Trademarks

Telebit, NetBlazer, and TrailBlazer are trademarks of Telebit Corporation.

Other brands or product names are trademarks or registered trademarks of their respective holders.

Software Copyright Notice

Printed in U.S.A.

How to Use This Manual 1

This manual is intended as a companion document to the *NetBlazer Configuration Guide*. It provides detailed information for each of the NetBlazer's commands, including an overview, syntax, options, and examples of use.

This manual is organized such that each command has its own chapter. This streamlines usability and facilitates future updating and the possible addition of new commands.

Audience

This guide is intended as a reference aid to the System Administrator. It provides specific NetBlazer command usage information to the system administrator performing NetBlazer configuration or network maintenance activities.

Related Documentation

Hardware installation of the NetBlazer products are detailed in the *NetBlazer Installation Guides*.

Once the installation of the NetBlazer has been completed, software configuration must be performed. This procedure involves establishing the IP addresses and routes and is covered in the *NetBlazer Configuration Guide*. It also covers network "fine-tuning" and management activities.

Once the NetBlazer has been configured, the system administrator can turn his or her attention to getting the users up to speed with the NetBlazer. The *NetBlazer User's Guide* provides a user-oriented overview of the NetBlazer operation, its user-level commands, and shows step-by-step procedures for performing basic user functions.

NetBlazer Training

Telebit's Customer Services group offers excellent NetBlazer training on topics such as "Introduction to Internetworking and TCP/IP," and "NetBlazer Technical Training." This comprehensive course combines easy-to-read training materials with insightful lecture and labs. For more information and class schedules, contact Telebit Education and Training Services at 1-800-TELEBIT.

Table of Contents

Command Reference INTRO-1
Typographic Conventions INTRO-1
Hierarchical Command Set INTRO-1
Common Commands INTRO-2

Alias Commands ALI-1
alias add [<alias>] [<alias-string>] ALI-1
alias delete <alias1> [<alias2> ...] ALI-2
alias list [<alias>] ALI-2

ARP Commands ARP-1
arp add <hostid> ether|token <mac-addr> ARP-1
arp delete <hostid> [<interface>] ARP-2
arp list ARP-2
arp proxy <hostid> ether|token|<interface> <mac-addr> ARP-2

Configure Command CON-1

Debug Command DEB-1
debug alloc [<value>] DEB-1
debug args DEB-2
debug autoconfigure DEB-2
debug clear DEB-2
debug core DEB-2
debug db <address> [<count>] DEB-2
debug devs DEB-2
debug dh <address> [<count>] DEB-2
debug display <address> [<count>] DEB-3
debug dl <address> [<count>] DEB-3
debug dump DEB-3
debug files DEB-3
debug filter DEB-3
debug gdt <start> [<end>] DEB-3
debug hardware DEB-3
debug hex <filename> DEB-4
debug hogmemory <bytes> DEB-4
debug idle DEB-4
debug idt <start> [<end>] DEB-4
debug inb <port> <count> DEB-4
debug irq <value> DEB-4
debug maplist <addr> <count> DEB-5
debug memory <command> DEB-5
debug memset <addr> <char> <length> DEB-5

debug noautoconfigure ... DEB-5
debug nocore ... DEB-5
debug outb <port> <value> ... DEB-5
debug printf <fmt> <value ...> ... DEB-5
debug probe <device> ... DEB-6
debug process [<pid>] ... DEB-6
debug ps ... DEB-6
debug psignal <value> <count> ... DEB-6
debug remote [off] [on] [file <filename>] [myip <myip>] [list] [host <ip>] [ether <hardware-addr>] [test] ... DEB-6
debug show ... DEB-7
debug sleep <delay> ... DEB-7
debug status [arguments] ... DEB-7
debug timers ... DEB-10
debug unmap <address> <count> ... DEB-10
debug watch <addr> <type> [124] [WRE] ... DEB-10
debug writebyte <address> <value> ... DEB-10
debug writelong <address> <value> ... DEB-10
debug writeshort <address> <value> ... DEB-11

Destination Commands ... DES-1
destination add [<dest-group>] [<short-name>] ... DES-1
destination delete <dest-group> [<dest-number>] ... DES-2
destination host <dest-group> <dest-number> <dialout>|<telnet-name> ... DES-2
destination list [<dest-group>] ... DES-3
destination longname <dest-group> <dest-number> <long-name-string> ... DES-3
destination type <dest-group> <dest-number> dial | telnet ... DES-3

Dialout Commands ... DIA-1
dialout add [<dialout>] [<phone-number>] [<characteristics>] [<timeout>] ... DIA-1
dialout characteristics <dialout> [#<item-number>] <group> ... DIA-2
dialout delete <dialout> [#<item-number>] ... DIA-2
dialout ditto <dialout> [#<item-number>] <phone-number> ... DIA-3
dialout list [<dialout>] [#<item-number>] ... DIA-3
dialout maxwait <dialout> [#<item-number>] <seconds> ... DIA-3
dialout minwait <dialout> [#<item-number>] <seconds> ... DIA-3
dialout phone <dialout> [#<item-number>] <phone-number> ... DIA-4
dialout timeout <dialout> [#<item-number>] <timeout-minutes> ... DIA-4

Disk Commands ... DIS-1
disk chmod <mode|+w|-w> <file> ... DIS-2
disk copy <source-file> <destination-file> ... DIS-2
disk delete <filename> ... DIS-2
disk dir [<filename.ext>] ... DIS-2
disk diskcopy ... DIS-3
disk format ... DIS-4
disk rename <oldfile> <newfile> ... DIS-4

Domain Commands DOM-1
domain add <servername> DOM-1
domain cache DOM-1
domain delete <servername> DOM-1
domain list DOM-2
domain suffix [<suffix-string>] DOM-2

Dynamic Commands DYN-1
dynamic add [<interface>] [<uname>] [<password>] [<crypto>] [<chat>] [slip|ppp] [<max-lines>] [<mru>] DYN-1
dynamic auto-arp <interface> <add|delete> <ip_addr ...> DYN-2
dynamic auto-route <interface> <add|delete> <ip_addr[/bits] ...> DYN-2
dynamic chat <interface> <chat-script-name> DYN-4
dynamic crypto <interface> [<crypto-key>] DYN-5
dynamic delete <interface> DYN-5
dynamic framing <interface> slip|ppp DYN-5
dynamic list [<interface>] DYN-5
dynamic max-lines <interface> <max-lines> DYN-5
dynamic password <interface> <remote-password> DYN-6
dynamic queue-max <interface> <queue-length> DYN-6
dynamic timeout <interface> <timeout-seconds> DYN-6
dynamic trigger <interface> <trigger-depth> DYN-6
dynamic uname <interface> <remote-uname> DYN-7

Echo Command ECH-1

Filter Commands FIL-1
filter add <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto> [port]] [source|destination] [in|out] permit|deny FIL-1
filter delete <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto>] [source|destination] [in|out] FIL-4
filter deny <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto> [port]] [source|destination] [in|out] FIL-6
filter flush [<interface>] FIL-6
filter list <interface> FIL-6
filter lookup <address>[/<bits>] <address>[/<bits>] <interface> [<proto> [port]] FIL-7
filter notify [on|off] FIL-7
filter permit <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto> [port]] [source|destination] [in|out] FIL-7

Grep Command GRE-1

Group Command GRO-1
group add <group-name> GRO-1
group delete <group-name> GRO-1
group modify <group-name> add | delete <string>
group modify <group-name> delete #<string-number> GRO-1
group list [<group-name>] GRO-2

History Commands HIS-1
history status HIS-1
history on [<number-of-lines>] HIS-1
history off HIS-2
history prompt HIS-2
history list [<number-of-lines>] HIS-2
Referencing Previous Commands HIS-3
Repeating Previous Commands HIS-3
Building New Commands from Previous Commands HIS-4
Editing Previous Commands HIS-5
History Command Summary HIS-6

Hostname Command HOS-1

ICMP List Command ICM-1

Interface Commands INT-1
interface attach <line-name> slip|ppp <interface> [<mru>] [ip_addr] INT-1
interface delete <interface> INT-3
interface dump-mib <interface> INT-3
interface list [<interface>] INT-3
interface mru [<interface>] [<mru>] INT-3
interface mtu [<interface>] [<mtu>] INT-4
interface rif <interface> [<node_address>] INT-4
interface start <interface> INT-4
interface stop <interface> INT-4
interface sync-backup [<interface>] INT-4
interface watchdog <interface> on|off|<IP address> [<seconds>] INT-5

IP Commands IP-1
ip address [<ip-addr>[/bits]] [<interface>|<line> [<port>]] IP-1
ip farside <interface> IP-3
ip ipdial IP-3
ip list IP-5
ip rtimer [<reassemble-time>] IP-5
ip trace [<interface>] [<options ...> [<filename>]] IP-6
ip ttl [<time-to-live>] IP-8
ip vj [<interface>] [on|off|auto] IP-8
Using VJ Header Compression on a SLIP Interface IP-8
Using VJ Header Compression on a PPP Interface IP-9

Line Commands LIN-1
line access <line-name> [packet|all] LIN-1
line auto <line-name> [on|off|config] LIN-2
line callin <line-name> [on|off] LIN-3
line command <line-name> <command> [<arguments>] LIN-4
line characteristics <line-name> [<group-name-of-characteristics>] LIN-5
line dcdtime <line-name> [<seconds>] LIN-5
line delay <line-name> [<milliseconds>] LIN-6
line dsrtime <line-name> [<seconds>] LIN-6
line dtrtime <line-name> [<milliseconds>] LIN-7
line flowcontrol <line-name> [none|auto|xon|cts] LIN-7
line idle <line-name> [minutes] LIN-8
line list [<line-name>] LIN-9
line modem <line-name> [<modem-name>] LIN-10
line request-call <line-name> [on|off] LIN-11
line reset <line-name> LIN-11
line speed <line-name> <speed> LIN-11
line terminal <line-name> [<terminal-type>] [-rows <rows>] [-cols <cols>] LIN-12
line test-request <line-name> LIN-13
line trace <line-name> [on|off] LIN-13

List Command LIS-1
list commands LIS-1
list ethernet LIS-1
list mem LIS-2
list pgroup LIS-2
list ps LIS-2
list slip LIS-2
list socket [<number>] LIS-2
list stack LIS-2
list udp LIS-2
list uptime LIS-3
list version LIS-3

Logout Command LOG-1

Message Command MES-1
message hello [<message>] MES-1
message list MES-1
message login [<message>] MES-2
message shutdown [<message>] MES-2

More Command MOR-1

Msglog Command MSG-1

Passwd Command PAS-1

Phonelist Commands PHO-1
phonelist delete <phonenumber...> PHO-1
phonelist deny default | <phonenumber ...> PHO-1
phonelist flush PHO-2
phonelist permit default | <phonenumber ...> PHO-2
phonelist list PHO-2
phonelist test <phonenumber> [<phonenumber> ...] PHO-2

PPP Commands PPP-1
ppp activeopen <interface> PPP-1
ppp close <interface> PPP-1
ppp list [<interface>] PPP-1
ppp passiveopen <interface> PPP-1
ppp reset <interface> PPP-2

Reboot Command REB-1

RIP Commands RIP-1
rip accept <gateway> RIP-1
rip add <destination> <interval> [<split-horizon>] RIP-1
rip delete <destination> RIP-2
rip list RIP-2
rip merge [on | off] RIP-2
rip refuse <gateway> RIP-2
rip request <gateway> [<port-number>] RIP-2
rip timeout <interface> <seconds> RIP-3

Route Commands ROU-1
route add <dest-addr>[/<bits>] <interface> [<gateway>] [<metric>] ROU-1
route addprivate <dest-addr>[/<bits>] <interface> [<gateway>] [<metric>] ROU-2
route change <dest-addr>[/<bits>] <interface> [<gateway>] [<metric>] ROU-2
route delete <dest-addr>[/bits] [<interface>] ROU-3
route flush ROU-3
route list ROU-3
route listlocal ROU-4
route lookup <dest-addr> ROU-4
route metric <dest-addr>[/<bits>] <metric> ROU-5
route private <dest-addr>[/bits] <interface> ROU-5
route public <dest-addr>[/bits] <interface> ROU-5

Save Command SAV-1

Security Commands SEC-1
security write [enable|disable] [user|priv|status] SEC-1
security login [enable|disable] SEC-2

Servers Commands SER-1
servers list SER-1
servers start <server-name> SER-2
servers stop <server-name> SER-2

Sessions Commands SES-1
session abort [<session-number>] SES-1
session bang <host> [<length> <count> <interval> <outstanding>] SES-1
session binary <session-number> [on|off] SES-2
session close [<session-number>] SES-2
session connect <destination-name>|<destination-number> SES-2
session continue [<session-number>] SES-3
session dial <site-name> SES-3
session disconnect [<session-number>] SES-3
session eol [standard | unix] SES-3
session escape [<character>] SES-4
session fg [<session-number>] SES-4
session finger [<user>] [@<host>] SES-4
session ftp [<address>] SES-5
session jobs SES-5
session kill [<session-number>] SES-5
session list SES-6
session ping <host> [<length> <count> <interval> <outstanding>] SES-7
session rlogin [-b|-8] [-noesc] [-l username] [<address>] [<port>] SES-7
session send <session-number> ao|ayt|brk|ec|el|esc|ga|ip|nop|synch SES-8
session talk [<line-name>] SES-9
session telnet [-b|-8|-raw] [-noesc] [<address>] [<port>] SES-9
session terminal [<terminal-type>] [-rows <rows>] [-cols <cols>] SES-10

Setconsole Command SET-1

Shutdown Command SHU-1

SNMP Command SNM-1
snmp get-community [<community-name>] SNM-1
snmp list SNM-1
snmp set-community [<community-name>] SNM-1
snmp sysContact <sysContact-string> SNM-2
snmp sysDescr <description-string> SNM-2
snmp trap-community <community_name> SNM-2

SNMP Trap Commands SNM-3
snmp trap add <manager-address> SNM-3
snmp trap authentication on | off SNM-3
snmp trap delete <manager-address> SNM-3
snmp trap list SNM-3
snmp trap startup on | off SNM-4

Source Command SOU-1

Syslog Commands SYS-1
syslog host off | on [<loghostname>] [<facility>] SYS-2
syslog console off | on [<level>|<=level>] SYS-4
syslog terminal off | on [<level>|<=level>] SYS-5
syslog buffer off | on [<level>|<=level>] | list [<file name>] | size <sizeK> SYS-6
syslog list SYS-7

TCP Commands TCP-1
tcp irtt [<int-rnd-trip-time>] TCP-1
tcp list [<tcb>] TCP-1
tcp mss [<max-seg-size>] TCP-2
tcp reset <tcb> TCP-2
tcp window [<max-window-size>] TCP-2

Time Command TIM-1

TTY Command TTY-1

User Command USE-1
user add [<userid>] USE-1
user changeable [on|off] USE-3
user crypto <userid> [<crypto-key>] USE-3
user delete [<userid>] USE-3
user destgroup <userid> <destination-group> USE-4
user epassword <userid> <encrypted-password> USE-4
user list USE-4
user load <passwd> USE-4
user password <userid> [<password>] USE-5
user privilege <userid> [-]sysconf | [-]sysstat USE-6
user restrict <userid> [-]dial | [-]telnet USE-6
user save USE-6

Wall Command WAL-1

Who Command WHO-1

Write Command WRI-1

Command Reference

This manual provides a complete description of all the commands used by the NetBlazer. Each command description shows the command format followed by an explanation of the command and its associated parameters.

Typographic Conventions

The following typographic conventions are used in this manual.

keyword Key words shown in bold type are NetBlazer commands. You only need to enter enough of the key word to uniquely distinguish it from other key words used by the NetBlazer. For example, entering **host** is equivalent to entering **hostname**.

<parameter-name> Parameter names enclosed in angle brackets serve as placeholders for information which you must supply. Substitute an appropriate value for the named parameter.

[optional parameters] Parameters enclosed in square brackets are optional. A series of optional parameters, such as [<param-1>] [<param-2>] [<param-3>], must be entered in sequence. In other words, <param-1> must be entered before <param-2>, and <param-2> must be entered before <param-3>.

choice1 | choice2 Vertical bars indicate a choice between two or more items. You must choose one of the items unless all of the items are enclosed in square brackets, e.g. [choice1 | choice2 | choice3].

Hierarchical Command Set

As shown in Figure 6-1, the NetBlazer commands are arranged in a tree-structured hierarchy, starting at the top command level, and working down through several subcommand levels. A summary of the commands by functional group is provided in table.

To go to a new command level, simply enter the name of the new command level. Remember that you only need to enter enough of the command name to uniquely distinguish it from other commands. For example, to get to the Configure IP Route command level from anywhere in the command tree, enter either **route** or **ro** at the command prompt.

When a command is entered, the NetBlazer first checks the current command level for the command. If it is a valid command for that level, the NetBlazer invokes the function associated with the command at that level. If not, the NetBlazer searches the command tree for the specified command. To illustrate, assume that you are currently at the Configure User command level. Entering **list** displays a list of users; whereas entering **route list** displays the list of routes without leaving the current command level.

Common Commands

Three commands are common to every level in the command tree: **top**, **up** and **?**.

The **?** command displays a list of possible commands at the current command level. If you enter a command before the **?**, the syntax for that command is displayed.

The **up** command takes you up to the next higher level in the command tree. For instance, if you are currently at the Configure IP command level, entering an **up** command moves you up one level to the Configure command level.

The **top** command takes you to the top level of the command tree.

The remainder of this manual provides a description of each NetBlazer command arranged in alphabetical order by submenu. For example, the **route** commands which are at the Configure IP Route command level are described under the heading Route Commands.

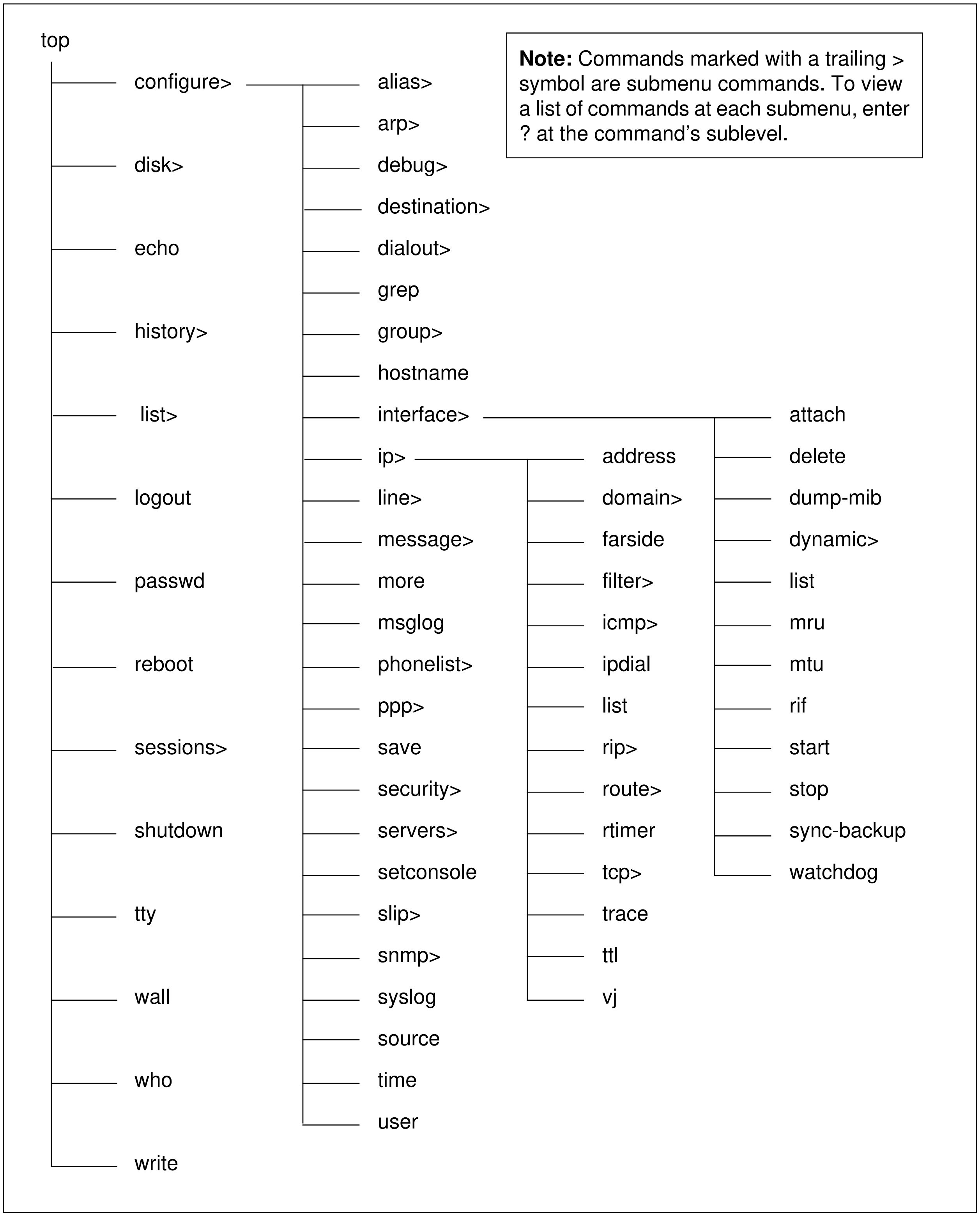


Figure INTRO-1. NetBlazer Command Tree

Table INTRO-1. Command Summary

Command	Function
Disk	
chmod	Disable/enable write protection for a specified file.
copy	Copy a specified file.
delete	Delete a specified file.
dir	List a directory of files on the diskette.
diskcopy	Create a duplicate copy of the NetBlazer diskette.
format	Initialize a blank diskette for use with the NetBlazer.
rename	Rename a specified file.
IP Packet Routing	
hostname	Assign a host name to the NetBlazer.
ip address	Assign a global IP address to the NetBlazer, and assign an IP address to each of the static interfaces. Entering an **ip address** <ip-addr> command assigns a global IP address to the NetBlazer. Entering an **ip address** <ip-addr>/<bits> <interface> command assigns an IP address to a static interface, adds an ARP broadcast address to the ARP table, and adds a network route to the routing list.
interface commands	Interface control and maintenance.
domain commands	Configure the NetBlazer as a domain name server client.
arp commands	ARP table maintenance.
route commands	Route table control and maintenance.
rip commands	Control and maintain destination list for RIP advertisements sent by the NetBlazer. Also, maintain a list of source addresses from which the NetBlazer will not accept RIP information.
filter commands	Maintain a list of IP packet filters used to restrict IP packet routing through a particular interface.
Synchronous Interface	
interface commands	Synchronous interface (syn0) control and maintenance.
route commands	Control routing IP packets through the syn0 interface.
ppp commands	Control the point-to-point framing protocol used to transfer packets over the synchronous link.
Dedicated SLIP or PPP Link	
interface commands	Attach a packet mode interface to an asynchronous line.
ppp commands	Control the point-to-point framing protocol used to transfer packets over the dedicated PPP link.

Table INTRO-1. Command Summary (Continued)

Command	Function
Dynamic (Dial-Up) Interface	
ipdial	Start a dialog that steps you through the process of creating a dynamic dial-up interface to a remote site. The NetBlazer uses information obtained from ipdial to create: • a dynamic interface record for the remote site. • one or more dialout records used to call the remote site. • a user record to authenicate a dial-in connection from the remote site. • add routes for the remote site.
dynamic commands	Set up and maintain dynamic interfaces, used to route IP packets over the public telephone network.
route commands	Maintain routes to remote networks through the dynamic interfaces.
dialout commands	Maintain a list of dial-out destinations used to establish a connection with a remote site.
users commands	Maintain a list of users authorized to log into the NetBlazer.
Users and Destination Groups	
user commands	Set up and maintain list of authorized users.
destination commands	Create destination groups, which allows authorized user to easily start a telnet or dial-out session with a predefined destination.
alias commands	Set up a special command alias that automati cally connects a user to a pre-defined host when the user logs in to the NetBlazer.
Sessions	
bang	Equivalent to the ping command except instead of printing a line of statistics for each response, only a single ! character is displayed.
binary	Turn binary mode on or off for a specified telnet session.
close	End a telnet session with a graceful, negotiated termination.
connect	Start a telnet or dialout session t using a predefined destination name.
continue/fg	Reactivate a session after it has been suspended.
dial	Establish a character mode connection with a remote site specified by a dial-out record.
disconnect	Terminate a character mode dial-out connection.
eol	Unix end-of-line marker. Functionally equivalent to a Return or Enter key-stroke. Used when a remote Unix host does not respond to a normal Return.
finger	Get information about a specific user on the network.
ftp	Transfer files between the NetBlazer and a remote host that supports FTP.
list/jobs	Display a list of sessions currently active on the NetBlazer.
kill	Force an abrupt termination of a session.

Table INTRO-1. Command Summary (Continued)

Command	Function
ping	Send ICMP echo requests to a specified host, then measures the time taken for the host to respond.
send	Issue a predefined character sequence to a remote host, such as an escape sequence.
telnet	Establishes an interactive session with an TCP/IP host.
terminal	Send terminal information to a UNIX host.
Diagnostic/Network Management	
snmp commands	A group of Simple Network Mangement Protocol commands used to obtain statistical and other network management information.
syslog commands	Send NetBlazer activity information to a specified UNIX host, to a terminal, to a buffer in memory, or to the console.
icmp	Display a list of error and control statistics maintained by the NetBlazer.
tcp commands	Set up and manage specific TCP (Transmission Control Protocol) parameters used by the NetBlazers.
list commands	Display various NetBlazer status information.
debug commands	A suite of network and NetBlazer troubleshooting facilities.
Servers	
discard	Accepts and acknowledges data received on TCP port 9. Data is discarded.
echo	Accepts and acknowledges TCP segments received at port 7. Data is echoed back to the original source.
finger	Allows other hosts to query the NetBlazer for information about NetBlazer users using the finger protocol.
raw	Allows a raw TCP connection from a host to the modem or other asynchronous device attached to a NetBlazer line.
ftp	Enables using FTP (File Transfer Protocol) to transfer new software and configuration files to the NetBlazer from a remote host.
rip	Allows the NetBlazer to learn routes to other networks from other routers on the network.
snmp	Used for network management tasks.
telnet	Allows users to log into the NetBlazer from a TCP/IP host.
v25bis	Must be turned on to use the synchronous line with a ISDN terminal adapter.
Productivity Tools	
alias	Sets up and maintain command aliases.
grep	Search a file for a specified string. Can also pipe output of a list to grep.

Table INTRO-1. Command Summary (Continued)

Command	Function
history commands	Maintain and display a list of the last 20 commands entered into the NetBlazer. Provides a “shorthand” method of entering commands based on the history list.
message commands	Enter and maintain messages displayed at startup, login and shutdown.
more	Display the contents of NetBlazer files on diskette a page (24 lines) at a time. The output of a list can also be piped to more.
source	Run NetBlazer commands stored in a file.
who	List all users currently logged onto the NetBlazer.
write	Write a message to a user currently logged onto the NeBlazer.
wall	Write to all users currently logged onto the NetBlazer.

Alias Commands

The **alias** command takes you to the Configure Alias command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Alias command level.

```
NB:Top> alias
NB:Top>Configure>Alias> ?
Available commands:
add             delete          list            top>
up>             ?
```

Commands available at this level are used to maintain a list of aliases that can be used as abbreviations of standard NetBlazer commands.

Caution: *Do not use an existing command word as an alias. Doing so will render the command inaccessible.*

alias add [<alias>] [<alias-string>]

The **alias add** command assigns a command string to an alias key word. The <alias> parameter is an abbreviation that is used to invoke the NetBlazer command specified by the <alias-string> parameter.

The following example shows how the **update** alias was created at Telebit:

```
NB:Top> alias
NB:Top>Configure>Alias> add update ftp 143.191.20.1
NB:Top>Configure>Alias> list
Alias                   Expansion
update                  ftp 143.191.20.1
```

The **update** alias shown above is a short hand way to start an FTP session at 143.191.20.1 which is used to download the latest NetBlazer software.

If the **alias add** command is entered without any parameters, the system prompt you for the alias key word and its associated command string.

