LBI-38967

ERICSSON

Mobile Communications

EDACS® BCU/CAL

Billing Correlation Unit/ Centralized Activity Logger

User Interface Manual

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

1. OVERVIEW

This document provides a thorough discussion of the Enhanced Digital Access Communications System (EDACS) Billing Correlation Unit/Centralized Activity Logger (BCU/CAL) User Interface. System topics, command syntax, and examples are provided. The majority of the User Interface is very "UNIX-like" in nature. Many of the commands will be familiar to those with experience in the UNIX environment.

The BCU/CAL User Interface is a command shell with terse and powerful syntax. Advantages of this approach are cross-product commonality, a rich command set, and most importantly, a high degree of flexibility for the BCU/CAL end user. Disadvantages are that randomly hacking at commands can result in significant disruption of the BCU/CAL operation. The User Interface described herein has been designed to strike a reasonable compromise between potency and problematic potential.

Section 2 provides general information required to effectively employ the User Interface. Topics include volume naming conventions, directory structures, and important system daemons. Section 3 discusses commands and syntax. These commands and their syntax are common to several Ericsson GE VMEbus-based EDACS products, such as the Jessica Private Branch Exchange (PBX) Gateway and EDACS Data Gateway (EDG). Where applicable, examples are annotated specifically towards BCU/CAL usage. Appendix A discusses the BCU-CAL Configuration Service (BCS) program. BCS is essentially a small, self-contained user interface dedicated to BCU/CAL system configuration issues. Appendix B presents telnet protocol features, including a description of telnet and a discussion of the telnet commands. Appendix C covers file transfer protocol (FTP) features, with an overview of FTP and a discussion of FTP commands. Appendix D defines the management information base (MIB) variables which may be examined with BCU/CAL. Appendix E contains error codes associated with BCU/CAL networking daemon PNAD.

Additional information on BCU/CAL may be found in the following publications:

- LBI-38703, EDACS VAX/VMS System Manager Technical Reference Manual
- LBI-38965, EDACS Billing Correlation Unit/Centralized Activity Logger System and Installation Manual
- RFC-1213, Management Information Base for Network Management (MIB-II) (Internet documentation)
- RFC-793, Transmission Control Protocol (TCP) (Internet documentation)

2. GENERAL INFORMATION

The following section provides general information to assist the user in understanding the User Interface and its effective operation.

2.1. VOLUME NAMES

Several BCU/CAL internal devices (such as the system disk) are identified as a "volume." Volume naming conventions are similar to those used in the UNIX operating system. Specifically, a volume label is of the form *Major_Number.Minor_Number*. The Major_Number may be thought of as identifying the type of device, and the Minor_Number as identifying a specific device of the major type.

Of specific interest to the User Interface are the volume names associated with the BCU/CAL's mass storage devices. The Major_Number for the BCU/CAL's storage devices is 1, which identifies them as being devices connected on the BCU/CAL's small computer systems interface (SCSI) bus. The Minor_Number identifies a particular device's address on the SCSI bus.

The table below defines the volume names for the BCU/CAL's storage devices.

Table 1 - BCU/CAL Volume Names

Volume	Storage Device
1.1	Floppy disk drive. MS-DOS 1.44Mb format.
1.2	System hard disk drive.
1.3	System tape drive (if any).

Throughout this document, reference will occasionally be made to a "pHILE+" formatted volume or directory. The term "pHILE+" simply refers to the internal file format of the BCU/CAL system hard disk, analogous to the way MS-DOS compatible is used to describe the format of a floppy diskette.

2.2. SYSTEM DIRECTORIES

The table that follows describes the system directories mandatory for correct BCU/CAL operation. These directories are verified each time the system boots. If they do not exist, they will be created. However, under no circumstances should a user attempt to delete or modify these directory paths. To do so may result in indeterminate or catastrophic system behavior.

The user is free to create any other (sub)directories on the BCU/CAL system disk to suit his or her requirements. To assure compatibility with future software releases, it is recommended that users not create subdirectories or store user data beneath any of the directories specified in the following table.

Table 2 - BCU/CAL Root Directory Structure

Directory	Contents
/etc	System administration (hosts, networks, passwd, etc. files).
/tmp	Temporary file repository, such as created by tmpfile.
/bin	Binary files, system utilities.
/usr	Mount point for users (static sharable files).
/var	Subtree for varying files.
/export	NFS exports (default root of exported file system).
/mnt	Default mount point for (NFS) file systems.
/loads	Application S-Record files, such as BCU_A.SX and BCU_B.SX.
/cnfg	Application configuration files, such as the BCU/CAL's IP.DAT and SYSTEM.BIN.
/backup	Backup files. When a new software version is installed, the previous version(s) will be moved to this directory.
/activity	Repository for system catastrophic events, diagnostic report files, and CAL disk queues.
/cdr	Default directory where call detail records (CDRs) are maintained.
/rar	Default directory where raw activity records (RARs) are maintained.
/log	Directory where BCU/CAL information and log files are maintained.

2.3. SYSTEM FILES

The table below describes the three ordinary (non-directory) files contained in the root directory of the BCU/CAL system disk. The files BITMAP.SYS and FLIST.SYS are system write access only, and are thus protected from accidental modification by the user. The file LOADER.SX is copied from floppy disk when the BCU/CAL application is installed or upgraded. As such, it is read/write accessible by a user. However, under no (normal) circumstances should it be modified or deleted. To do so will prevent the BCU/CAL from booting.

Table 3 - BCU/CAL Root Directory Files

File Name	Contents
BITMAP.SYS	Volume bitmap. Maps which disk blocks are in use.
FLIST.SYS	File descriptor list. Maps files/directories on the disk.
LOADER.SX	BCU/CAL application loader file, mandatory for system bootstrap loading.

Several files may be generated by the BCU/CAL in the /log directory. These include communication error logging and system behavior information for the user. The user is free to access and/or remove files in this directory (those with a .log extension) at his or her discretion.

In the event of a catastrophic system error (i.e., a software bug or internal hardware error), the BCU/CAL will automatically initiate a full system reboot. Prior to doing so, an attempt will be made to isolate and log the source of the error. If this diagnostic sequence is successful, a file named "fatal.log" will be present in the /activity directory. The user should copy this file to a floppy diskette and forward it to the Ericsson GE Service Representative for analysis.

2.4. SYSTEM DAEMONS

The following table provides the names and a brief description of system daemons which have relevance to the User Interface. Under normal circumstances, these daemons are of no consequence to the user. They are provided primarily for reference, and to assist the network administrator in security administration.

For example; if the network administrator suspects malicious login attempts via a telnet connection, he or she could suspend the telnet server daemon (tnpd) while the user is being investigated. After security has been restored, the server daemon can been resumed. Commands related to suspended, resuming, and priority adjustment of daemon tasks are provided in the sections below.

Table 4 - Important System Daemons

Daemon	Function
pshc	Console login shell.
pshd	Listens for connection requests from the telnet server daemon (tnpd), and dynamically spawns additional shell daemons to handle the telnet login (one per session).
tnpd	Telnet server.
ftpd	FTP server.
PNAD	TCP/IP network management.
pmap	NFS port mapper.
mntd	Mount server. Handles requests for mounting and listing exported directories.
nfsd	Export server. Handles all NFS requests after exported directories have been mounted.

3. COMMANDS AND SYNTAX

This section of the document provides detailed information on specific commands, their syntax, and examples. The commands should be typed at the prompt `pSH+>`. Please note that the commands are case-sensitive.

3.1. SYSTEM ADMINISTRATION COMMANDS

The following commands support system administration functions of the BCU/CAL.

3.1.1. bcs

bcs -- invokes the BCS program

USAGE:

bcs

DESCRIPTION:

bcs invokes the BCU/CAL Configuration Service. Refer to Appendix A for a complete discussion of BCS and its syntax.

OPTIONS:

None.

3.1.2. date

date -- displays or sets the date

USAGE:

date [yyyymmddhhmm [.ss]]

DESCRIPTION:

If no argument is given, ***date*** displays the current date and time. Otherwise, the current date will be set. ***date*** should only be used to display the BCU/CAL's current time and date.

yyyy is the four digits of the year; the first ***mm*** is the month number; ***dd*** is the day number in the month; ***hh*** is the hour number (24-hour system); the second ***mm*** is the minute number; ***.ss*** (optional) specifies the seconds. The year may be omitted; the current value is supplied as the default.

The command ***settime*** should be used to set the system's time and date.

3.1.3. passwd

passwd -- performs password management

USAGE:

passwd [login_name]

DESCRIPTION:

passwd changes (or installs) a password associated with the user's username (your own by default). When changing a password, ***passwd*** prompts for the old password and then for the new one. You must supply both, and the new password must be typed twice to preclude mistakes. Only the owner of the name or the super-user may change a password; the owner must prove he knows the old password. The super-user can change any password, and is the account authorized to install a new user.

OPTIONS:

None.

NOTES:

1. Login and Password entries are case-sensitive.
2. To delete an account, (1) copy the file 1.2/etc/passwd to floppy disk, (2) edit the file with any ASCII text editor (simply delete the line which contains the user account to be removed), (3) convert the file on floppy back to UNIX format (i.e., LF terminates each line, rather than the CR/LF), and if necessary, (4) copy the file from floppy back to 1.2/etc. Changes will be effective upon the next system reboot. Note that the filename 1.2/etc/passwd is case-sensitive. You may need to use the ***mv*** command to rename the file to lower case if it is copied from a floppy disk.

3.1.4. product

product -- examines/modifies product feature(s)

USAGE:

usage: product -elsw

DESCRIPTION:

product is a utility to allow the end user to selectively activate (or deactivate) particular features for which their BCU/CAL has been licensed.

OPTIONS:

- e Displays which of the unit's licensed features are enabled.
- l Lists the feature license(s) installed on this unit.
- s Sets which of the licensed features are to be enabled on the next system boot.
- w Shows the licenses that may be purchased for this platform.

EXAMPLES:

```
pSH+> product -le
This unit is licensed for the following product feature(s):
        BCU:  Billing Correlation Unit
        NFS:  Network File System
This unit has the following product feature(s) enabled:
        BCU:  Billing Correlation Unit
        NFS:  Network File System
pSH+>

pSH+> product -s
Enable Billing Correlation Unit (BCU) feature [Y/N]?y
Enable Network File System (NFS) feature [Y/N]?n
Product configuration successful!
Change(s) effective on next system reboot.
```

3.1.5. reboot

reboot -- reboots the system

USAGE:

reboot [-h]

DESCRIPTION:

reboot performs an immediate and complete system reboot

OPTIONS:

- h Performs immediate full system reboot (hard).

NOTES:

The ***sync*** command should be issued prior to reboot to assure that all user data are flushed to the hard disk. This command should only be used to reboot the BCU/CAL under “panic” situations.

3.1.6. settime

settime -- sets date and time in BCU/CAL non-volatile storage

USAGE:

settime YYYY MM DD HH MM SS

DESCRIPTION:

settime sets the date and time in the BCU/CAL battery-backed non-volatile storage unit. The time format is specified in the following table. Use the ***date*** command to inspect the current date and time.

YYYY	Year, A.D.
MM	Month, 01-12
DD	Day of month, 01-31
HH	Hour, 00-23
MM	Minute, 00-59
SS	Second, 00-59

3.1.7. version

version -- displays software versions

USAGE:

version

DESCRIPTION:

version displays the current software revisions of the BCU/CAL application and operating system components.

OPTIONS:

None.

EXAMPLE:

```
pSH+> version
BCU/CAL Application Software Versions:
File: LOADER.SX Version: V00.07 Path: 01.02/LOADER.SX
File: BC_A.SX   Version: V00.07 Path: 01.02/loads/BC_A.SX
File: BC_B.SX   Version: V00.07 Path: 01.02/loads/BC_B.SX

Operating System Software Version:
ROM OS Version: V01.03  Board Type: MVME-147
```

3.2. FILE MAINTENANCE AND UTILITY COMMANDS

The following commands provide the maintenance and utility function for the BCU/CAL file system. These commands apply to both the system hard disk (volume 1.2) and floppy disk (volume 1.1) drives.

3.2.1. cat

cat -- concatenates and displays

USAGE:

cat [-bents] filename...

DESCRIPTION:

cat reads each ***filename*** in sequence and displays it on the standard output.

OPTIONS:

- b Numbers the lines as **-n** but omits the line numbers from blank lines.
- e Displays non-printable characters as **-v** and in addition displays a $ character at the end of each line.
- n Precedes each line output with its line number.
- s Substitutes a single blank line for multiple adjacent blank lines.
- t Displays non-printable characters as **-v** and in addition displays TAB characters such as ^I (CTRL-I).
- v Displays non-printable characters (with the exception of TAB and NEWLINE characters) so that they are visible. Control characters print such as ^X for CTRL-X; the DEL character (octal 0177) prints as "^?". Non-ASCII characters (with the high bit set) are displayed as M-x, where M- stands for "meta" and x is the character specified by the seven low order bits.

NOTES:

Using ***cat*** to redirect output of a file to the same file, such as ***cat filename1 > filename1*** or ***cat filename1 >> filename1***, does not work. This type of operation should be avoided at all times since it may cause the system to enter an indeterminate state.

Once started, ***cat*** cannot be aborted. It may, however, be suspended and resumed using flow control characters ^S and ^Q.

3.2.2. cd

cd -- changes working directory

USAGE:

cd [directory]

DESCRIPTION:

directory becomes the new working directory.

