to obtain a list of available processes, see the **show process list** command.

The table includes the following information:

Name	The name of the process.
Slot	Slot number.
Used	Hundredths of seconds used by each process.
%Used	Percentage of CPU time used by each process.
Idle	CPU idle time in hundredths of seconds.
Max	Total CPU time in hundredths of seconds.

list [detail | total]

Displays a list of all the processes running on each slot.

show process list [**detail** | **total**]

The table includes the following information:

Name	The name of the process.
Slot	Slot number.

memory [detail | total]

Displays the number of bytes and the percentage of memory used by all processes on the router.

detail	Displays details about the amount of memory used by each process running on the router.
total	Displays only memory usage totals.

The table includes the following information:

Name	The name of the process (if you specify the detail option).
Slot	Slot number.
Used	Number of memory bytes used by each process.
%Used	Percentage of memory used by each process.
Free	Number of free buffers on this slot (if you specify the total option).

show system

The **show system** *<option>* commands display information about the overall system state.

The **show system** command supports the following subcommand options:

buffers	memory
drivers	protocols
information	tasks

buffers

Displays the current buffer usage for all active slots on the router. Because buffers circulate rapidly through the system, a low free percentage does not necessarily indicate a buffer shortage; it may be a transient condition.

The table includes the following information:

Slot	Slot number.
Used	Number of buffers used by each process.
%Free	Percentage of free buffers on each slot.
Total	Total number of buffers available.
Free	Total number of free buffers.

drivers

Displays link modules and drivers installed on all slots. If the configuration displayed differs from that expected, your configuration file may be incorrect (wrong module type specified, for example) or there may be a problem loading the software.

information

Displays general system information (system name, contact, node location, image data, MIB version, and total uptime since last cold boot).

memory

Displays the global memory usage for all active slots in the system. Memory usage is not as volatile as buffer usage, so a low free percentage may indicate that you need more memory.

The table includes the following information:

Slot	Slot number.
Total	Total number of memory (DRAM) bytes available on each slot.
Used	Number of memory bytes used on each slot.
Free	Amount of free memory on each slot.
%Free	Percentage of free memory on each slot.

protocols

Displays the protocols installed on all slots in the system. If the configuration displayed differs from that expected, your configuration file may be incorrect (wrong protocol specified, for example) or there may be a problem loading the software.

tasks

Displays the number of tasks scheduled to run on all slots. This number is highly volatile and a large In Queue value does not necessarily indicate a problem.

The table includes the following information:

Slot	Slot number.
Total	Total number of tasks running on each slot.
In_Queue	Number of tasks scheduled to run.
% in Queue	Percentage of tasks scheduled to run.

Appendix D
Syntax for Module Location

Table D-1 lists the syntax for specifying the physical location of a module for each Bay Networks device that the BCC supports.

Table D-1. Syntax for Specifying Module Location per Device

Platform	Syntax
AN/ANH	***<interface> <slot> <connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, etc. • *<slot>* = 1 (AN/ANH is a one-slot device) • *<connector>* numbering starts with connector 1. ***Example:*** **ethernet slot 1 connector 3** This is an Ethernet interface configured on AN/ANH connector 3, which exists on an Ethernet adapter module. (Connectors 1 and 2 are on the base module.)
ASN	***<interface> <slot> <module> <connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, etc. • *<slot_number>* = 1, 2, 3, or 4, depending on the setting of the module ID switch on each ASN. • *<module>* numbering corresponds to net module numbering (1 through 4) on each ASN. • *<connector>* numbering starts at 1 on each net module (per media type). ***Example:*** **ethernet slot 2 module 3 connector 2** This is an Ethernet interface on connector 2 of net module 3, in ASN 2 of a stacked ASN configuration.

(continued)