Caution: *Do not enter an* ***alias add*** *command with just the <alias> parameter as this will create an alias for a carriage return. If you do this inadvertently, delete the alias using an* ***alias delete ""*** *command.*

The **alias add login_**<user> <alias-string> command automatically invokes a command, such as telnet, when a user logs in to the NetBlazer. For example, the telnet command assigned to the login_gary alias, shown below, is automatically invoked when gary logs in to the NetBlazer.

```
NB:Top>Configure>Alias> add login_gary telnet 143.191.167.254
NB:Top>Configure>Alias> list
Alias                   Expansion
login_gary              telnet 143.191.167.254
update                  ftp 143.191.20.1
```

Note: *Unless the user has configuration privileges, he or she is logged out after the login command is executed. This way the user does not need to log out of the NetBlazer after completing the telnet session.*

If you want more than one command to be invoked when a user logs in, you can create a source file containing the commands, and then use the **source** <filename> command in the user's login alias.

alias delete <alias1> [<alias2> ...]

The **alias delete** command removes one or more alias key words and their associated command strings from the list of valid aliases.

```
NB:Top>Configure>Alias> list
Alias                   Expansion
bye                     logout
quit                    logout
update                  ftp 143.191.20.1
NB:Top>Configure>Alias> delete quit bye
NB:Top>Configure>Alias> list
Alias                   Expansion
update                  ftp 143.191.20.1
```

alias list [<alias>]

The **alias list** command displays the specified alias key words and their associated command string. If this command is entered without a <alias> parameter, all aliases are displayed.

```
NB:Top>Configure>Alias> list
Alias                   Expansion
bye                     logout
quit                    logout
update                  ftp 143.191.20.1
NB:Top>Configure>Alias> list quit bye
Alias                   Expansion
            quit        logout
             bye        logout
```

ARP Commands

The **arp** command takes you to the Configure ARP command level. To view a list of the commands available at this level, type a question mark (?) at the Configure ARP command level.

```
NB:Top> arp
NB:Top>Configure>ARP> ?
Available commands:
add             delete          list            proxy
top>            up>             ?
```

The ARP, Address Resolution Protocol, commands are used to maintain a cache of IP addresses and corresponding ethernet addresses which allows the NetBlazer to find the ethernet address of a specific host on the same network given only the destination host's IP address.

Note: *When RIP is used to dynamically maintain routing information, the broadcast addresses of each ethernet network connected directly to the NetBlazer must be added to the ARP list. The preferred method for adding broadcast addresses to the ARP list is to assign an IP address to the ethernet interface using a subnet bit mask. Refer to the* ***ip address*** *command description for additional information.*

arp add <hostid> ether|token <mac-addr>

The **arp add** command adds an IP address and its associated ethernet or token ring address to the ARP list. This command can be used to provide support for systems that do not support ARP or to work around a routing problem.

The <hostid> may be a domain name or IP address of a specific host, or the broadcast IP address for a network.

The ether <mac-addr> or token <mac-addr> parameter specifies the physical address that is mapped to the <hostid>.

The following example adds a static ARP entry to the ARP list that maps the ethernet address 00:80:ad:02:13:00 to the IP address 143.191.167.254.

```
NB:Top>Configure>ARP> add 143.191.167.254 ether 00:80:ad:02:13:00
NB:Top>Configure>ARP> list
received 29 badtype 0 bogus addr 0 reqst in 12 replies 17 reqst out 20
IP addr         Type  Interface   Time Q Addr              RIF
143.191.167.254 ether any         0      00:80:ad:02:13:00 none
143.191.166.255 ether any         0      ff:ff:ff:ff:ff:ff none
143.191.167.255 ether any         0      ff:ff:ff:ff:ff:ff none
```

arp delete <hostid> [<interface>]

The **arp delete** command is used to remove an entry in the ARP list. This command may be used when an IP address assigned to an ethernet address is no longer valid which may occur when an ethernet board is changed or the IP address has been assigned to a different host or router.

The next example removes the static ARP entry that was previously added to the ARP list.

```
NB:Top>Configure>ARP> delete 143.191.167.254
```

Note that the <interface> parameter is only required when deleting a proxy ARP entry from the list.

arp list

The **arp list** command displays the current ARP status and a list of related IP and ethernet addresses. The ARP list contains two types of entries: static entries, which are manually entered by **arp add** commands, and dynamic entries, which are automatically added to the list in responds to an ARP request.

The following ARP list shows that IP address 143.191.167.254 is mapped to ethernet address 00:80:ad:02:13:00 and that the entry will be discarded in 543 seconds. The last two entries are static entries, which only can be removed via an **arp delete** command.

```
NB:Top>Configure>ARP> list
received 29 badtype 0 bogus addr 0 reqst in 12 replies 17 reqst out 20
IP addr         Type  Interface   Time Q Addr              RIF
143.191.167.254 ether any         543    00:80:ad:02:13:00 none
143.191.166.255 ether any         0      ff:ff:ff:ff:ff:ff none
143.191.167.255 ether any         0      ff:ff:ff:ff:ff:ff none
```

arp proxy <hostid> ether|token|<interface> <mac-addr>

The **arp proxy** command allows the NetBlazer to respond to ARP requests intended for another machine by supplying its own physical address. The NetBlazer then receives packets for that host and forwards them to the proper destination.

Using the **arp proxy** <hostid> <interface> command syntax is the preferred way to add a proxy ARP entry to the ARP list. This allows the NetBlazer to automatically get the physical address of the specified interface.

The **arp proxy** <hostid> ether|token <mac-addr> command syntax is provided for backward compatibility with earlier versions of the NetBlazer software.

To illustrate how proxy ARP is used, assume that you have a host with an IP address on the same network as the NetBlazer, but you must move that host to a remote location that is accessible via a dynamic connection. See Figure ARP-1 below.

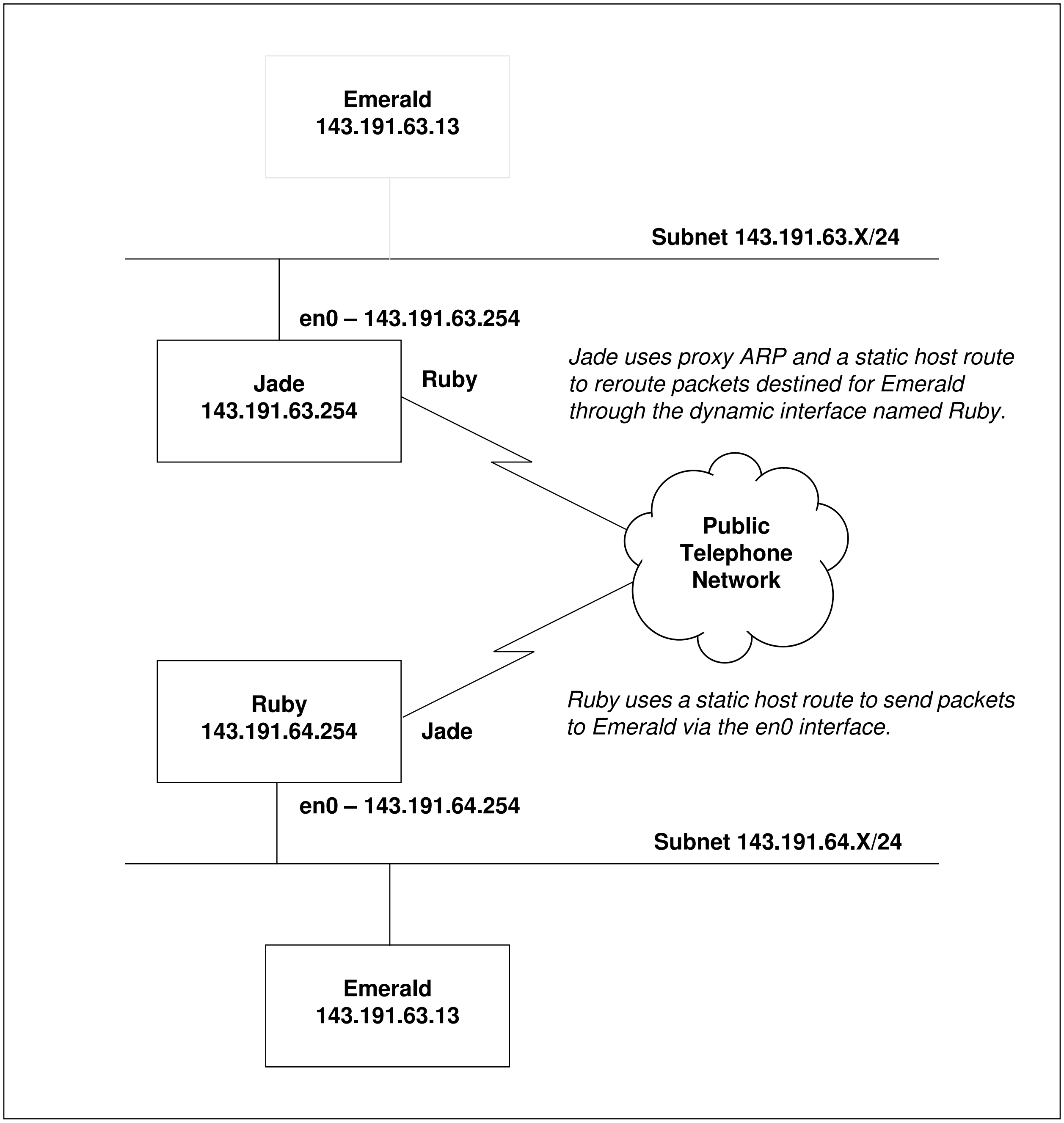


Figure ARP-1. Proxy ARP Example

To reroute packets destined for the host named Emerald to the remote site, you must add a proxy ARP and static route at the local NetBlazer, Jade.

```
Jade:Top> arp proxy 143.191.63.13 en0
Jade:Top> arp list
received 2 badtype 0 bogus addr 0 reqst in 2 replies 0 reqst out 10
IP addr         Type  Interface   Time Q Addr              RIF
143.191.63.255  ether any         0      ff:ff:ff:ff:ff:ff none
143.191.63.1    ether any         131    00:80:ad:02:61:c1 none
143.191.63.13   ether en0         0      00:80:ad:02:13:00 none (published)
Jade:Top> route add 143.191.63.13 Ruby
Jade:Top> route lookup 143.191.63.13
Dest            Len Interface    Gateway          Metric  P Timer  Use
143.191.63.13   32  Ruby                          1         0      0
```

An alternative method is to add an auto-arp entry to the dynamic interface named Ruby.

```
Jade:Top> dynamic
Jade:Top>Configure>Interface>Dynamic> auto-arp Ruby add 143.191.63.13
Jade:Top>Configure>Interface>Dynamic> list Ruby
Name                  IP address   MRU   MTU type    line      pkts in   pkts out
Ruby              143.191.63.254  1006  1006 Dialout Down            0          0
Idle: In 0:01:40:37   Out 0:01:40:37                IP:          0          0
Auto Arp's:143.191.63.13
TCP/IP header compression allowed
Dynamic SLIP interface, Uname=Jade, Password=jade
Dial-out=Yes, Dial-in=Yes
Chat script=ics, Crypto key=jade, Max-lines=1, MRU=1006, MTU=1006
Trigger level=2, Timeout=180 s, Queue-max=8
```

This creates a temporary proxy ARP entry and static route for Emerald when a dynamic packet mode connection is established from Ruby to Jade.

The final step, in either case, is to add a route to the remote NetBlazer, named Ruby, that sends packets to Emerald through Ruby’s en0 interface.

```
Jade:Top> telnet 143.191.64.254
Type ^] (decimal 29) <CR> to return to NetBlazer
Trying 143.191.64.254:23...
Telnet session 0 connected to 143.191.64.254


Telebit's NetBlazer Version 1.40

Ruby login: root
Password:
Ruby:Top> route add 143.191.63.13 en0
Ruby:Top> route lookup 143.191.63.13
Dest            Len Interface    Gateway          Metric  P Timer  Use
143.191.63.13   32  en0                           1         0      0
Ruby:Top> logout
Telnet session 0 closed: EOF
```

Configure Command

The **configure** command takes you to the Configure command level. To view a list of the commands available at this level, type a question mark (?) at the Configure command prompt.

```
NB:Top> configure
NB:Top>Configure> ?
Available commands:
accounting      alias>          arp>            asy>
debug>          destination>    dialout>        grep
group>          hostname        interface>      ip>
line>           message>        more            msglog
phonelist>      ppp>            save            security>
servers>        setconsole      slip>           snmp>
syslog>         syslogto        source          time
top>            up>             user>           ?
```

For a description of the above configuration commands, refer to the specific command heading. For example, to read about the **dialout** commands look under Dialout Commands in this manual.

All commands at or beneath the Configure command level are <u>only</u> available to users who have been given sysconf or sysstat privileges. User with sysconf can use any of the commands at or below this level to configure the NetBlazer; whereas users with sysstat privileges are restricted to displaying information about the current configuration of the NetBlazer, they cannot change the configuration.

Note: *The* **accounting** *and* **syslogto** *commands have been obsoleted by the* **syslog** *command level. To log user activity information to a disk or host system use the* **syslog** *commands.*

The ***asy*** *command functions are now provided at the* ***line*** *command level. Use the* ***line*** *commands to control the operation of the asynchronous ports.*

Caution: *The* ***debug*** *commands are provided primarily to aid Telebit in isolating problems in the NetBlazer. Do not use the* ***debug*** *commands without first contacting Telebit.*

Debug Command

The **debug** command takes you to the Configure Debug command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Debug command level.

```
NB:Top> debug
NB:Top>Configure>Debug> ?
Available commands:
alloc           args            autoconfigure   core
clear           devs            display         db
dh              dl              dump            files
filter>         gdt             idle            idt
inb             irq             hardware        hex
hogmemory       memory          misc            maplist
memset          noautoconfigure nocore          notnclient
notnserver      outb            printf          probe
process         profile         ps              psignal
remote          show            sleep           status
timers          tnclient        tnserver        top>
unmap           watch           writelong       writebyte
writeshort      up>             ?
```

Caution: *These commands, provided to help Telebit resolve software problems, are unsupported and likely to change without notice. Since many of these commands may have undesirable effects, you should not use commands at this level unless instructed to do so by Telebit's Technical Support department.*

debug alloc [<value>]

The **debug alloc** command sets the allocation debug level, which is only available if memory allocation debugging is compiled into the NetBlazer software. The <value> parameter is defined as follows:

1	Set and check a fence value on malloc and free
2	Set memory after free
4	Set memory after alloc
8	Set watch point on fence after data
16	Set watch point on fence

Note that settings 2 and 4 degrade performance.

If the <value> parameter is not entered, the current allocation debug options are displayed.

debug args

The **debug args** command displays the argument list.

debug autoconfigure

The **debug autoconfigure** command sends debug messages to the console from line autoconfigure function. The **debug noautoconfigure** command stops sending line autoconfigure messages.

debug clear

The **debug clear** command clears any watch points.

debug core

The **debug core** command turns on core dumps. The **debug nocore** command turns off core dumps.

debug db <address> [<count>]

The **debug db** command is the same as the **display** command, except that the memory is displayed as 8 bit hex quantities.

debug devs

The **debug devs** command lists logical names of devices that the NetBlazer software supports.

debug dh <address> [<count>]

The **debug dh** command is the same as the **display** command,except that the memory is displayed as 16 bit hex quantities.

debug display <address> [<count>]

The **debug display** command displays the specified memory address in 32 bit hex words and ASCII for the <count> values. Addresses that are invalid are rejected.

debug dl <address> [<count>]

The **debug dl** command is the same as the **display** command.

debug dump

The **debug dump** command forces a core dump and then reboots. This command is useful if the system is in a bad state and should be recorded for later analysis by Telebit Technical Support.

debug files

The **debug files** command displays a list of open files.

debug filter

The **debug filter** command takes you to the Configure Debug Filter command level. These commands, which work the same as the standard packet filtering commands, are used to filter the **ip trace** output for one or more interfaces. Refer to the **ip trace** and **filter** command descriptions for additional information.

debug gdt <start> [<end>]

The **debug gdt** command displays the global descriptor table entries ranging from <start> to <end>.

debug hardware

The **debug hardware** command provides information about the NetBlazer's hardware configuration.

debug hex <filename>

The **debug hex** command produces a hex dump listing of the specified file.

debug hogmemory <bytes>

The **debug hogmemory** command grabs the specified number of bytes of memory.

Caution: *This command only should be used in a test environment to force memory shortages for testing memory stress situations.*

debug idle

The **debug idle** command displays the cpu idle percentage (over the last second) and how many times timerproc was skipped.

debug idt <start> [<end>]

The **debug idt** command displays the interrupt descriptor table entries ranging from <start> to <end>.

debug inb <port> <count>

The **debug inb** command displays the contents of an I/O port (or ports if count is specified).

Caution: *This can be dangerous as some ports do unpredictable things if they are read. This can bring down the system or confuse the drivers.*

debug irq <value>

The **debug irq** command displays the status of the I/O interrupt specified by value (value must be between 0 and 15).

debug maplist <addr> <count>

The **debug maplist** command displays the memory management map for the given address and count (in pages).

debug memory <command>

The **debug memory** command invokes memory commands, which are now available as debug commands. For example, **debug alloc** and **memory alloc** are equivalent commands.

debug memset <addr> <char> <length>

The **debug memset** command sets the memory at <address> to the hex byte value specified by <char> for the number of bytes specified by the <length> parameter.

Caution: *This command should never be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

debug noautoconfigure

The **debug noautoconfigure** command stops sending line autoconfigure messages to the console.

debug nocore

The **debug nocore** command turns off core dumps.

debug outb <port> <value>

The **debug outb** command sends <value> to the I/O port specified by the <port> parameter.

debug printf <fmt> <value ...>

The **debug printf** command displays <value> in the specified format using the tprintf routine.

debug probe <device>

The **debug probe** command probes for the named device. If found, the device is reloaded or reinitialized. See the **debug dev** command.

debug process [<pid>]

The **debug process** command displays the information in the proc structure for the process specified by <pid> or for the current process if <pid> is not entered.

debug ps

The **debug ps** command invokes the **list ps** command. This command is here so that **ps** does not get interpreted as an abbreviated **psignal** command.

debug psignal <value> <count>

The **debug psignal** command wakes up the <count> processes waiting on <value>.

Caution: *This command should not be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

debug remote [off] [on] [file <filename>] [myip <myip>] [list] [host <ip>] [ether <hardware-addr>] [test]

The **debug remote** command controls sending core dumps to a remote host.

If you just turn remote dumping on, and do not supply the host, host's hardware address and filename, the bootp protocol is used to determine that information. If there is no bootp server, the bootp server is down, or the server is on the other side of a gateway then the network dump will fail, and a normal dump to floppy will be taken.

The actual dump is taken by means of tftp, which means that there must be a tftp server running on the machine on which dumps are to be stored. Note that tftp requires that the file exist, and be writable by everybody.

If you wish to take dumps over the network without using bootp, you must supply the necessary information (the ip address, the hardware address, and the filename) on the command. To send core dumps through a gateway you must supply the gateway's hardware address instead of the tftp server machine.

The **debug remote on|off** command turns the remote core dump function on or off.

The **debug remote host** <ip> **ether** <hardware-addr> command specifies the ip address or domain name, and the ethernet address of that host or the gateway to reach the host.

The **debug remote list** command displays the current remote core dump information.

The **debug remote myip** <ip> command specifies the ip address of this NetBlazer. Normally, this command is not required because the NetBlazer's global ip address is used.

The **debug remote debug** <value> command specifies the debugging message level to be used (defaults to 0).

The **debug remote test** command tests the remote core dump function. The NetBlazer is disabled until the test finishes.

debug show

The **debug show** command shows the internal console screen log, which keeps the last 4096 bytes that would have been written to that screen. This is particularly useful from remote locations and when there is no external console terminal.

debug sleep <delay>

The **debug sleep** command stops the processor for the number of seconds specified by the <delay> parameter. This is useful for setting a delay in scripts.

debug status [arguments]

The **debug status** command runs a status monitoring session on a terminal or the internal console screen. The status display currently must be run on a terminal that supports the ANSI escape sequences.

Issuing a **debug status start** command starts a standard status monitoring session, which displays the status of all items available on this NetBlazer, as illustrated in the following example.

```
host     Pearl                          time     08:38:33   date     01/13/92
        *---------------------------------------------------*  Value  % Units
cpu     |                                                   |    1    1%
ints    |CCCCC                                              |  109   10%
line00  |TTT                                                |  142    7% C/S
line01  |                                                   |    0    0% C/S
en0     |                                                   |    0    0% P/S
en1     |                                                   |    0    0% P/S
ibuf    |SSSSSSSSSSSSSSSSSSLLLLLLLLLL                       |  300   60% #
mbuf    |                                                   |    9    1% #
memfree |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                 |  992K  69%
fd      |                                                   |    0    0%
        *---------------------------------------------------*
```

By default, this display is updated each second and is refreshed (redrawn) after every tenth update.

To stop the status monitoring session, escape out of the session (e.g. enter ctrl-] <return>), then issue a **session close** or **debug status clear** command to close the session and reset the status monitor.

You can control how often the status display is updated and refreshed using the **delay** <value> and **refresh** <value> arguments. The delay value specifies the delay in seconds between updates, and the refresh value specifies the number of updates to display before refreshing (redrawing) the display. For example, a **debug status start delay 2 refresh 5** command starts a standard status monitoring session that is updated every two seconds and refreshed after every fifth update.

You can redirect the status monitor output to an internal video board (if present) by including the **internal** augment in the **debug status** command (e.g. **debug status start internal**).

The **debug status list** command displays the current settings of the status monitor.

You can build a custom status display using the following command syntax:

```
debug status <item> [line <line>] [col <column>] [notitle] [maximum <value>]
```

Each item is displayed in the order entered. You can set the position of an item using the **line** <line> and **col** <column> parameters.

The **notitle** option suppresses the title that is normally displayed.

The **maximum** <value> parameter specifies a value to override the default for this item. If no explicit maximum is specified then the appropriate default is used (these are fixed except for asynchronous lines which default to the maximum transmission rate).

Items that can be shown in the status display are defined in Table DEB-1.

Table DEB-1. Status Display Items

Item	Description
host	Shows the host name of the NetBlazer.
time	Shows the current time.
date	Shows the current date.
text <string>	Displays the text string. This string must be enclosed in quotes.
cpu	Shows the CPU time used.
ints	Shows the interrupts status.
memfree	Shows the current available memory.
ibuf	Shows the interrupt buffers.
mbuf	Shows the mbufs.
fd	Shows the floppy disk I/O.
asyXX asy <line-name>	Shows the transmit and recieve status of a specified line. To display all of the asynchronous lines use the asys command.
iface <interface> interface <interface>	Shows the packets transmitted for a specified interface. To display all of the interfaces use the nets command.
asys	Shows the status for all of the NetBlazer's asynchronous lines.
nets	Shows the status for all of the NetBlazer's network adapters.

The following screen shows the debug status commands used to build the standard screen.

```
NB:Top>Configure>Debug> status standard
config debug status host line 1
config debug status time line 1 col 40
config debug status date line 1 col 60
config debug status text "*----------------------------------------------------
*"
col 10
config debug status text "Value  % Units" col 64 line 2
config debug status cpu
config debug status ints
config debug status asys
configure debug status line00
configure debug status line01
config debug status nets
configure debug status interface en0
configure debug status interface en1
config debug status ibuf
config debug status mbuf
config debug status mem
config debug status fd
config debug status text "*----------------------------------------------------
*"
col 10
```

debug timers

The **debug timers** command displays the current pending timers.

debug unmap <address> <count>

The **debug unmap** command unmaps the specified address from the address space. *Don't even try it!* About the only address that can usefully be unmapped is page zero, which is already unmapped.

Caution: *This command should not be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

debug watch <addr> <type> [124] [WRE]

The **debug watch** command set a watch point at address of the given type. The type should consist of [erw][124] (access type, length).

Caution: *This command should not be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

debug writebyte <address> <value>

The **debug writebyte** command writes a byte value into the specified address.

Caution: *This command should not be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

debug writelong <address> <value>

The **debug writelong** command writes a long value into the specified address.

Caution: *This command should not be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

debug writeshort <address> <value>

The **debug writeshort** command writes a short value into the specified address.

Caution: *This command should not be used unless you are specifically instructed to do so by Telebit's Technical Support department.*

Destination Commands

The **destination** command takes you to the Configure Destination command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Destination command level.

```
NB:Top> destination
NB:Top>Configure>Destination> ?
Available commands:
add             gid             delete          host
list            longname        top>            type
up>             ?
```

This set of commands allows you to create and maintain destination groups for the users. Assigning the users to a destination group has the following advantages:

- Simplifies the user's interaction with the NetBlazer by allowing a user to select a telnet or dial-out destination from a list of destinations assigned to that user.
- Allows you to restrict the user's access to a limited group of telnet hosts and/or dial-out destinations.

The **user destgroup** command assigns a user to a defined destination group. To start a telnet or dial-out session with a destination contained in the user's assigned destination group, he or she issues a **session connect** command.

destination add [<dest-group>] [<short-name>]

The **destination add** command creates a new destination group or adds a new destination name to an existing destination group.

The <dest-group> parameter is the name of the group which is to be created when the <short-name> parameter is omitted. If the <short-name> variable is entered, the new destination name specified by <short-name> will be added to the designated destination group.

The <short-name> parameter is used in the **session connect** command to connect to the destination defined by this entry.

If you enter the **destination add** command without any parameters the NetBlazer will begin a dialog. This dialog collects the information normally acquired by the **destination add**, **host**, **longname**, and **type** commands. This is done to make configuration easier and helps prevent mistakes caused by forgetting to enter a necessary command.

First, you are prompted for a destination group name. If the group name does not exist, the NetBlazer verifies that you want to create a new destination group. If the group name does exist, the NetBlazer displays a list of information that it currently has on the designated destination group.

The NetBlazer then prompts for the short name to be added to the destination group. Enter the destination name which is to be used in a **session connect** command.

The NetBlazer then requests the destination's long name. Enter a descriptive name for the machine which will be accessed through this destination.

Next, the NetBlazer prompts for the destination type which must be either "dial" or "telnet." If a dial-out destination type is specified, the NetBlazer prompts for a dial-out destination name which must be created by a **dialout add** command before this destination can be used. If a telnet destination type is specified, the NetBlazer requests a telnet hostname which may be either a domain name or IP address.

When all the information for this destination is supplied, the NetBlazer again prompts you for a new destination short name. At this point you can either enter a destination as described above, or press the return key to quit the dialog.

destination delete <dest-group> [<dest-number>]

The **destination delete** command deletes an entire destination group when only the <dest-group> parameter is entered. If both parameters are entered, this command removes a destination entry from the specified group. The <dest-group> variable identifies the group to modify and the <dest-number> parameter identifies the particular entry in the group to delete.

destination host <dest-group> <dest-number> <dialout>|<telnet-name>

The **destination host** command is used to add or change the dial-out or telnet destination information of a destination group entry.

The first two parameters, <dest-group> and <dest-number>, identify which particular entry within a specified group to assign a dial-out destination or telnet host ID.

The last parameter, <dialout> | <telnet-name>, identifies the dial-out destination or telnet host ID, depending on if the destination entry is a dial or telnet type destination. Note that the telnet host ID may be either a domain name or IP address.

destination list [<dest-group>]

The **destination list** command displays a list of the available destination groups, or displays a list of the destinations in a particular group, depending on whether or not the <dest-group> parameter is entered.

destination longname <dest-group> <dest-number> <long-name-string>

The **destination longname** command is used to assign a more descriptive name to a particular destination entry.

The first two parameters, <dest-group> and <dest-number>, identify which particular entry within a specified group to assign a new description.

The <long-name-string> parameter is the descriptive text to associate with the designated destination entry.

destination type <dest-group> <dest-number> dial | telnet

The **destination type** command specifies the type of destination, i.e. dial or telnet, associated with a particular destination group entry.

The first two parameters, <dest-group> and <dest-number>, identify which particular entry within a specified group to modify.

The last parameter must be either “dial” or “telnet”. Note that the host assigned to the designated entry must be appropriate for the type selected.

Dialout Commands

The **dialout** command takes you to the Configure Dialout command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Dialout command level.

```
NB:Top> dialout
NB:Top>Configure>Dialout> ?
Available commands:
add             characteristics delete          ditto
list            maxwait         minwait         phone
timeout         top>            up>             ?
```

Commands at this level are used to maintain a list of dial-out destinations which are used to establish a connection with a remote site in either packet or character mode.