OPTIONS:

None.

3.2.3. clone

clone -- copies system files (wildcards supported)

USAGE:

clone -q source_spec destination_path

DESCRIPTION:

clone is a file copy command similar to **cp**, which supports the use of the wildcard character "*****" in the source specification. ***source_spec*** specifies the path and file(s) to be copied, which may include wildcards. ***destination_path*** specifies the target directory or volume to which ***source_spec*** will be copied, which may be a directory or a volume. If a directory specified by ***destination_path*** does not exist, it will be created. Usage of ***clone*** is restricted to the root user account.

OPTIONS:

- q Quiet. Suppresses summary information from being displayed during the copy process.

EXAMPLE:

The following example copies all files with a .DAT extension in the /cnfg directory to a floppy disk.

```
pSH+> pcmount 1.1
pSH+> clone cnfg/*.DAT 1.1/
clone: ./cnfg/EXPORTS.DAT copied to 1.1/EXPORTS.DAT  1606 bytes copied
clone: ./cnfg/IP.DAT copied to 1.1/IP.DAT  915 bytes copied
clone: ./cnfg/ROUTES.DAT copied to 1.1/ROUTES.DAT  784 bytes copied
clone: ./cnfg/CAL.DAT copied to 1.1/CAL.DAT  862 bytes copied
4 files copied.
Clone complete.
pSH+> umount 1.1
```

3.2.4. cmp

cmp -- performs a byte-by-byte comparison of two files

USAGE:

cmp [-ls] filename1 filename2 [skip1] [skip2]

DESCRIPTION:

cmp compares ***filename1*** and ***filename2***. With no options, ***cmp*** makes no comment if the files are the same. If they differ, it reports the byte and line number at which the difference occurred, or, that on file is an initial subsequence of the other. ***skip1*** and ***skip2*** are initial byte offsets into ***filename1*** and ***filename2***, respectively. These offsets may be either octal or decimal, where a leading 0 denotes octal.

OPTIONS:

- l Prints the byte number (in decimal) and the differing bytes (in octal) for all differences between the two files.
- s Silent. Prints nothing for differing files.

3.2.5. cp

cp -- copies files

USAGE:

cp [-i] filename1 filename2
cp -rR [-i] directory1 directory2
cp [-irR] filename ... directory

DESCRIPTION:

cp copies the contents of ***filename1*** onto ***filename2***. If ***filename1*** is a symbolic link, or a duplicate hard link, the contents of the file that the link refers to are copied; links are not preserved.

In the second form, ***cp*** recursively copies ***directory1***, along with its contents and subdirectories, to ***directory2***. If ***directory2*** does not exist, ***cp*** creates it and duplicates the files and subdirectories of ***directory1*** within it. If ***directory2*** does exist, ***cp*** makes a copy of the ***directory1*** directory (along with its files and subdirectories) within ***directory2*** (as a subdirectory).

In the third form, each ***filename*** is copied to the indicated ***directory***; the basename of the copy corresponds to that of the original. The destination ***directory*** must already exist for the copy to succeed.

OPTIONS:

- i Interactive. Prompt for confirmation whenever the copy would overwrite an existing file. A "y" in answer to the prompt confirms that the copy should proceed. Any other answer prevents ***cp*** from overwriting the file.
- r or R Recursive. If any of the source files are directories, copy the directory along with its files (including subdirectories and their files). The destination must be a directory.

NOTES:

cp refuses to copy a file onto itself.

The wildcard character "*" is not supported.

3.2.6. df

df -- displays file system usage

USAGE:

df

DESCRIPTION:

df displays the amount of disk space occupied by currently mounted file systems, the amount of used and available space, and the portion of the file system's total capacity that has been used.

EXAMPLE:

```
pSH+> pcmount 1.1
pSH+> df
Filesystem    kbytes      used     avail  capacity  Mounted on
01.01           1440      1244       196   86.389%  01.01/
01.02        1211728      7133   1204595    0.589%  /
```

3.2.7. du

du -- displays the number of disk blocks used per directory or file

USAGE:

du [-sa] [filename ...]

DESCRIPTION:

du gives the number of 512-byte disk blocks contained in all the files and, recursively, directories within each specified directory of file ***filename***. If ***filename*** is missing, "." (the current directory) is used. If no option is selected, entries are generated only for each directory.

OPTIONS:

- s Only displays the grand total for each of the specified filenames.
- a Generates an entry for each file.

3.2.8. head

head -- displays the first few lines of specified files

USAGE:

head [-n] [filename ...]

DESCRIPTION:

head copies the first ***n*** lines of each ***filename*** to the standard output. If no ***filename*** is given, ***head*** copies lines from the standard input. The default value of ***n*** is 10 lines.

When more than one file is specified, the start of each file will appear as follows:

```
==> filename <==
```

Thus, a common way to display a set of short files, identifying each one, is as follows:

```
pSH+> head -1 filename1 filename2 ...
```

EXAMPLE:

```
pSH+> head -3 junk1 junk2 junk3

==> junk1 <==

Line 1 of junk1...
Line 2 of junk1...
Line 3 of junk1...

==> junk2 <==

Line 1 of junk2...
Line 2 of junk2...
Line 3 of junk2...

==> junk3 <==

Line 1 of junk3...
Line 2 of junk3...
Line 3 of junk3...
```

3.2.9. ls

ls -- lists the contents of a directory

USAGE:

ls [-aACdfFgilqrRs1] filename

DESCRIPTION:

For each ***filename*** that is a directory, ***ls*** lists the contents of the directory; for each ***filename*** that is a file, ***ls*** repeats its name and any other information requested. By default, the output is sorted alphabetically. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents.

OPTIONS:

- a Lists all entries. In the absence of this option, entries having names beginning with a "." are not listed.
- A Same as **-a**, except that "." and ".." are not listed.
- C Forces multi-column output, with entries sorted down the columns; for ***ls***, this is the default when output is to a terminal.
- d If the argument is a directory, lists only its name (not its contents); often used with **-l** to obtain the status of a directory.
- f Forces each argument to be interpreted as a directory and lists the name found in each slot. This option turns off **-l**, **-s**, and **-r**, and turns on **-a**; the order is the order in which entries appear in the directory.
- F Marks directories with a trailing slash (/) and executable files with a trailing asterisk (*).
- g For ***ls***, shows the group ownership of the file in a long output.
- i For each file, prints the **i**-number in the first column of the report.
- l Lists in long format, providing mode, owner, size in bytes, and time of last modification for each file. If the time of last modification is greater than six months ago, it is shown in the format "month date year"; files modified within six months show "month date time".
- q Displays non-graphic characters in filenames as the character "?"; for ***ls***, this is the default when output is to a terminal.
- r Reverses the order of sort to receive reverse alphabetic or oldest first, as appropriate.
- R Recursively lists the subdirectories encountered.
- s Provides the size of each file, including any indirect blocks used to map the file, in kilobytes.
- 1 Forces single-column output.

3.2.10. mkdir

mkdir -- makes a directory

USAGE:

mkdir [-p] dir_name ...

DESCRIPTION:

mkdir creates the directory ***dir_name***.

OPTIONS:

- p Allows missing parent directories to be created as needed.

3.2.11. more

more -- browses or pages through a text file

USAGE:

more [-l lines] file1 ... filen

DESCRIPTION:

more is a filter that displays the contents of a text file on the terminal, one screen at a time. It pauses after each screen, and prints `--More--` at the bottom of the screen. To continue browsing the file, enter the CR (<Return>) character (i.e., Enter or Return). To terminate **more**, enter the character "q", followed by CR.

OPTIONS:

- l Displays the file in increments of line_count lines (default is 23 lines).

EXAMPLE:

```
pSH+> more -l 15 /loads/BC_A.SX
################################################
# Copyright (c) Ericsson GE Mobile Communications
# All rights reserved. 1993, 1994
#
# FILE:    BC_A.SX
# VERSION: V00.05
# DATE:    Thru Feb  3 10:05:22 EST 1994
# PATH:    /files0/home/jay/bcu/build
################################################

S00600004844521B
S30A00148B7C706D6170002C
S31400148B820A3C4544473E2053797374656D20535E
S31400148B9174617274757020486 16C7465640A009F
S31400148BA030312E30322F636E66672F535953546C
--More-- q
pSH+>
```

3.2.12. mv

mv -- moves or renames files

USAGE:

mv [-fi] filename1 filename2
mv [-fi] directory1 directory2
mv [-fi] filename ... directory

DESCRIPTION:

mv moves files and directories around in the file system. A side effect of ***mv*** is to rename a file or directory. The three major forms of ***mv*** are shown in the usage synopsis above.

The first form of ***mv*** moves (changes the name of) ***filename1*** to ***filename2.*** If ***filename2*** already exists, it is removed before ***filename1*** is moved.

The second form of ***mv*** moves (changes the name of) ***directory1*** to ***directory2***, only if ***directory2*** does not already exist; if it does, the third form applies.

The third form of ***mv*** moves one or more ***filename*(s)** (may also be directories) with their original names, into the last ***directory*** in the list.

OPTIONS:

- f Force. Overrides any mode restrictions and the **-i** option. The **-f** option also suppresses any warning messages about modes which would potentially restrict overwriting.
- i Interactive mode. ***mv*** displays the name of the file or directory followed by a question mark whenever a move would replace an existing file or directory. If a line starting with "y" is typed, ***mv*** moves the specified file or directory, otherwise ***mv*** does nothing with that file or directory.

NOTES:

mv refuses to move a file or directory onto itself.

mv will not move a directory from one file system to another.

3.2.13. purge

purge -- purges system files

USAGE:

purge -q [-d YYYY MM DD HH] file_spec

DESCRIPTION:

purge is a file deletion utility which supports wildcard characters (*) in the file specification ***file_spec***. Its access is restricted to the root user account.

OPTIONS:

- q Quiet. User confirmation for file deletion is suppressed.
- d Only files older than the date specified by YYYY MM DD HH.

NOTES:

When ***purge*** is used from a remote login (i.e., telnet), some connection types may require that <RETURN> be entered twice when prompted for file deletion confirmation. ***purge*** may be aborted during a confirmation prompt by entering **q**, (rather than **y** or **n**), followed by <RETURN>.

EXAMPLES:

This example quietly purges CDR files that start with t01, and are older than the indicated date.

```
pSH+> purge -q cdr/t01*.CDR 1994 2 1 0
```

3.2.14. rm

rm -- removes (unlink) files

USAGE:

rm [-fir] filename

DESCRIPTION:

rm removes (directory entries for) one or more files. If an entry was the last link to the file, the contents of that file are lost.

OPTIONS:

- f Forces files to be removed without displaying permissions, asking questions, or reporting errors.
- i Asks whether to delete each file, and, under **-r**, whether to examine each directory. Sometimes called the "interactive option."
- r Recursively deletes the contents of a directory, its subdirectories, and the directory itself.

NOTES:

Removing the file ".." is forbidden to avoid inadvertently causing a problem such as "rm -r .*".

3.2.15. rmdir

rmdir -- removes (unlinks) directories

USAGE:

rmdir directory

DESCRIPTION:

rmdir removes each named ***directory***. ***rmdir*** only removes empty directories.

OPTIONS:

None.

3.2.16. sync

sync -- forces changed blocks to disk

USAGE:

sync

DESCRIPTION:

sync brings a mounted volume up to date. It does this by writing to the volume all modified file information for open files, and flushing cache buffers containing physical blocks that have been modified.

This call is superfluous under immediate write synchronization mode, and is not allowed on an NFS mounted volume.

OPTIONS:

None.

3.2.17. tail

tail -- displays the last part of a file

USAGE:

tail +|-number [lc] filename

DESCRIPTION:

tail copies ***filename*** to the standard output beginning at a designated place.

OPTIONS:

Options are not specified separately with their own "-" signs.

- +number Begins copying at distance number from the beginning of the file. number is counted in units of lines or characters, according to the appended option **-l** or **-c**. When no units are specified, counting is by lines. If number is not specified, the value 10 is used.
- -number Begins copying at distance number from the end of the file. number is counted in units of lines or characters, according to the appended option **-l** or **-c**. When no units are specified, counting is by lines. If number is not specified, the value 10 is used.
- l number is counted in units of lines.
- c number is counted in units of characters.

3.2.18. touch

touch -- updates the access and modification times of a file

USAGE:

touch [-cf] filename

DESCRIPTION:

touch sets the access and modification times of each ***filename*** argument to the current time. ***filename*** is created if it does not exist (default).

OPTIONS:

- c Does not create filename if it does not exist.
- f Attempts to force the ***touch*** in spite of read and write permissions on filename.

3.3. FLOPPY DISK COMMANDS

The following commands are specifically associated with the floppy disk drive of the BCU/CAL.

3.3.1. pcmkfs

pcmkfs -- initializes a volume for an MS-DOS file system

USAGE:

pcmkfs [-i] volume_name format

DESCRIPTION:

pcmkfs initializes (i.e., formats) the volume ***volume_name*** for the MS-DOS disk type specified by ***format***; where ***format*** is one of the following:

1 = 360 Kbyte (5-1/4" double density)
2 = 1.2 Mbyte (5-1/4" high density)
3 = 720 Kbyte (3-1/2" double density)
4 = 1.4 Mbyte (3-1/2" high density)

OPTIONS:

- i Calls device driver initialization procedure.