Table D-1. Syntax for Specifying Module Location per Device *(continued)*

Platform	Syntax
ARN	***<interface> <slot> <connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, etc. • *<slot>* = 1 • *<connector>* numbering depends on the port type (LAN or WAN). LAN connector numbering starts at 1 on the base module, which contains only LAN ports. LAN connector numbering continues in ascending order, starting with the first LAN port on an ARN expansion module. (The ARN expansion module plugs into the ARN base module.) WAN connector numbering starts with connector 1 on WAN adapter module 1, continues with connector 2 on WAN adapter module 2, and ascends sequentially with WAN connectors 3 through *n* on the ARN expansion module. ***Example:*** **ethernet slot 1 connector 2** This interface is configured on LAN connector 2, which exists physically on an Ethernet expansion module. (Ethernet connector 1 is on the base module.) ***Example:*** **serial slot 1 connector 3** This is a serial (WAN) interface configured on WAN connector 3, which exists physically on the ARN expansion module.
BN	***<interface> <slot> <connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, etc. • *<slot>* = 2 through 5 (BLN) or 1 through 14 (BCN). • *<connector>* numbering starts with connector 1 on each slot (per media type). ***Example:*** **ethernet slot 8 connector 3** This is an Ethernet interface configured on connector 3 of slot 8 of a BCN router.
System 5000	***<interface> <slot> <module> <connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, atm, etc. • *<slot_number>* = 2 through 13 in a System 5000 chassis • *<module>* numbering corresponds to net module 1 or 2, which plugs into a System 5000 base module. A System 5000 base module is always module 3. • *<connector>* numbering starts at 1 on each net module. The base module always contains a connector 1, which plugs into a System 5000 backplane (for example, Ethernet, token ring, or ATM backplane). ***Example:*** **ethernet slot 2 module 2 connector 1** This is an Ethernet interface on connector 1 of the Ethernet net module (module 2) in slot 2. **atm slot 2 module 3 connector 1** This is an ATM interface on connector 1 of slot 2, a Model 5782 Virtual Network Router (VNR) base module (module 3). Note that the VNR does not accommodate any net modules.

Index

Symbols

{, 3-6

A

acronyms, xvi

B

backplane hardware, C-5
BCC
  accessing, 2-1
  command files, 3-10
  commands
    entering, 3-1
    show, C-1
    system, A-1
  configuration cycle, 4-4
  defined, 1-1
  Help, 2-11, A-4
  platform requirements, 1-2
  terminology and concepts, 1-6
board, defined, 1-8
bootstrap PROM, revision date, C-6
box-wide objects, 1-7
buffers, system, C-10

C

class, defined, 1-6
commands
  ?, 2-11
  back, 2-3
  config, 1-2
  configuration, 3-5
  continuing, 3-4
  editing, 3-3
  entering, 3-1
  Help, 2-11
  lso, 2-10
  pwc, 2-3
  show config, 2-8, 2-10
  show console, C-2
  show hardware, C-5
  show process, C-7
  show system, C-10
  source, 3-12
  system, A-1
connector, defined, 1-9
console subcommands, C-2
context
  changing, 2-3
  defined, 1-5
  displaying, 2-3
continuing a command line, 3-4
conventions, text, xiv
creating a new configuration, 4-1

D

delete command, 3-10, 4-17
derived parameters, defined, 1-10
diagnostics PROM, revision date, C-6
disable command, 3-10, 4-15

E

editing commands, 3-3
educational services, xvii
enable command, 3-10, 4-16

entering the BCC, 2-1
exiting the BCC, 2-1

G

global objects, defined, 1-7

H

hardware information, C-5
Help commands, 2-11, A-4
history buffer, 3-2

I

image, software, C-5
instance identifier, 1-7, 2-6
interface, defined, 1-9

L

line, defined, 1-9
location in configuration hierarchy, 2-3
login privileges and procedures, 2-1
lso command, 2-10

M

Manager/User privilege levels, 2-2
memory
  hardware, C-6
  system, C-11

N

navigating
  using configuration commands, 2-4
  using the back command, 2-3

O

objects
  defined, 1-6
  deleting, 4-17
  disabling, 4-15
  enabling, 4-16
online Help, BCC, 2-11
optional parameters, defined, 1-10

P

parameters
  defined, 1-10
  Help, 2-11
  specifying values for, 3-8
path, specifying, 2-6, 2-7
port, defined, 1-9
product support, xvii
PROM information, hardware, C-6
protocols, show system command, C-11
publications
  Bay Networks, xvi
pwc command, 2-3

R

reading commands from a file, 3-12
recalling commands, 3-2
required parameters, defined, 1-10

S

saving configuration commands, 3-11
serial port
  configuration, C-2
  error statistics, C-4
show commands
  config, 2-8
  console, C-2
  hardware, C-5
  process, C-7
  system, C-10
show config command, 2-8
slot, defined, 1-8
slots, hardware, C-6
source command, 3-12
specifying a path, 2-6, 2-7

statistics, console, C-4
support, Bay Networks, xvii
syntax
  configuration commands, 3-5, D-1
  Help, 2-11
system information, C-11

T

TCL support, B-1
technical publications, xvi
technical support, xvii
Technician Interface, 2-1
text conventions, xiv

U

User privileges, 2-2