Each dial-out destination can include multiple dial-out entries. If a given dial-out destination name includes more than one entry (phone number), the NetBlazer chooses the least recently used phone number to place a call. This prevents the NetBlazer from always trying to dial the first number, when it already has been used and is busy.

dialout add [<dialout>] [<phone-number>] [<characteristics>] [<timeout>]

The **dialout add** command is used to create a new entry in the dial-out list.

The <dialout> parameter is a single word name, which identifies the dial-out destination. If the dial-out destination is another NetBlazer with more than one modem connected, you should enter the same destination name for each telephone line connected to the remote NetBlazer.

The <phone-number> parameter is the telephone number of the remote destination.

The <characteristics> parameter specifies the characteristics group assigned to the dial-out destination record. The chosen characteristics group must be previously defined via the **group** commands.

The <timeout> parameter determines the number of minutes that an inactive connection will be maintained. When a line is inactive for the number of minutes specified by the <timeout> value, the NetBlazer commands the modem to drop the telephone connection.

If this command is entered without any parameters, the NetBlazer prompts you for the necessary information as illustrated below.

```
Top>Configure>Dialout> add
Name of dialout: branch
Phone number: 555-1234
Line characteristics (dialout):
Char mode timeout (in minutes) (60):
Dialout: name=branch, phone=555-1234, characteristics=dialout
Okay (yes|no|quit) [y]?
```

dialout characteristics <dialout> [#<item-number>] <group>

The **dialout characteristics** command sets or changes the characteristics group assigned to a dial-out destination.

The <dialout> and optional #<item-number> parameters identify the dial-out destination to assign the designated characteristics group.

The <group> parameter identifies the characteristics group assigned to the selected dial-out destination. The chosen group name must be previously defined via the group commands. If no <group> is entered, the characteristics group for this dial-out destination is set to <none>.

The NetBlazer comes with six standard dial-out characteristic groups: dialout, v32dialout, pepdialout, request, raw, and raw_dial. The dialout or v32dialout groups should be used for most destinations. If you encounter problems with connections failing because the modems unexpectedly hang up and you have PEP-compatible modems at both sites, try the pepdialout group.

The request characteristics group is automatically assigned to a line that is designated as a callback request line.

The raw and raw_dial characteristics groups are used to configure a line for a raw connection. Raw only allows raw connections on the line. Raw_dial allows both raw connections and dialout connections if a modem is attached to the line.

dialout delete <dialout> [#<item-number>]

The **dialout delete** command deletes the dial-out destination specified by the <dialout> and #<item-number> variables. The #<item-number> parameter is only required when more than one entry share the same dial-out destination name.

Note: *Deleting a dynamic interface record removes dialout and user records with the same name as the interface.*

dialout ditto <dialout> [#<item-number>] <phone-number>

The **dialout ditto** command is used to copy a dial-out destination record, and assign a new telephone number to the duplicate record. This provides a convenient way to create the dial-out records for a destination site with multiple telephone lines.

The <dialout> and optional #<item-number> parameters select the dial-out record to copy.

The <phone-number> parameter is the telephone number assigned to the duplicate record.

dialout list [<dialout>] [#<item-number>]

The **dialout list** command either displays a list of the currently defined dial-out destinations, or if <dialout> and #<item-number> are specified displays the settings of the dial-out destination specified by the optional <dialout> and #<item-number> parameters.

dialout maxwait <dialout> [#<item-number>] <seconds>

The **dialout maxwait** command sets the maximum number of seconds that a dial-out record waits between calls. When a dialout record is created the maximum wait time is set to a default value of 0, which disables the wait time function.

The <dialout> and optional #<item-number> parameters identify the dial-out record to modify.

The <seconds> parameter specifies the maximum number of seconds that a dial-out record must wait before it can be used to place another call.

dialout minwait <dialout> [#<item-number>] <seconds>

The **dialout minwait** command sets the minimum number of seconds that a dial-out record waits between calls.

The <dialout> and optional #<item-number> parameters identify the dial-out record to modify.

The <seconds> parameter specifies the minimum number of seconds that a dial-out record must wait before it can be used to place another call.

dialout phone <dialout> [#<item-number>] <phone-number>

The **dialout phone** command is used to change the telephone number of a dial-out destination.

The <dialout> and optional #<item-number> parameters identify the dial-out record to modify.

The <phone-number> parameter is the new telephone number assigned to the selected dial-out destination.

dialout timeout <dialout> [#<item-number>] <timeout-minutes>

The **dialout timeout** command sets the amount of time in minutes that a NetBlazer will maintain an inactive dial-out connection to the destination associated with a designated interface.

The <dialout> and optional #<item-number> parameters identify the dial-out record to modify.

The <timeout-minutes> parameter determines the number of minutes that an inactive connection will be maintained. When a line is inactive for the number of minutes specified by the <timeout-minutes> value, the NetBlazer commands the modem to drop the telephone connection.

Disk Commands

The **disk** command takes you to the Disk command level. To view a list of the commands available at this level, type a question mark (?) at the Disk command level.

```
NB:Top> disk
NB:Top>Disk> ?
Available commands:
chmod          copy           delete         dir
diskcopy       format         rename         top>
up>            ?
```

The NetBlazer uses the following files, which are contained on the system diskette:

TELEBIT.SYS contains the operating system used to load and execute the NetBlazer's executable code.In order for the NetBlazer to boot correctly, this file must be the first file on the diskette.

FRED is the NetBlazer's executable code that runs under TELEBIT.SYS.

MODEMCAP.TXT defines the modem capabilities for modems that are supported by the NetBlazer. This information is used in conjunction with the dialout and line characteristics to automatically configure the modems.

CHAT.TXT controls the log-in and authentication interchange that takes place between the NetBlazer and a remote device (usually another NetBlazer) when establishing a dynamic packet mode connection.

STARTUP.CNF contains configuration commands that are loaded when the NetBlazer is powered up or rebooted. Issuing a **configure save** command updates this file with the current configuration information.

DEFAULT.CNF contains the default configuration commands set by Telebit. You can restore the NetBlazer to its default configuration by copying this file to the STARTUP.CNF file and rebooting the system.

PASSWD contains login IDs, passwords and permissions for authorized NetBlazer users. The record format of this file is similar to the password file format used by UNIX systems. This allows you to FTP one or more UNIX password files to the NetBlazer and use that information to create or update the NetBlazer's list of authorized users. The PASSWD file is updated by entering either a **user save** or **configure save** command.

In addition to the standard NetBlazer files, defined above, you also can put-command files on your diskette. These files contain a series of NetBlazer commands that are run in sequence by the **source** <command-file> command.

To use a command file, you must first create the file using a text editor on an external system, and then copy the file to the NetBlazer system diskette.

Although the NetBlazer has its own operating system, the NetBlazer system diskette file format is compatible with MS-DOS. Therefore as an alternative to using the NetBlazer's **disk** commands, you can use an MS-DOS system that supports 3.5 inch high-density diskettes to perform the following disk maintenance operations:

- Format a diskette
- Copy the NetBlazer system diskette
- Copy, rename or delete files
- Edit the STARTUP.CNF, MODEMCAP.TXT, CHAT.TXT and PASSWD files, or create command files using a text editor.

If you do not have access to an MS-DOS system, you can use FTP to copy a NetBlazer file to other system, which has a text editor, edit the file, and then use FTP to copy the file back to the NetBlazer diskette.

disk chmod <mode|+w|-w> <file>

The **disk chmod** command write protects a file on the system diskette by entering a **chmod -w** <file> or **chmod 1** <file> command where <file> identifies the file to be write-protected. Entering a **chmod +w** <file> or **chmod 2** <file> command turns write-protection off for the specified file.

disk copy <source-file> <destination-file>

The **disk copy** command copies the file specified by the <source-file> parameter to a file specified by the <destination-file> parameter.

disk delete <filename>

The **disk delete** command deletes the file specified by the <file-name> parameter from the diskette.

disk dir [<filename.ext>]

The **disk dir** command displays a list of files on the NetBlazer system diskette.

You can display all files on the diskette, a single file referenced by name, or a selected group of files using the * wildcard character to represent an arbitrary string of characters. Several examples are shown below.

```
NB:Top>Disk> dir
Directory of A:/

TELEBIT  SYS      14844  07-22-91   15:08
FRED            603648  02-25-92   16:25
MODEMCAP TXT      17095  02-25-92   16:26
CHAT     TXT       1588  02-25-92   16:26
DEFAULT  CNF       1265  02-25-92   16:26
STARTUP  CNF       3670  03-04-92   11:32
PASSWD               32  03-04-92   11:32
UPDATE   CNF        177  01-24-92   18:56
STARTUP  SAV       3670  03-08-92   10:09
9 File(s) (645989 bytes) 808448 bytes free
NB:Top>Disk> dir st*
Directory of A:/

STARTUP  CNF       3670  03-04-92   11:32
STARTUP  SAV       3670  03-08-92   10:09
2 File(s) (7340 bytes) 808448 bytes free
NB:Top>Disk> dir *.cnf
Directory of A:/

DEFAULT  CNF       1265  02-25-92   16:26
STARTUP  CNF       3670  03-04-92   11:32
UPDATE   CNF        177  01-24-92   18:56
3 File(s) (5112 bytes) 808448 bytes free
```

disk diskcopy

The **disk diskcopy** command makes a duplicate copy of the NetBlazer system diskette. The destination diskette must be a formatted 3.5 inch high-density diskette. Any files on the destination diskette are overwritten.

The **diskcopy** routine tells you when to insert the appropriate diskette as shown in the following example.

```
NB:Top>Disk> diskcopy
0% done...Insert source diskette and press return
Insert destination diskette and press return

47% done...Insert source diskette and press return
Insert destination diskette and press return

94% done...Insert source diskette and press return
Insert destination diskette and press return

Disk copy complete
Insert system diskette and press return
```

Caution: *The **diskcopy** routine does not flush the typeahead buffer – **do not press the return key until you are instructed to do so**.*

disk format

The **disk format** command initializes a blank 3.5 inch high-density diskette for use with the NetBlazer.

disk rename <oldfile> <newfile>

The **disk rename** command renames the file specified by the <oldfile> parameter to the filename specified by the <newfile> parameter.

Domain Commands

The **domain** command takes you to the Configure IP Domain command level. To view a list of the commands available at this level, type a question mark (?) at the Configure IP Domain command level.

```
NB:Top> domain
NB:Top>Configure>IP>Domain> ?
Available commands:
add             cache           delete          list
suffix          top>            up>             ?
```

These commands control the domain name service client. If your network includes one or more domain name servers, you can configure the NetBlazer to query the server(s) for the IP address of a specified host.

domain add <servername>

The **domain add** command adds a domain name server to the list of name servers which are queried by the NetBlazer when it needs to resolve a name to an IP address.

The <servername> parameter is the IP address of the name server which is to be added to the list. If you have another name server running, you can enter the domain name of the new name server to be added to the list.

domain cache

The **domain cache** command displays a list of names and IP address which have been cached locally. If a name has been resolved by a remote name server, its associated address is cached locally. This reduces the load on the network because the local cache is searched for an address resolution before the remote name server is queried.

domain delete <servername>

The **domain delete** command removes a domain name server from the list of name servers which are queried by the NetBlazer when it needs to resolve a name to an IP address.

The <servername> parameter is the IP address of the name server which is to be removed from the list.

domain list

The **domain list** command displays the list of IP addresses and status of the name servers that are queried by the NetBlazer for name resolutions.

domain suffix [<suffix-string>]

The **domain suffix** command specifies the <suffix-string> which is to be added to a name that does not include a suffix, i.e. any name that does not contain a dot (period). For example, Telebit's domain suffix, telebit.com, is set by issuing the following command to the NetBlazer:

```
NB:Top>Configure>IP>Domain> suffix telebit.com
```

When a domain suffix has been defined, names are expanded to include the domain suffix. In the example, references to a machine named robin would be expanded to robin.telebit.com.

If this command is entered with no parameter, the current domain suffix is displayed.

```
NB:Top>Configure>IP>Domain> suffix
telebit.com
```

To remove a suffix use a null string in the **suffix** command.

```
NB:Top>Configure>IP>Domain> suffix ""
```

Dynamic Commands

The **dynamic** command takes you to the Configure Interface Dynamic command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Interface Dynamic command level.

```
NB:Top> dynamic
NB:Top>Configure>Interface>Dynamic> ?
Available commands:
add              auto-arp         auto-route       chat
crypto           delete           framing          list
max-lines        password         queue-max        timeout
top>             trigger          uname            up>
?
```

Commands at this level are used to set up and maintain dynamic interfaces, which are used to route IP packets over the public telephone network.

Note: *The preferred method for creating a dynamic interface to a remote site is through the **ipdial** command. Using this command ensures that all of the information needed to set up a dynamic interface is provided.*

*If you decide to use the **dynamic add** command instead of **ipdial** to create a new dynamic interface, you must create a dialout and/or user record with the same name as the interface to allow the NetBlazer to establish outbound or inbound connections through the interface. You also must add one or more routes to the routing table to send IP packets destined for the remote site through the interface.*

dynamic add [<interface>] [<uname>] [<password>] [<crypto>] [<chat>] [slip|ppp] [<max-lines>] [<mru>]

The **dynamic add** command creates a new dynamic interface for a dial-up connection to a remote site. Each parameter in this command must be entered in sequence. If you omit any of the parameters, you are prompted for the remaining parameters.

The <uname> and <password> parameters are the user ID and password which will be sent to the remote destination when a dial-out connection is established. The remote NetBlazer must have a user ID and password in its user records that match the uname and password parameters of this interface.

The <crypto-key > parameter is the crypto key which is used to encrypt a crypto response when a crypto challenge is received from the remote site. When crypto handshaking is used, the remote NetBlazer must have a user record that matches the uname and crypto-key of this interface.

The <chat> parameter specifies which chat script in the CHAT.TXT file to use when attempting to establish a dial-out connection with the remote site. The ics chat script should be used when the interface is used to dial out to another NetBlazer at the remote site.

The slip|ppp parameter specifies which framing method to use when sending IP datagrams through this interface.

The <max-lines> determines the maximum number of connections which will be allowed through this interface.

The <mru> (maximum receive unit) parameter sets the number of bytes in the largest packet that may be received or transmitted through this interface.

dynamic auto-arp <interface> <add|delete> <ip_addr ...>

The **dynamic auto-arp** command configure the NetBlazer to automatically create proxy ARPs and related routes for one or more host IP addresses when a connection is established through a specified dynamic interface. This accommodates users who temporarily need to move their system from the local network to a remote site.

See the description of the **arp proxy** command for an example of how the **dynamic auto-arp** command is used.

The **dynamic auto-arp** <interface> **add** <ip_addr ...> command adds host IP addresses to the list of addresses that the NetBlazer creates proxy ARPs and related routes for when the specified dynamic interface is connected. The auto-ARP entries, shown as “published,auto” in the ARP list, and related routes, marked by an A in the P field of the route list, only remain while the connection is present. When the connection is dropped, the auto-ARP entries and routes are removed.

The **dynamic auto-arp** <interface> **delete** <ip_addr ...> command removes host IP addresses from the list of addresses that the NetBlazer creates proxy ARPs and related routes for when a dynamic connection is made.

dynamic auto-route <interface> <add|delete> <ip_addr[/bits] ...>

The **dynamic auto-route** command configures the NetBlazer to automatically add one or more host or network IP addresses to the routing table when a connection is established through a specified dynamic interface. When the connection is dropped, the dynamic interface’s routes, marked by the letter A in the P field of the route list, are removed. This helps to reduce the number of entries in the routing table by only listing routes through a dynamic interface when it is in use.

The **dynamic auto-route** <interface> **add** <ip_addr[/bits] ...> command adds IP addresses to the list of auto-route addresses for a specified interface.

The **dynamic auto-route** <interface> **delete** <ip_addr[/bits] ...> command removes IP addresses from the auto-route address list for the specified interface.

To illustrate how the **auto-route** command is used, assume that you have a remote network that you only want to be accessible when a connection has been established from the remote site. See Figure DYN-1.

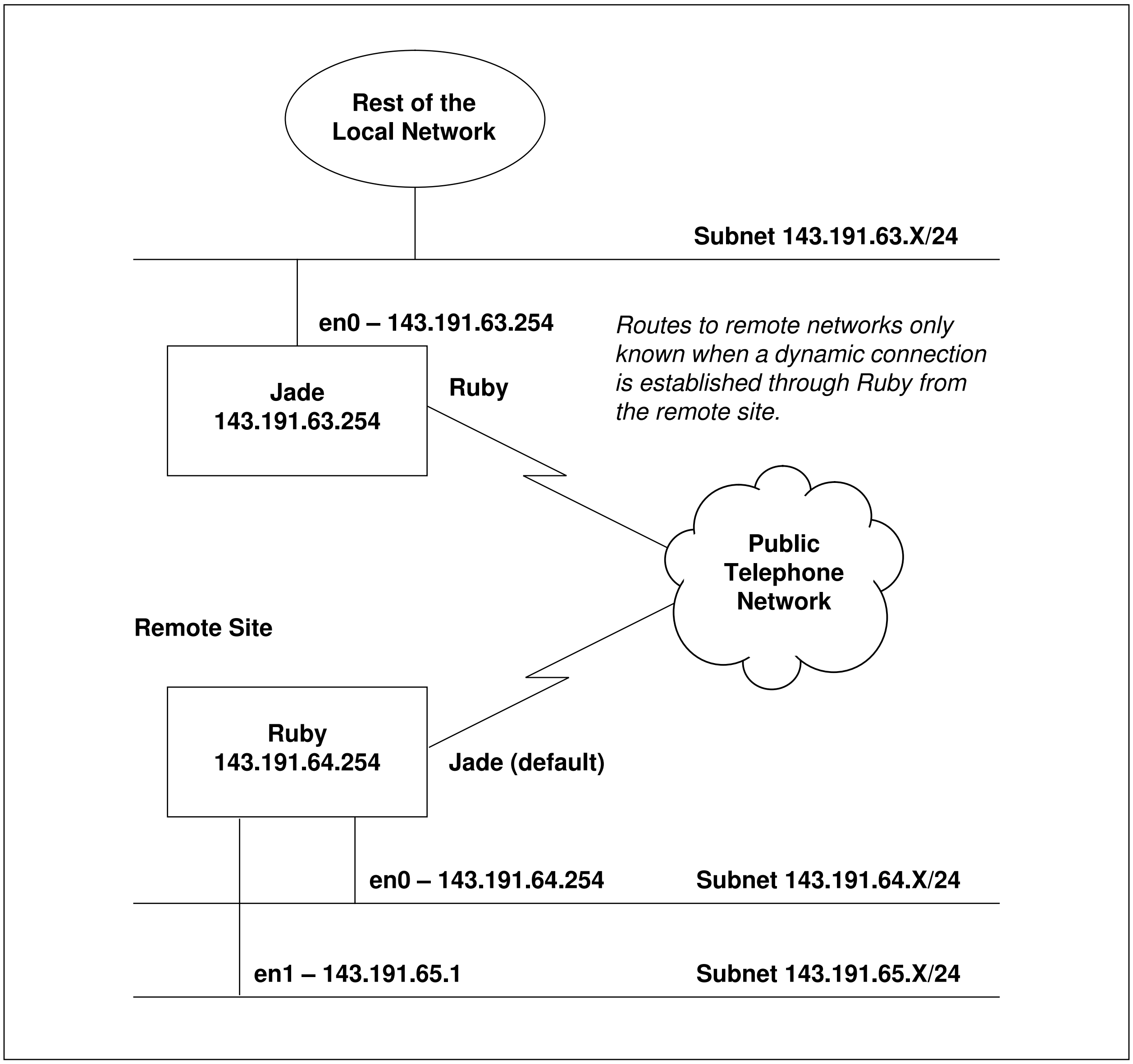


Figure DYN-1. Auto Route Example

First, use **ipdial** to create dynamic interfaces for Ruby and Jade. *Do not enter any static routes through Ruby on the local NetBlazer named Jade.*

Next, go to the Dynamic command level on Jade, and added auto routes for the networks at the remote site to the dynamic interface named Ruby.

```
Jade:Top> dynamic
Jade:Top>Configure>Interface>Dynamic> auto-route Ruby add 143.191.64/24
Jade:Top>Configure>Interface>Dynamic> auto-route Ruby add 143.191.65/24
Jade:Top>Configure>Interface>Dynamic> list Ruby
Name               IP address   MRU   MTU type    line      pkts in   pkts out
Ruby           143.191.63.254  1006  1006 Dialout Down         186        186
Idle: In 1:18:03:10   Out 1:18:03:10                  IP:        186        245
Auto IP's :143.191.64.0/24     143.191.65.0/24
TCP/IP header compression allowed (currently off)
Dynamic SLIP interface, Uname=Jade, Password=jade
Dial-out=No, Dial-in=Yes
Chat script=ics, Crypto key=jade, Max-lines=1, MRU=1006, MTU=1006
Trigger level=2, Timeout=180 s, Queue-max=8
```

At this point, the local NetBlazer named Jade should not have any routes to the remote site through the interfaced named Ruby. You can confirm this by entering the following **route list** command.

```
Jade:Top>Configure>Interface>Dynamic> route list |grep Ruby
Jade:Top>Configure>Interface>Dynamic>
```

When a dynamic interface connection is initiated from the remote site, the local NetBlazer (Jade) automatically adds temporary routes to the remote networks to its routing table as shown below.

```
Jade:Top>Configure>Interface>Dynamic> route list |grep Ruby
143.191.65.0    24  Ruby                          1       A 0      0
143.191.64.0    24  Ruby                          1       A 0      0
```

If Jade is advertising its routes using RIP, these routes to the remote site are learned by the rest of the local network through RIP.

When the interface named Ruby is allowed to timeout and disconnects, Jade discards the temporary routes, and the remote site is no longer accessible to the local network.

dynamic chat <interface> <chat-script-name>

The **dynamic chat** command sets or changes the chat script assigned to a dynamic interface.

The <chat-script-name> parameter specifies which chat script in the CHAT.-TXT file to use when attempting to establish a dial-out connection with the remote site. The ics chat script should be used when the interface is used to dial out to another NetBlazer at the remote site.

dynamic crypto <interface> [<crypto-key>]

The **dynamic crypto** command sets or changes the crypto key assigned to a dynamic interface.

The <crypto-key > parameter is the crypto key which is used to encrypt a crypto response when a crypto challenge is received from the remote site. When crypto handshaking is used, the remote NetBlazer must have a user record that matches the uname and crypto-key parameters assigned to this interface.

If no <crypto-key> is entered, the crypto key for this interface is removed.

dynamic delete <interface>

The **dynamic delete** command removes a specified dynamic interface from the interface list.

Note: *Any user, dialout destination, and/or routes with the same name as the interface are also deleted.*

dynamic framing <interface> slip|ppp

The **dynamic framing** command selects SLIP or PPP framing protocol for packets transferred through a specified dynamic interface.

dynamic list [<interface>]

The **dynamic list** command displays status information about the interfaces, which are currently used by the NetBlazer.

If the optional <interface> parameter is entered, additional information for the named interface is displayed. This information varies depending on the type of interface and its present status.

dynamic max-lines <interface> <max-lines>

The **dynamic max-lines** command sets the maximum number of lines that the NetBlazer may use to connect to a remote NetBlazer through a designated interface. This allows the NetBlazer to use more than one line to share the load thereby resulting in greater throughput.

When the queue depth exceeds the number of packets specified in the **dynamic trigger** command, the NetBlazer activates an additional line. This process is repeated whenever the queue depth exceeds the trigger level until the number of lines specified in <max-lines> are activated.

Setting the value of <max-lines> to 0 prevents the NetBlazer from calling the destination assigned to this interface.

dynamic password <interface> <remote-password>

The **dynamic password** command sets or changes the password that is sent to the remote system while attempting to establish a dial-out connection via the specified interface. The remote NetBlazer must have a user ID and password in its user records that match the uname and password parameters of the interface.

dynamic queue-max <interface> <queue-length>

The **dynamic queue-max** command sets the maximum number of packets that may be queued for a designated interface.

Once the queue reaches the value specified by the <queue-length> parameter, additional packets are discarded.

dynamic timeout <interface> <timeout-seconds>

The **dynamic timeout** command sets the amount of time in seconds that a NetBlazer will maintain an inactive dial-out connection to the destination associated with a designated interface.

The <timeout-secs> parameter determines the number of seconds that an inactive connection will be maintained. The NetBlazer starts a timer on each line it has dialed out for packet communication. This timer is reset every time the NetBlazer transmits a packet. When a line has been inactive (no packets have been transmitted) for the number of seconds specified by the <timeout-secs> value, the NetBlazer commands the modem to hang up the telephone.

dynamic trigger <interface> <trigger-depth>

The **dynamic trigger** command sets the trigger level for activating an additional telephone connection for a designated interface.

The <trigger-depth> parameter specifies the number of packets that can be queued before an additional connection to the remote site is activated. When the number of packets on the queue exceeds the <trigger-depth> value and the number of lines currently open for the selected destination is less than the value set with the **dynamic max-lines** command, the NetBlazer will select and dial an additional line.

dynamic uname <interface> <remote-uname>

The **dynamic uname** command sets or changes the user name that is sent to the remote system while attempting to establish a dial-out connection via the specified interface. The remote NetBlazer must have a user ID and password in its user records that match the uname and password parameters of the interface.

Echo Command

The **echo** [-n] <message> command prints a message to the user's screen. If you do not include the optional -n parameter, the message is followed by a new line. On the other hand if you include the -n parameter, the new line is not sent. A couple of examples are shown below.

```
Pearl:Top> echo Hello NetBlazer users
Hello NetBlazer users
Pearl:Top> echo –n Hello NetBlazer users
Hello NetBlazer usersPearl:Top>
```

Echo commands are useful in source files to get give users feedback as to what is happening when they run the **source** <filename> command.

Another use for the **echo** command is to restrict a user's access to the NetBlazer by using **echo Contact your System Administrator to regain access** as a login_user alias.

Filter Commands

The **filter** command takes you to the Configure IP Filter command level. To view a list of the commands available at this level, type a question mark (?) at the Configure IP Filter command level.

```
NB:Top> filter
NB:Top>Configure>IP>Filter> ?
Available commands:
add              delete           deny             flush
list             lookup           notify           permit
top>             up>              ?
```

Commands at this level are used to maintain a list of IP packet filters that are used to restrict IP packet routing through a particular interface. Before attempting to set up IP packet filtering, first make sure that the NetBlazer routing table is set up correctly.

filter add <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto> [port]] [source|destination] [in|out] permit|deny

The **filter add** command is used to add one or more IP packet permit or deny filters for a designated interface to the list of IP packet filters. For your convenience the **filter permit** or **filter deny** commands have also been provided to allow you to add permit or deny filters to the list of filters.

The following display shows several examples of the **filter add**, **permit** and **deny** commands.

```
NB:Top> filter
NB:Top>Configure>IP>Filter> add 143.191/16 en0 permit
NB:Top>Configure>IP>Filter> permit 143.191/16 syn0 source out
NB:Top>Configure>IP>Filter> permit 143.191/16 syn0 dest in
NB:Top>Configure>IP>Filter> deny 143.191.13/24 syn0 tcp <1024 !=23 !=25 so out
NB:Top>Configure>IP>Filter> deny 143.191.13/24 syn0 tcp <1024 !=25 dest in
NB:Top>Configure>IP>Filter> permit 128.10.2.1 143.191.13/24 syn0 in
NB:Top>Configure>IP>Filter> list
Source              Destination         Interface Protocol I/O  Permit/Deny
128.10.2.1/32       143.191.13.0/24     syn0               In   Permit
                    143.191.13.0/24     syn0      TCP      In   Deny
                                          port rules: !=25 <1024
                    143.191.0.0/16      syn0               In   Permit
143.191.13.0/24                         syn0      TCP      Out  Deny
                                          port rules: !=23 !=25 <1024
143.191.0.0/16                          syn0               Out  Permit
Default                                 syn0                    Deny
143.191.0.0/16                          en0                In   Permit
                    143.191.0.0/16      en0                In   Permit
143.191.0.0/16                          en0                Out  Permit
                    143.191.0.0/16      en0                Out  Permit
Default                                 en0                     Deny
```

Notice that input and/or output filtering for a specified interface can be performed on a source address, a destination address, or a source-destination address pair. You can further restrict the filter to only apply to a specified protocol and port number or range.