NOTES:

The BCU/CAL only supports ***format*** specification 4 above; 1.4 Mbyte (3-1/2" high density)

EXAMPLE:

The following example would correctly format a diskette installed in the BCU/CAL floppy disk drive.

```
pSH+> pcmkfs 1.1 4
Warning: this operation will destroy all data on the specified volume.
Do you wish to continue (y/n)? y
```

3.3.2. pcmount

pcmount -- mounts an MS-DOS file system.

USAGE:

pcmount volume_name [sync_mode]

DESCRIPTION:

pcmount will mount an MS-DOS volume ***volume_name***. A volume must be mounted before any file operations can be carried out on it. ***sync_mode*** specifies the file system synchronization method for the volume, which is one of the following:

0 = Immediate write synchronization mode.
1 = Control write synchronization mode.
2 = Delayed write synchronization mode (default).

OPTIONS:

None.

NOTES:

The ***volume_name*** for the BCU/CAL floppy disk drive is 1.1. A diskette should be inserted in the floppy drive prior to using ***pcmount***.

The ***umount*** command should be used prior to removing the diskette.

EXAMPLE:

The following example would mount a diskette in the BCU/CAL floppy disk drive.

```
pSH+> pcmount 1.1
```

3.3.3. umount

umount -- unmounts the file systems

USAGE:

umount volume

DESCRIPTION:

umount unmounts a previously mounted file system ***volume***. Unmounting a file system causes it to be synchronized (all memory-resident data will be flushed to the device).

OPTIONS:

None.

EXAMPLE:

The following example mounts the BCU/CAL floppy disk drive, changes directory to it, and lists the contents. The current directory is restored to the hard disk drive, and the floppy is unmounted.

```
pSH+> mount 1.1
pSH+> cd 1.1/
pSH+> ls
IP.DAT ROUTES.DAT EXPORTS.DAT CAL.DAT
pSH+> cd 1.2/
pSH+> umount 1.1
pSH+> cd 1.1/
1.1/: no such file or directory
```

3.4. TAPE DRIVE COMMANDS

The following commands relate to the optional Digital Analog Tape (DAT) drive on the BCU.

3.4.1. backup

backup -- manually initiates a backup procedure

USAGE:

backup

DESCRIPTION:

backup manually starts the tape archiving program, which will copy CDR files to tape and purge those files which have expired. This command is only applicable to BCU/CAL units which have the optional DAT tape drive installed.

OPTIONS:

None.

3.4.2. mt

mt -- magnetic tape utility

USAGE:

mt volume command [count]

DESCRIPTION:

mt is a magnetic tape utility which can be used to control the (optional) BCU/CAL's DAT drive. ***volume*** specifies the BCU/CAL volume number of the optional DAT tape drive, which is always 1.3. The ***command*** parameters are defined in the following table. ***count*** is an optional parameter for some of the ***command*** parameters, indicating how many of the appropriate ***command*** specifications are to be executed (if applicable). The default value of count is 1.

- eof Writes ***count*** end-of-file marks at the current position on the tape.
- weof Same as eof.
- fsf Forwards space over ***count*** EOF marks. The tape is positioned on the first block of the file.
- fsr Forwards space ***count*** records.
- bsf Back spaces over ***count*** EOF marks. The tape is positioned on the beginning-of-tape side of the EOF mark.
- bsr Back spaces ***count*** records.
- nbsf Back spaces ***count*** files. The tape is positioned on the first block of the file. This is equivalent to [***count*** + 1] bsf, followed by one fsf.
- asf Absolute space to ***count*** file number. This is equivalent to a rewind followed by an fsf count.
- eom Spaces to the end of the recorded media on the tape. This is useful for appending files onto previously written tapes.
- rewind Rewinds the tape.
- erase Erases the entire tape.
- eject Ejects the tape.
- offline Rewinds the tape and takes the drive unit off-line by unloading the tape

.

EXAMPLE:

The following examples rewind and eject the DAT tape, respectively.

```
pSH+> mt 1.3 rewind
pSH+> mt 1.3 eject
```

3.4.3. tar

tar -- UNIX tar utility

USAGE:

tar source_files destination

DESCRIPTION:

tar is an archive utility which copies files to tape (or to a file) in UNIX tape archive (tar) format, which can be read on a UNIX-based computer. ***source_files*** specifies the source file(s) or directories to be processed to tar format. The ***source_files*** specification may contain wildcard characters (*), and directories are recursively descended. ***destination*** specifies the target for the tar output, which may be a file or volume specification.

OPTIONS:

None.

EXAMPLES:

The following example uses the floppy disk (volume 1.1) as the destination for the tar file.

```
pSH+> tar cnfg/*.DAT 1.1
a ./cnfg/EXPORTS.DAT, 1606 bytes, 4 tape blocks
a ./cnfg/IP.DAT, 915 bytes, 2 tape blocks
a ./cnfg/ROUTES.DAT, 784 bytes, 2 tape blocks
a ./cnfg/CAL.DAT, 862 bytes, 2 tape blocks
a ./cnfg/TX15RX16.DAT, 729 bytes, 2 tape blocks
a ./cnfg/SITE14.DAT, 664 bytes, 2 tape blocks
a ./cnfg/FOUR.DAT, 862 bytes, 2 tape blocks
a ./cnfg/CAL_LAB.DAT, 4241 bytes, 9 tape blocks
a ./cnfg/CAL1.DAT, 665 bytes, 2 tape blocks
```

The following example produces a tar file named my_file.tar in the current working directory.

```
pSH+> tar cnfg/*.DAT my_file.tar
a ./cnfg/EXPORTS.DAT, 1606 bytes, 4 tape blocks
...
a ./cnfg/CAL1.DAT, 665 bytes, 2 tape blocks
pSH+> ls -als my_file.tar
  39 -rwxrwxrwx   1 root          19456 Jul 07 1994 14:05 my_file.tar
```

The following example uses the optional DAT tape drive (volume 1.3) as the destination for the tar file.

```
pSH+> tar cnfg/*.DAT 1.3
```

The following example would tar the entire system disk into the file backup.tar, located in the root directory. This file could be used as a complete backup of the BCU/CAL system disk.

```
pSH+> cd /
pSH+> tar . backup.tar
```

3.5. NETWORKING COMMANDS

The following commands provide statistical and diagnostics services associated with the TCP/IP networking capability of the BCU/CAL.

3.5.1. mac

mac -- displays/sets MAC (ethernet) address

USAGE:

mac [-d] | [-s xx:xx:xx:xx:xx:xx]

DESCRIPTION:

mac provides the ability to display or modify the MAC (ethernet station) address. All users may display the current MAC address. However, only the super-user (root) can modify the MAC address. Attempts by any users other than root will result in a "Permission denied" warning. Setting of the MAC address should only be used if the battery backup RAM TOD module on the TVME-147 has been replaced. The MAC address to be reprogrammed is labeled on the internal side of the TVME-147. The MTBF of the TOD module is 5 years. Thus, use of this command will be rare, if ever.

OPTIONS:

- d Displays the current MAC address.
- s Sets the MAC address, where xx:xx:xx:xx:xx:xx is the hexadecimal value of the address.

EXAMPLE:

```
pSH+> mac -d
Current MAC Address is 08:00:3E:20:E0:68

pSH+> mac -s 8:0:3e:21:5c:f1
MAC Address set to 08:00:3E:21:5C:F1
Change effective upon next system reboot.
```

3.5.2. netstat

netstat -- displays network statistics

USAGE:

netstat topic [-as]

DESCRIPTION:

netstat displays a variety of statistical information regarding network activity. Refer to Appendix B and Management Information Base for Network Management (RFC-1213) for further discussion of the statistics group variables. This command can be especially valuable in characterizing network performance, particularly with the CAL application.

topic specifies the network statistical entity of interest, which may be one of the following:

if	Interface group statistics
icmp	Internet Control Message Protocol (ICMP) group statistics
ip	Internet Protocol (IP) group statistics
tcp	Transmission Control Protocol (TCP) group statistics
upd	User Datagram Protocol (UDP) group statistics

OPTIONS:

- a Displays all information available within the statistics group, including any "special" information.
- s Displays only "special" information available with the statistics group, if any.

EXAMPLES:

IF EXAMPLE:

The following example displays the interface group statistics on the BCU/CAL. These parameters reflect the activity on the ethernet network interface. The ***-a*** option requests that any "special" information regarding the interface be displayed. In this example, the special information reflects the lower level chip statistics associated with the ethernet interface.

```
pSH+> netstat if -a
MIB Interface Table: ifIndex = 1
ifDesc = Ericsson GE BCU/CAL: MVME-147, AM7990
ifAdminStatus   = 1          ifOperStatus      = 1
ifType          = 6          ifMtu             = 1500       ifSpeed = 10485760
ifInOctets      = 726822     ifInUcastPkts     = 1999
ifInNUcastPkts  = 8994       ifInDiscards      = 0
ifInErrors      = 0          ifInUnknownProtos = 5895
ifOutOctets     = 86298      ifOutUcastPkts    = 1356
ifOutNUcastPkts = 336        ifOutDiscards     = 0
ifOutErrors     = 0          ifOutQLen         = 0

ETHERNET DRIVER LOW LEVEL STATISTICS:
GENERAL CHIP (AM-7990) ACTIVITIES:
Rcv Interrupts  = 12149       Xmt Interrupts = 1871
Chip Babble     = 0           Chip Restarts  = 0
Heartbeat       = 1870        Missed Packets = 0
RECEIVER ERRORS:
Framing         = 0           Overflow       = 0
CRC             = 0           No Buffers     = 0
TRANSMITTER ERRORS:
Late Collisions = 0           Lost Carrier   = 0
No Buffers      = 0           Underflow      = 0
Retries         = 0
```

ICMP EXAMPLE:

The following example displays the ICMP group statistics on the BCU/CAL.

```
pSH+> netstat icmp
icmpInMsgs           = 2          icmpInErrors      = 2
icmpInDestUnreachs   = 0          icmpInTimeExcds   = 0
icmpInParmProbs      = 0          icmpInSrcQuenchs  = 0
icmpInRedirects      = 0          icmpInEchos       = 0
icmpInEchoReps       = 0          icmpInTimestamps  = 0
icmpInTimestampReps  = 0          icmpInAddrMasks   = 0
icmpInAddrMaskReps   = 0

icmpOutMsgs          = 76         icmpOutErrors     = 0
icmpOutDestUnreachs  = 7          icmpOutTimeExcds  = 0
icmpOutParmProbs     = 0          icmpOutSrcQuenchs = 45
icmpOutRedirects     = 24         icmpOutEchos      = 0
icmpOutEchoReps      = 0          icmpOutTimestamps = 0
icmpOutTimestampReps = 0          icmpOutAddrMasks  = 0
icmpOutAddrMaskReps  = 0
```

IP EXAMPLE:

The following example displays the IP group statistics on the BCU/CAL.

```
pSH+> netstat ip
ipForwarding      = 1          ipDefaultTTL      = 30
ipInReceives      = 5902       ipInHdrErrors     = 0
ipInAddrErrors    = 45         ipForwDatagrams   = 1391
ipInUnknownProtos = 0          ipInDiscards      = 0
ipInDelivers      = 4511       ipOutRequests     = 3903
ipOutDiscards     = 0          ipOutNoRoutes     = 0
ipReasmTimeout    = 60         ipReasmReqds      = 0
ipReasmOKs        = 0          ipReasmFails      = 0
ipFragOKs         = 0          ipFragFails       = 0
ipFragCreates     = 0          ipRoutingDiscards = 0
```

TCP EXAMPLE:

Note that instances of object types that represent information about a particular TCP connection are transient; they persist only as long as the connection in question exists. Refer to the ***tcpcon*** command for specific information regarding TCP connections.

```
pSH+> netstat tcp
tcpRtoAlgorithm = [4]: Van Jocobson
tcpRtoMin       = 1000       tcpRtoMax       = 64000
tcpMaxConn      = -1         tcpActiveOpens  = 4
tcpPassiveOpens = 9          tcpAttemptFails = 0
tcpEstabResets  = 0          tcpCurrEstab    = 4
tcpInSegs       = 4705       tcpOutSegs      = 3996
tcpInErrs       = 0          tcpOutRsts      = 0
tcpRetransSegs  = 0
```

UDP EXAMPLE:

The following example displays the UDP group statistics on the BCU/CAL

```
pSH+> netstat udp
udpInDatagrams = 22         udpNoPorts      = 9
udpInErrors    = 0          udpOutDatagrams = 18
```

3.5.3. ping

ping -- sends ICMP ECHO_REQUEST packets to network hosts

USAGE:

ping [-s] host_address [timeout]

DESCRIPTION:

ping utilizes the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified host, or network gateway. ECHO_REQUEST datagrams, or "pings," have an IP and ICMP header, followed by a timeval structure, and then an arbitrary number of bytes to pad out the packet. If ***host_address*** responds, ***ping*** will print out a message indicating that the host is alive, then exit. Otherwise, after ***timeout*** seconds, it will print out a message indicating that no answer was received, then exit. The default value of ***timeout*** is 10 seconds.

If the ***-s*** option is specified, ***ping*** sends one datagram per second, and prints one line of output for every ECHO_RESPONSE it receives. No output is produced if there is no response from ***host_address***. The default datagram size is 64 bytes (8 byte ICMP header + 56 data bytes).

When using ***ping*** for fault isolation, first "**ping**" the local host (127.0.0.1) to verify that the local network interface is running.

host_address must be specified in Internet dotted-decimal notation.