The permit|deny parameter determines if the packet filter is a permit or deny filter. A permit filter allows IP packets which match the filter criteria to pass through the designated interface; whereas IP packets which match a deny filter's criteria are discarded and an ICMP Destination Unreachable message is sent back to the source. Packets which do not meet the criteria of a filter will be passed on to the next filter in the list.

The <interface> parameter specifies the name of the interface to which the filter applies. Packet filtering is done on a per interface basis. If no filters exist for a particular interface, filtering for that interface is considered off, and all packets routed through that interface will be permitted. When the first filter for an interface is added to the list of filters, packet filtering is turned on for that interface, and all packets routed into or out of the interface are checked for permission to continue.

Note: *If an interface is assigned filters in only one direction (i.e. all inbound or all outbound filters), the default filter for that interface only applies to packets traveling in the same direction. Packets going in the opposite direction are automatically permitted.*

The <address>[/<bits>] parameters specify the address or address pair which will be trapped by the filter. Entering the keyword **default** in place of the address parameters allows you to change the permission setting of the default filter which is automatically created for a designated interface.

When a pair of addresses are entered, the first address identifies the source network (or host) address, and the second address specifies the destination address. If only one address is entered, the source|destination parameter determines whether this address is to be treated as a source or destination address.

Note: *The source|destination parameter should not be used when creating a filter for a source and destination address pair.*

If the source|destination parameter is not specified for a single address, the NetBlazer will create both source and destination filters for that address.

The optional in|out parameter determines if the specified filter will filter IP packets entering or leaving the NetBlazer through the designated interface. If this parameter is not specified, filters for both inbound and outbound packets will be created.

The optional <proto> and <port> parameters specify the criteria for matching an IP packet's protocol and destination port number. If the <port> parameter is entered, the <proto> parameter must also be specified.

The <proto> parameter is any of the following keywords: **tcp**, **udp** or **icmp**.

Note: *The <port> parameter should not be used with the* ***icmp*** *protocol.*

The <port> parameter can be just the port number, or a range of port numbers as defined below:

$=P_n$	Any port number equal to P_n. Port numbers entered without a modifier have an implied equal to characteristic.
\!P_n	Excludes any port number equal to P_n. If you enter a exclude command (!) from the keyboard with the **history** mechanism turned on, you must escape the ! by preceding it with a \ (e.g. **filter add 143.191.10.0/24 en0 \!25 permit**). If you enter the command from a file or with history turned off, you do not need the escape character (\) before the exclude character (!).
$!=P_n$	Excludes any port number equal to P_n. This is same as the previous command except that the exclude character (!) does not need to be escape when it is followed by an equal sign (=).
$>P_n$	Any port number greater than P_n.
$>=P_n$	Any port number greater than or equal to P_n.
$<P_n$	Any port number less than P_n.
$<=P_n$	Any port number less than or equal to P_n.
$P_1<P_2$	Any port number greater than P_1 and less than P_2.
P_1-P_2	Any port number equal to or greater than P_1 and less than or equal to P_2. This is the same as P_1<= <=P_2.
P_1<= <=P_2	Any port number equal to or greater than P_1 and less than or equal to P_2. A space is required between the beginning and ending range parameters.
P_1< <=P_2	Any port number greater than P_1 and less than or equal to P_2. The space is required.
P_1<= <P_2	Any port number equal to or greater than P_1 and less than P_2. The space is required.

To specify port numbers 20, 21, 23, and 25, you could enter any of the following port number range specifications.

20<= <=25 !=22 !=24
19<26 !=22 !=24
19< <=25 !=22 !=24
20<= <26 !=22 !=24

Refer to RFC 1010 for a list of well-known port numbers used by the TCP and UDP protocols. Services available on UNIX systems can often be found in the /etc/services file.

filter delete <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto>] [source|destination] [in|out]

The **filter delete** command removes one or more filters from the list of filters depending on how specific you are when entering the **filter delete** command parameters. The more specific the command the fewer number of filters will be removed from the list.

To illustrate how the **filter delete** command is used, assume that the filter list includes the following filters for the en0 interface.

```
NB:Top>Configure>IP>Filter> list en0
Source              Destination         Interface Protocol I/O  Permit/Deny
143.191.12.0/24                         en0                In   Permit
143.191.13.0/24                         en0                In   Permit
                    143.191.12.0/24     en0                In   Permit
                    143.191.13.0/24     en0       TCP      In   Permit
                                          port rules: =25 >1023
                    143.191.13.0/24     en0                In   Permit
143.191.12.0/24                         en0                Out  Permit
143.191.13.0/24                         en0       TCP      Out  Permit
                                          port rules:  20<= <=25  >1023
143.191.13.0/24                         en0                Out  Permit
                    143.191.12.0/24     en0                Out  Permit
                    143.191.13.0/24     en0                Out  Permit
Default                                 en0                     Deny
```

Entering a **filter delete 143.191.12/24 en0** command removes the four filters with a source or destination address of 143.191.12.0/24.

```
NB:Top>Configure>IP>Filter> delete 143.191.12/24 en0
NB:Top>Configure>IP>Filter> list en0
Source              Destination         Interface Protocol I/O  Permit/Deny
143.191.13.0/24                         en0                In   Permit
                    143.191.13.0/24     en0       TCP      In   Permit
                                          port rules: =25 >1023
                    143.191.13.0/24     en0                In   Permit
143.191.13.0/24                         en0       TCP      Out  Permit
                                          port rules:  20<= <=25  >1023
143.191.13.0/24                         en0                Out  Permit
                    143.191.13.0/24     en0                Out  Permit
Default                                 en0                     Deny
```

Notice that the list of filters for the en0 interface now only includes filters with a source or destination address of 143.191.13.0/24.

The next example illustrates how the source|destination and in|out parameters can be used to limit the number of filters which are removed from the list.

```
NB:Top>Configure>IP>Filter> delete 143.191.13/24 en0 source in
NB:Top>Configure>IP>Filter> list en0
Source              Destination         Interface Protocol I/O  Permit/Deny
                    143.191.13.0/24     en0       TCP      In   Permit
                                          port rules: =25 >1023
                    143.191.13.0/24     en0                In   Permit
143.191.13.0/24                         en0       TCP      Out  Permit
                                          port rules:  20<= <=25  >1023
143.191.13.0/24                         en0                Out  Permit
                    143.191.13.0/24     en0                Out  Permit
Default                                 en0                     Deny
NB:Top>Configure>IP>Filter> delete 143.191.13/24 en0 out
NB:Top>Configure>IP>Filter> list en0
Source              Destination         Interface Protocol I/O  Permit/Deny
                    143.191.13.0/24     en0       TCP      In   Permit
                                          port rules: =25 >1023
                    143.191.13.0/24     en0                In   Permit
143.191.13.0/24                         en0       TCP      Out  Permit
                                          port rules:  20<= <=25  >1023
Default                                 en0                     Deny
```

The first **delete** command in the above example removed the inbound filter with source address of 143.191.13.0/24 from the en0 list of filters.

The second **delete** command removed all outbound filters with a source or destination address of 143.191.13.0/24 except for the one with a specific protocol parameter (TCP) assigned to it. To delete one or more filters which have an assigned protocol, you must include the protocol parameter in the **filter delete** command as shown below.

```
NB:Top>Configure>IP>Filter> delete 143.191.13/24 en0 tcp
NB:Top>Configure>IP>Filter> list en0
Source              Destination         Interface Protocol I/O  Permit/Deny
                    143.191.13.0/24     en0                In   Permit
Default                                 en0                In   Deny
```

At this point only one filter other than the default filter for the en0 interface remains. When the last filter is deleted the default filter will also be removed and filtering for the en0 interface will be turned off.

```
NB:Top>Configure>IP>Filter> delete 143.191.13/24 en0
NB:Top>Configure>IP>Filter> list en0
Source              Destination         Interface Protocol I/O  Permit/Deny
NB:Top>Configure>IP>Filter>
```

filter deny <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto> [port]] [source|destination] [in|out]

The **filter deny** command adds one or more deny filters to the list of filters. Refer to the **filter add** command for a description of the command parameters.

filter flush [<interface>]

The **filter flush** command removes all filters assigned to a specified interface from the list of filters. If this command is entered without the <interface> parameter, you will be asked if you want to remove all filters.

filter list <interface>

The **filter list** command displays a list of packet filters currently used for the specified interface. Entering this command without the optional <interface> parameter displays a list of filters for all interfaces.

If the IP packet filter list for a particular interface is empty, all packets routed through that interface will be allowed to pass through unimpeded. If one or more filters have been applied to an interface, only the IP packets that meet the criteria of a permit filter will be allowed to continue through the interface.

The list of filters for a particular interface are arranged in the following order:

- **Inbound filters** – apply to packets entering the NetBlazer through the designated interface.
- **Outbound filters** – apply to packets leaving the NetBlazer through the interface.
- **Default filter** – applies to any packet which was not trapped by one of the inbound or outbound filters. Note that if an In or Out is listed in the I/O field of the default filter, the filter only applies to packets traveling in that direction. Packets going in the opposite direction are automatically permitted.

Filters within the above grouping are listed in the order of precedence. For example, a filter at the top of the inbound filter takes precedence over filters further down in the list of inbound filters for a particular interface. The same is true for outbound filters.

filter lookup <address>[/<bits>] <address>[/<bits>] <interface> [<proto> [port]]

The **filter lookup** command is used to determine whether or not a packet will be allowed to pass through the NetBlazer based on the packet's source address, destination address, protocol and destination port number.

The two <address>[/<bits>] parameters specify the source and destination addresses of the packet in that order.

The <interface> parameter identifies the <u>input</u> interface of the packet.

The optional <proto> and <port> parameters indicate the protocol (i.e. **tcp**, **udp** or **icmp**) and destination port number of the packet.

To determine if a packet will be allowed to pass through the NetBlazer, it performs the following lookup operations:

- Filter lookup for permission to route the packet through the input interface.
- Routing table lookup to determine if a route exists for the destination address.
- Filter lookup for permission to send the packet out through the output interface.

All three conditions must be satisfied to allow the packet to continue through the NetBlazer toward its destination.

filter notify [on|off]

The **filter notify** command controls whether or not a Destination Unreachable message is sent when a packet is denied by the NetBlazer filters.

Entering this command without specifying on or off displays its current setting. The NetBlazer is shipped with notify set to send a Destination Unreachable message when a packet is denied. Entering a **filter notify off** command configures the NetBlazer to silently discard packets that are denied.

filter permit <address>[/<bits>] [<address>[/<bits>]] <interface> [<proto> [port]] [source|destination] [in|out]

The **filter permit** command adds one or more permit filter to the list of filters. Refer to the **filter add** command for a description of the command parameters.

Grep Command

The **grep** command, located at the Configure command level, searches a file or the piped output of a command for all occurrences of a specified string.

To display all lines in a file that contain a specified string, use the **grep** <string> <filename> command . For example, the following command displays a list of all the route commands in the STARTUP.CNF file.

```
NB:Top> grep route startup.cnf
#add static routes:
configure ip route addprivate 143.191.20.1/32    tbupdate
configure ip route addprivate 127.0.0.1/32    loopback
configure ip route add 143.191.10.0/24    en0
configure ip route addprivate 143.191.22.0/24    en0  143.191.10.29  3
configure ip route add default en0  143.191.10.253  4
```

To scan the output of a NetBlazer command for lines containing a specified string, use the **|grep** <string> command. The next example displays a list of routes that use IP address 143.191.10.29 as a gateway.

```
NB:Top> route list |grep " 143.191.10.29"
143.191.13.0    24  en0          143.191.10.29    2       P 0      0
143.191.22.0    24  en0          143.191.10.29    3       P 0      0
```

When searching for a <string> that includes one or more spaces, you must enclose the string in quotes as illustrated in the above screen.

When searching for a <string> that includes an exclamation point, you must enter it as \! to avoid confusion with a history reference.

Not all commands can be piped through **grep**; in particular any command that starts a session (e.g. **ping** or **telnet**) and **trace** commands.

Group Command

The **group** command takes you to the Configure Group command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Group command level.

```
NB:Top> group
NB:Top>Configure>Group> ?
Available commands:
add              delete           list             modify
top>             up>              ?
```

These commands are used to set up and maintain line characteristics groups. Line characteristic groups are used to select an appropriate asynchronous line when dialing out to a particular destination by comparing the dial-out destination characteristics to the characteristics assigned to each line. Only lines which exactly match the dial-out destination requirements can be used to call that destination.

group add <group-name>

The **group add** command adds a group name to the line characteristics group list. After adding a new group name to the group list, you should assign characteristics to the group using the **group modify** command.

group delete <group-name>

The **group delete** command removes the group specified by the <group-name> parameter from the line characteristics group list.

group modify <group-name> add | delete <string>
group modify <group-name> delete #<string-number>

The **group modify** command is used to add or delete characteristics assigned to a line characteristics group.

The <group-name> parameter is the name of the group previously created through a **group add** command.

When adding a characteristic tag to a group, the <string> parameter is the characteristic tag assigned to the specified group. This can be a characteristic that is defined in the MODEMCAP.TXT file, or a user defined string.

When deleting a characteristic from a group using the first form of the command, the <string> parameter is the characteristic which is to be removed from the specified group. In the second form of the command, the #<string-number> parameter is the number assigned to the characteristic which can be obtained by performing a **group list** <group-name> command.

The following example shows you how to create a group named telebit that includes two characteristics tags, dialout and branch.

```
NB:Top>Configure>Group> add telebit
NB:Top>Configure>Group> modify telebit add dialout
NB:Top>Configure>Group> modify telebit add branch
NB:Top>Configure>Group> list telebit
telebit:
 0: dialout          1: branch
```

The next two examples show you how to remove a characteristic tag from the group telebit using either of the **group modify delete** command formats.

```
NB:Top>Configure>Group> modify telebit delete #0
NB:Top>Configure>Group> modify telebit delete branch
```

group list [<group-name>]

The **group list** command displays the specified group and its assigned characteristics. The optional <group-name> parameter specifies which group to display. If this command is entered without a <group-name> variable, a list of group names is displayed.

History Commands

The **history** command takes you to the History command level. To view a list of the commands available at this level, type a question mark (?) at the History command level.

```
NB:Top> history
NB:Top>History> ?
Available commands:
?                list             status           top>
on               off              prompt           up>
```

These commands control the NetBlazer history mechanism, which saves a list of recent NetBlazer commands. You can use this list to avoid repetitive typing by using a shorthand notation to enter commands based on previous commands. Refer to the **history list** command description on the next page for instruction on how to use the history list while entering commands.

history status

The **history status** command displays the current settings of the history mechanism.

```
NB:Top>History> status
history is on: max=20 count=4 first=1 prompt=off debug=0
```

The above example shows the NetBlazer's default history settings. The NetBlazer is shipped with history turned on. The maximum number of history lines displayed by the **history list** command is 20, and the **history prompt** is turned off.

The count field shows the current number of entries in the history list, and the first field shows the number of the first entry in the list. The debug field is used for maintenance purposes only.

history on [<number-of-lines>]

The **history on** command turns on the history mechanism and optionally sets the maximum number of lines in the list.

```
NB:Top>History> on 10
NB:Top>History> status
history is on: max=10 count=6 first=1 prompt=off debug=0
```

If you do not specify the number of lines in the **history on** command, the NetBlazer uses the current maximum.

history off

The **history off** command turns off the history mechanism, and clears the list of recent commands.

```
NB:Top>History> off
NB:Top>History> status
history not on
```

history prompt

The **history prompt** command instructs the NetBlazer to display an event number at the end of each command prompt when history is on.

```
NB:Top>History> prompt
NB:Top>History> on
NB:Top>History> [1] top
NB:Top> [2] ip address
Global IP address is 143.191.10.28
NB:Top> [3] escape
current escape character is ^[ (decimal 27)
NB:Top> [4]
```

Notice that the event numbers are incremented with each command. These event numbers are used to repeat or modify a recent command referenced by its event number as described in the following **history list** command.

history list [<number-of-lines>]

The **history list** command displays a list of recent commands (events).

```
NB:Top> [5] history list
    1 top
    2 ip address
    3 escape
    4 ping judy 128 5 0 1
    5 history list
```

The above example shows five commands, which were entered since the last time history was turned on.

You can display just part of the history list by entering the number of lines to display.

```
NB:Top> [6] history list 4
    3 escape
    4 ping judy 128 5 0 1
    5 history list
    6 history list 4
```

This example displays the last four commands in the history list.

Note: *If you do not specify the number of lines to display in the* ***history list*** *command, the NetBlazer displays up to 20 lines. You can display more than 20 lines by specifying a number greater than 20 lines.*

Referencing Previous Commands

The history mechanism uses the exclamation mark (!) followed by a variable to reference a previous command. The reference variable may be a second exclamation mark (!) to reference the previous command, a number to reference a recent command by its event number, or a character string to reference the most recent command that begins with or contains the character string.

Caution: *To prevent the history mechanism from misinterpreting an exclamation mark (!) used in other NetBlazer commands when history is on, you must escape the exclamation mark by preceding it with a backslash (\). For example, !23 in a filter command should be enter as \!23 when history is on.*

Repeating Previous Commands

To reissue the last command (**history list 4** in the previous example), enter **!!** at the command prompt.

```
NB:Top> [7] !!
history list 4
    4 ping judy 128 5 0 1
    5 history list
    6 history list 4
    7 history list 4
```

To reissue some other recent command, enter **!** followed by the event number of the command.

```
NB:Top> [8] !5
history list
    1 top
    2 ip address
    3 escape
    4 ping judy 128 5 0 1
    5 history list
    6 history list 4
    7 history list 4
    8 history list
```

You can repeat a command by matching the first word or part of the first word in the command string. The following example shows you how to reissue the last **escape** command entered.

```
NB:Top> [9] !e
escape
current escape character is ^[ (decimal 27)
```

Entering **!e** repeats the most recent command that starts with the letter e. If the history list contains other commands that begin with e, you may need to enter additional characters to distinguish one command from another.

You also can repeat a command by matching a character string contained in the command string. The next example shows you how to reissue the last command that contained the word judy.

```
NB:Top> [10] !?judy?
ping judy 128 5 0 1?
PING judy(143.191.10.29): 128 bytes      min/avg/max/mdev
Type ^[ (decimal 27) <CR> to return to NetBlazer
128 bytes from 143.191.10.29 icmp_seq=0 time=20 ms  20/20/20/0
128 bytes from 143.191.10.29 icmp_seq=1 time=130 ms  20/34/130/24
128 bytes from 143.191.10.29 icmp_seq=2 time=0 ms  0/30/130/26
128 bytes from 143.191.10.29 icmp_seq=3 time=120 ms  0/41/130/39
128 bytes from 143.191.10.29 icmp_seq=4 time=0 ms  0/36/130/38
----judy PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max/mdev = 0/36/130/38
```

Notice that you must surround the reference string with question marks (?) to reference the last command containing the string.

Building New Commands from Previous Commands

You can build a new command by extending a previous command as illustrated below.

```
NB:Top> [11] !e
escape
current escape character is ^[ (decimal 27)
NB:Top> [12] !! ctrl-z
escape
NB:Top> [13] !!:0
escape
current escape character is ^Z (decimal 26)
```

In the above example, we added a ctrl-z to the end of the most recent **escape** command, thereby changing the escape character to ^Z (decimal 26).

On line 13, we confirmed that the escape character was changed by directing the NetBlazer to issue a command that consisted of the first word (field 0) of the last command in the history list. This illustrates how you can build new commands by using selected fields from previous commands.

The history mechanism treats each command line as a series of numbered fields starting at 0. The fields are separated by spaces. For example, the **route add** command shown below consists of 6 fields, numbered 0-5.

```
  0     1          2         3          4         5
route add 143.191.13/24 en0 143.191.10.29 2
```

You can refer to individual fields in a command by entering **:**<field-number> after the event identification. The **:**<field-number> identifies which field to use in the new command. For example, assume that you have just added a route, and you want to verify that the route is correct by doing a route lookup.

```
NB:Top> [14] route add 143.191.13/24 en0 143.191.10.29 2
NB:Top> [15] !!:0 lookup !!:2
route lookup 143.191.13/24
Dest            Len Interface    Gateway          Metric  P Timer  Use
143.191.13.0    24  en0          143.191.10.29    2          0      0
```

The **!!:0** and **!!:2** in line 15 refer to the first and third field of the previous command.

You can also specify a range of fields to use in the new command. For example, if you want to enter a second route that uses the same interface and gateway as the previous **route add** command, you could enter the following command.

```
NB:Top> [16] !14:0-1 143.191.14/24 !14:3-5
route add 143.191.14/24 en0 143.191.10.29 2
```

Notice how this command uses the range of fields from history line 14 to build the new **route add** command.

Editing Previous Commands

You can build new commands by editing the previous command. This is useful for correcting typing errors or changing words within the command. The command syntax for editing the previous command is:

^old string^new string^

The caret symbol (^) is used to separate the text that you want to replace from the new text. The rest of command line remains unchanged.

To illustrate, assume that you want to change the route entered on line 14 to private. First, make the command on line 14 current.

```
NB:Top> [17] !14:p
route add 143.191.13/24 en0 143.191.10.29 2
```

The **:p** in the above command makes the command specified by **!14** current, but does not execute the command.

Next, enter the following command to change the word **add** to **addprivate** and execute the new command.

```
NB:Top> [18] ^add^addprivate^
route addprivate 143.191.13/24 en0 143.191.10.29 2
```

Finally, verify that the route has been marked private.

```
NB:Top> [19] route lookup !!:2
route lookup 143.191.13/24
Dest            Len Interface    Gateway          Metric  P Timer  Use
143.191.13.0    24  en0          143.191.10.29    2       P 0      0
```

History Command Summary

The following table summarizes the syntax of the history commands.

Command Syntax	Description
history on [<number of lines>]	Turns on the history mechanism and optionally sets the maximum number of lines in the history list.
history off	Turns off the history mechanism and clears the history list.
history prompt	Displays an event number at the end of each command prompt when history is on.
history status	Displays the current settings of the history mechanism.
history list [<number of lines>]	Displays a list of recent commands (events).
!!	Repeats the last command entered.
!<event number>	Invokes the command specified by the event number.
!<string>	Invokes the most recent command beginning with <string>.
!?<string>**?**	Invokes the most recent command containing <string>.
!<event>**:p**	Displays and makes current the specified event, but does not invoke the command.
!<event>**:**<number>	References field <number> of the specified event.
!<event>**:**<start>**–**<end>	References fields ranging from <start> to <end> of specified event.
^<old string>**^**<new string>**^**	Replaces <old string> with <new string> in last command.

Hostname Command

The **hostname** [<hostname>] command is used to display or set the hostname of the NetBlazer which will appear in the NetBlazer command prompt. If the <hostname> parameter is omitted in the command, the current hostname is displayed.

ICMP List Command

The **icmp list** command produces a list of ICMP statistics, which is maintained as part of the Management Information Base (MIB) used by ICMP. A sample listing follows.

```
NB:Top> icmp list
( 1)icmpInMsgs               104     (14)icmpOutMsgs               17
( 2)icmpInErrors               0     (15)icmpOutErrors              0
( 3)icmpInDestUnreachs         0     (16)icmpOutDestUnreachs        0
( 4)icmpInTimeExcds            0     (17)icmpOutTimeExcds           0
( 5)icmpInParmProbs            0     (18)icmpOutParmProbs           0
( 6)icmpInSrcQuenchs           2     (19)icmpOutSrcQuenchs          2
( 7)icmpInRedirects            0     (20)icmpOutRedirects           0
( 8)icmpInEchos               85     (21)icmpOutEchos              17
( 9)icmpInEchoReps            17     (22)icmpOutEchoReps           85
(10)icmpInTimestamps           0     (23)icmpOutTimestamps          0
(11)icmpInTimestampReps        0     (24)icmpOutTimestampReps       0
(12)icmpInAddrMasks            0     (25)icmpOutAddrMasks           0
(13)icmpInAddrMaskReps         0     (26)icmpOutAddrMaskReps        0
```

The left hand column lists all the ICMP messages that have been received. The right hand column lists all the ICMP message that have been sent. The following is a list of the field names and definitions:

icmpInMsgs – The total number of ICMP messages received by the NetBlazer since it was started.

icmpInErrors – The number of error messages received.

icmpInDestUnreachs – The number of destination unreachable messages received.

icmpInTimeExcds – The number of time exceeded messages received.

icmpInParmProbs – The number of parameter problem messages received.

icmpInSrcQuench – The number of source quenches received.

icmpInRedirects – The number of redirect messages received.

icmpInEchos – The number of echo requests (pings) received.

icmpInEchoReps – The number of echo replies (responses to pings) received.

icmpInTimestamps – The number of time stamp requests received.

icmpInTimestampReps – The number of time stamp replies received in response to our time stamp requests.

icmpInAddrMasks – The number of address masks received.

icmpInAddrMaskReps – The number of address mask replies received in response to our address mask requests.

Interface Commands

The **interface** command takes you to the Configure Interface command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Interface command level.

```
NB:Top> interface
NB:Top>Configure>Interface> ?
Available commands:
attach          delete          dump-mib        dynamic>
list            mru             mtu             rif
start           stop            sync-backup     top>
up>             watchdog        ?
```

Commands available at this level are used to configure and control the interfaces used by the NetBlazer.

The NetBlazer uses three types of interfaces: static, dynamic, and dedicated SLIP or PPP interfaces.

The Ethernet interfaces (en0, en1, and en2), Token Ring (tr0, tr1, and tr2) and the synchronous interface (syn0) are static interfaces. These interfaces are automatically created for each Ethernet, Token Ring or synchronous interface board in the NetBlazer when the system is booted. However, before using any of these interfaces, you must preconfigure the interfaces as described in the *NetBlazer Configuration Guide*.

Dynamic interfaces, used to route IP packets over the switched telephone network, are created by the **ipdial** or **dynamic add** commands. Refer to Dynamic Commands in this manual for a description of the commands used to control dynamic interface connections.

Dedicated SLIP or PPP interfaces, used to send IP packets through a serial RS-232 connection, are created by the **interface attach** command.

interface attach <line-name> slip|ppp <interface> [<mru>] [ip_addr]

The **interface attach** command configures an asynchronous serial line for transmitting IP datagrams over a dedicated SLIP or PPP link. As illustrated in Figure INT-1, a dedicated SLIP or PPP link may be used to interconnect two NetBlazers, or to attach a workstation, running SLIP or PPP software, to the NetBlazer.

Note: *Creating a dedicated interface automatically turns off **callin**, **autoconfigure**, **dtrtime** and **dcdtime** for the line that is used by that interface. Deleting the interface restores the default values for these line parameters.*

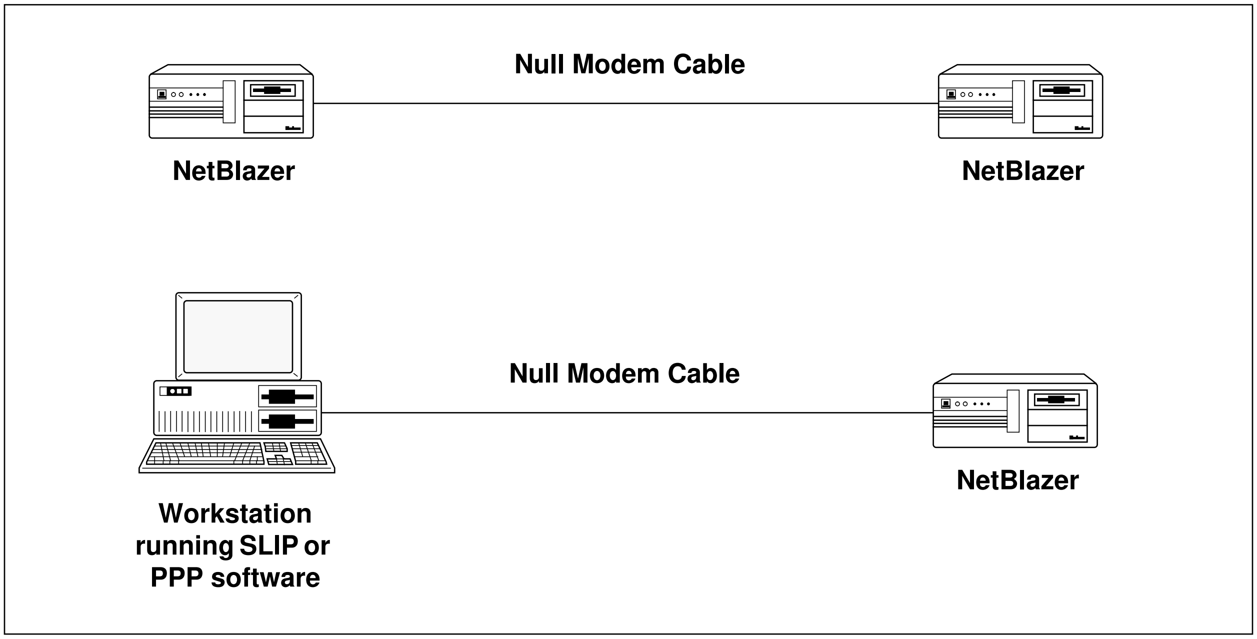


Figure INT-1. Dedicated SLIP or PPP Links

The <line-name> parameter specifies which asynchronous line to use as a dedicated SLIP or PPP link.

The slip | ppp parameter determines the framing protocol used to transmit IP datagrams over the serial link.

The <interface> parameter is the name that you assign to the new interface created by this command.

The optional <mru> (maximum receive unit) parameter sets the number of bytes in the largest packet that may be received or transmitted over the dedicated link. This parameter defaults to 1006 for a SLIP link, and 1500 for a PPP link.

The optional <IP-address> parameter assigns an IP address to the named interface. If this parameter is not entered, the NetBlazer global IP address, set by the **ip address** command, is assigned to the interface.

The following example illustrates how to attach a PPP interface, named bob, to line11.

```
NB:Top>Configure>Interface> attach line11 ppp bob
line11: stopping login and disabling autoconfigure
line11: interface=bob, protocol=PPP, MRU=1500, ip address=143.191.12.26
```

Before you can actually use this dedicated PPP link, you must set up a route and do an activeopen on the link.

```
NB:Top>Configure>Interface> route add 192.124.95/24 bob
NB:Top>Configure>Interface> ppp activeopen bob
NB:Top>Configure>Interface> list
Name                IP address   MRU   MTU type    line      pkts in   pkts out
bob               143.191.12.26  1500  1500 PPP     Up             16         17
tbupdate          143.191.12.26  1500  1500 Dialout Down            0          0
syn0              143.191.12.26  1500  1500 PPP     Down            0         20
en0               143.191.12.26  1500  1500 Ether                   0          1
loopback              127.0.0.1 65535 65535                         0          0
```

The other end of the link also must be active and have a valid route across the link.

interface delete <interface>

The **interface delete** command deletes a specified virtual interface from the interface list. You cannot delete the loopback or static interfaces.

Note: *Any user, dialout destination, and/or routes associated with the interface are also deleted.*

interface dump-mib <interface>

The **interface dump-mip** command displays the variables maintained by each interface for SNMP as defined in RFC-1213.

interface list [<interface>]

The **interface list** command displays status information about the interfaces, which are currently used by the NetBlazer.

If the optional <interface> parameter is entered, additional information for the named interface is displayed. This information varies depending on the type of interface and its present status.

interface mru [<interface>] [<mru>]

The **interface mru** command sets the maximum receive unit (i.e. number of octets or bytes) that may be transmitted through a specified interface.

If this command is entered without the <interface> or <mru> parameters, the system will prompt you for the required information.

We recommend that you use the default MRU setting: 1500 for PPP or 1006 for SLIP connections.

interface mtu [<interface>] [<mtu>]

The **interface mtu** command sets the maximum transmit unit (i.e. number of octets or bytes) that may be transmitted through a specified interface.

If this command is entered without the <interface> and <mtu> parameters, the system will prompt you for the required information.

You cannot set the MTU for a PPP interface. This value is negotiated based on the peer MRU.

We recommend that you use the default MTU setting of 1006 for SLIP connections.

interface rif <interface> [<node_address>]

The **interface rif** command displays the contents of the Routing Information Fields, which are currently used only in Token Ring environments. For additional information, refer to RFC 1042, IEEE 802.5 (Token Ring Media Access Control), or the IBM Token-Ring Network Architecture Reference Manual.

interface start <interface>

The **interface start** command restarts an interface that has been previously turned off by a **interface stop** command.

interface stop <interface>

The **interface stop** command turns off a specified interface. This command only works for static or dedicated interfaces. It cannot be used to stop a dynamic interface.

interface sync-backup [<interface>]

The **interface sync-backup** command specifies which dynamic interface to use instead of syn0 when the **watchdog** function determines that the synchronous interface line is down.

When the synchronous interface (syn0) goes down, and the NetBlazer switches to the backup interface, all routes using syn0 are changed to the backup interface. When the synchronous interface line returns, the routes are returned to syn0.

Entering the **sync-backup** command without specifying an interface, turns off the synchronous backup function.

interface watchdog <interface> on|off|<IP address> [<seconds>]

The **interface watchdog** command starts checking a specified interface, which uses the synchronous port (syn0), to detect if the link goes down. The NetBlazer uses either PPP echo requests or ICMP pings to check the link.

Issuing a **watchdog on** command uses PPP echo requests to check the synchronous link; whereas issuing a **watchdog** <ip address> sends ICMP pings to the specified IP address to check the link.

The optional <seconds> parameter determines how frequently the NetBlazer checks or attempts to restart the synchronous link. If the number of seconds is not specified, a default value of 120 seconds is used.

When the NetBlazer detects that the synchronous link is down, it switches the routes for the synchronous interfaces to the backup interface set by the **sync-backup** command, and attempts to restart the link with active PPP packets.

Issuing a **watchdog off** command turns off the watch dog function.

IP Commands

The **ip** command takes you to the Configure IP command level. To view a list of the commands available at this level, type a question mark (?) at the Configure IP command level.

```
NB:Top> ip
NB:Top>Configure>IP> ?
Available commands:
address          domain>          farside          filter>
icmp>            ipdial           list             rip>
route>           rtimer           tcp>             top>
trace            ttl              vj               up>
?
```

Commands at this level are used to configure and manage the IP (Internet Protocol) facilities. Commands followed by an angle bracket (>) in the above list take you to a subcommand level. Refer to the specific command heading for a description of these commands. For example, for information about the **route** commands, look under Route Commands.

ip address [<ip-addr>[/bits]] [<interface>|<line> [<port>]]

The **ip address** command sets the global IP address assigned to the NetBlazer or sets the IP address assigned to a specified interface or line.

Entering this command without any parameters displays the global IP address of the NetBlazer.

```
NB:Top>Configure>IP> address
Global IP address is 0.0.0.0
```

The NetBlazer is shipped with the global IP address set to 0.0.0.0. The next example illustrates how to set the NetBlazer's global IP address.

```
NB:Top>Configure>IP> address 143.191.166.1/24
Set IP address(es) of 3 interface(s)
Set IP address(es) of 2 lines(s)
NB:Top>Configure>IP> address
Global IP address is 143.191.166.1
NB:Top>Configure>IP> interface list
Name               IP address   MRU   MTU type    line      pkts in   pkts out
tbupdate        143.191.166.1  1500  1500 Dialout Down            0          0
en1             143.191.166.1  1500  1500 Ether                   0          1
en0             143.191.166.1  1500  1500 Ether                   0          1
loopback            127.0.0.1 65535 65535                         0          0
```

The global IP address is automatically assigned to each interface and line.

The /24 in the previous example tells the NetBlazer that the first 24 bits are the network part of the IP address. It uses this information to automatically create a route through en0 and ARP broadcast entry for the 143.191.166.0 network.

```
NB:Top>Configure>IP> route list
Dest             Len Interface    Gateway          Metric  P Timer  Use
143.191.20.1     32  tbupdate                      1       P 0      0
127.0.0.1        32  loopback                      1       P 0      0
143.191.166.0    24  en0                           1         0      0
NB:Top>Configure>IP> arp list
received 0 badtype 0 bogus addr 0 reqst in 0 replies 0 reqst out 2
IP addr          Type  Interface   Time Q Addr              RIF
143.191.166.255 ether any          0      ff:ff:ff:ff:ff:ff none
```

Each ethernet interface must have an unique IP address. The next example illustates how to set the IP address for a specific interface.

```
NB:Top>Configure>IP> address 143.191.167.1/24 en1
Set IP address of interface 'en1'
```

The /24 parameter in this example tells the NetBlazer to create a route through en1 and ARP broadcast entry for the 143.191.167.0 network.

```
NB:Top>Configure>IP> route list
Dest             Len Interface    Gateway          Metric  P Timer  Use
143.191.20.1     32  tbupdate                      1       P 0      0
127.0.0.1        32  loopback                      1       P 0      0
143.191.166.0    24  en0                           1         0      0
143.191.167.0    24  en1                           1         0      0
NB:Top>Configure>IP> arp list
received 0 badtype 0 bogus addr 0 reqst in 0 replies 0 reqst out 3
IP addr          Type  Interface   Time Q Addr              RIF
143.191.167.255 ether any          0      ff:ff:ff:ff:ff:ff none
143.191.166.255 ether any          0      ff:ff:ff:ff:ff:ff none
```

An **ip address** <ip-addr> <line> command is used to set the IP address of a specified line to an address that is different than the NetBlazer's global IP address.

```
NB2:Top> ip address 143.191.167.253 line10
```

This allows you to telnet to the line from another host that has a route to the line's host or network address.

```
NB:Top>Sessions> telnet 143.191.167.253
Type ^] (decimal 29) <CR> to return to NetBlazer
Trying 143.191.167.253:23...
Telnet session 0 connected to 143.191.167.253

Type ^] (decimal 29) <CR> to return to NetBlazer
Talking to line10
```

When the connection is established, you can issue modem commands, such as **ate1v1** or **at&f**, just as if you were directly connected to the line through a talk session.

Issuing an **ip address** <ip-addr> <line> <port> command changes the TCP port number that is assigned to a line for raw TCP connections as described in the *NetBlazer Version 1.4 Release Notes*.

ip farside <interface>

The **ip farside** command displays a list of recommended settings to use when configuring a NetBlazer on the remote side of a specified dynamic interface. This information is also displayed at the end of an **ipdial** routine.

ip ipdial

The **ipdial** command starts a dialog that steps you through the process of creating a dynamic dial-up interface to a remote site. Before starting **ipdial**, you will need the following information:

- The serial link protocol (SLIP or PPP) used to encapsulate IP packets sent between the two sites. The same protocol must be used at both sites.
- The user IDs, passwords, and crypto keys used to authenticate an inbound and/or outbound connection.
- The telephone number(s) used to call the remote NetBlazer.
- The IP addresses of each remote network that can be accessed either directly or indirectly through this interface.

An example of a typically **ipdial** dialog follows:

When you start the **ipdial** dialog, you are asked for the name of the remote NetBlazer, and to specify if this interface will be used to for dial-out and/or dial-in connections.

```
Eden:Top> ipdial
Name of remote (other) system: Paradise
Allow dial-out from this interface [y]?
Allow dial-in to this interface [y]?
```

To allow both outbound and inbound connections through the interface, press the return key or enter “y” in response to each of the above prompts.

The NetBlazer then prompts you for information that is required to create the interface record, which varies depending on how you responded to the previous prompts. This example assumes that you are allowing both dial-out and dial-in connections.

```
Begin adding interface 'Paradise' ...
User ID to send to remote system [Eden]:
Want to enter a password [y]?
Password to send to remote system (no echo):
Re-enter the password:
Will remote system issue a crypto handshake [n]? y
Enter the crypto key (no echo):
Re-enter the crypto key:
Chat script [ics]:
SLIP or PPP (s|p) [s]:
Maximum lines to connect to remote system [1]:
What is the Maximum Receive Unit (MRU) [1006]:
Add dynamic interface=Paradise, user=Eden, chat=ics, using SLIP
    with crypto handshaking, max-lines=1, MRU=1006
Okay (yes|no|quit) [y]?
```

Note that the remote NetBlazer must have a user record with the same user ID, password and crypto key entered in this interface record to authenticate a connection initiated from this interface.

If dial-out connections are allowed through the interface, the NetBlazer prompts you for information required to create one or more dial-out records for the interface.

```
Begin adding dial-out 'Paradise' ...
Phone number: 9-745-3864
Line characteristics (dialout):
Dialout: name=Paradise, phone=9-745-3864, characteristics=dialout
Okay (yes|no|quit) [y]?
Add another dial-out [n]? y
Phone number: 9-745-3863
Line characteristics (dialout):
Dialout: name=Paradise, phone=9-745-3863, characteristics=dialout
Okay (yes|no|quit) [y]?
Add another dial-out [n]?
```

If dial-in connections are allowed through the interface, the NetBlazer prompts you for information required to create the user record that is used to authenticate a call from the remote NetBlazer.

```
Please add the information that the remote system 'Paradise'
        will use to contact this NetBlazer
Want to enter a password [y]?
Enter the password (no echo):
Re-enter the password:
Issue a crypto handshake [n]? y
Enter the crypto key (no echo):
Re-enter the crypto key:
User: Paradise         privs:     dest-group:
Okay (yes|no|quit) [y]?
```

Next, the NetBlazer gives you the opportunity to add routes for networks at the remote site to its routing table.

```
Add routes for interface [y]?
Begin adding route thru 'Paradise' ...
Destination name or address/bits (Return to finish routes): 143.191.168/24
Metric [1]:
Private [n]:
Add route destination=143.191.168.0, bits=24, metric=1, not private
Okay (yes|no|quit) [y]?
Route added
Begin adding route thru 'Paradise' ...
Destination name or address/bits (Return to finish routes): 143.191.169/24
Metric [1]:
Private [n]:
Add route destination=143.191.169.0, bits=24, metric=1, not private
Okay (yes|no|quit) [y]?
Route added
Begin adding route thru 'Paradise' ...
Destination name or address/bits (Return to finish routes):
```

When you have finished adding routes, the NetBlazer displays information that is used to set up a corresponding dynamic interface at the remote site.

```
... Done adding dial IP
For proper 2-way communications, remote system should have:
1) Dynamic interface=Eden, user=Paradise, chat=ics, using SLIP
   with crypto handshaking, max-lines=1, MTU<=1006
2) Dialout=Eden
3) User=Eden
4) Route destination=143.191.167.254, interface=Eden, gateway=none, metric=1
```

ip list

The **ip list** command produces a list of IP statistics, which is maintained as part of the Management Information Base (MIB) used by SNMP.

ip rtimer [<reassemble-time>]

The **ip rtimer** command sets or displays the maximum time in seconds that IP will wait for all the fragments to reassemble fragments into the original datagram.

If this command is entered without the <reassemble-time> parameter, the current value is displayed.

ip trace [<interface>] [<options ...> [<filename>]]

The **ip trace** command is used to view incoming and outgoing information, and to control the format of that information. Trace output is sent to only one port at a time (either a telnet session or a serial connection). That port can trace several different interfaces, but turning off any tracing automatically turns off all tracing.

If the trace output falls too far behind (i.e. more than 250 packets in the queue), trace information will be discarded to avoid tying up too much memory. If syslog is enabled, an informational message is logged indicating how many trace packets were lost.

Entering an **ip trace** command without any parameters displays the current status of tracing for each interface.

The <interface> parameter specifies which interface is to be traced. This is the name assigned to a selected interface.

The <options ...> parameters defines the form and content of the trace output as described in the following table.

Table IP-1. Trace Options

Option	Description
on or **all**	The **on** option is equivalent to specifying both the **in** and **out** options. The **all** option is equivalent to specifying **in out comp uncomp ascii hex** and **raw**.
in and/or **out**	The **in** option traces inbound packets, and the **out** option traces outbound packets. You must specify at least one of these two options to start tracing on a specified interface.
ascii or **hex**	The **ascii** option shows an ASCII dump of the packet's data. The **hex** option shows a HEX and ASCII dump of the data.
comp and/or **uncomp**	These two options are only valid when tracing SLIP interfaces. The **comp** option only shows TCP packets that have been compressed using VJ header compression. The **uncomp** option shows all uncompressed packets.
raw	The **raw** option shows raw data in HEX and ASCII format while tracing a SLIP interface. This option is not valid for non-SLIP interfaces.
off	The **off** option turns tracing off for all interfaces regardless of which one is specified by the <interface> parameter. This will probably change in a later release.

Notes:

Subsequent trace commands add to the trace options set for a specified interface. They do not turn off previously set trace options. To turn off trace options, you must use the **trace** <interface> **off** command.

To trace a SLIP interface, you must specify at least one of the **comp**, **uncomp** or **raw** options.

Normally, trace output is sent to the device from which the trace command is issued. You can redirect the trace output to a diskette file by entering the file name at the end of the command. For example, **trace en0 on trace.out** sends the trace output for en0 to the file named trace.out.

Note: *To view a trace file, you must turn off tracing then enter a* ***more*** *<filename> command.*

You can filter trace information using the **debug filter** commands, which work the same as the standard packet filtering commands. For example if you only want to view trace information on the en0 interface for packets that have a source or destination address of 143.191.64.1, enter the following **debug filter** and **ip trace** commands.

```
Jade:Top> debug filter
Jade:Top>Configure>Debug>Filter> permit 143.191.64.1 en0
Jade:Top>Configure>Debug>Filter> list
Source             Destination        Interface Protocol I/O  Permit/Deny
143.191.64.1/32                       en0                In   Permit
                   143.191.64.1/32    en0                In   Permit
143.191.64.1/32                       en0                Out  Permit
                   143.191.64.1/32    en0                Out  Permit
Default                               en0                     Deny
notify on
Jade:Top>Configure>Debug>Filter> ip trace en0 on ascii
en0: input output (ASCII dump)
```

Note: *Non-IP packets cannot explicitly be filtered, but are controlled by the default filter for that particular interface.*

If you start a trace from a telnet session, which uses the same interface that is being traced, debug filters are automatically set up to filter out the telnet session's packets. An example of the filters that are automatically added to the debug filter list is shown below.

```
Jade:Top> debug filter list
Source             Destination        Interface Protocol I/O  Permit/Deny
143.191.63.1/32    143.191.63.254/32  en0       TCP      In   Deny
                                        port rules: =23
143.191.63.254/32  143.191.63.1/32    en0       TCP      Out  Deny
                                        port rules: =1086
Default                               en0                     Permit
notify on
```

Caution: *Do not delete these filters while tracing is active as traffic will increase exponentially until the connection is unusable. Also, make sure that the default filter's Permit/Deny field is set correctly for your requirements.*

ip ttl [<time-to-live>]

The **ip ttl** command sets the value placed in the time-to-live field of an IP datagram originating from within the NetBlazer.

If this command is entered without the <time-to-live> parameter, the current value is displayed.

ip vj [<interface>] [on|off|auto]

The **ip vj** command controls whether or not Van Jacobson TCP/IP header compression is used for a specified SLIP or PPP interface.

Using VJ Header Compression on a SLIP Interface

Table IP-2 shows the possible VJ header compression setting for a SLIP connection. We recommend that you leave one end of the link set to its default value of auto and use the other end to turn VJ compression on or off.

Table IP-2. VJ Header Compression for SLIP Connections

Local	Remote	Result and Comments
auto	auto	VJ header compression is off. However, it can be turned on by issuing a **vj on** command from either the local or remote end.
	off	VJ header compression is disabled. It can be turned on by issuing a **vj on** command from the remote end.
	on	VJ header compression is enabled. When header compression is turned on, it cannot be turned off for the current connection.
off	auto	VJ header compression is disabled. It can be turned on by issuing a **vj on** command from the local end.
	off	VJ header compression is disabled.
	on	This is an illegal setting, which causes one end of the link to attempt to use header compression while the opposite end is not using header compression. To avoid this condition, set at least one end of the link to auto.
on	auto	VJ header compression is enabled. When header compression is turned on, it cannot be turned off for the current connection.
	off	This is an illegal setting, which causes one end of the link to attempt to use header compression while the opposite end is not using header compression. To avoid this condition, set at least one end of the link to auto.
	on	VJ header compression is enabled. When header compression is turned on, it cannot be turned off for the current connection.

By default, VJ header compression is set to auto, which means that the interface starts with header compression turned off, but it is automatically turned on when a compressed packet arrives from the remote site. Once header compression is turned on, it is not turned off until the connection is dropped.

Using VJ Header Compression on a PPP Interface

VJ header compression for a PPP connection is negotiated between the local and remote end while establishing a connection. Once header compression is negotiated on or off, it remains in effect for the duration of the connection.

Table IP-3 shows the possible VJ header compression settings and results for a PPP connection.

Table IP-3. VJ Header Compression for PPP Connections

Local	Remote	Result and Comments
auto	auto	VJ header compression is negotiated to **on** when a connection is established.
	off	VJ header compression is negotiated to **off** when a connection is established.
	on	VJ header compression is negotiated to **on** when a connection is established.
off	auto	VJ header compression is negotiated to **off** when a connection is established.
	off	VJ header compression is negotiated to **off** when a connection is established.
	on	VJ header compression is negotiated to **off** when a connection is established.
on	auto	VJ header compression is negotiated to **on** when a connection is established.
	off	VJ header compression is negotiated to **off** when a connection is established.
	on	VJ header compression is negotiated to **on** when a connection is established.

Line Commands

The **line** command takes you to the Configure Line command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Line command prompt.

```
NB:Top> line
NB:Top>Configure>Line> ?
Available commands:
access          auto            callin          command
characteristics dcdtime         delay           dsrtime
dtrtime         flowcontrol     idle            list
modem           request-call    reset           speed
terminal        test-request    trace           top>
up>             ?
```

These commands control the operating characteristics of the NetBlazer's RS-232 asynchronous ports.

line access <line-name> [packet|all]

The **line access** command is used to allow only packet mode connections from the outside on a specified line. This prevents callers from using the line as a terminal server connection to the NetBlazer. Refer to Figure LIN-1.

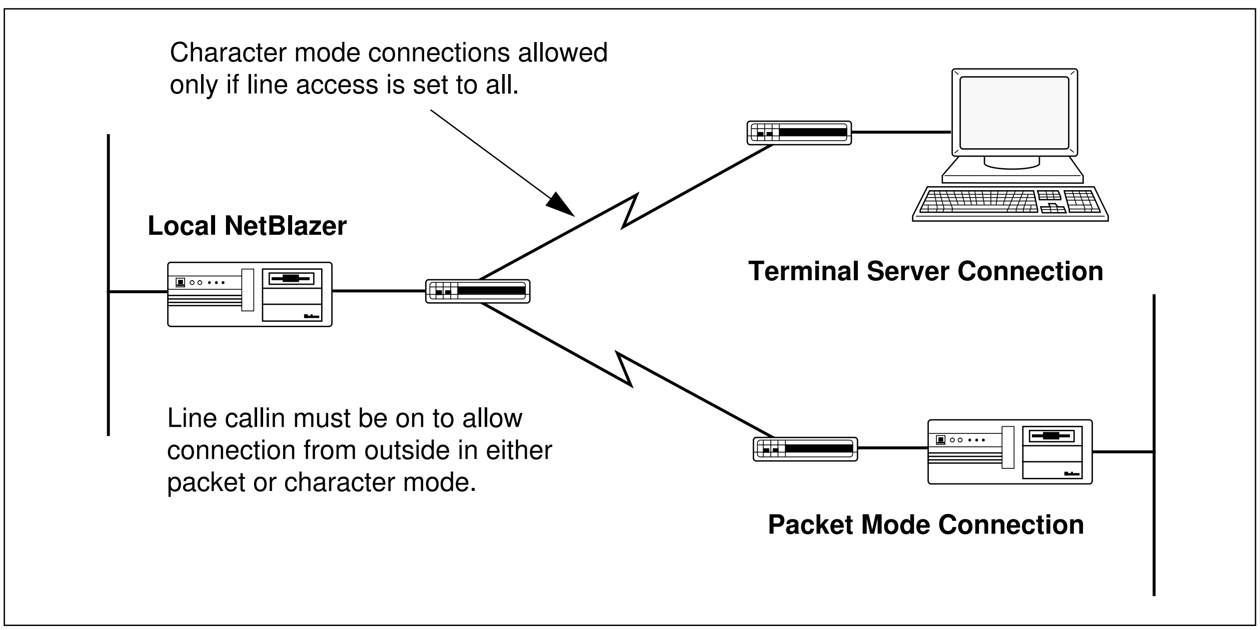


Figure LIN-1. Restricting Line Access

If the optional packet|all parameter is omitted, the current access setting for the specified line is displayed.

```
NB:Top>Configure>Line> access line01
'line01' access is all
```

Normally, line access is set to all, which allows any outside call to establish a connection with the NetBlazer.

Issuing a **line access** <line-name> **packet** command sets the line specified by the <line-name> parameter to only allow packet mode connections for incoming calls.

```
NB:Top>Configure>Line> access line01 packet
'line01' accesscontrol is packet only
```

Issuing a **line access** <line-name> **all** command sets the line to allow all incoming calls.

```
NB:Top>Configure>Line> access line01 all
'line01' accesscontrol is all
```

line auto <line-name> [on|off|config]

When the NetBlazer is turned on or rebooted, it automatically detects the type of modem connected to the line, and preconfigures the line and the modem based on information contained in the MODEMCAP.TXT file. Any line that does not have a modem attached to it is treated as a log-in line.

If a line's DSR signal is dropped and then raised (e.g. the connected device is turned off and then on), the NetBlazer queries the line again to determine the type of device that is attached to the line.

The **line auto** command controls automatic modem detection and configuration for a specified line.

If the optional on|off|config parameter is omitted, the current autoconfigure setting for the specified line is displayed.

```
NB:Top>Configure>Line> auto line01
Autoconfigure of  line01 is on
```

Normally, autoconfigure is set to on, which allows the NetBlazer to automatically determine the type of modem attached to the line.

Note: *Attaching a dedicated SLIP or PPP interface to a line automatically turns off auto-configure for that line.*

Issuing a **line auto** <line-name> **off** command turns off auto-configure for the specified line.

```
NB:Top>Configure>Line> auto line01 off
Autoconfigure of line01 is off
```

Issuing a **line auto** <line-name> **config** command turns off automatic modem type detection, but allows auto-configuration for the modem type defined by the **line modem** command.

```
NB:Top>Configure>Line> modem line01 t3000
NB:Top>Configure>Line> auto line01 config
Autoconfigure of line01 config
```

Issuing a **line auto** <line-name> **on** command returns the specified line to its default state, which allows automatic modem detection and configuration.

```
NB:Top>Configure>Line> auto line01 on
Autoconfigure of line01 is on
```

line callin <line-name> [on|off]

The **line callin** command is used to designate a line specified by the <line-name> parameter as dialout only line. When a line's callin setting is off, the NetBlazer no longer accepts incoming calls on that line.

Refer to the **line access** command description for additonal information on controlling incoming call connections to the NetBlazer.

If the optional on|off parameter is omitted, the current callin setting for the specified line is displayed.

```
NB:Top>Configure>Line> callin line01
Callin on line 'line01' is on
```

Normally, line callin is set to on, which allows remote users to use the line for inbound and outbound calls.

Issuing a **line callin** <line-name> **off** command designates the specified line for outbound calls only.

```
NB:Top>Configure>Line> callin line01 off
Callin on line 'line01' is off
```

Issuing a **line callin** <line-name> **on** command returns the specified line to its default state, which allows both inbound and outbound calls.

```
NB:Top>Configure>Line> callin line01 on
Callin on line 'line01' is on
```

line command <line-name> <command> [<arguments>]

The **line command** command allows you to assign a NetBlazer command string to a line. This command is invoked whenever a call is received on the line.