OPTIONS:

- s Sends one "ping" per second to host_address.

EXAMPLE:

```
pSH+> ping 190.1.2.3
PING (190.1.2.3): 56 data bytes
190.1.2.3 is alive.
```

3.5.4. route

route -- examines/modifies network routes

USAGE:

route [-f] [-sh] add|delete [host|net] destination gateway

DESCRIPTION:

route manually manipulates the network routing tables normally maintained by the system routing daemon, the configuration file ROUTES.DAT, or through default routes, and redirect messages from routers. ***route*** allows the super-user to operate directly on the routing table for the specific host or network indicated by ***destination***. The ***gateway*** argument indicates the network gateway to which packets should be addressed.

The ***add*** command instructs ***route*** to add a route to ***destination***. ***delete*** deletes a route. ***destination*** and ***gateway*** must be specified in Internet dotted-decimal notation. Any user may display the current routes using the **-s** or **-h** options. Only the super-user "root" may add or delete routes.

Routes to a particular host must be distinguished from those to a network. The optional keywords ***net*** and ***host*** force the ***destination*** to be interpreted as a network or a host, respectively. If neither the ***net*** or ***host*** keywords are supplied, the route is presumed to be to a host.

ERROR MESSAGES:

Permission denied. -- Attempt by non-super-user to add or delete a route entry.

Invalid arguments. -- Incorrect route parameters were entered.

The route specified cannot be found. -- (1) Attempting to delete a route with an incorrect host or net keyword. The host/net specification must match the route type. (2) Attempting to delete a non-existent route.

Network is unreachable. -- An attempt to add a route failed because the gateway listed was not on a directly connected network. Give the next-hop gateway instead.

Internal routing table out of space. -- An add operation was attempted, but the system was unable to allocate memory to create the new entry.

OPTIONS:

- f Flushes the network routing table.
- s Displays routes in Internet dotted-decimal notation.
- h Display routes in hexadecimal notation.
- host Specifies destination as an IP host address (default).
- net Specifies destination as an IP network address.

EXAMPLES:

```
pSH+> route -s
Destination:     Next hop:       IF:     Type:           Subnet Mask:
127.0.0.1        127.0.0.1       7       DIRECT          0xFF000000
147.117.32.0     147.117.37.240  1       DIRECT          0xFFFFF000

pSH+> route add host 147.117.1.2  147.117.37.245
pSH+> route add net  147.100.0  147.117.32.2
```

```
pSH+> route -s
Destination:     Next hop:       IF:     Type:           Subnet Mask:
127.0.0.1        127.0.0.1       7       DIRECT          0xFF000000
147.117.32.0     147.117.37.240  1       DIRECT          0xFFFFF000
147.117.1.2      147.117.37.245  1       INDIRECT        0xFFFFF000
147.100.0.0      147.117.32.2    1       INDIRECT        0xFFFFF000
```

3.5.5. tcpcon

tcpcon -- examines the TCP connection table

USAGE:

tcpcon -alrhH [-i ip_addr] [-p port_num] [-c conn_state]

DESCRIPTION:

tcpcon allows the user to examine the current TCP connection table entries. Several options are available which allow the table to be searched for only particular parameters of interest. In addition, the table search can be specified for either the local (BCU/CAL) or remote (network) side of a connection. In the absence of other qualifiers, the default is to search both sides of a connection for any parameter specification(s).

Note that TCP connections are transient in nature. Table entries are present only as long as the connection in question exits. Also, recall that TCP implements a psuedo three-way handshaking to communicate over and shut down a connection. This is mentioned since the TCP connection table may appear quite dynamic in nature.

tcpcon can be extremely useful in diagnosing CAL network connectivity problems. The CAL site simulation feature requires a correct mapping of TCP port numbers to terminal server IP address(es), which may be readily examined with this command.

OPTIONS:

- a Examines all current TCP connection table entries. Overrides any other options.
- l Searches the local connection side for a specific IP address (**-i**) and/or TCP port number (**-p**).
- r Searches the remote connection side for a specific IP address (**-i**) and/or TCP port number (**-p**).
- h Provides help. Supplies a brief syntax and options explanation.
- H Provides extended help. Furnishes a list of connection state values which may be used with **-c**.
- i Searches for connection(s) involving the IP address specified by ***ip_addr***.
- p Searches for connection(s) involving the TCP port number specified by ***port_num***.
- c Searches for connections(s) which are currently in the state specified by ***conn_state***.

conn_state is a decimal value representing a specific TCP connection state. The possible values of ***conn_state*** are provided below. The intricacies of TCP connection state transitions are beyond the scope of this document. The explanations given are limited for the sake of brevity.

- 1 Closed. A Transmission Control Block (TCB) has been allocated, but is not currently in use.
- 2 Listen. A server daemon associated with the local port is awaiting connection(s) from clients.
- 3 Sync sent. A sync has been sent, and the connection side is awaiting acknowledgment.
- 4 Sync received. A sync has been received, or a sync has been sent without receiving an acknowledgment.
- 5 Established. Both sides are ready to exchange data and acknowledgments.
- 6 Final wait 1. One side of the connection has requested to close it , and is waiting for a response.
- 7 Final wait 2. One side of the connection has received a request to close it, and is ready to proceed with the shutdown.
- 8 Close wait. A connection shutdown is in progress for reasons other than a close request.
- 9 Last acknowledge. Awaiting final acknowledgment in closing down the connection.

- 10 Closing. Both sides of the connection have agreed to shut down.
- 11 Time wait. A graceful shutdown of the connection has been completed.
- 12 Delete TCB. A TCB allocated for the connection is being returned to the system pool.

EXAMPLES:

The following example displays all current entries in the TCP connection table. In this example, a user at 147.117.37.243 has an active telnet session to the BCU/CAL. Additionally, the CAL feature is active with four sites configured.

```
pSH+> tcpcon -a
TCP Connection Table: Wed May 25 09:07:44 1994
       Local-IP       Remote-IP  Local-Port  Remote-Port  State
 147.117.37.227  147.117.37.227        1028         1030  Established
 147.117.37.227  147.117.37.227        1030         1028  Established
 147.117.37.227  147.117.37.243          23         1064  Established
        0.0.0.0         0.0.0.0          21            0  Listen
        0.0.0.0         0.0.0.0          23            0  Listen
        0.0.0.0         0.0.0.0        1028            0  Listen
 147.117.37.227   147.117.37.11        1027         5016  Established
 147.117.37.227   147.117.37.11        1026         5017  Established
 147.117.37.227   147.117.37.10        1025         5007  Established
 147.117.37.227   147.117.37.10        1024         5006  Established
        0.0.0.0         0.0.0.0         111            0  Listen
```

The following example searches the remote side of the connection table for a TCP port number of 5006.

```
pSH+> tcpcon -r -p 5006
TCP Connection Table: Wed May 25 09:09:00 1994
       Local-IP       Remote-IP  Local-Port  Remote-Port  State
 147.117.37.227   147.117.37.10        1024         5006  Established
```

The following example searches the remote side of the connection table for an IP address of 147.117.37.10.

```
pSH+> tcpcon -r -i 147.117.37.10
TCP Connection Table: Wed May 25 09:08:15 1994
       Local-IP       Remote-IP  Local-Port  Remote-Port  State
 147.117.37.227   147.117.37.10        1025         5007  Established
 147.117.37.227   147.117.37.10        1024         5006  Established
```

The following example searches the connection table for any entries in the Established (5) state (discussed on the preceding page).

```
pSH+> tcpcon -c 5
TCP Connection Table: Wed May 25 09:08:32 1994
       Local-IP       Remote-IP  Local-Port  Remote-Port  State
 147.117.37.227  147.117.37.227        1028         1030  Established
 147.117.37.227  147.117.37.227        1030         1028  Established
 147.117.37.227  147.117.37.243          23         1064  Established
 147.117.37.227   147.117.37.11        1027         5016  Established
 147.117.37.227   147.117.37.11        1026         5017  Established
 147.117.37.227   147.117.37.10        1025         5007  Established
 147.117.37.227   147.117.37.10        1024         5006  Established
```

The following example searches for any entry involving both IP address 147.117.37.245 and port 1024. The resultant message indicates that no entries with this combination were found.

```
pSH+> tcpcon -i 147.117.37.245 -p 1024
TCP Connection Table: Wed May 25 09:09:39 1994
       Local-IP       Remote-IP  Local-Port  Remote-Port  State
<< No connections found that match specified search option(s). >>
```

3.6. UTILITY COMMANDS

The following commands provide general purpose utilities.

3.6.1. clear

clear -- clears the terminal screen

USAGE:

clear

DESCRIPTION:

clear attempts to clear the current terminal screen. It is an alternative to the ***clr*** command. User preference between ***clr*** and ***clear*** depends on the terminal characteristics and network connection type.

See also ***clr***.

OPTIONS:

None.

3.6.2. clr

clr -- clears the terminal screen

USAGE:

clr

DESCRIPTION:

clr clears a terminal display, such as a VT100, or xterm connection. It is an alternative to the **clear** command. User preference between ***clr*** and ***clear*** depends on the terminal characteristics and network connection type.

OPTIONS:

None.

3.6.3. exit

exit -- exits the shell

USAGE:

exit

DESCRIPTION:

exit exits (i.e., logs out) the user from the shell.

OPTIONS:

None.

3.6.4. help

help -- provides help about shell commands

USAGE:

help [command_name]

DESCRIPTION:

help prints to the console information about shell commands. If no ***command_name*** is given, ***help*** prints out a list of available commands. If a valid ***command_name*** is given, help prints out information about that command.

OPTIONS:

None.

EXAMPLE:

```
pSH+> help
ftp     telnet
cat     cmp     echo    help    mkfs    pcmount pwd     setenv  suspend umount
cd      cp      getid   kill    mount   ping    resume  setid   sync
clear   date    getpri  ls      mv      popd    rm      setpri  tail
console du      head    mkdir   pcmkfs  pushd   rmdir   sleep   touch
bcs     more    tar     scsi    clone   netstat reboot  cal     product
passwd  clr     mt      df      mac     tcpcon  settime cam     loopback
version lp      backup  purge   route   stats   wan


pSH+> help cat
cat      - concatenate and display (reentrant, not locked)
```

NOTES:

help on an individual command provides two additional types of information: whether the command is reentrant, and whether it is currently locked. If a command is indicated as reentrant, it may be used simultaneously by multiple shell users. A command indicated as not being reentrant is only available to one user at a time. Lock status indicates if a non-reentrant command is currently in use by another user. If a user attempts to execute a command that is currently locked, a message indicating "`Command not reentrant`" will be displayed.

3.6.5. lp

lp -- prints a text file

USAGE:

lp file_name

DESCRIPTION:

lp queues the text file specified by ***file_name*** to be output on the optional BCU/CAL printer. ***file_name*** may be either an absolute or relative file specification on the BCU/CAL system disk. ***lp*** does not support printing files from a floppy disk.

OPTIONS:

None.

3.7. STATISTICS AND DIAGNOSTICS COMMANDS

The following commands provide statistical and diagnostics services associated with the BCU/CAL and its operation within an EDACS infrastructure.

3.7.1. cal

cal -- maintenance command for the CAL application

USAGE:

cal -rcphtg [-s site/x] [-l on/off]

DESCRIPTION:

cal provides statistical information and logging for the CAL feature of the BCU/CAL. This command allows access to general status information of all CAL sites and detailed information on a per-site basis. The status of the system manager communications link, activity queue, error detection, and link log can be displayed.

The activity queue(s) may be purged for individual sites or all sites and the activity dump threshold may be set.

Logging of the system manager communications link may be enabled or disabled. The communications link may be reset using the CAL command; this would initiate the link recovery mechanism without waiting for a link timeout.

OPTIONS:

- r Resets site statistics. Option **-s** must be used.
- c Restarts site-system manager communications link(s).
- p Purges site activity for a particular site or all sites. Option **-s** must be used.
- h Provides help. Supplies a brief description of command options.
- t Sets activity dump threshold. Defines the level at which the CAL site will initiate an activity download to the system manager. Must be followed by threshold value 1-20000 (default = 1000).
- g Provides general information. This is equivalent to ***cal*** without any options. Communications link status and activity level are displayed for all CAL sites.
- s Specifies a particular site of interest. ***site*** may range from 1 to 32, inclusive. A ***site*** value of **x** indicates that the operation(s) should be performed for all sites.

- l Logs site-system manager communications. Log info is appended to file 1.2/log/siteXX.tmp where XX corresponds to site ID 01-32. This option must be followed by "on" to enable logging and "off" to disable. The current state of logging is provided in the detailed site statistics.

3.7.2. cam

cam -- logs CAM messages

USAGE:

cam -ocvqpPtThw [-s site] [-m mask]

DESCRIPTION:

cam allows packets being received from the IMC CAM module to be logged in ASCII format. Received packets may be logged to a disk file, displayed on the BCU/CAL test port, or both. ***cam*** is essentially a diagnostics utility, and its access is restricted to the root user account.

If a reproducible problem is discovered with the BCU/CAL, CAM packets should be logged to disk. The log file (1.2/log/cam.hex) may then be forwarded to Ericsson GE for problem analysis. This utility also applies to diagnosing anomalous IMC call activity.