For example, assigning a **telnet** command to a line automatically passes a caller through to the host specified by the **telnet** command argument.

```
NB:Top>Configure>Line> command line01 telnet robin
```

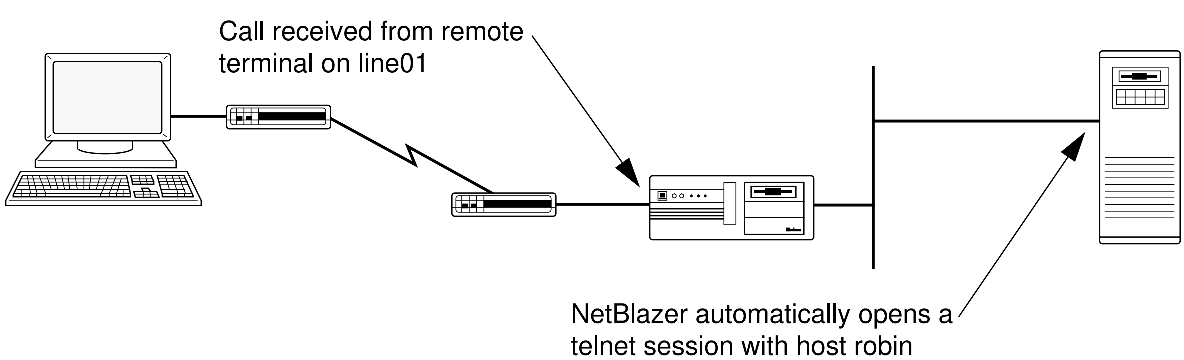


Figure LIN-2. Automatic Telnet

Caution: *If you assign a telnet command to a line, as shown in the above example, the line only can be used for establishing a telnet session with the designated host from a remote terminal.* ***Attempting to use the line for a dynamic (packet mode) connection will have unpredictable results.***

If you assign a command to a line that is attached to a terminal with the DCD signal held high, the command is executed each time the line is not in use. For example, if you assign a telnet robin command to a line with a terminal attached, the NetBlazer establishes a telnet session between the host named robin and the terminal. This allows a user to log in to robin. When the user logs out of robin, another telnet session is automatically established with robin so that the next user can log in to robin from the terminal.

Entering a **line command** command with just a <line-name> parameter removes the NetBlazer command string assigned to the specified line.

```
NB:Top>Configure>Line> command line01
```

line characteristics <line-name> [<group-name-of-characteristics>]

The **line characteristics** command assigns a named group of characteristics to a designated line.

The <group-name-of-characteristics> parameter is the name of the characteristics group that contains the desired characteristics. This group must be an existing characteristics group.

One application of the **line characteristics** command is to assign a line to a characteristic group that includes the raw characteristic. This configures the line for use by the raw server.

```
NB:Top>Configure>Line> group list
dialout         v32dialout      request         pepdialout      raw
raw_dial
NB:Top>Configure>Line> characteristics line01 raw_dial
NB:Top>Configure>Line> list
Line     Modem Type Char's   Use         Command   Flags
line00   <<None>>   <<None>> login      <<None>>   520e <CTS,DCD,DSR,AF,CON,BSY>
line01   t1500      raw_dial <<None>>   <<None>>   40ca <CTS,DSR,NOL,CB,BSY>
```

line dcdtime <line-name> [<seconds>]

The **line dcdtime** command sets the amount of time that a modem must hold DCD low before the NetBlazer decides that the connection has been dropped.

Entering this command without specifying a time displays the current DCD lock time.

```
NB:Top>Configure>Line> dcdtime line01
'line01' dcd lock time is 2 second
```

The default DCD lock time for a line is 2 seconds. You can extend (or reduce) this time as shown in the next example.

```
NB:Top>Configure>Line> dcdtime line01 5
'line01' dcd lock time is 5 seconds
```

If you set the DCD lock time to 0, the NetBlazer ignores DCD signal changes.

```
NB:Top>Configure>Line> dcdtime line01 0
'line01' dcd lock time is 0 seconds
```

line delay <line-name> [<milliseconds>]

The **line delay** command sets the amount of time that the NetBlazer pauses after the modem has accepted a command and returned a result code.

Entering this command without specifying a time displays the current delay time.

```
NB:Top>Configure>Line> delay line01
'line01' delay time is 0 milliseconds
```

The default delay time is 0 milliseconds. If a Telebit modem is attached to the line, you do not need to change this value. However, if some other modem is attached to the line, you may need to increase the delay time as shown in the following example.

```
NB:Top>Configure>Line> delay line01 50
'line01' delay time is 50 milliseconds
```

line dsrtime <line-name> [<seconds>]

The **line dsrtime** command sets the minimum time that modem must hold DSR on for the NetBlazer to assume that the modem is ready.

Issuing this command without specifying a time duration displays the current setting of the DSR lock time.

```
NB:Top>Configure>Line> dsrtime line01
'line01' dsr lock time is 3 seconds
```

The default DSR lock time is 3 seconds. To change this time to 2 seconds for example, issue the following **line dsrtime** command.

```
NB:Top>Configure>Line> dsrtime line01 2
'line01' dsr lock time is 2 seconds
```

line dtrtime <line-name> [<milliseconds>]

When the NetBlazer needs to use a modem, it first drops DTR and issues a series of AT commands to bring the modem back to a known state.

The **line dtrtime** command sets the time duration that the NetBlazer holds down the DTR signal for a specified line.

Issuing this command without specifying a time duration displays the current amount of time that DTR is held low when resetting a modem attached to the specified line.

```
NB:Top>Configure>Line> dtrtime line01
'line01' dtr off time is 1100 milliseconds
```

The default DTR off time is 1100 milliseconds (1.1 seconds). To extend this time to 2 seconds for example, issue the following **line dtrtime** command.

```
NB:Top>Configure>Line> dtrtime line01 2000
'line01' dtr off time is 2000 milliseconds
```

To prevent the NetBlazer from dropping DTR to reset a modem on a specified line, set the dtrtime value to 0.

```
NB:Top>Configure>Line> dtrtime line01 0
'line01' dtr off time is 0 milliseconds
```

Setting a line's dtrtime to 0 also prevents the NetBlazer from issuing AT commands to bring the attached modem back to a known state.

line flowcontrol <line-name> [none|auto|xon|cts]

The **line flowcontrol** command sets the flow control method used by the NetBlazer to regulate the flow of data between the NetBlazer and the device attached to a specified line.

You can configure a line to use XON-XOFF, CTS-RTS, or no flow control; or you can configure the line to automatically switch between XON-XOFF and CTS-RTS or no flow control.

The default flow control setting for a line is auto flow control. This method uses the following rules to determine when to use XON-XOFF, CTS-RTS or no flow control:

- Use CTS-RTS flow control when the NetBlazer detects a modem on the line during the auto-configure process.

- Use XON-XOFF flow control when a user is logged in on the line, and has a command prompt from the command interpreter.

- Turn off XON-XOFF flow control when the user starts a data communications session, such as telnet or dial, to allow the XON and XOFF characters to pass through the NetBlazer. If the CTS signal is present use CTS-RTS flow control; otherwise leave flow control turned off.

Note: *To prevent lost or garbled data, both the NetBlazer and the attached device must agree on the flow control method used. Telebit modems connected to the NetBlazer are configured to use CTS-RTS flow control. Therefore, lines with Telebit modems attached should use auto flow control or CTS-RTS flow control.*

Issuing this command without specifying a flow control method displays the current flow control setting of the line.

```
NB:Top>Configure>Line> flowcontrol line01
'line01' flow control is auto
```

The following example shows you how to set a line to use XON-XOFF flow control.

```
NB:Top>Configure>Line> flowcontrol line01 xon
'line01' flow control is xon/xoff
```

The next example shows you how to set a line to use CTS-RTS flow control.

```
NB:Top>Configure>Line> flowcontrol line01 cts
'line01' flow control is cts/rts
```

To set a line to not use flow control, specify “none” as shown below.

```
NB:Top>Configure>Line> flowcontrol line01 none
'line01' flow control is none
```

To configure a line to automatically switch between XON-XOFF and CTS-RTS or no flow control specify “auto.”

```
NB:Top>Configure>Line> flowcontrol line01 auto
'line01' flow control is auto
```

line idle <line-name> [minutes]

The **line idle** command configures the specified line to automatically reset if it is idle for a set number of minutes.

Entering this command without specifying the number of minutes shows the current line idle setting.

```
NB:Top>Configure>Line> idle line01
line01 idle reset disabled
```

The NetBlazer is shipped with the line idle function disabled for all lines.

The next example shows you how to configure line01 to automatically reset when it is idle for 30 minutes.

```
NB:Top>Configure>Line> idle line01 30
NB:Top>Configure>Line> idle line01
line01 idle reset after 30 minutes
```

Setting line idle to 0 minutes disables the line idle function for the specified line as shown below.

```
NB:Top>Configure>Line> idle line01 0
NB:Top>Configure>Line> idle line01
line01 idle reset disabled
```

line list [<line-name>]

Entering a **line list** command without the optional <line-name> parameter displays a list of all the lines including the type of modem (if any) attached to the line, the characteristics group name assigned to the line, the current usage of the line, the NetBlazer command (if any) assigned to the line, and the line flags, which includes information about the current status of the line.

```
NB:Top>Configure>Line> list
Line     Modem Type Char's   Use        Command   Flags
line00   <<None>>   <<None>> login      <<None>>   520e <CTS,DCD,DSR,AF,CON,BSY>
line01   t1500      dialout  <<None>>   <<None>>   420a <CTS,DSR,AF,BSY>
```

To get additional information concerning a specific line, enter the **line list** command with a <line name> parameter. This listing shows the current line parameter settings. For definitions of each parameter refer to its associated command description.

The IP address parameter shows the IP address assigned to the line by a **ip address** <ip-addr> <line> [<port>] command. The NetBlazer's global IP address is initially assigned to all lines as a default. If you assign a line a unique ip address, you can telnet to the device attached to the specified line. This can be used to "talk" with a modem over an IP network.

Note: *Configuring a line for a raw connection automatically sets the IP address and port number assigned to a line.*

```
NB:Top>Configure>Line> list line01
Line     Modem Type Char's   Use        Command   Flags
line01   <<None>>   <<None>> login      <<None>>   5200 <AF,CON,BSY>
DSR: OFF  DCD: OFF  CTS: OFF  DSR_CHANGE: OFF
speed:           9600
callin:          allowed
autoconfigure:   on
access:          all
flow control:    auto
delay time: 0 milliseconds
dcd lock time: 0 seconds
dsr lock time: 3 seconds
dtr off time: 1100 milliseconds
locks               RI        DCD        DSR
                     0         0          0
miscellaneous: DEBUG: OFF  CONSOLE: ON  BUSY: YES  IGNORE_STATUS: OFF
callback monitor: OFF
IP address: 143.191.11.5
Terminal type: none
Rows: not set   Cols: not set   Nulls: not set
```

line modem <line-name> [<modem-name>]

If a line is set to auto-configure, the NetBlazer automatically determines the type of modem attached to the line whenever the DSR signal is raised.

The **line modem** command is used to specify the type of modem attached to a specific line if for some reason it fails to properly identify the modem type.

Before changing the modem type, make sure that the line is set to auto-configure,then try resetting the line by issuing a **line reset** command. If that doesn't work, use the **line modem** command to set the line's modem type.

```
NB:Top>Configure>Line> modem line01 t1500
NB:Top>Configure>Line> list
Line     Modem Type Char's   Use        Command   Flags
line00   <<None>>   <<None>> login      <<None>>   520e <CTS,DCD,DSR,AF,CON,BSY>
line01   t1500      dialout  <<None>>   <<None>>   420a <CTS,DSR,AF,BSY>
```

Note: *The <modem-name> parameter must point to a modem type that is defined in the MODEMCAP.TXT file.*

Issuing a **line modem** command without the <modem-name> parameter sets the line's modem type to <<None>>.

```
NB:Top>Configure>Line> modem line01
NB:Top>Configure>Line> list
Line     Modem Type Char's   Use        Command   Flags
line00   <<None>>   <<None>> login      <<None>>   520e <CTS,DCD,DSR,AF,CON,BSY>
line01   <<None>>   dialout  <<None>>   <<None>>   420a <CTS,DSR,AF,BSY>
```

line request-call <line-name> [on|off]

The **line request-call** command is used to designate a line specified by the <line-name> parameter as a call back request monitor line. This is the line that a remote user should call to request a call back.

If the optional on|off parameter is omitted the current request-call setting for the specified line is displayed.

```
NB:Top>Configure>Line> request-call line01
Request-callbacks on line 'line01' is off
```

Normally, a line's request-call setting is off. To designate a line as a call-back request line, set the line request-call parameter to on.

```
NB:Top>Configure>Line> request-call line01 on
Request-callbacks on line 'line01' is on
```

To return a call-back request line to normal use, set the line request-call parameter to off.

```
NB:Top>Configure>Line> request-call line01 off
Request-callbacks on line 'line01' is off
```

line reset <line-name>

The **line reset** command resets the line specified by the <line-name> parameter. Entering this command causes the NetBlazer to disconnect a call in progress, and then automatically reconfigure the line.

```
NB:Top>Configure>Line> reset line01
```

This is a way to reset a modem that has become unresponsive for any reason.

Note: *If the line's auto-configure is turned off, a reset line command resets the line, but does not attempt to reconfigure the line.*

line speed <line-name> <speed>

The **line speed** command is used to set the data transfer rate on a specified serial port.

The <speed> parameter may be set to any of the following standard data transfer rates: 300, 1200, 2400, 4800, 9600, 19200, 38400 or 57600.

Note: *The speed of the connected device must match the line speed.*

The following display shows you how to change the speed of line01 to 19200 bps.

```
Top>Configure>Asy> speed line01 19200
```

If you are using a terminal connected to the specified line, the command prompt displayed after this command will be scrambled. Set your terminal to match the speed of the line (e.g. 19200), and press the return key. The prompt should then be displayed correctly.

line terminal <line-name> [<terminal-type>] [-rows <rows>] [-cols <cols>]

The **line terminal** command provides information about the terminal, which is attached to the specified line, to a telnet or rlogin host. Typically a telnet server running on a UNIX host uses this information to set the "TERM" environment variable. The desirable <terminal-type> values in that case are those that are listed in the host's /etc/termcap file or /etc/terminfo database.

Entering a line terminal <line-name> command displays the current terminal characteristics for the specified line.

```
NB:Top>Configure>Line> terminal line01
Terminal type: none
Rows: not set   Cols: not set
```

The next example shows you how to set the default terminal type for line01 to vt100.

```
NB:Top>Configure>Line> terminal line01 vt100
NB:Top>Configure>Line> terminal line01
Terminal type: "vt100"
Rows: not set   Cols: not set
```

To remove a terminal type parameter use a **line terminal ""** command.

```
NB:Top>Configure>Line> terminal line01 ""
NB:Top>Configure>Line> terminal line01
Terminal type: none
Rows: not set   Cols: not set
```

The -rows and -cols parameter defines the row and column characteristics of your terminal.

```
NB:Top>Configure>Line> terminal line01 -rows 24 -cols 80
NB:Top>Configure>Line> terminal line01
Terminal type: none
Rows: 24   Cols: 80
```

line test-request <line-name>

The **line test-request** command, assigned to a line's command field by the **request-call** command, is used to call back the destination specified by the caller. You can also use this command to test the callback function for a particular destination.

line trace <line-name> [on|off]

The **line trace** command turns the line tracing function on or off for a specified line. This command is useful for monitoring chat script execution.

Entering the command without the on|off parameter, displays whether or not tracing is currently enabled for the specified line.

```
NB:Top>Configure>Line> trace line01
trace of line01 off
```

Normally, line tracing is turned off. To turn tracing on, enter the following **line trace** command.

```
NB:Top>Configure>Line> trace line01 on
```

An example of how line tracing works when starting a telnet session with a remote system is shown on the following page.The line trace function shows the data that was sent and received over the specified line while the connection was being made.

To turn line tracing off issue the **line trace** <line-name> **off** command.

```
NB:Top>Configure>Line> trace line01 off
```

```
NB:Top>Configure>Line> telnet 143.191.22.32
Type ^[ (decimal 27) <CR> to return to NetBlazer
Trying 143.191.22.32:23...
sent: ats51=5\r
got: 0\r
sent: AT\r
got: 0\r
sent: atdt 9-745-3725\r
got: 12\r
got: CARRIER ON line01\n\r
got: \r
got: Telebit's NetBlazer Version 1.5        \r
got: \n\r
got: remote login:
sent: local\r
got: local\r
got: Challenge:841114292A3C1429:\r
got: Response:
sent: B1F65F1558C85D09\r
got: B1F65F1558C85D09\r
got: \r
got: Packet mode enabled\r
Telnet session 0 connected to 143.191.22.32

Telebit's NetBlazer Version 1.5

remote login:
```

List Command

The **list** command takes you to the List command level which is comprised of various commands to display statistical information.

```
NB:Top> list
NB:Top>List> ?
Available commands:
alias           arp             asy             commands
destination     dialout         domain          dynamic
ethernet        filter          group           icmp
history         interface       ip              line
mem             pgroup          ppp             ps
rip             route           servers         session
slip            snmp            socket          stack
tcp             top>            trap            udp
up>             uptime          user            version
?
```

Commands at this level are only available to users who have system status or configure privileges.

The following list functions are also available at their corresponding command level: **alias**, **arp**, **asy**, **destination**, **dialout**, **domain**, **dynamic**, **filter**, **group**, **icmp**, **interface**, **ip**, **line**, **ppp**, **rip**, **route**, **servers**, **sessions**, **snmp**, **tcp**, **trap**, **udp**, and **user**. For information on using these list functions refer to the specific command level. For example, the **list route** command is described under the **route list** command description.

The remainder of this section documents those **list** commands which are not documented elsewhere in this manual.

list commands

The **list commands** command displays a complete list of the available NetBlazer commands.

list ethernet

The **list ethernet** command displays the status of the ethernet adapter boards installed in the NetBlazer.

list mem

The **list mem** command displays the NetBlazer memory usage status. This information is primarily used for debugging purposes.

list pgroup

The **list pgroup** command displays the process groups, e.g., processes that are grouped together by user (a user may have more than one process running at a time). This information is primarily used for debugging purposes.

list ps

The **list ps** command shows all the processes running in the NetBlazer. This information is primarily used for debugging purposes.

list slip

The **list slip** command displays SLIP statistics.

list socket [<number>]

The **list socket** command shows all the sockets (logical connections between transport services and applications) in the NetBlazer. This information is primarily used for debugging purposes.

list stack

The **list stack** command displays usage statistics for the NetBlazer memory stack. This information is primarily used for debugging purposes.

list udp

The **list udp** command displays UDP statistics.

list uptime

The **list uptime** command displays the current date and time, the number of hours and minutes the NetBlazer has been running since the last time it was started, and the number of users currently logged onto the system.

list version

The **list version** command displays information about the current version of the NetBlazer software.

Logout Command

The **logout** command is used to log out of the NetBlazer. If this command is issued from a terminal connected directly to the NetBlazer through a null modem cable, the system displays the login prompt to allow another user to log into the NetBlazer.

If the connection into the NetBlazer is via telnet, the telnet session is closed. If the connection into the NetBlazer is via dial-up modem connection, the NetBlazer instructs the modem to disconnect and hang up the phone line.

Message Command

The **message** command takes you to the Configure Message command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Message command level.

```
NB:Top> message
NB:Top>Configure>Message> ?
Available commands:
hello           list            login           shutdown
top>            up>             ?
```

Commands at this level are used to enter messages which are sent to users who are logged in to the NetBlazer.

message hello [<message>]

The **message hello** command allows you to enter a greeting message that is displayed before the login prompt. You can enter a single line message or a multi-line message consisting of up to 1024 characters.

The following example shows you how to enter a single line message.

```
NB:Top>Configure>Message> hello 'Telebit's NetBlazer Version %s'
```

The next example shows you how to enter a multi-line message.

```
NB:Top>Configure>Message> hello
Enter the message for 'hello' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: Telebit Corporation Publications Department
Line 2: NetBlazer Version %s
Line 3: Connected to %s from %s
Line 4: .
```

The %s in both of the above examples is a special variable which is replaced by the current system version number, host name, and port (in that sequence) when the message is displayed.

message list

The **message list** command displays the current contents of the hello, login, and shutdown messages.

message login [<message>]

The **message login** command is used to enter a message which is presented to every user after they log into the NetBlazer. This is similar to the motd (message of the day) used by UNIX systems.

You can enter a single line message (enclosed in quotes) or a multi-line message consisting of up to 1024 characters. An example of a multi-line message follows.

```
NB:Top>Configure>Message> login
Enter the message for 'login' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: Telebit Proprietary Information
Line 2: Unauthorized Use is Prohibited.
Line 3: .
```

message shutdown [<message>]

The **message shutdown** command is used to enter a message which is sent to every user a specified number of seconds before the NetBlazer is shut down by a **shutdown** command.

You can enter a single line message (enclosed in quotes) or a multi-line message consisting of up to 1024 characters. An example of a multi-line message follows.

```
NB:Top>Configure>Message> shutdown
Enter the message for 'shutdown' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: *** WARNING ***
Line 2: The NetBlazer will be shut down in %s seconds.
Line 3: Please logout
Line 4: .
```

The %s in the above example is a required variable that is substituted by the <interval in seconds> parameter in the **shutdown** command.

When entering a single line message that contains an exclamation point, enter it as \! to avoid confusion with a history reference.

More Command

The **more** command, located at the Configure command level, displays the content of a file or the piped output of a command one screenful (23 lines) at a time. When the screen is full you are prompted to press any key except the "q" key to display the next screenful. Pressing the "q" key stops the display and returns the command prompt.

To display the contents of a file use the **more** <file-name> command. For example, entering **more passwd** displays the contents of the PASSWD file one screenful at a time.

To display the output of a NetBlazer command one screenful at a time, use the **|more** command. For example, entering **route list | more** pipes the output of the **route list** command through **more**.

Not all commands can be piped through **more**; in particular any command that starts a session (e.g. **ping** or **telnet**) and **trace** commands.

Msglog Command

The **msglog** <message> command is used to send a message to *syslogd* on the host system specified by the **syslog host** command. This command is useful for verifying that messages from the NetBlazer are reaching the appropriate log file on the *syslog* host system.

The <message> parameter may be a single word or several words enclosed in quotes such as: “Test message from the NetBlazer.”

If the message sent by **msglog** does not reach the host log file, refer to the **syslog host** command description for troubleshooting instructions.

Passwd Command

The **passwd** [<username>] command at the Top command level allows all users (privileged or non-privileged) to change their password, provided the system administrator has enabled this feature by issuing a **user changeable on** command.

Normal (non-privileged) users can change their password by logging into the NetBlazer and entering the **passwd** command at the Top command level. They are then prompted for their old password and the new password.

```
Telebit's NetBlazer Version 1.5

NB login: carol
Password:
NB:Top> passwd
Changing password for carol
Enter old password :
New Password :
Retype Password :
```

The optional <username> parameter may only be enter by a user with sysconf privileges to change the password of another user.

```
Telebit's NetBlazer Version 1.5

NB login: root
Password:
NB:Top> passwd carol
Changing password for carol
New Password :
Retype Password :
```

This has the same result as using the **user password** command to change a user's password.

Phonelist Commands

The **phonelist** command takes you to the Configure Phonelist command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Phonelist command level.

```
NB:Top> phonelist
NB:Top>Configure>Phonelist> ?
Available commands:
delete          deny            flush           permit
list            test            top>            up>
?
```

Commands available at this level are used to maintain a list of telephone numbers that is used to accept or reject an incoming call by matching the telephone number provided by an ISDN terminal adapter attached to the syn0 line.

Initially the phone list is empty, and all incoming calls are accepted. When the first telephone number is added to the list by either the phonelist deny or permit commands, the default is changed to the opposite state. For example, if the first phone number added to the list is permitted, all other phone numbers are rejected.

phonelist delete <phonenumber...>

The **phonelist delete** command removes one or more telephone numbers from the phone list.

phonelist deny default | <phonenumber ...>

The **phonelist deny** command specifies one or more telephone numbers that the NetBlazer will not accept a call from through an ISDN terminal adapter.

If the first number added to the phone list is added with a **deny** command, the default is to permit all other phone numbers not specifically denied.

If you enter **phonelist deny default**, all numbers which are not specifically permitted in the list are denied.

phonelist flush

The **phonelist flush** command removes all entries from the phone list except the default entry.

phonelist permit default | <phonenumber ...>

The **phonelist permit** command specifies one or more telephone numbers that the NetBlazer will accept a call from through an ISDN terminal adapter.

If the first number added to the phone list is added with a **permit** command, the default is to deny all other phone numbers not specifically permit.

If you enter **phonelist permit default**, all numbers which are not specifically denied in the list are permitted.

phonelist list

The **phonelist list** command displays the phone number list.

phonelist test <phonenumber> [<phonenumber> ...]

The **phonelist test** command checks the permit/deny status of one or more phone numbers.

PPP Commands

The **ppp** command takes you to the Configure PPP command level. To view a list of the commands available at this level, type a question mark (?) at the Configure PPP command level.

```
NB:Top> ppp
NB:Top>Configure>PPP> ?
Available commands:
activeopen      close           list            passiveopen
reset           top>            trace           up>
?
```

These commands control the operation of PPP interfaces.

ppp activeopen <interface>

The **ppp activeopen** command starts PPP running on the specified interface. This command must be issued before any packets can be sent through the interface.

ppp close <interface>

The **ppp close** command terminates the link specified by <interface>. This stops the flow of packets over the link until the **ppp activeopen** command is again issued.

ppp list [<interface>]

The **ppp list** command displays information about the PPP interfaces. If <interface> is excluded, information about all PPP interfaces is displayed. If the <interface> is specified, detailed information about that interface is displayed.

ppp passiveopen <interface>

The **ppp passiveopen** command opens the specified interface in a passive state. This allows the other side of the link to start a PPP connection via the **ppp activeopen** command.

ppp reset <interface>

The **ppp reset** command puts the link specified by <interface> into a passive open state. A close connection message is not sent to the other end of the link. This stops the flow of packets over the link until a **ppp activeopen** command is issued from either end.

Reboot Command

The **reboot** [fast] command restarts the NetBlazer using the configuration information saved in the STARTUP.CNF file. If this command is enter without the **fast** option, the NetBlazer attempts to close any sessions in progress before rebooting the system. A **reboot fast** command immediately reboots the system without attempting to close the sessions.

The **shutdown** command is a more friendly way to reboot the NetBlazer in that it notifies users that you are about to reboot the NetBlazer, and gives them a chance to finish what they are doing prior to the reboot.

Before issuing a **reboot** command, verify that you have a NetBlazer system diskette installed. Also, make sure that there are no other users logged in to the NetBlazer by issuing a **who** command. If there are users logged in, you can send them a message informing them that you are about to reboot the NetBlazer by issuing a **write** or **wall** command.

RIP Commands

The **rip** command takes you to the Configure IP RIP command level. To view a list of the commands available at this level, type a question mark (?) at the Configure IP RIP command level.

```
NB:Top> rip
NB:Top>Configure>IP>RIP> ?
Available commands:
accept          add             delete          list
merge           refuse          request         timeout
top>            up>             ?
```

Commands at this level are used to configure and manage RIP (Routing Information Protocol). RIP learns about routes to other indirect networks by receiving routing table entries from other routers. The NetBlazer may participate in this process by transmitting its own routing tables using the **rip add** command.

Normally RIP accepts routing information from any router. There may be cases where you may not wish the NetBlazer to accept routing information from a particular router or routers. In that case the **rip refuse** command may be used to force the NetBlazer to ignore RIP transmissions from specific routers.

rip accept <gateway>

The **rip accept** command starts accepting routing information from a gateway from which information was previously being refused. This reverses the effects of a **rip refuse** command.

rip add <destination> <interval> [<split-horizon>]

The **rip add** command adds a destination IP address as the recipient for RIP information.

The <destination> parameter specifies the IP address of the destination router/gateway. This is usually a broadcast IP address but may be a specific IP address for another router. Make sure that any router/gateway specified is on the same network as the NetBlazer.

The <interval> parameter specifies the number of seconds between routing table transmissions to the destination router/gateway. This should be approximately every 30 seconds for an Ethernet connection.

The optional [<split-horizon>] parameter enables or disables split-horizon, a technique that improves the performance of RIP when a router disappears. The default value for this parameter is “yes” (split horizon enabled). Entering “no” disables split horizon.

rip delete <destination>

The **rip delete** command removes an IP address from the RIP destination list. The <destination> parameter specifies which IP address should be removed from the RIP destination list.

rip list

The **rip list** command displays the list of destinations to which the NetBlazer is sending routing information.

rip merge [on | off]

The **rip merge** command causes routing table entries to be compacted or merged as much as possible. This potentially reduces the size of the routing table. This function can be turned on or off by the on | off parameter. The NetBlazer is shipped with RIP merging turned off.

When you turn on RIP merging, you should flush the routing table for the merge to take effect.

rip refuse <gateway>

The **rip refuse** command causes the NetBlazer to ignore routing information sent from the specified gateway IP address. The <gateway> parameter is the IP address of the gateway from which routing information will be refused.

rip request <gateway> [<port-number>]

The **rip request** command causes the NetBlazer to immediately transmit a request for routing information to a specified gateway rather than waiting for the next scheduled RIP advertisement.

rip timeout <interface> <seconds>

The **rip timeout** command causes routes acquired on the specified interface to time out in the specified number of seconds. If the route is not updated via RIP within the time allowed the route is discarded. The default timeout value is 240 seconds.

Caution: *Make sure that an interface's RIP timeout value is greater than the RIP updating interval; otherwise the routes acquired through the interface will not be available from the time the routes time out until the next RIP update is received.*

Route Commands

The **route** command takes you to the Configure IP Route command level. To view a list of the commands available at this level, type a question mark (?) at the Configure IP Route command level.

```
NB:Top> route
NB:Top>Configure>IP>Route> ?
Available commands:
add             addprivate      change          delete
flush           list            listlocal       lookup
metric          private         public          top>
up>             ?
```

Commands at this level are used to maintain the NetBlazer routing table.

Static routes are added to the routing table by the **route add** or **addprivate** commands. The **ip address** and **ipdial** commands also add static routes to the routing table. Static routes that are designated as private are not advertised by RIP to other routers on the network.

Dynamic routes are added to the routing table through RIP when the RIP server is running.

Temporary routes are automatically added to the routing table when a dynamic interface, which has auto-routes set by the **dynamic auto-route** or **auto-arp** commands, is connected. These routes are removed when the dynamic interface connection is dropped.

route add <dest-addr>[/<bits>] <interface> [<gateway>] [<metric>]

The **route add** command is used to add a static route to the routing table.

The <dest-addr> parameter is the destination network (or host) IP address, and the optional /<bits> parameter specifies the number of bits assigned to the network and subnetwork part of the IP address. This parameter must be preceded by a slash (/) with no spaces between the destination IP address and the number of bits.

The <interface> parameter specifies the name of the interface through which the packets are routed.

The optional <gateway> and <metric> parameters specify the IP address of the gateway that the packets should be sent to for further routing, and the number of hops required to reach the destination network.

Caution: *If you add a route with the same destination address as an existing route, the new route is added to the routing table below the existing route. Because the NetBlazer uses the first route it finds that meets the destination address routing criteria, the older route takes precedence over the new route, and the new route is never used. You should delete the old route to use the new route.*

To change the interface, gateway or metric of an existing route, use the **route change** command. To change a public route (one that is advertised by RIP) to a private route, use the **route private** command. To change a private route (one that is not advertised by RIP) to a public route, use the **route public** command.

The following commands add three new static routes to the routing table.

```
NB:Top>Configure>IP>Route> add 143.191.167/24 en0
NB:Top>Configure>IP>Route> add 143.191.13/24 en0 143.191.167.254 2
NB:Top>Configure>IP>Route> addprivate default en0 143.191.167.1 2
```

The first command adds a route to a network that is connected directly to the NetBlazer through the en0 interface.

The second command adds a route to a non-adjacent network which is also accessed through the en0 interface. However, IP packets destined for this network are sent to the gateway with an IP address of 143.191.167.254. The last parameter in this command indicates that the destination network is 2 hops away.

The last command adds a private default route which is used to route packets that do not have a specific route defined.

route addprivate <dest-addr>[/<bits>] <interface> [<gateway>] [<metric>]

The **route addprivate** command is equivalent to the **route add** command with one exception: routes added to the routing table through this command are not advertised to other routers by RIP.

route change <dest-addr>[/<bits>] <interface> [<gateway>] [<metric>]

The **route change** command changes the interface, gateway and metric of an existing route specified by the <dest-addr>[/<bits>] parameter.

The <interface> parameter specifies the name of the interface through which the packets are routed.

The optional <gateway> and <metric> parameters specify the IP address of the gateway that the packets should be sent to for further routing, and the number of hops required to reach the destination network.

The following example illustrates how to change the routing for a specified destination.

```
NB:Top>Configure>IP>Route> list |grep 143.191.63.0
143.191.63.0    24  syn0                        1         0      0
NB:Top>Configure>IP>Route> change 143.191.63/24 en0 143.191.65.254 3
NB:Top>Configure>IP>Route> !l
list |grep 143.191.63.0
143.191.63.0    24  en0          143.191.65.254   3         0      0
```

The above **change command** changes the route used to send packets to the 143.191.63/24 subnetwork from sending the packets through the syn0 interface to send the packets to the 143.191.65.254 gateway through the en0 interface. Notice that this example also illustrates how to use the **grep** and **history** features to display the route for a specific destination address.

route delete <dest-addr>[/bits] [<interface>]

The **route delete** command removes a specified route from the routing table. The <dest-addr>/[<bits>] and <interface> parameters identifies which route to remove.

route flush

The **route flush** command removes the routing table entries that were created by RIP. This leaves only the static routing table entries (those entered by hand). This command is useful immediately after restructuring a network so that the NetBlazer will not have to wait for the routing table entries to expire (timeout) before learning new routes.

route list

The **route list** command displays the current routing table as illustrated in the following example:

```
NB:Top>Configure>IP>Route> list
Dest            Len Interface    Gateway          Metric  P Timer  Use
127.0.0.1       32  loopback                      1       P 0      0
143.191.20.1    32  tbupdate                      1       P 0      0
143.191.65.0    24  en0                           1         0      1
143.191.64.0    24  en0          143.191.65.254   2       R 220    0
default         0   en0          143.191.65.254   2       R 220    55
```

Dest – Indicates the IP address of the destination. The default destination is used to route any IP packet that does fall within a specific destination range.

Len – Indicates how many bits represent the network and subnetwork part of the IP address.

Interface – Identifies the interface or device through which the packets will be routed.

Gateway – Identifies any router or gateway that a packet should be sent to for further routing.

Metric – Indicates the number of hops required to reach the destination or network.

P – Indicates if a route is designated as private (P), has been acquired through RIP (R) or has been automatically (A) added to the table by a dynamic interface connection.

Timer – Indicates the number of seconds before this route is discarded. A non-zero value indicates that the route was generated by RIP and will expire when the timer reaches zero. A value of 0 indicates that this is a static route and should remain forever.

Use – Indicates the number of IP packets that have used this route since the route was created.

route listlocal

The route **listlocal** command displays a list of static routes that were not acquired through RIP.

route lookup <dest-addr>

The **route lookup** command is used to manually check for a valid route to a destination address. If a valid route exists, the NetBlazer will return the routing table entry used to route packets to that destination. If no route exists, the NetBlazer will return a message indicating that the destination is unreachable.

The <dest-address> parameter specifies the IP address of the destination network. If the NetBlazer is using the domain name system, you can enter a domain name in place of the <dest-address> parameter.

route metric <dest-addr>[/<bits>] <metric>

The **route metric** command changes the metric (hop count) of the route specified by the <dest-addr>[/<bits>] parameter.

route private <dest-addr>[/bits] <interface>

The **route private** command is used to change a public route (i.e. a route that advertised by RIP to other routers on the network) to a private route. The <dest-addr>[/<bits>] and <interface> parameters identifies which route to make private.

route public <dest-addr>[/bits] <interface>

The **route private** command is used to change a private route (i.e. a route that is not advertised by RIP to other routers on the network) to a public route. The <dest-addr>[/<bits>] and <interface> parameters identifies which route to make public.

Save Command

The **save** command saves the current configuration settings into the STARTUP.CNF and USERS.CNF files on the NetBlazer system diskette. To view the contents of these files, issue a **more** <filename> command to the NetBlazer.

Anytime you modify the NetBlazer configuration, you should make sure that you have the NetBlazer system diskette inserted in the NetBlazer, and issue a **save** command.

To save just the user information, issue a **user save** command to the NetBlazer.

Security Commands

The **security** commands control user write privileges, and the login prompt in situations where the password is not found. To view a list of the commands available at this level, type a question mark (?) at the Configure Security command level.

```
Pearl:Top>Sessions> security
Pearl:Top>Configure>Security> ?
Available commands:
login          write          top>           up>
?
```

security write [enable|disable] [user|priv|status]

The **security write** command controls whether or not a group of user is allowed to use the **write** command based on their privilege level.

Entering a **security write** command without any parameters shows the current write permissions for each group.

```
Pearl:Top>Configure>Security> write
User      :enabled
StatPriv  :enabled
ConfPriv  :enabled
```

The NetBlazer is shipped with the **write** command enabled for all groups of users. If you do not want normal user to use the **write** command enter a **write disable user** command at the Configure Security level.

```
Pearl:Top>Configure>Security> write disable user
Pearl:Top>Configure>Security> write
User      :disabled
StatPriv  :enabled
ConfPriv  :enabled
```

The next example shows you how to enable write permissions for normal users.

```
Pearl:Top>Configure>Security> write enable user
Pearl:Top>Configure>Security> write
User      :enabled
StatPriv  :enabled
ConfPriv  :enabled
```

security login [enable|disable]

The **security login** command enables or disables the prompt for a password when a user attempts to login with an unauthorized user ID.

Enter this command without specifying enable or disable shows the current setting.

```
Pearl:Top>Configure>Security> login
disabled
```

The NetBlazer is shipped with **security login** disabled, which means that a password prompt is not displayed when a user attempts to login with an invalid user ID. If you enable the security login function as shown below, the password prompt is displayed even when the user has entered an invalid user ID.

```
Pearl:Top>Configure>Security> login enable
```

The user is given another opportunity to enter a valid user ID and password.

Servers Commands

The **servers** command takes you to the Configure Servers command level where you can start or stop various servers in the NetBlazer. To view a list of the commands available at this level, type a question mark (?) at the Configure Servers command level.

```
NB:Top> server
NB:Top>Configure>Servers> ?
Available commands:
list            start>          stop>           top>
up>             ?
```

servers list

The **servers list** command displays the current status of the NetBlazer servers. The following servers are currently supported:

- **discard** – This server is used for network testing. When this server is turned on the NetBlazer accepts and acknowledges data received on TCP port 9. The data is discarded.
- **echo** – This server is used for network testing. When this server is turned on the NetBlazer accepts and acknowledges TCP segments received at port 7. The data received by the NetBlazer is echoed back to the original source.
- **finger** – The finger server enables other hosts to query the NetBlazer for information about NetBlazer users using the finger protocol.
- **raw** – The raw server allows you to establish a raw TCP connection from a host to the modem or other asynchronous device attached to a NetBlazer line.
- **ftp** – When the FTP server is running the system administrator may use FTP (File Transfer Protocol) to automatically transfer new software and configuration files to the NetBlazer from a remote host.
- **rip** – The RIP (Routing Information Protocol) server is used by the NetBlazer to learn routes to other networks from RIP advertisements sent by other routers on the network.
- **snmp** - The SNMP (Simple Network Management Protocol) server is used for network management tasks.
- **telnet** – The telnet server allows users and administrators to log into the NetBlazer remotely via the network.

- **v25bis** – This server must be turned on to use the synchronous line with a ISDN terminal adapter.

servers start <server-name>

The **servers start** command starts a servers specified by the <server-name> parameter. See **servers list** for a description of the servers supported by the NetBlazer.

servers stop <server-name>

The **servers stop** command is used to disable a server that has been started by a previous start command. The <server-name> parameter specifies which server to stop. Servers may be started and stopped at any time, but beware that if you are connected to the NetBlazer through a telnet session, there may be no way to restart the telnet server without resetting the NetBlazer.

Sessions Commands

The **sessions** command takes you to the Sessions command level. To view a list of the commands available at this level, type a question mark (?) at the Sessions command level.

```
NB:Top> session
NB:Top>Sessions> ?
Available commands:
abort           bang            binary          close
connect         continue        dial            disconnect
eol             escape          fg              finger
ftp             jobs            kill            list
ping            rlogin          send            talk
telnet          terminal        top>            up>
?
```

Commands at this level allow authorized users to start, stop and control communication sessions with local or remote machines.

session abort [<session-number>]

The **session abort** command aborts an FTP data transfer operation. It is the only command permitted during an FTP session.

If no <session-number> parameter is entered, the NetBlazer displays a list of active sessions and prompts you for the session number.

session bang <host> [<length> <count> <interval> <outstanding>]

The **session bang** command sends ICMP echo requests to the specified remote host. This command is equivalent to a **session ping** command except that instead of printing a line of statistics for each response, **session bang** only prints a single character (!). Lost or out of sequence packets are shown as a dot (.) and ICMP messages are shown as an X in the **bang** display. Refer to the **session ping** command for an explanation of the parameters used by both these commands.

session binary <session-number> [on|off]

The **session binary** command turns on or off the telnet binary option. This allows the NetBlazer to transmit and receive all 256 possible character codes. This is useful for such operations as X-terminals and DEC VT-200 series terminals. If binary mode is not enabled the NetBlazer will transfer only a subset of the 7-bit ASCII code.

If the on or off parameter is omitted, the current telnet binary setting for the specified session number is displayed.

session close [<session-number>]

The **session close** command closes the session specified by the <session-number> parameter. This is the preferred way to terminate a session.

If no <session-number> is entered, the NetBlazer displays a list of active sessions and prompts you for the session number.

session connect <destination-name>|<destination-number>

The **session connect** command starts a telnet or dial-out session with a predefined destination. The NetBlazer first looks up the destination name in the user's assigned destination group to get the connection mode (telnet or dial) and address (domain name, IP address, or telephone number), and then makes the appropriate connection.

You can enter a destination number assigned to a destination in place of the destination name parameter. To get a list of destination numbers and names, press the return key when you are at the NetBlazer Sessions command level.

```
Top>Sessions> press the Return key
Accessible sites:
1  marketing        Marketing Computer
2  databank         Data Bank Service
```

The above example shows that two sites, marketing and databank, are accessible through the **connect** command for this user. Entering **connect 1** is equivalent to entering **connect marketing**. If you get an "All sites are accessible" message, you probably do not have a destination group assigned to your user ID account.

For more information about setting up destinations used by the **connect** command see the **destination** command descriptions.

session continue [<session-number>]

The **session continue** command returns to the session specified by the <session-number> parameter.

If no <session-number> parameter is entered, the NetBlazer prompts you for the session number.

You also can use the **session fg** command to continue a session.

session dial <site-name>

The **session dial** command establishes a character mode connection with the remote destination specified by the <site-name> parameter. The NetBlazer looks up the <site-name> in the dialout list to get the information needed to make the connection.

Refer to the **dialout** command descriptions for instructions on setting up dial-out destinations.

session disconnect [<session-number>]

The **session disconnect** command terminates the session specified by the <session-number> parameter. It is intended for dial-out connections, but it can be used to close any session.

If no <session-number> parameter is entered, the NetBlazer prompts you for the session number.

session eol [standard | unix]

The **session eol** command selects the proper end-of-line processing during a telnet session when the operator presses the return key. The default setting of "standard" should be used for most systems. The "unix" option should only be used when the remote system fails to recognize the carriage return key as indicating end-of-line.

session escape [<character>]

The **session escape** command displays or changes the keystroke used in a escape sequence to interrupt a session to get back to the command prompt. The escape sequence consists of the escape character followed by a carriage return.

Changing the escape character on the NetBlazer is desirable, for example, when telnet on your computer uses the same escape character as the NetBlazer. In this case, telnet would trap the escape character which may have been intended for the NetBlazer. By changing the escape character on the NetBlazer to a value that does not match the telnet escape character, telnet allows the escape character to pass through to the NetBlazer.

Entering the command without the <character> parameter, displays the current escape character, normally ctrl-].

The optional <character> parameter specifies which control character should be used as the escape character. Be sure to enter the desired keystroke exactly as you would use it. For example, if you decide to change the escape character to ctrl-/, you should hold down the control key and press the / key.

The escape character is set on a per session basis, i.e. the escape character used by one user may differ from the escape character used by other users logged into the system.

If a user with configuration privileges issues a **configure save** command after changing the escape character, that character becomes the default escape character when the NetBlazer is rebooted. The escape character setting for all other users only applies to their current login session.

session fg [<session-number>]

Like the **session continue** command, the **session fg** (foreground) command continues a session specified by the <session-number> parameter.

If no <session-number> is entered, the NetBlazer prompts you for the session number.

session finger [<user>] [@<host>]

The **session finger** command invokes the finger client. The finger client permits a user to request information about a user at a host. The field <user> is a valid user ID on the system being queried. The field @<host> is the @ character immediately followed by an IP address or a domain name.

If <user> is specified without a host specification the local (NetBlazer) finger server is queried about that user of the local NetBlazer. If just @<host> field is specified the host being queried will usually display a list of possible users or a list of currently active users. This depends on the remote host and is not under the control of the NetBlazer.

Here are a couple of examples:

```
session finger brian@marketing.telebit.com
session finger @marketing.telebit.com
```

The first example returns information about the user brian on the host named marketing.telebit.com. The second example returns information about the current users at the host named marketing.telebit.com.

session ftp [<address>]

The **session ftp** command starts an FTP session with a host specified by the <address> parameter. The address is the host's IP address or domain name.

Once connected to the remote host, the standard set of FTP commands apply. These will vary slightly from host to host so see the documentation for the particular remote host to which the connection is being established.

Caution: *Interrupting an FTP session while copying a file to the NetBlazer diskette might make the diskette unusable. Before starting the FTP session, make sure that you have a backup of your current system diskette. Do not interrupt the FTP session while it is copying a file to the NetBlazer diskette.*

session jobs

Like the **session list** command, the **session jobs** command displays a list of sessions currently running on the NetBlazer.

session kill [<session-number>]

The **session kill** command terminates the session specified by the <session-number> parameter without notifying the remote end.

If no <session-number> is entered, the NetBlazer displays a list of active sessions and prompts you for the session number.

This command is used as a last attempt to close a session with a host that is no longer responding. Do not use this command to terminate a session with a remote host that is responding because the NetBlazer will terminate the session without telling the remote end that the connection is closed.

You should always first try to **close** a session, and only if that fails **kill** the session.

session list

The **session list** command displays a list of sessions currently running on the NetBlazer. An example of a session status list is shown below:

```
Top>Sessions> list
 #   S#  Type     Rcv-Q Snd-Q State        Remote socket     Pgroup
 0   6  Telnet      0     0 Established  napa (143.191.10.100:23)143.191.63.1:
1026
```

The # field is the session number. An asterisk (*) next to the session number denotes the current session.

The S# field identifies the socket number. This field is use primarily for debugging purposes.

The Type field identifies the type of session, i.e. Telnet, FTP, Ping, etc.

The Rcv-Q displays the number of characters received on that session but not displayed. Continuing the session displays the stored data.

The Snd-Q displays the number of bytes transmitted but not yet acknowledged by the remote host.

The State field displays the current state of the TCP session. The most significant state is "Established" indicating that a connection exists with the remote host and that information may be transferred to/from that host. A state containing the word "SYN" indicates that the NetBlazer is attempting to establish a connection. A state containing the phrase '"FIN wait" or "Time wait" indicates that the session is being closed.

The Remote socket identifies with whom the NetBlazer has established a session. The first part of the entry, that part preceding the colon, is the host name or IP address of the remote host. The second part of the entry, the part following the colon, is the remote TCP port number and can be used to determine which remote application is involved in the session.

session ping <host> [<length> <count> <interval> <outstanding>]

The **session ping** command sends ICMP echo requests to the specified remote host. The NetBlazer then measures the amount of time required for the remote host to send a response.

The <host> parameter may be either a domain name or an IP address.

The optional <length> parameter is used to pad the request to a larger size specified in bytes. This parameter must be greater than 32 bytes (the minimum length allowed) to take effect. However, a value less than 32 can be used as placeholder in the command.

The optional <count> parameter specifies the number of ICMP requests to send to the remote host. If this parameter is not specified or 0 is entered as a placeholder, the NetBlazer will continue issuing requests to the host until the ping session is stopped by pressing ctrl-C, or by issuing a close, disconnect, or kill command.

The optional <interval> parameter specifies in seconds the time to wait between sending ICMP echo requests. This value may also be set to 0 and the NetBlazer will send pings with 10 ms separation (approximately 100 pings per second).

The optional <outstanding> parameter specifies the number of echo requests (pings) that may be transmitted before the NetBlazer stops and waits for their responses.

The ability to set the ping interval to 0 (10 ms between pings) and to set the number of outstanding pings is a useful tool to place a significant load on a network without overloading the network.

session rlogin [-b|-8] [-noesc] [-l username] [<address>] [<port>]

The **session rlogin** command is used to establish an interactive session on remote host using current NetBlazer user ID as remote host login ID. If remote host has an entry for NetBlazer in /etc/hosts.equiv, and NetBlazer user ID matches remote host user ID, user is simply logged into remote host without requiring additional user name and password authentication on remote host.

The -b option starts a binary telnet session. This has the same result as escaping out of a rlogin session, turning on binary for that session, and then continuing the session.

The -8 option starts an rlogin session using 8-bit data in non-binary mode.

The -noesc option disables escape processing. This prevents the NetBlazer from enter command mode when an escape sequence is entered. Instead, the escape sequence is passed through to the remote end.

The -l option is used to overwrite the default user name. This is normally used to specify a different user name when logging onto a remote host that does not recognize the users NetBlazer login.

Note: *The -noesc option may not be used if there are other sessions open for the same login (pgroup). Other sessions for the same login must be closed before an* ***rlogin -noesc*** *command will be accepted.*

The only way to return to the NetBlazer's command mode from an rlogin session that was started with the -noesc option is to close the session at the remote end (e.g. logout).

The optional <port> parameter is used to start a telnet with the host system using a port number other than telnet's well known port number 23.

session send <session-number> ao|ayt|brk|ec|el|esc|ga|ip|nop|synch

The **session send** command sends the special character sequences defined below to a remote host.

ao	Sends the telnet ao (abort output) sequence, which should cause the remote system to flush all output from the remote system to the user's terminal.
ayt	Sends the telnet ayt (are you there) sequence, to which the remote system may or may not choose to respond.
brk	Sends the telnet brk (break) sequence, which may have significance to the remote system.
ec	Sends the telnet ec (erase character) sequence, which should cause the remote system to erase the last character entered.
el	Sends the telnet el (erase line) sequence, which should cause the remote system to erase the line currently being entered.
esc	Sends the current telnet escape character. The CR or LF that would be expected by a NetBlazer as part of the escape sequence are not sent by this command; they may be sent by the user if required.
ga	Sends the telnet ga (go ahead) sequence, which likely has no significance to the remote system.
ip	Sends the telnet ip (interrupt process) sequence, which should cause the remote system to abort the currently running process.
nop	Sends the telnet nop (no operation) sequence.

synch	Sends the telnet synch sequence. This sequence causes the remote system to discard all previously typed but not yet read input. This sequence is sent as TCP urgent data and may not work if the remote system is a 4.2 BSD system -- if it does not work, a lower case "r" may be echoed on the terminal.

session talk [<line-name>]

The **session talk** command is for debugging purposes. It allows you to connect to an asynchronous line and issue modem commands to an attached modem.

The <line-name> parameter must be one of the lines listed by a **line list** command that is connected to a modem and not in use.

To stop talking to the modem, send the escape character (control-]) to the NetBlazer, and issue a **close**, **disconnect** or **kill** command.

session telnet [-b|-8|-raw] [-noesc] [<address>] [<port>]

The **session telnet** command is used to establish a telnet session with a host specified by the <address> parameter. The address is the host's IP address or domain name.

The -b option starts a binary telnet session. This has the same result as escaping out of a telnet session, turning on binary for that session, and then continuing the session.

The -8 option starts a telnet session using 8-bit data in non-binary mode.

The -raw option implies the -b and -noesc options, and eliminates all Telnet IAC, CR/LF/NUL, option, and suboption processing.

The -noesc option disables escape processing. This prevents the NetBlazer from enter command mode when an escape sequence is entered. Instead, the escape sequence is passed through to the remote end.

Note: *The -noesc option may not be used if there are other sessions open for the same login (pgroup). Other sessions for the same login must be closed before a **telnet -noesc** command will be accepted.*

The only way to return to the NetBlazer's command mode from a telnet session that was started with the -noesc option is to close the session at the remote end (e.g. logout).

The optional <port> parameter is used to start a telnet with the host system using a port number other than telnet's well known port number 23.

session terminal [<terminal-type>] [-rows <rows>] [-cols <cols>]

The **session terminal** command sets or displays the terminal characteristics for the current login. Typically a telnet server running on a UNIX host uses this information to set the "TERM" environment variable. The desirable <terminal-type> values in that case are those that are listed in the host's /etc/termcap file or /etc/terminfo database.

Entering a session terminal command without any parameters displays the current login terminal characteristics.

```
NB:Top>Sessions> terminal
Terminal type: none
Rows: not set   Cols: not set
```

The next example shows you how to set the login terminal type to vt100.

```
NB:Top>Sessions>  terminal vt100
NB:Top>Sessions> terminal
Terminal type: "vt100"
Rows: not set   Cols: not set
```

To remove a terminal type parameter use a **session terminal ""** command.

```
NB:Top>Sessions> terminal ""
NB:Top>Sessions> terminal
Terminal type: none
Rows: not set   Cols: not set
```

The -rows and -cols parameter defines the row and column characteristics of your terminal.

```
NB:Top>Sessions> terminal -rows 24 -cols 80
NB:Top>Sessions> terminal
Terminal type: none
Rows: 24   Cols: 80
```

Setconsole Command

The **setconsole** [internal|<line-name>] command selects the line where the NetBlazer displays debugging information.

If **setconsole internal** is entered, the debugging messages are sent to an internal monochrome display adapter, provided one is present.

If **setconsole** <line-name> is entered, the debugging messages are displayed on a terminal attached to that line (line00, line01, etc.).

Note: *The NetBlazer is shipped from the factory configured to send console message to line00, which is intended to only be used with an asynchronous terminal. Modems should be connected to the multi-port lines, then to line01. If you must attach a modem to line00, you should issue a **setconsole internal** command to the NetBlazer in order to prevent console messages from adversely affecting the operation of the modem.*

Shutdown Command

The **shutdown** [<interval in seconds>] command sends a shutdown message to all users warning them that the NetBlazer will be shut down in the number of seconds specified by the [<interval in seconds>] parameter.

```
NB:Top> shutdown 60

Message from root on 192.124.95.18:11329


****System will go down in 60 seconds!!****


Message from root on 192.124.95.18:11329


****System will go down in 30 seconds!!****


Message from root on 192.124.95.18:11329


****System will go down in 15 seconds!!****


Message from root on 192.124.95.18:11329


****System will go down in 7 seconds!!****

Rebooting will immediately take down the system.
There are no other users logged on.
Are you sure you want to do this? (y/n)
```

Notice in the above example that the NetBlazer repeats the shutdown message in half-live intervals. When the shut down period has timed out, the NetBlazer gives you an opportunity to reboot or abort the shutdown process.

If you do not specify a shut down interval the NetBlazer uses the default value of 20 seconds. The message shown in the above example is the default message used by the NetBlazer. You can change the message sent by the shutdown command via the **message shutdown** command.

SNMP Command

The **snmp** command takes you to the Configure SNMP command level. To view a list of the commands available at this level, type a question mark (?) at the Configure SNMP command level.

```
NB:Top> snmp
NB:Top>Configure>SNMP> ?
Available commands:
get-community   list          set-community   sysContact
sysDescr        top>          trap>           trap-community
up>             ?
```

This set of commands controls the configuration of the SNMP (Simple Network Management Protocol) server.

snmp get-community [<community-name>]

The **snmp get-community** command sets the community name used by management stations to get information from the NetBlazer. Think of community name as a password needed to fetch statistical and other information from the NetBlazer. The default get-community name is "public."

snmp list

The **snmp list** command shows the status and configuration settings for the NetBlazer.

snmp set-community [<community-name>]

The **snmp set-community** command sets the community name used by management stations to set or change the configuration of the NetBlazer. Think of community name as a password needed to modify the configuration of the NetBlazer. For security reasons this community name should not be freely distributed. The default value of set-community name is "private."

snmp sysContact <sysContact-string>

The **snmp sysContact** command sets the string that is transmitted to a remote management station when the sysContact variable is queried. This should be set to the name and phone number of the person responsible for the operation of the NetBlazer. The default value for <sysContact-string> is “sysContact not set.”

snmp sysDescr <description-string>

The **snmp sysDescr** command sets the string that will be transmitted to a remote management station when the sysDescr variable is queried.

snmp trap-community <community_name>

The **snmp trap-community** command sets the community name used by the NetBlazer when sending trap event information to a management station. Think of the trap community name as a password sent to the remote management station. For security reasons, this community name should not be freely distributed.