OPTIONS:

- o Opens the CAM logging file 1.2/log/cam.hex.
- c Closes the CAM logging file.
- v Enables the verbose mode. If CAM logging is going to disk, a brief explanation of each packet will be provided in the file. The explanations are denoted as comments, i.e., preceded with a "#" character.
- q Quiet. Disables verbose mode.
- p Enables test port logging. Received CAM packets will be sent to the BCU/CAL test port (port 2 on the BCU/CAM MVME-712 translation module). The test port may be connected to a "dump terminal," or other serial device. The test port operates at 9600 baud, 1 start bit, 1 stop bit, and no parity. Flow control XON/XOFF characters are ignored on the test port.
- P Disables test port logging.
- t Enables time stamping in the verbose mode. The CAM packet time stamp will be provided in human readable format.
- T Disables time stamping.
- h Provides help. Displays a brief explanation of ***cam*** options.
- w Displays the current ***cam*** option settings.
- s Specifies site filtering of CAM packets. Only packets associated with ***site*** will be logged. ***site*** may range from 1 to 32 to designate a specific site. A site value of **x** indicates that packets from all sites should be logged.
- m Specifies the type of CAM packets that should be logged. ***mask*** is a hexadecimal value which may range from 0 to FFFF, inclusive. Each bit position of the mask value is associated with a particular type of CAM packet. The default value is 01FF.

NOTES:

If CAM logging is enabled, packets detected with errors are preceded with an "#ERROR:" comment.

EXAMPLES:

```
pSH+> cam -ov -s 5 -m 0
pSH+> cam -w
CAM log file   : Open
Test port echo : Off
Verbose mode   : On
Time stamping  : Off
Site logging   : Site 5
Msg trace mask : 0x0000

pSH+> cam -cqp -s x -m 7F
pSH+> cam -w
CAM log file   : Closed
Test port echo : On
Verbose mode   : Off
Time stamping  : Off
Site logging   : All sites
Msg trace mask : 0x007F
```

3.7.3. loopback

loopback -- performs a loopback test on the WanServer ports

USAGE:

loopback port_number

DESCRIPTION:

loopback performs a LAPB/HDLC communications loopback test between WanServer port 0 and the port specified by ***port_number*** (1-3). This command may be used to test the I/O function of the WanServer board, as well as the associated internal wiring of the BCU/CAL. A loopback cable must be connected between WanServer port 0 and the port specified by ***port_number*** for this command's function to be valid. Refer to LBI-38965 for loopback cable specifications. The baud rate of the loopback LAPB connection is 512Kbps.

OPTIONS:

None.

NOTES:

loopback will try for up to 60 seconds to establish a LAPB connection between the designated ports. The subsequent display will be either

```
Connection established between port 0 and <port_number>
```

or

```
Unable to establish connection
```

where "`connection established`" indicates that the loopback test was successful.

3.7.4. scsi

scsi -- queries SCSI bus for active devices

USAGE:

scsi -n

DESCRIPTION:

scsi queries the SCSI bus for an active device, where ***-n*** is the SCSI bus address ID of the target device. Device information, such as the vendor, model number, and storage capacity will be displayed on the terminal. ***n*** = 0 implies that all seven (1-7) SCSI addresses should be queried for devices.

OPTIONS:

None.

EXAMPLES:

The following example queries all seven SCSI IDs for active devices.

```
pSH+> scsi -0
SCSI ID: 1      LUN: 0  Removable:  YES System volume: 1.1
Blocks:  2879           Block Size: 512 bytes
Vendor:  TEAC           Model: FC-1     HF   11
Type:    Direct access (disk)
SCSI ID: 2      LUN: 0  Removable:  NO  System volume: 1.2
Blocks:  2423456        Block Size: 512 bytes
Vendor:  MAXTOR         Model: MXT-1240S
Type:    Direct access (disk)
SCSI ID: 3      LUN: 0  Removable:  YES System volume: 1.3
Blocks:  0              Block Size: 512 bytes
Vendor:  ARCHIVE        Model: Python 28388-XXX
Type:    Sequential access (tape)
SCSI ID: 4      No information available.
SCSI ID: 5      No information available.
SCSI ID: 6      No information available.
SCSI ID: 7      No information available.
```

NOTES:

In the case of removable storage devices (i.e., floppy disk or tape), no information will be available unless the media (i.e., diskette) is installed in the target device.

3.7.5. stats

stats -- displays system statistics

USAGE:

stats -icuqekghrfQ* [-s site]

DESCRIPTION:

stats provides for examination and control of the BCU/CAL MIB statistics at the user interface level. A subset of the BCU/CAL MIB statistics, normally examined by a Network Management station, may be presented in a human readable format.

The primary options indicate the statistics group(s) of interest. By default, these are presented at an aggregate, i.e., IMC level. Several options may be overloaded with a site specification. If a site is specified, statistics group(s) of interest for ***site*** will be displayed.

OPTIONS:

- i Displays the IMC interface group statistics.
- c Displays the call assignment type distribution at the IMC level. If overloaded with a site specification option (**-s**), call type distribution on that site(s) will be displayed.
- u Displays system utilization at the IMC level, which provides a summary of site activity. If overloaded with a site specification, channel utilization on that site(s) will be displayed.
- q Displays system queuing and access information at the IMC level. If overloaded with a site specification, the access distribution on that site(s) will be displayed.
- e Displays a message error summary at the IMC level. If overloaded with a site specification, error distribution on that site(s) will be displayed.
- k Displays a summary of keying events for the site specified by ***site***. Provides a count of channel assignments, drops, and unkeys occurring on the specified site.
- g Displays a general summary of packets, errors, and CDRs at the IMC level. The CDR count is only of significance for the BCU feature.
- h Provides help. Furnishes a brief explanation of the command line options.
- r Resets the statistics counters to zero. Only the super-user root may reset the statistics.
- f Logs the current statistics to a binary file, 1.2/log/profile.bin. The record is written to the file in append mode, and time stamped as to when the statistics were saved.
- Q Quiet. This option is useful with an **-s x** option, which displays statistics for all sites (i.e., 1 to 32 by default, 33 to 64 if **-*** option is specified). The **-Q** options suppress output for sites which contain no statistical information.
- * StarGate sites. The **-*** option overloads site-related options, such as **-u**, to show statistics for sites 33 through 64. The default is to show statistics for sites 1 through 32.
- s Specifies a site of interest for the selected statistic group(s). ***site*** may range from 1 to 64, inclusive. A ***site*** value of **x** indicates that the selected statistics group(s) for all sites should be queried. This value (**x**) is primarily intended for use with the file redirection operators “>” and “>>.”

NOTES:

If a statistics reset (**-r** option) is requested in the presence of other options, the reset request will be serviced last. The time of the last statistics reset is used to calculate the sampling interval presented with ***stats*** inquiries.

The **-u** and **-q** options can be extremely useful in EDACS system tuning and diagnostics. Examples are provided with annotation towards this type of usage.

Examples

IMC Interface Group

The following example provides a high-level summary of message activity between the IMC-CAM and the BCU/CAL WanServer LAPB interface.

```
pSH+> stats -i
IMC Packet Statistics: Wed Jun  1 08:36:37 1994
Sampling Interval: 4 Days, 16 Hours, 32 Minutes, 16 Seconds

imcInPkts    : 114555          imcInOctets    : 2374124
imcInErrors  : 1588            imcInDiscards  : 3618
imcAssigns   : 52772           imcDrops       : 28753
imcUnkeys    : 11328           imcRadioStatus : 5889
imcStarSites : 875             imcPhoneDigits : 1482
imcQueues    : 476             imcDenieds     : 46
imcSysBusy   : 381             imcCnvtCallee  : 136
imcCAMResets : 2               imcTimeChanges : 7209
```

Diagnostic Tip

The example above can serve as a quick check to verify that the BCU/CAL is receiving packets from the IMC-CAM. Issue "***stats -i***" several times. If the communication link is up, the **imcInPkts** count will be increasing.

Call Type Distribution

The following example shows the call type distribution at the IMC level. The display is an indication of the various types of valid channel assignments being received. The values reflect channel activity, and should not be confused with the BCU definition of a "call."

```
pSH+> stats -c
IMC Assignment Distribution: Wed Jun  1 08:36:44 1994
Sampling Interval: 4 Days, 16 Hours, 32 Minutes, 23 Seconds

Assignment Type                Percentage
imcUnitCalls    : 6257         11.86%
imcGroupCalls   : 10709        20.29%
imcRadioToPhone : 16655        31.56%
imcPhoneToRadio : 718          1.361%
imcPhoneToGroup : 166           0.31%
imcDataICalls   : 15288        28.97%
imcDataGCalls   : 26            0.04%
imcConventCalls : 19            0.03%
imcTestICalls   : 2934          5.56%
```

The following example shows how a site specification overloads the **-c** option.

```
pSH+> stats -c -s 1
Site [1] Assignment Statistics: Wed Jun  1 08:36:49 1994
Sampling Interval: 4 Days, 16 Hours, 32 Minutes, 28 Seconds

Assignment Types :              Percentage
siteUnitCalls    : 5729         26.27%
siteGroupCalls   : 6025         27.63%
siteRadioToPhone : 8124         37.26%
sitePhoneToRadio : 324          1.486%
sitePhoneToGroup : 48            0.22%
siteDataICalls   : 55            0.25%
siteDataGCalls   : 0                0%
siteConventCalls : 0                0%
siteTestICalls   : 1499         6.875%
```

System Usage

The following example shows system site usage at the IMC level.

```
pSH+> stats -u
IMC Site Usage Distribution: Wed Jun  1 08:37:18 1994
Sampling Interval: 4 Days, 16 Hours, 32 Minutes, 57 Seconds

Site  Usage    Assigns    Airtime       Site  Usage    Assigns    Airtime
 01: 41.32%      21804      81261        17:     0%          0          0
 02: 7.277%       3840       6238        18:     0%          0          0
 03:     0%          0          0        19:     0%          0          0
 04:     0%          0          0        20:     0%          0          0
 05:  0.02%         11          0        21:     0%          0          0
 06:  0.29%        156        464        22:     0%          0          0
 07: 32.08%      16930       9708        23:     0%          0          0
 08:     0%          0          0        24:     0%          0          0
 09:  0.32%        170        363        25:     0%          0          0
 10:     0%          0          0        26:     0%          0          0
 11:     0%          0          0        27:     0%          0          0
 12:     0%          0          0        28:     0%          0          0
 13: 16.63%       8775      35691        29:     0%          0          0
 14:     0%          0          0        30:     0%          0          0
 15:     0%          0          0        31:     0%          0          0
 16:     0%          0          0        32: 2.058%       1086       2567
```

System Tuning Tips

1. The Usage column expresses the number of channel assignments as a percentage of the total number of assignments for all sites. In its simplest form, this shows the operator which sites are being most heavily utilized. Issuing "***stats -u***" for that site will show whether it is experiencing a significant percentage of inaccessibility (queuing, busy, etc.). If so, this would be an indication that the site in question may be in need of additional audio channels.

2. In the example above, site 13 is the Jessica ISDN PBX site. In addition, its air time is almost one half of that of site 1, which is a standard transmission trunked site. Recall that the **-c** option provides call type distribution statistics. While **-c** is extremely useful, it does not provide the entire system usage picture. However, when used in conjunction with **-u**, the operator can obtain a sense of the call types that are consuming the most system air time.

3. In the example above, site 5 is a conventional site. Note that the air time is zero. Specifically, the BCU/CAL *does not* calculate air time for conventional sites. If a transmission trunked site is exhibiting a statistically low amount of RF usage, it may be an indication that channel (de)allocation messages are being errantly serviced or presented.

   The following example shows how a site specification overloads the **-u** option.

```
pSH+> stat -u -s 1
Site [1] Channel Usage: Wed Jun  1 08:37:26 1994
Sampling Interval: 4 Days, 16 Hours, 33 Minutes, 5 Seconds

Chan  Usage    Assigns    Airtime       Chan  Usage    Assigns    Airtime
 01: 1.004%        219        491        13:     0%          0          0
 02: 23.29%       5079      28231        14:     0%          0          0
 03: 25.71%       5606      22754        15:     0%          0          0
 04: 23.71%       5170      14789        16:     0%          0          0
 05: 26.28%       5730      14994        17:     0%          0          0
 06:     0%          0          0        18:     0%          0          0
 07:     0%          0          0        19:     0%          0          0
 08:     0%          0          0        20:     0%          0          0
 09:     0%          0          0        21:     0%          0          0
 10:     0%          0          0        22:     0%          0          0
 11:     0%          0          0        23:     0%          0          0
 12:     0%          0          0        24:     0%          0          0
```

Diagnostics Tips

The example above illustrates a useful diagnostic capability of the ***stats*** command. The Usage column expresses the number of channel assignments as a percentage of the total assignments for that site. An approximately equal distribution of channel assignment usage indicates that the site is properly rotating its channels.

The Airtime column indicates the number of seconds of actual RF time a channel has provided. Note that for RF time to be calculated, a channel assignment must be balanced with a channel drop, each containing the same call party entity descriptions.