SNMP Trap Commands

The **snmp trap** command takes you to the Configure SNMP Trap command level. To view a list of the commands available at this level, type a question mark (?) at the Configure SNMP Trap command level.

```
NB:Top> snmp trap
NB:Top>Configure>SNMP>Trap> ?
Available commands:
add              authentication  delete           list
startup          top>            up>              ?
```

Commands at this level are used to maintain a list of IP addresses for up to five remote management stations, and to enable or disable sending specific SNMP traps.

snmp trap add <manager-address>

The **snmp trap add** command adds an IP address to the list of IP addresses for the SNMP management stations.

The <manager-address> parameter may be either a domain name or an IP address.

snmp trap authentication on | off

The **snmp trap authentication** command enables/disables the transmission of an authentication trap by the NetBlazer when a query is received that has an invalid community name.

snmp trap delete <manager-address>

The **snmp trap delete** command removes an IP address from the list of IP addresses for the SNMP management stations. The <manager-address> parameter may be either a domain name or an IP address.

snmp trap list

The **snmp trap list** displays the current status of the SNMP traps.

snmp trap startup on | off

The **snmp trap startup** command enables/disables the transmission of a startup trap when the NetBlazer SNMP server is turned on.

Source Command

The **source** <command-file> command automatically runs a series of commands contained in the file specified by the <command-file> parameter.

To use this command, you must first create a command file using a text editor on an external system, and then copy the file to the NetBlazer system diskette. Two ways to do this are:

- Create the command file using the text editor of host system that can be reached by the NetBlazer through the network, then use FTP to copy the file to the NetBlazer diskette.
- Create the command file on an MS-DOS system, which supports 3.5 inch high-density diskettes, then use the MS-DOS copy utility to copy the file onto the NetBlazer diskette.

Commands in the command file must include all required parameters and run from the Top command level.

Do not use commands, like **ipdial**, which require operator input. Also, avoid using commands, such as **reboot** or **shutdown**, which may have adverse results.

The order that the commands are entered is important. Since the commands are executed in sequence starting at the top and working down through the file, commands that define parameters used by other commands should be entered earlier in the list.

Syslog Commands

The **syslog** command takes you to the Configure Syslog command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Syslog command level.

```
NB:Top> syslog
NB:Top>Configure>Syslog> ?
Available commands:
host            console         terminal        buffer
list            top>            up>             ?
```

These commands are used to designate where NetBlazer syslog messages are sent. The NetBlazer is shipped from the factory with syslog turned off (i.e. no sylog messages are reported). You can configure the NetBlazer to send syslog messages to a UNIX host, the NetBlazer console, your login terminal, and/or an internal buffer in the NetBlazer's memory. Messages stored in memory can be listed to a disk file or to your terminal screen.

Each **syslog** destination command except **syslog host** allows you to filter messages sent to the destination based on the severity level of the messages. Filtering of syslog messages sent to a UNIX host is controlled by the *syslog.conf* configuration file on the UNIX system.

Recognized message severity levels are listed below in descending order of severity.

Level	Message Severity	
0	emerg	panic conditions requiring immediate attention
1	alert	conditions that should be corrected immediately
2	crit	critical error conditions
3	err	other error conditions
4	warning	warning messages
5	notice	non-error conditions requiring special handling
6	info	informational messages
7	debug	messages that are used only for debugging

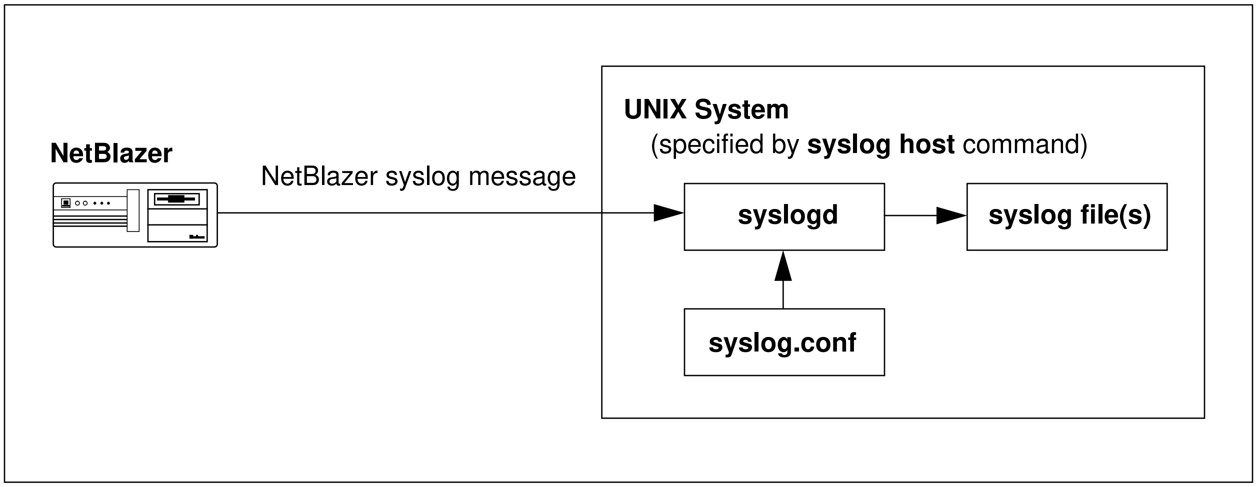
The <level> and <=level> parameters in the **syslog** destination commands determine which syslog messages are sent to the specified destination.

If you specify a level without the equal sign (=), all messages at that level and above are sent to the syslog message destination. For example, a **syslog console on 5** command directs all syslog messages with a severity level of 0 through 5 to the console.

If you specify a level with the equal sign (=), only messages at that level are sent to the syslog message destination (e.g. **sylog console on =5** sends only syslog message with a severity level equal to 5 to the console).

syslog host off | on [<loghostname>] [<facility>]

The **syslog host** command controls sending NetBlazer activity information to a UNIX system's *syslogd* daemon, which then forwards this information to the appropriate system log files and/or users. The */etc/syslog.conf* file on the UNIX system controls where the NetBlazer messages are sent.



Also see: For information on setting up your UNIX host system, see the following UNIX manual pages: *syslog*, *syslogd*, *syslog.conf*.

The <loghostname> parameter identifies which UNIX system to send the NetBlazer activity information. This parameter may be an IP address or a domain host name.

The <facility> parameter specifies which *syslog* facility to use when sending accounting log messages to a host system. Refer to the *syslog.conf* description in your UNIX Manual for a list of recognized *syslog* facilities.

The **syslog host off** command stops sending syslog messages to a UNIX system.

```
NB:Top>Configure>Syslog> host off
Syslog on host is now off
```

If you enter a **syslog host on** command without specifying a host or facility, the NetBlazer uses the host name and/or facility settings previously defined by the **syslog host on** [<loghostname>] [<facility>] command.

```
NB:Top>Configure>Syslog> host on
sending syslog messages to 143.191.10.100, facility = local0
```

The above example shows the default host name and facility that the NetBlazer uses if these two parameters have never been set.

The **syslog host on** <loghostname> <facility> command allows the NetBlazer to start sending syslog messages to a specified host and facility.

```
NB:Top>Configure>Syslog> host on byron local0
sending syslog messages to byron(143.191.10.243), facility = local0
```

You can verify that syslog messages from the NetBlazer are reaching the right file on the host system by issuing a **msglog** <message> command. The <message> parameter may be a single word or several words enclosed in quotes as shown in the following example.

```
NB:Top>Configure>Syslog> msglog "message from NetBlazer"
```

If the message sent by the **msglog** command does not reach the host log file, try the following:

1. Enter a **syslog list** command to verify that the the NetBlazer's syslog messages are being sent to the correct UNIX host.

2. Enter a **trace** <interface> **101** command where <interface> specifies the name of the interface that syslog messages will be sent through. Then, send your syslog message again to verify that the message is going out.

   If you do not see trace information about the message, verify that you have the interface and route set correctly to reach the host.

3. Verify that your */etc/syslog.conf* configuration file on the host is set up correctly to route messages from the NetBlazer to the appropriate log file. You should have a line in your */etc/syslog.conf* file similar to the following:

   ```
   local0.debug            /var/log/netblazer
   ```

4. Make sure that the log file, /var/log/netblazer in the above example, is present on your host system.

5. Make sure that your host system is only running one *syslogd* process.

6. Try running *syslogd* with the *-d* (debug) option.

syslog console off | on [<level>|<=level>]

The **syslog console on** [<level>|<=level>] command controls sending syslog messages to the NetBlazer's console, defined by the **setconsole** command. The optional <level> and <=level> parameters specify the severity level(s) of messages that are sent to the console. Refer to page 1 for a description of the syslog message severity levels.

If you do <u>not</u> include a message severity level in the **syslog console on** command, the NetBlazer uses the default value of 7, which passes all syslog messages to the console.

```
NB:Top> syslog
NB:Top>Configure>Syslog> console on
NB:Top>Configure>Syslog> list
syslog to host is off
syslog to console is on ; level <= 7
syslog to terminal is off
syslog to buffer is off
(Levels: 0-emerg,1-alert,2-crit,3-err,4-warning,5-notice,6-info,7-debug)
```

To stop sending syslog messages to the console, enter a **syslog console off** command.

```
NB:Top>Configure>Syslog> console off
Syslog on console is now off
```

Note: *To change the severity level of messages sent to the console, you must turn off syslog to the console.*

The next example shows you how to start sending all syslog messages except debug messages to the console.

```
NB:Top>Configure>Syslog> console on 6
NB:Top>Configure>Syslog> list
syslog to host is off
syslog to console is on ; level <= 6
syslog to terminal is off
syslog to buffer is off
(Levels: 0-emerg,1-alert,2-crit,3-err,4-warning,5-notice,6-info,7-debug)
```

Notice that the NetBlazer now is configure to send all syslog messages with a severity level less than or equal to 6 to the console. Debug messages (level 7) are not sent to the console.

syslog terminal off | on [<level>|<=level>]

The **syslog terminal on** [<level>|<=level>] command controls sending syslog messages to your login terminal. The optional <level> and <=level> parameters specify the severity level(s) of messages that are sent to the console. Refer to page 1 for a description of the syslog message severity levels.

Note: *The syslog to terminal function is only active for your current login session, and is automatically turned off when you logout.*

If you do not include a message severity level in the **syslog terminal on** command, the NetBlazer uses the default value of 7, which passes all syslog messages to your terminal.

```
NB:Top> syslog
NB:Top>Configure>Syslog> terminal on
NB:Top>Configure>Syslog> list
syslog to host is off
syslog to console is off
syslog to terminal is on ; level <= 7
syslog to buffer is off
(Levels: 0-emerg,1-alert,2-crit,3-err,4-warning,5-notice,6-info,7-debug)
```

To stop sending syslog messages to your terminal, enter a **syslog terminal off** command.

```
NB:Top>Configure>Syslog> terminal off
Syslog on terminal is now off
```

Note: *To change the severity level of messages sent to your terminal, you must turn off syslog to the terminal.*

The next example shows you how to start sending all syslog messages except debug messages to your terminal.

```
NB:Top>Configure>Syslog> terminal on 6
NB:Top>Configure>Syslog> list
syslog to host is off
syslog to console is off
syslog to terminal is on ; level <= 6
syslog to buffer is off
(Levels: 0-emerg,1-alert,2-crit,3-err,4-warning,5-notice,6-info,7-debug)
```

Notice that the NetBlazer now is configure to send all syslog messages with a severity level less than or equal to 6 to your terminal. Debug messages (level 7) are not sent to the terminal.

syslog buffer off | on [<level>|<=level>] | list [<file name>] | size <sizeK>

The **syslog buffer** command is used to start or stop sending syslog messages to a buffer in NetBlazer's internal memory, set the size of the buffer, and list the current contents of the buffer to your screen or to a file on diskette.

The default size of the syslog buffer is 20K bytes. To increase or decrease the amount of memory allocated to the syslog buffer, issue a **syslog buffer size** <sizeK> command where <sizeK> is the number of Kbytes to use for the syslog buffer.

```
NB:Top> syslog
NB:Top>Configure>Syslog> buffer size 40
```

The **syslog buffer on** [<level>|<=level>] command starts sending syslog messages to an internal memory buffer. The optional <level> and <=level> parameters specify the severity level(s) of messages that are sent to the console. Refer to page 1 for a description of the syslog message severity levels.

If you do <u>not</u> include a message severity level in the **syslog buffer on** command, the NetBlazer uses the default value of 7, which sends all syslog messages to the buffer.

```
NB:Top>Configure>Syslog> buffer on
NB:Top>Configure>Syslog> list
syslog to host is off
syslog to console is off
syslog to terminal is off
syslog to buffer is on ; level <= 7
Syslog internal buffer size = 40K
(Levels: 0-emerg,1-alert,2-crit,3-err,4-warning,5-notice,6-info,7-debug)
```

You can list the contents of the buffer to your terminal screen or to a file on diskette.

```
NB:Top>Configure>Syslog> buffer list
Thu Nov 14 20:58:21 1991 - Exit Pid 27 CPU  0.0 [telser_in]
Thu Nov 14 20:58:21 1991 - Exit Pid 26 CPU  1.6 [in_telnet]
Thu Nov 14 20:58:21 1991 -  bac on 143.191.10.29:1054 at Nov 14 20:54 for 227 s
econds
NB:Top>Configure>Syslog> buffer list message.log
NB:Top>Configure>Syslog> more message.log
Thu Nov 14 20:58:21 1991 - Exit Pid 27 CPU  0.0 [telser_in]
Thu Nov 14 20:58:21 1991 - Exit Pid 26 CPU  1.6 [in_telnet]
Thu Nov 14 20:58:21 1991 -  bac on 143.191.10.29:1054 at Nov 14 20:54 for 227 s
econds
```

Caution: *If you list the buffer to an existing file name, the current contents of that file are overwritten. Choose a file name that is not used by the NetBlazer.*

To stop sending syslog messages to the buffer, enter a **syslog buffer off** command.

```
NB:Top>Configure>Syslog> buffer off
Syslog on buffer is now off
```

Note: *Turning off syslog to the buffer deletes the current contents of the buffer.*

syslog list

The **syslog list** command displays the current status for each of the syslog destinations.

```
NB:Top>Configure>Syslog> list
sending syslog messages to 143.191.10.100, facility = local0
syslog to console is off
syslog to terminal is on ; level <= 6
syslog to buffer is on ; level <= 4
Syslog internal buffer size = 20K
(Levels: 0-emerg,1-alert,2-crit,3-err,4-warning,5-notice,6-info,7-debug)
```

The above example shows that all NetBlazer syslog messages are sent to a UNIX host, messages with a severity level of 0 through 6 are sent to the login terminal, and messages with a severity level of 0 through 4 are saved in the NetBlazer's internal memory buffer.

TCP Commands

The **tcp** command takes you to the Configure IP TCP command level. To view a list of the commands available at this level, type a question mark (?) at the Configure IP TCP command level.

```
NB:Top> tcp
NB:Top>Configure>IP>TCP> ?
Available commands:
irtt            list            mss             reset
top>            up>             window          ?
```

These commands are used to configure and manage TCP (Transmission Control Protocol).

tcp irtt [<int-rnd-trip-time>]

The **tcp irtt** command sets or displays the initial round trip time in milliseconds used by TCP when opening a new TCP session with a remote host. If this command is entered without a parameter, the current IRTT value is displayed.

tcp list [<tcb>]

The **tcp list** command produces a list of TCP statistics including an entry for each TCP connection. The connection display is preceded by the address of the TCP control block (tcb). Here is a sample display:

```
( 1)tcpRtoAlgorithm            4     ( 2)tcpRtoMin                  10
( 3)tcpRtoMax         2147483647     ( 4)tcpMaxConn         4294967295
( 5)tcpActiveOpens             1     ( 6)tcpPassiveOpens             9
( 7)tcpAttemptFails            0     ( 8)tcpEstabResets              0
( 9)tcpCurrEstab               1     (10)tcpInSegs                1687
(11)tcpOutSegs              1571     (12)tcpRetransSegs             68
    &TCB Rcv-Q Snd-Q  Local socket         Remote socket          State
   e9ab0     0     0  0.0.0.0:21            0.0.0.0:0            Listen (S)
```

If a particular tcb is specified, all the information about that particular TCP connection is displayed. Here is a sample display of the information about the tcb at address e9ab0:

```
Local: 0.0.0.0:21 Remote: 0.0.0.0:0 State: Listen
    Init seq    Unack     Next Resent CWind Thrsh  Wind  MSS Queue     Total
Send:      0        0        0      0   512 65535     1  512     0         0
Recv:      0                 0      0              2048          0         0
Timer stopped SRTT 5000 ms Mean dev 0 ms
```

tcp mss [<max-seg-size>]

The **tcp mss** command sets or displays the maximum segment size for a TCP session. If this command is entered without a parameter, the current MSS value is displayed. MSS is the number of bytes that a NetBlazer will tell a remote host that it may send as a single segment.

tcp reset <tcb>

The **tcp reset** command resets the TCP session specified by the <tcb> parameter.

The <tcb> parameter is derived from the **tcp list** command.

The **session kill** command is a better choice for killing a session where the remote host is no longer responding.

tcp window [<max-window-size>]

The **tcp window** command sets or displays the maximum window size that will be presented to a remote TCP during a session. The actual window size varies as a function of the TCP protocol, but it will never exceed the value set by this command. If this command is entered without a parameter, the current maximum window size is displayed.

Time Command

The **time** command allows you to set the date and time used by the NetBlazer. When this command is entered, the NetBlazer will prompt you for the current date and time as shown below.

```
Current time is Wed Nov 21 08:33:39 1990
Enter the year [90]:
Enter the month number (0 to 12) [11]:
Enter the day of the month [21]:
Enter the hour (0 to 23) [8]:
Enter the minute [33]:
Enter the current second [39]:
Verify: Wed Nov 21 08:33:39 1990
```

Entering a **time list** command displays the current date and time.

TTY Command

The **tty** command displays the line or port that you are currently logged in through.

```
NB:Top> tty
143.191.12.26:10935
```

To get a list of users that are currently logged into the NetBlazer, use the **who** command.

User Command

The **user** command takes you to the Configure User command level. To view a list of the commands available at this level, type a question mark (?) at the Configure User command level.

```
NB:Top> user
NB:Top>Configure>User> ?
Available commands:
add             changeable      crypto          delete
destgroup       epassword       list            load
password        privilege       restrict        save
top>            up>             ?
```

Commands at this level are used to maintain a list of users authorized to log into the NetBlazer.

Every user that will be logging in to the NetBlazer, either through a telnet session or by calling in from a remote terminal, must have an assigned user ID and password. User records may be entered manually, via the **user add** command, or by loading a copy of a UNIX password file, as described under the **user load** command.

Before adding users to the list, consider setting up user destination groups for the users. Refer to the **destination** command description for information on creating destination groups.

You can create login aliases that automatically invokes a command, such as telnet to specific host, when a user logs in to the NetBlazer. See the **alias add** command description for additional information about this feature.

user add [<userid>]

The **user add** command adds a new user ID to the list of authorized users.

To add a new user, enter the command followed by a single word user ID. For example:

```
NB:Top>Configure>User> add mike
NB:Top>Configure>User> list
USERID      PASSWORD          CRYPTO          USE       PRIVS   DEST-GROUP
mike                                          login
root        abmKjJ.a/qDYk                     login     SC
```

This command creates a new user, named mike, with no privileges or restrictions.

If you enter the **user add** command without any parameters, the NetBlazer will begin a dialog, prompting you for the following information.

```
NB:Top>Configure>User> add
Enter the uname (1–16 chars): carol
Want to enter a password [y]? y
Enter the password (no echo):
Re-enter the password:
```

Enter a single word name that uniquely identify this user. Enter a password that the user can remember but is not easily guessed. Since the password is not displayed as you enter it, you are asked to enter the password again for confirmation.

```
Issue a crypto handshake [n]?
```

Crypto handshaking is normally used for dynamic interface packet mode connections. Unless the user has some way to generate the appropriate response to a crypto challenge, press the return key to accept the default "n" response.

```
Allow system configuration privileges (n/y) [n]?
Allow system status privileges (n/y) [n]?
```

Only a few authorized system administrators should be granted system configuration or status privileges. Unless this user is a system administrator, press the return key to accept the default "n" response for each of these prompts.

```
Restrict dialout privileges (n/y) [n]?
Restrict telnet privileges (n/y) [n]?
```

Answering "y" to these prompts limit the user's dialout or telnet capabilities to only those destinations which are in the user's assigned destination group. If the user is not assigned a destination group, then the dialout and/or telnet restrictions applies to any dialout/telnet session.

Pressing the return key or answering "n" allows the user to dial out or telnet to any destination.

```
Assign to a destination group? (y/n) [n]
```

Answer "y" to this prompt only if you have created a destination group for this user. Otherwise press the return key to accept the default "n" response. You can assign the user a destination group later with the **user destgroup** command.

```
User: carol              privs:     dest-group:
Okay (yes|no|quit) [y]?
```

The NetBlazer ends the dialog by displaying your responses and asking if you really want to add this user. Press the return key to add the user.

user changeable [on|off]

The **user changeable** command determines whether or not non-privileged user are allowed to change their password. Entering this command without a on or off parameter shows the current state of this function.

```
NB:Top>Configure>User> changeable
Normal users cannot change their passwords
```

The NetBlazer is shipped to not allow normal (non-privileged) users to change their passwords.

Issuing a **user changeable on** command allows <u>all</u> NetBlazer users to change their passwords.

```
NB:Top>Configure>User> changeable on
Normal users can change their passwords
```

Issuing a **user changeable off** command forbids non-privileged NetBlazer users from changing their passwords.

```
NB:Top>Configure>User> changeable off
Normal users cannot change their passwords
```

user crypto <userid> [<crypto-key>]

The **user crypto** command assigns a crypto key to a specified user. Entering this command without a crypto key removes the crypto key for that user.

A crypto key should only be assigned to dynamic interface packet mode user IDs or dial-in users who have the capability to respond to a crypto challenge.

user delete [<userid>]

The **user delete** command removes a user ID from the list of authorized users.

user destgroup <userid> <destination-group>

The **user destgroup** command assigns a destination group to a specified user ID. The destination group must be previously created by a **destination add** command.

Assigning a user to a destination group allows you to restrict that user's telnet and/or dialout access to only those destination that are defined in the user's destination group. Refer to the **user restrict** command description for instructions on how to restrict a users telnet and/or dialout privileges.

Another advantage of assigning users to a destination group is that users can use the **session connect** command to select a telnet or dialout destination from a list of assigned destinations.

user epassword <userid> <encrypted-password>

The **user epassword** command is used by the NetBlazer to save the encrypted password assigned to a specified user to diskette. The command is described here for reference purposes only.

user list

The **user list** command displays a list of authorized users. Included in this listing is the encrypted password, crypto key, privileges and destination group assigned to each authorized user.

user load <passwd>

The **user load** command loads user records into the NetBlazer from a UNIX password file that has been copied to the NetBlazer diskette through an FTP session.

Caution: *When a password file is loaded into the NetBlazer, the user records in the password file overwrite any current user records that have userids that are identical to the password file's userids.*

Before loading a password file into the NetBlazer, make sure that you understand how the NetBlazer will treat each record in the password file. Refer to the *NetBlazer Configuration Guide* for a detailed description of the NetBlazer PASSWD file format.

If you want a group of users to be automatically assigned to a particular destination group, make sure that a destination group with a matching gid number exists.

Perform the following steps to load user information from a password file into the NetBlazer.

1. Using FTP, transfer a copy of the password file from your UNIX system to the NetBlazer.

   If you do not want to overwrite your current PASSWD, choose a name other than PASSWD when transferring a UNIX password file to the NetBlazer diskette.

2. Go to the Configure User command level, and enter a **load** <filename> command as shown below.

```
NB:Top> user
NB:Top>Configure>User> load passwd.01
```

   This loads and merges the user information from the password file (passwd.01 in the example) with the current user records in memory.

   If you have more than one password file to load, repeat steps 1 and 2 for each additional password file.

3. Verify that the root user ID has system configuration privileges. The root record should have "SC" listed in its privileges field. If it doesn't, assign **sysconf** and **sysstat** privileges to root.

4. Verify that you can still log into the NetBlazer as root by starting a telnet session to the NetBlazer's loopback address (127.0.0.1).

   *If you cannot log in as root, escape out of the telnet session, and change the root password by issuing a **password root** <password> command.*

5. Save the user information to disk by issuing a **save** command at the Configure User command level.

user password <userid> [<password>]

The **user password** command assigns a password to a specified user ID.

If the optional <password> parameter is not entered, you are prompted for the new password.

```
NB:Top>Configure>User> password carol
Changing password for carol
New Password :
Retype Password :
```

Pressing return at both prompts removes the password assigned to the specified user.

user privilege <userid> [-]sysconf | [-]sysstat

The **user privilege** command grants or removes system configuration and/or status privileges for a specified user. Entering sysconf or sysstat after the user ID grants the user system configuration or status privileges. If either of these two parameters are preceded by a “-” the privilege is removed for the specified user.

user restrict <userid> [-]dial | [-]telnet

The **user restrict** command controls whether or not a specified user is restricted to the destinations contained in the user’s assigned destination group.

Entering dial or telnet after the user ID restricts the user’s dialout or telnet privileges to the destination defined in the user’s destination group. Preceding either of these two parameters with a “-” removes the specified restriction.

If the user is not assigned a destination group, then the dialout and/or telnet restrictions applies to any dialout/telnet session.

user save

The **user save** command saves the current list of users to the PASSWD file on the NetBlazer system diskette.

Wall Command

The **wall** [<message>] command immediately sends a message to all users currently logged in to the NetBlazer. You can enter a single line message or a multi-line message consisting of up to 1024 characters.

The following example shows you how to enter a single line message.

```
NB:Top> wall "Hello NetBlazer users."

Message from root on 192.124.95.18:11329
Hello NetBlazer users.
```

When entering a single line message that contains an exclamation point, you must enter it as \! to avoid confusion with a history reference.

The next example shows you how to enter a multi-line message.

```
NB:Top> wall
Enter the message for 'wall' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: The NetBlazer will be shut down 30 minutes from now.
Line 2: Please finish what you are doing and log out.
Line 3: .

Message from root on 192.124.95.18:11329
The NetBlazer will be shut down 30 minutes from now.
Please finish what you are doing and log out.
```

Note that the **shutdown** command also sends a warning message to notify all users that the NetBlazer is being shut down.

To send a messages to specific users, use the **write** command.

Who Command

The **who** command displays a list of users who are currently logged into the NetBlazer.

```
NB:Top> who
user      time          from
root      Dec 16 14:41  line00
root      Dec 17 14:03  143.191.12.26:10935
```

If the same account is used multiple times as shown above, and you want to know which line or port you are logged in on, you can enter a **who am i** command to get just your login information.

```
NB:Top> who am i
user      time          from
root      Dec 17 14:03  143.191.12.26:10935
```

Write Command

The **write** <user> [<location>] command immediately sends a message to all users logged in with the user ID specified by the <user> parameter, unless the <location> is specified, in which case only the user logged in from the specified location is sent the message.

For example, to send a message to all users logged in as root, issue a **write root** command as illustrated below.

```
NB login:  root

NB:Top> who
user      time          from
root      Dec 16 14:41  line00
root      Dec 18 05:50  143.191.12.26:12104
NB:Top> write root
Enter the message for 'write' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: Root users please login using your own IDs
Line 2: so that we know who you are.
Line 3: .

Message from root on 143.191.12.26:12104
Root users please login using your own IDs
so that we know who you are.
```

The next example shows you how to send a message to just the user who is logged in from a specific location.

```
NB:Top> who am i
user      time          from
root      Dec 18 05:50  143.191.12.26:12104
NB:Top> write root 143.191.12.26:12104
Enter the message for 'write' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: This message is to myself.
Line 2: .

Message from root on 143.191.12.26:12104
This message is to myself.
```

To send a message to all logged in users, regardless of their user IDs, use the **wall** [<message>] command.