System Errors

```
pSH+> stats -e
IMC Error Distribution: Wed Jun  1 08:38:22 1994
Sampling Interval: 4 Days, 16 Hours, 34 Minutes, 1 Seconds

Site Errors Assign   Drop  Unkey  Other  Site Errors Assign   Drop  Unkey  Other
 01:  0.13%  8.22%  46.6%  9.59%  35.6%   17:     0%     0%     0%     0%     0%
 02:  0.04%    50%    50%     0%     0%   18:     0%     0%     0%     0%     0%
 03:     0%     0%     0%     0%     0%   19:     0%     0%     0%     0%     0%
 04:     0%     0%     0%     0%     0%   20:     0%     0%     0%     0%     0%
 05:  4.35%   100%     0%     0%     0%   21:     0%     0%     0%     0%     0%
 06:  0.23%   100%     0%     0%     0%   22:     0%     0%     0%     0%     0%
 07:  5.16%    50%    50%     0%     0%   23:     0%     0%     0%     0%     0%
 08:     0%     0%     0%     0%     0%   24:     0%     0%     0%     0%     0%
 09:  0.48%   100%     0%     0%     0%   25:     0%     0%     0%     0%     0%
 10:     0%     0%     0%     0%     0%   26:     0%     0%     0%     0%     0%
 11:     0%     0%     0%     0%     0%   27:     0%     0%     0%     0%     0%
 12:     0%     0%     0%     0%     0%   28:     0%     0%     0%     0%     0%
 13:     0%     0%     0%     0%     0%   29:     0%     0%     0%     0%     0%
 14:     0%     0%     0%     0%     0%   30:     0%     0%     0%     0%     0%
 15:     0%     0%     0%     0%     0%   31:     0%     0%     0%     0%     0%
 16:     0%     0%     0%     0%     0%   32:  0.08%   100%     0%     0%     0%
```

In the example on the preceding page, the Errors column indicates the percentage of messages received in error. The other columns show the relative percentages of the message types for which the errors occurred.

The following example shows how a site specification overloads the **-e** option.

```
pSH+> stats -e -s 1
Site [1] Error Statistics: Wed Jun  1 08:38:29 1994
Sampling Interval: 4 Days, 16 Hours, 34 Minutes, 8 Seconds

siteInPkts      : 54117         siteInErrors    : 73
siteErrAssigns  : 6             siteErrDrops    : 34
siteErrUnkeys   : 7             siteErrOther    : 26

siteNullCallee  : 18            siteNullCaller  : 29
siteBadCallee   : 0             siteBadCaller   : 0
siteNullChannel : 29            siteBadChannel  : 0
siteBadGroup    : 0             siteBadCallType : 0
siteBadDate     : 0             siteBadDigits   : 26
```

In the example above, the first portion of the display provides the aggregate count of encountered errors. The second portion summarizes the type of errors detected. Note that a message may contain more than one error, as indicated in the example above.

Keying Activity

The following example shows how the radio keying event types may be examined on a particular site.

```
pSH+> stats -k -s 1
Site [1] Keying Activity: Wed Jun  1 08:39:02 1994
Sampling Interval: 4 Days, 16 Hours, 34 Minutes, 41 Seconds

Chan    Assigns      Drops     Unkeys   Chan    Assigns      Drops     Unkeys
 01:        219        106        153    13:          0          0          0
 02:       5079       3325       2211    14:          0          0          0
 03:       5606       3613       2432    15:          0          0          0
 04:       5170       3172       2418    16:          0          0          0
 05:       5730       3618       2517    17:          0          0          0
 06:          0          0          0    18:          0          0          0
 07:          0          0          0    19:          0          0          0
 08:          0          0          0    20:          0          0          0
 09:          0          0          0    21:          0          0          0
 10:          0          0          0    22:          0          0          0
 11:          0          0          0    23:          0          0          0
 12:          0          0          0    24:          0          0          0
```

System Accessibility

```
pSH+> stats -q
Site Queueing Distribution: Wed Jun  1 08:37:53 1994
Sampling Interval: 4 Days, 16 Hours, 33 Minutes, 32 Seconds

Site  Queued    Busy  Denied Convert  Site  Queued    Busy  Denied Convert
 01:   2.01%   1.27%   0.01%   0.58%   17:      0%      0%      0%      0%
 02:   0.48%    0.3%   1.07%    0.1%   18:      0%      0%      0%      0%
 03:      0%      0%      0%      0%   19:      0%      0%      0%      0%
 04:      0%      0%      0%      0%   20:      0%      0%      0%      0%
 05:      0%      0%      0%      0%   21:      0%      0%      0%      0%
 06:      0%   1.27%      0%      0%   22:      0%      0%      0%      0%
 07:      0%   0.44%      0%      0%   23:      0%      0%      0%      0%
 08:      0%      0%      0%      0%   24:      0%      0%      0%      0%
 09:      0%      0%      0%      0%   25:      0%      0%      0%      0%
 10:      0%      0%      0%      0%   26:      0%      0%      0%      0%
 11:      0%      0%      0%      0%   27:      0%      0%      0%      0%
 12:      0%      0%      0%      0%   28:      0%      0%      0%      0%
 13:      0%   0.04%      0%      0%   29:      0%      0%      0%      0%
 14:      0%      0%      0%      0%   30:      0%      0%      0%      0%
 15:      0%      0%      0%      0%   31:      0%      0%      0%      0%
 16:      0%      0%      0%      0%   32:      0%   0.09%      0%      0%
```

The following example shows how a site specification overloads the **-q** option.

```
pSH+> stats -q -s 1
Site [1] Access Statistics: Wed Jun  1 08:37:59 1994
Sampling Interval: 4 Days, 16 Hours, 33 Minutes, 38 Seconds

22684 Channel Assignment Attempts.

siteAssigns    :      21804     => 96.12% Successful assignment
siteQueues     :        457     => 2.015% Queueing
siteSysBusy    :        287     => 1.265% System Busy
siteDenieds    :          4     =>  0.01% Denied
siteCnvtCallee :        132     =>  0.58% Converted to Callee
```

In the example above, the Assigns column indicates the “success” rate of the number of channel assignment attempts made.

General Information

```
pSH+> stats -g
Site Activity Summary: Wed Jun  1 08:39:25 1994
Sampling Interval: 4 Days, 16 Hours, 35 Minutes, 4 Seconds

Site  RAR's      CDR's     Errors       Site  RAR's      CDR's     Errors
 01 5.4117e+04 2.7740e+03 7.3000e+01     17 0.0000e+00 0.0000e+00 0.0000e+00
 02 8.5280e+03 1.6500e+02 4.0000e+00     18 0.0000e+00 0.0000e+00 0.0000e+00
 03 0.0000e+00 0.0000e+00 0.0000e+00     19 0.0000e+00 0.0000e+00 0.0000e+00
 04 0.0000e+00 0.0000e+00 0.0000e+00     20 0.0000e+00 0.0000e+00 0.0000e+00
 05 2.3000e+01 0.0000e+00 1.0000e+00     21 0.0000e+00 0.0000e+00 0.0000e+00
 06 4.1700e+02 1.0000e+00 1.0000e+00     22 0.0000e+00 0.0000e+00 0.0000e+00
 07 2.9216e+04 4.9000e+01 1.5070e+03     23 0.0000e+00 0.0000e+00 0.0000e+00
 08 0.0000e+00 0.0000e+00 0.0000e+00     24 0.0000e+00 0.0000e+00 0.0000e+00
 09 2.0700e+02 3.0000e+00 1.0000e+00     25 0.0000e+00 0.0000e+00 0.0000e+00
 10 0.0000e+00 0.0000e+00 0.0000e+00     26 0.0000e+00 0.0000e+00 0.0000e+00
 11 0.0000e+00 0.0000e+00 0.0000e+00     27 0.0000e+00 0.0000e+00 0.0000e+00
 12 0.0000e+00 0.0000e+00 0.0000e+00     28 0.0000e+00 0.0000e+00 0.0000e+00
 13 1.2823e+04 0.0000e+00 0.0000e+00     29 0.0000e+00 0.0000e+00 0.0000e+00
 14 0.0000e+00 0.0000e+00 0.0000e+00     30 0.0000e+00 0.0000e+00 0.0000e+00
 15 0.0000e+00 0.0000e+00 0.0000e+00     31 0.0000e+00 0.0000e+00 0.0000e+00
 16 0.0000e+00 0.0000e+00 0.0000e+00     32 1.1380e+03 2.0000e+00 1.0000e+00
```

stdout Redirection

The following examples show the ***stats*** command when used with the UNIX redirection operators. Recall that the "**>**" and "**>>**" operators may be used to redirect output normally destined for the standard output device (i.e., terminal) to a file. The operation "**>** ***file_name***" redirects output to the file specified by ***file_name***. If ***file_name*** exists, it will be overwritten. The operation " >> ***file_name*** " also redirects output to the file specified by ***file_name***. In this case, however, if ***file_name*** exists, output will be appended to the file.

The ***stats*** command uses a ***site*** specification of "**x**" to indicate all sites. This is not particularly useful at the console since the information presented would be scrolling off of the screen. In the following example, the call types, errors, and queuing information for all 32 sites would be written to the file my_file.dat. Note that if my_file.dat already existed, it would be overwritten.

```
pSH+> stats -ceq -s x  >  my_file.dat
```

In the following example, the IMC group level and utilization statistics information would be appended to the file my_file.dat.

```
pSH+> stats -iu  >> my_file.dat
```

The file my_file.dat may now be examined at leisure, with the ***more*** command, or copied to floppy disk, for example.

3.7.6. wan

wan -- displays WanServer status

USAGE:

wan -s

DESCRIPTION:

wan provides a brief summary of the WanServer to IMC-CAM communications linkup time.

OPTIONS:

- s Shows WanServer information

EXAMPLE:

```
pSH+> wan -s
WanServer Statistics   : Tue Jul  5 12:04:31 1994
System boot time       : Tue Jun 28 15:07:24 1994

Total WanServer resets : 1
Last WanServer reset   : Tue Jun 28 15:07:38 1994

Total IMC-CAM resets   : 3
Last IMC-CAM reset     : Thu Jun 30 16:15:20 1994
```

3.8. ENVIRONMENT COMMANDS

The following commands are associated with subtle characteristics of a user's operational environment. These commands are seldom, if ever, used.

3.8.1. getid

getid -- obtains user ID and group ID

USAGE:

getid

DESCRIPTION:

getid displays the user ID (uid) and group ID (gid) of the shell user.

OPTIONS:

None.

EXAMPLE:

```
pSH+> getid
uid: 20, gid: 100
```

3.8.2. popd

popd -- pops the directory stack

USAGE:

popd

DESCRIPTION:

popd pops the directory stack, and changes the current working directory to the new top directory.

OPTIONS:

None.

EXAMPLE:

```
pSH+> pushd cdr
pSH+> pwd
1.2/cdr
pSH+> popd
pSH+> pwd
1.2/
```

3.8.3. pushd

pushd -- pushes the current directory onto the directory stack

USAGE:

pushd directory

DESCRIPTION:

pushd pushes ***directory*** onto the directory stack, and changes the current working directory to that directory.

OPTIONS:

None.

EXAMPLE:

```
pSH+> pwd
1.2/
pSH+> pushd activity
pSH+> pwd
1.2/activity
```

3.8.4. setenv

setenv -- sets environment variables

USAGE:

setenv variable_name value

DESCRIPTION:

setenv changes a shell's variables to a new value. If used without any arguments, ***setenv*** prints a list of the shell variables and their current values.

OPTIONS:

None.

EXAMPLE:

```
pSH+> setenv
CVOL=1.2
CDIR=/
SOFLIST=5
LOGNAME=root
IND=0
OUTD=0
TERM=sun
pSH+> setenv TERM vt100
CVOL=1.2
CDIR=/
SOFLIST=5
LOGNAME=root
IND=0
OUTD=0
TERM=vt100
```

NOTES:

Currently, the only variable that can be changed is TERM.

3.8.5. setid

setid -- sets user ID and group ID

USAGE:

setid uid gid

DESCRIPTION:

setid changes the shell user's user ID to ***uid***, and group ID to ***gid***.

OPTIONS:

None.

EXAMPLE:

```
pSH+> getid
uid: 20, gid: 100
pSH+> setid 20 169
pSH+> getid
uid: 20, gid: 169
```

3.9. ADVANCED COMMANDS

The following section discusses commands which, in general, should not be used on the BCU/CAL. Reasons for avoiding their use are summarized below. The commands are provided for factory troubleshooting.

1. Similar functions are provided which are uniquely tailored to the functional requirements of BCU/CAL.

2. Their function is superfluous to the operations of the BCU/CAL. That is, the function that the command provides either is not applicable or is provided under different context. For example, the ***mkfs*** and ***mount*** commands are somewhat meaningless since the BCU/CAL's system disk is maintained under application control.

3. Their incorrect use may result in degraded or catastrophic system behavior (e.g., incorrect use of the **kill** command may result in the BCU/CAL's crashing).

3.9.1. console

console -- redirects the console output to a telnet session

USAGE:

console [-r] [task_name]

DESCRIPTION:

console redirects output going to the BCU/CAL's system console to a telnet session. The default is to redirect all output to the telnet session. If a ***task_name*** is given, only the output from that task will be redirected to the telnet session.

OPTIONS:

- r Redirects input from the telnet session. Note that if a task is currently waiting for console input when this command is issued, the task's input redirection will take effect only after it returns from the waiting.

EXAMPLE:

Telnet into the BCU/CAL and enter: **`pSH+> console -r ROOT`**

All output from the task "ROOT" will be redirected to the telnet session, and the task will get its input from the telnet session.

NOTES:

There is no graceful way to undo ***console*** redirection. Use of the console command should be avoided.

3.9.2. echo

echo -- echoes arguments to the standard output

USAGE:

echo [-n] [argument ...]

DESCRIPTION:

echo writes its arguments on the standard output. ***argument***(s) must be separated by SPACE characters or TAB characters, and terminated by a NEWLINE character.

OPTIONS:

- n Does not add the NEWLINE to the output.

3.9.3. getpri

getpri -- displays the priority of a task

USAGE:

getpri task_name | -task_id

DESCRIPTION:

getpri displays the priority of a task named ***task_name***, or with a task ID of ***task_id***.

OPTIONS:

None.

EXAMPLE:

The following example provides the current priority of the BCU/CAL's telnet server daemon.

```
pSH+> getpri tnpd
tnpd task priority = 50
```

3.9.4. kill

kill -- terminates a task

USAGE:

kill task_name | -task_id

DESCRIPTION:

kill terminates a task named ***task_name***, or with an ID of ***task_id***. It does this by calling t_restart with a second argument of **1**. The task must be designed to read this second argument and perform its own resource cleanup, then terminate.

OPTIONS:

None.

EXAMPLE:

The following example kills the ftp server daemon on the BCU/CAL.

```
pSH+> kill ftpd
```

3.9.5. mkfs

mkfs -- makes a file system (volume initialization)

USAGE:

mkfs [-i] volume_name label size num_of_fds

DESCRIPTION:

mkfs initializes a file system volume ***volume_name*** and labels it with ***label***. Its size will be ***size*** and the number of file descriptors will be ***num_of_fds***.

OPTIONS:

- i Calls device driver initialization procedure.

3.9.6. mount

mount -- mounts a file system volume

USAGE:

mount volume_name [sync_mode]

DESCRIPTION:

mount mounts a pHILE+ formatted volume ***volume_name***. A volume must be mounted before any file operations can be carried out on it. ***sync_mode*** specifies one of the following file system synchronization methods for the volume:

0 = Immediate write synchronization mode.
1 = Control write synchronization mode.
2 = Delayed write synchronization mode (default).

Permanent (i.e., non-removable media) volumes need only be mounted once. Removable volumes must be mounted and unmounted as required.

OPTIONS:

None.

3.9.7. resume

resume -- resumes a suspended task

USAGE:

resume task_name | -task_id

DESCRIPTION:

resume will resume a task named ***task_name***, or with an ID of ***task_id***, that was previously suspended.

OPTIONS:

None.

EXAMPLE:

The following example would resume the BCU/CAL telnet server daemon, had it been previously suspended.

```
pSH+> resume tnpd
```

3.9.8. setpri

setpri -- sets task priority

USAGE:

setpri task_name | -task_id new_priority

DESCRIPTION:

setpri sets the priority of a task named ***task_name***, or with an ID of ***task_id***, to ***new_priority***.

OPTIONS:

None.

EXAMPLE:

The following example adjusts the priority of the BCU/CAL's telnet server daemon.

```
pSH+> getpri tnpd
tnpd task priority = 50
pSH+> setpri tnpd 100
pSH+> getpri tnpd
tnpd task priority = 100
```

3.9.9. sleep

sleep -- suspends execution for the specified interval

USAGE:

sleep time

DESCRIPTION:

sleep suspends execution for ***time*** seconds.

OPTIONS:

None.

3.9.10. suspend

suspend -- suspends a task

USAGE:

suspend task_name | -task_id

DESCRIPTION:

suspend suspends a task named ***task_name***, or with an ID of ***task_id***.

OPTIONS:

None.

EXAMPLE:

The following example would suspend the BCU/CAL telnet server daemon.

```
pSH+> suspend tnpd
```

ERICSSON

Ericsson GE Mobile Communications Inc.
Mountain View Road • Lynchburg Virginia 24502

Printed in U.S.A.

APPENDIX A
BCS-BCU/CAL CONFIGURATION SERVICE

This section discusses the BCU/CAL Configuration Service (BCS) user interface. BCS is an interactive program which is used to configure and control the main functions of the BCU and several CAL functions. BCS is invoked from the pSH+> prompt by entering ***bcs***. Once started, BCS is an autonomous command interpreter, with its own prompt, command syntax, help facility, and error messages.

The following example shows invocation of the BCS program. The welcome banner will advise the user of his security access level. System Administrator level is granted to either the "root" or "admin" user accounts, which allows complete access to all BCS command functions. All other user accounts are *not* privileged for system administration. Specifically, non-administrative accounts may query the BCS configuration parameters, and execute the help facilities. They cannot, however, alter any of the parameters.

```
pSH+> bcs

Welcome to the BCU-CAL Configuration Service (BCS)

System Administrator privilege acknowledged.
   - or -
This account is not privileged for System Administration.

BCS>
```

Important Notes

1. If BCS is to be executed from a remote login (i.e., telnet session), the session *must* be established through an ***xterm***, or VT100 compatible, connection point. BCS does not currently support remote shells, such as the UNIX “C-Shell” (/bin/csh). Attempts to invoke BCS from other than the aforementioned remote connection methods may cause unpredictable system interface behavior. A remote user is warned of this possibility by a login message similar to the following:

   ```
   WARNING - REMOTE CONSOLE USER!
           BCS ONLY SUPPORTS XTERM-VT100 COMPATIBLE SESSION WINDOWS.
           ACCESS FROM SunOS /bin/csh CMDTOOL WILL NOT WORK CORRECTLY.

           IF YOUR SESSION IS NOT AN XTERM-VT100 CONNECTION TYPE,
           EXIT IMMEDIATELY!
   ```

2. The BCS utility is restricted to a single user at a time. If someone else is currently running BCS when a new user attempts to execute it, messages similar to the those shown below will be displayed. The message indicates to the denied user which account is currently running BCS, and whether the account is executing from a local or remote console. The PID value indicates the process ID associated with the current BCS user.

   ```
   pSH+> bcs
   BCS is currently in use by login account: admin
   admin is logged in at a remote terminal. PID = 0x008E0000
   pSH+>
   ```

   In the example above, user *admin* is currently executing BCS from a remote telnet session.

   ```
   pSH+> bcs
   BCS is currently in use by login account: root
   root is logged in at the local console. PID = 0x00870000
   pSH+>
   ```

   In the example above, user *root* is currently executing BCS at the local terminal.

1. BCS COMMAND SYNTAX

The BCS user interface operates similar to VAX VMS Digital Command Language (DCL), but it is not necessary to know DCL to use BCS. A command line may be composed of a command verb, command noun, qualifying statements, and an argument list. A definition of the command line syntax as well as the relationship of these command elements to one another is described in the following sections.

The command verb defines the intent of the command line. BCS allows the following command verbs:

- SHOW
- SET
- HELP
- QUIT
- EXIT
- LOG
- HALT
- RESTART
- CLOSE

The command noun specifies the object of a preceding command verb. BCS allows the following command nouns:

- SYSTEM
- GROUP
- UNIT

Command qualifiers further define the noun portion of the command. Qualifiers are specified as a keyword, preceded by a forward slash ("/"). If a qualifier is to alter a parameter (such as the SET command), the keyword is followed by an equals sign ("="), and then by the desired value. Multiple qualifiers may be used with a single command verb and noun statement.

The argument list specifies which unit or group IDs are to be affected by the command statement. The list specification may contain single values, ranges of values, or a combination thereof. A range of IDs is specified as two numbers separated by a hyphen ("-"). Single numbers and/or ranges are separated by a comma (","). If the argument list is omitted, the full range of IDs is implied (default).

All commands may be abbreviated using the minimal number of unique characters required to distinguish them from other command elements.

Spaces before and after "/," "=," "," and "-" delineators are not required and are ignored if present. The syntax is not case-sensitive.

Syntax errors are reported with the incorrect portion of the command line shown in angle brackets, ("<...>").

2. BCS COMMAND SUMMARY

This section summarizes the command verbs, nouns, and qualifiers implemented by BCS.

2.1. COMMAND VERBS

Table A.1 - BCS Command Verbs

Command Verbs	Meaning
SHOW	Displays parameters associated with a command noun. Valid command nouns are SYSTEM, UNIT, and GROUP.
SET	Sets parameters associated with a command noun. Valid command nouns are SYSTEM, UNIT, and GROUP.
HELP	Obtains interactive on-line help.
CLOSE	Closes an open billing file (.CTM extension) in the "1.2/cdr" directory. The closed file will have a .CDR extension.
HALT	Closes the communication port to the IMC CAM. Any queued activity records in the system are processed, and all open billing-related files are closed. This command should be used to shut down the BCU/CAL gracefully.
RESTART	Used after the HALT command to reopen communication with the IMC CAM. This command is not used under normal operating conditions.
LOG	Ages any open log files in the "1.2/log" directory. Opened log files are closed and saved to a .log extension, and new log files are opened with a .tmp extension.
EXIT	Exits BCS and returns to the pSH+> prompt. A synonym for QUIT.
QUIT	Quits BCS and returns to the pSH+> prompt. A synonym for EXIT.

2.2. COMMAND NOUNS

Table A.2 - BCS Command Nouns

Command Nouns	Meaning
SYSTEM	Specifies the command related to the BCS system parameters file - SYSTEM.BIN.
UNIT	Specifies the command related to the BCS unit parameters file - UNIT.BIN.
GROUP	Specifies the command related to the BCS group parameters file GROUP.BIN.

2.3. SYSTEM QUALIFIERS

Table A.3 - BCS System Qualifiers

System Parameters	Meaning
UNIT_HANGTIME	Default pseudo hang time to be used for individual call billing. The hang time value is specified in seconds, and may range from 0 to 60.
GROUP_HANGTIME	Default pseudo hang time to be used for group call billing. The hang time value is specified in seconds, and may range from 0 to 60.
BILLING	Default billing mode for group calls. Displayed as either Caller or Callee. For SET, the specified value may be `CALLEE` (to bill the group) or `NORMAL` to bill the Caller.
DATA_HANGTIME	Pseudo hang time for landline (EDG) data calls. The hang time value is specified in seconds, and may range from 0 to 60.
DATA_BILLING	Enables/disables CDR generation for landline (EDG) data calls.

Table A.3 - BCS System Qualifiers (Cont.)

System Parameters	Meaning
CDR_FILE_SIZE	Maximum number of CDRs that may be written to a single file. For SET, the value is specified in number of records, and may range between 0 and 10000, inclusive. A file size of zero indicates that a single CDR will be allowed to grow until the time of day specified by OFFLOAD_TIME.
SEQUENCE_WRAPPING	Controls whether CDR Radix-64 sequence numbers "wrap" across consecutive CDR files (On), or if each new CDR file starts at sequence number 1 (Off).
PHONE_DIGITS_NEEDED	Allows suppression of CDRs for radio-to-landline calls that do not contain the phone digits dialed (On). Off indicates that air time for incomplete calls will be billed.
LOG_ACTIVITY	State of the RAR logging feature (either ON or OFF).
RAR_FILE_SIZE	Maximum number of RARs that may be written to a single file. For SET, the value is specified in number of records, and may range between 1 and 10000, inclusive.
TAPE_LOG	State of DAT tape logging feature (either ON or OFF). Should *only* be enabled if the BCU/CAL is configured with an internal tape drive (customer option).
OFFLOAD_TIME	Time of day closed CDR files will be copied to the tape drive if TAPE_LOG is enabled. If CDR_FILE_SIZE is zero, the current CDR file will be closed daily at this time. For SET, the time is specified as `hh:mm:ss`, where; `hh` is hours (00 to 23), `mm` is minutes (00 to 59), and `ss` is seconds (00 to 59). Leading zeros must be provided for values less than 10.
ARCHIVE_LIFETIME	Number of days an archived CDR (*.CDP) file will remain on the system, if TAPE_LOG is enabled. BCS deletes a .CDP file from the system when it has aged to the specified archive lifetime. The value is defined in days and may range from 0 to 30, inclusive.
SYSTEM_ID	Network system number of the area for which the BCU/CAL is providing service. The value may range from 0 to 255.
NODE_ID	Local node number of the BCU/CAL. The value may range from 0 to 255.
IMC_TIME_SYNC	Allows the BCU/CAL system time and date to be synchronized to IMC-CAM time updates (either ON or OFF).
NIM_SLOT	The IMC slot number that this node's NIM occupies. This parameter is only of significance if the BCU/CAL is operating within a StarGate configuration. The value may range from 0 to 32, inclusive. A value of 0 indicates that the IMC does not have a NIM installed.
PIM_SLOT	The IMC slot number that this node's PIM occupies. This parameter is only of significance if the BCU/CAL service area (IMC) has a Jessica ISDN PBX installed. The value may range from 0 to 32, inclusive. A value of 0 indicates that the IMC does not have a PIM installed.

NOTES:

If RAR logging is enabled, Raw Activity Records received from the IMC CAM -- and subsequently used for CDR generation -- will be saved to disk. RAR logging will consume disk space normally available for CDR storage. RAR log files are not automatically archived and purged as CDR files are. These data are a copy of IMC Central Activity Monitoring and are not post-processed to conform to EDACS System Manager format. Use of the RAR logging feature is highly discouraged.

2.4. UNIT QUALIFIERS

Table A.4 - BCS Unit Qualifiers

Unit Parameters	Meaning
HANGTIME	Pseudo hang time to be used for this individual's call billing. The hang time value is specified in seconds, and may range from 0 to 60.
DEFAULT	The system default pseudo hang time for individual call billing.

2.5 GROUP QUALIFIERS

Table A.5 - BCS Group Qualifiers

Group Parameters	Meaning
HANGTIME	Pseudo hang time to be used for this group's call billing. The hang time value is specified in seconds, and may range from 0 to 60.
BILLING	The assigned value (SET command verb) may be CALLEE (to bill the group) or NORMAL to bill the Caller.
DEFAULT	The system default pseudo hang time and billing mode for group call billing.

3. BCS EXAMPLES

The following sections provide examples of BCS commands. The actual operator entries are show in **bold** print The SHOW SYSTEM, SHOW UNIT, and SHOW GROUP example subsections were performed immediately after initial software installation. Thus, the values shown represent the default parameters used when the system files were created.

3.1. SHOW SYSTEM

The following example displays all of the BCS system parameters.

```
BCS> show system
Default unit pseudo hangtime = 10
Default group pseudo hangtime = 10
Default group billing mode = Bill callee
Data call pseudo hangtime = 10
Data call billing = On
Phone digits mandatory = Off
CDR maximum file records = 1024
CDR sequence wrapping = On
Offload time = 00:00:00
Tape logging = Off
CDR archive(s) lifetime = 7
IMC time synchronization = Off
RAR activity logging = Off
RAR maximum file records = 10000
System ID = 0
Node ID = 0
NIM slot = 0
PIM slot = 0
```

The following example shows how individual entries in the system table can be displayed.

```
BCS> show system /unit_hang /tape_log
Default unit pseudo hangtime = 10
Tape logging = Off
BCS>
```

3.2. SHOW UNIT

The following example displays how unit parameters are displayed when a list is not specified.

The `< Default values >` next to a unit ID indicates that the billing parameters are being obtained from the system table's default UNIT_HANGTIME value.

```
BCS> show unit
Unit ID = 1      < Default values >
Pseudo hangtime = 10

Unit ID = 2      < Default values >
Pseudo hangtime = 10

Unit ID = 3      < Default values >
Pseudo hangtime = 10

Unit ID = 4      < Default values >
Pseudo hangtime = 10

Unit ID = 5      < Default values >
Pseudo hangtime = 10

Unit ID = 6      < Default values >
Pseudo hangtime = 10

Unit ID = 7      < Default values >
Pseudo hangtime = 10

Continue display Y/N?n
BCS>
```

The following example shows how unit parameters are displayed when a unit list is provided.

```
BCS> show unit 100,500,1500-1503
Unit ID = 100    < Default values >
Pseudo hangtime = 10

Unit ID = 500    < Default values >
Pseudo hangtime = 10

Unit ID = 1500   < Default values >
Pseudo hangtime = 10

Unit ID = 1501   < Default values >
Pseudo hangtime = 10

Unit ID = 1502   < Default values >
Pseudo hangtime = 10

Unit ID = 1503   < Default values >
Pseudo hangtime = 10
```

The following example shows the default pseudo hang time value for individual calls. This command is analogous to entering "show system /unit_hangtime."

```
BCS> show unit /default
 Default unit pseudo hangtime = 10
```

3.3. SHOW GROUP

The following example displays how group parameters are displayed when a list is not specified.

The < Default values > next to a group ID indicates that the billing parameters are being obtained from the system table's default GROUP_HANGTIME and BILLING values.

```
BCS> show group
Group ID = 0      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 1      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 2      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 3      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 4      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Continue display Y/N?n
BCS>
```

The following example shows how group parameters are displayed when a list is specified.

```
BCS> show group 100,2010-2013
Group ID = 100    < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 2010   < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 2011   < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 2012   < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee
```

```
Group ID = 2013   < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Continue display Y/N?n
BCS>
```

The following example shows how the default group parameters may be displayed. This command is analogous to entering "show system /group_hangtime /billing."

```
BCS> show group /default
 Default group pseudo hangtime = 10
 Default billing mode = Bill callee
BCS>
```

The following example shows how only the group billing mode may be displayed.

```
BCS> show group /billing 1-3
Group ID = 1      < Default values >
Billing mode = Bill callee

Group ID = 2      < Default values >
Billing mode = Bill callee

Group ID = 3      < Default values >
Billing mode = Bill callee
BCS>
```

The following example shows how only the group pseudo hang time may be displayed.

```
BCS> show group /hangtime 1-3
Group ID = 1      < Default values >
Pseudo hangtime = 10

Group ID = 2      < Default values >
Pseudo hangtime = 10

Group ID = 3      < Default values >
Pseudo hangtime = 10
BCS>
```

3.4. SET SYSTEM

The following example sets several different system parameters, then uses "show system" to verify that they were set correctly. The values entered were randomly chosen, and are provided for example purposes ***only***.

```
BCS> set system /nim = 32 /pim = 16
BCS> set system /sys = 10 /node = 1
BCS> set system /cdr_file = 100
BCS> set system /tape_log = on /offload = 12:30:00
BCS> set system /log_act = on /rar_file = 5000
BCS> set system /archive = 14
BCS> set system /imc_time = On
```

```
BCS> show system
Default unit pseudo hangtime = 10
Default group pseudo hangtime = 10
Default group billing mode = Bill callee
Data call pseudo hangtime = 10
Data call billing = On
Phone digits mandatory = Off
CDR maximum file records = 100
CDR sequence wrapping = On
Offload time = 12:30:00
Tape logging = On
CDR archive(s) lifetime = 14
IMC time synchronization = On
RAR activity logging = On
RAR maximum file records = 5000
System ID = 10
Node ID = 1
NIM slot = 32
PIM slot = 16
```

3.5. SET UNIT

The following example shows how to set unit pseudo hang time value(s). It is followed by a "show unit" to clarify that the parameters have been adjusted as desired.

```
BCS> set unit/hangtime = 30 1,3,5-7
BCS> show unit
Unit ID = 1
Pseudo hangtime = 30

Unit ID = 2      < Default values >
Pseudo hangtime = 10

Unit ID = 3
Pseudo hangtime = 30

Unit ID = 4      < Default values >
Pseudo hangtime = 10

Unit ID = 5
Pseudo hangtime = 30

Unit ID = 6
Pseudo hangtime = 30

Unit ID = 7
Pseudo hangtime = 30

Continue display Y/N?n
BCS>
```

The following example shows how to set unit pseudo hang time value(s) back to the system default value. It is followed by a "show unit" to clarify that the parameters have been adjusted as desired.

```
BCS> set unit /default 3,6
BCS> show unit
Unit ID = 1
Pseudo hangtime = 30

Unit ID = 2      < Default values >
Pseudo hangtime = 10

Unit ID = 3      < Default values >
Pseudo hangtime = 10

Unit ID = 4      < Default values >
Pseudo hangtime = 10

Unit ID = 5
Pseudo hangtime = 30

Unit ID = 6      < Default values >
Pseudo hangtime = 10

Unit ID = 7
Pseudo hangtime = 30

Continue display Y/N?n
BCS>
```

The following examples show that a level of protection is given to the operator if a list was omitted. If the operator responded with "yes," all 16,383 unit IDs would have been set to the system default value.

```
BCS> set unit /default
All 16383 units have been implied for this operation
Are you sure [YES/NO]? no
Command aborted by user
BCS>
```

3.6. SET GROUP

The following example shows how group billing mode and pseudo hang time values may be set for various group IDs. It is followed by a "show group" to clarify that the parameters have been adjusted as desired.

```
BCS> set group /billing = callee /hangtime = 25 1,3-5
BCS> show group
Group ID = 0      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 1
Pseudo hangtime = 25
Billing mode = Bill callee

Group ID = 2      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee
```

```
Group ID = 3
Pseudo hangtime = 25
Billing mode = Bill callee

Group ID = 4
Pseudo hangtime = 25
Billing mode = Bill callee

Continue display Y/N?n
BCS>
```

The following example shows how just the group billing mode may be set for various group IDs. It is followed by a "show group" to clarify that the parameters have been adjusted as desired.

```
BCS> set group /billing = normal 1,3
BCS> show group
Group ID = 0      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 1
Pseudo hangtime = 25
Billing mode = Bill caller

Group ID = 2      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 3
Pseudo hangtime = 25
Billing mode = Bill caller

Group ID = 4
Pseudo hangtime = 25
Billing mode = Bill callee

Continue display Y/N?n
BCS>
```

The following example shows how just the group pseudo hang time values may be set for various group IDs. It is followed by a "show group" to clarify that the parameters have been adjusted as desired.

```
BCS> set group /hangtime = 30 1,3
BCS> show group
Group ID = 0      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 1
Pseudo hangtime = 30
Billing mode = Bill caller

Group ID = 2      < Default values >
Pseudo hangtime = 10
Billing mode = Bill callee

Group ID = 3
Pseudo hangtime = 30
Billing mode = Bill caller
```

```
Group ID = 4
Pseudo hangtime = 25
Billing mode = Bill callee

Continue display Y/N?n
BCS>
```

The following examples show that a level of protection is given to the operator if a list was omitted. If the operator responded with "yes," all 2,048 group IDs would have been set to the system default values.

```
BCS> set group /default
All 2048 groups have been implied for this operation
Are you sure [YES/NO]? no
Command aborted by user
BCS>
```

3.7. PERMISSION DENIAL

Any attempt by a non-administrative account to alter BCS's parameters or operational state will result in denial messages similar to the following examples. In these examples, the commands were entered from the "user" account.

```
BCS> set unit/hangtime = 30 507
 BCS: permission denied.
 This command requires system administrator privilege.

 BCS> close
 BCS: permission denied.
 This command requires system administrator privilege.
```

4. BCS ERROR MESSAGES

The following section lists the error, warning, and informational messages that BCS can produce and provides explanations of these messages. Portions of the message that are specific to the instance of the message are enclosed in angle brackets ("<...>"). These messages are displayed on the same terminal (local or remote) as the BCS> prompt.

Internal errors should not occur under normal circumstances. These indicate a possible problem with the BCU/CAL hardware or a possible defect in BCU/CAL software.

Table A.5 - BCS Error, Warning, and Informational Messages

WARNING: cnfg_uif.c <line> Failed to create file: <file_name>	Internal error.
WARNING: cnfg_uif.c <line> Failed to open file: <file_name>	Internal error.
WARNING: cnfg_uif.c <line> Failed to seek record in file: <file_name>	Internal error.
WARNING: cnfg_uif.c <line> Failed to read from file: <file_name>	Internal error.
WARNING: cnfg_uif.c <line> Failed to write to file: <file_name>	Internal error.
WARNING: cnfg_uif.c <line> Failed to create directory file: <file_name>	Internal error.
Invalid command verb <verb>	BCS did not recognize the first word of a command.
Invalid command noun <noun>	BCS did not recognize the second word of a command.

Table A.5 - BCS Error, Warning, and Informational Messages (Cont.)

Missing command noun	BCS expected to find a second word.
Invalid qualifier keyword </qualifier>	The qualifier specified is not one of the valid values for the entered command.
Missing qualifier keyword	BCS expected a qualifier for the entered command.
Qualifier value not numeric <qualifier>	BCS expected a numeric value for the qualifier for the entered command.
Qualifier value out of range <value>	The qualifier specified is not within a valid preset range (minimum/maximum).
Invalid qualifier value <value>	An unrecognized qualifier was specified for a command that requires an alphabetical qualifier from a fixed list of possibilities.
Missing qualifier value	A qualifier was entered without the required value (.../qualifier=**value**...).
List entry out of range <entry>	At least one of the entries in your list specification was less than the minimum or greater than the maximum allowable value.
Invalid list syntax <list>	The list specified did not conform to the valid list syntax. Use commas and hyphens.
Incomplete qualifier - supply more characters	A forward slash in a command was not followed by any qualifiers.
Invalid time entry <time>	The time value specified did not conform to the valid time syntax.
Invalid command structure - no value expected	A value was given where none was expected.
Communication error <task>	Internal error
Memory allocation error	Internal error
This account is not privileged for System Administration.	Informational only. BCS was started in User mode.
System Administrator privilege acknowledged.	Informational only. BCS was started in Admin mode.
Permission denied. This command requires administrator privilege.	An Admin mode command was attempted while BCS was in User mode.
You have requested a halt in the billing collection task BILLING DATA MAY BE LOST IF YOU EXECUTE THIS COMMAND	The administrator is about to halt the billing collection task, which will close the communications port from which activity data are being received, causing a probable loss in some billing data.
Restarting billing collection task.	Informational only. The air time correlation daemon was restarted using the RESTART command.
INFO MSG: Creating default system file...	Informational only. The SYSTEM.BIN file was missing, and BCS created a new one.
INFO MSG: Creating default unit parameters file	Informational only. The UNIT.BIN file was missing, and BCS created a new one.
INFO MSG: Creating default group parameters file	Informational only. The GROUP.BIN file was missing, and BCS created a new one.
INFO MSG: Created default LOG directory path <path>	Informational only. The 1.2/log directory was missing, and BCS created it.
INFO MSG: Created default CDR directory path <path>	Informational only. The 1.2/cdr directory was missing, and BCS created it.

Table A.5 - BCS Error, Warning, and Informational Messages (Cont.)

INFO MSG: Created default RAR directory path <path>	Informational only. The 1.2/rar directory was missing, and BCS created it.
Command aborted by user	A response other than YES to the "`Are you sure [YES/NO]?`" prompt was entered.
All <num> units have been implied for this operation.	Since no list was specified, all units are implied.
All <num> groups have been implied for this operation.	Since no list was specified, all groups are implied.
Invalid response, re-enter	A response other than YES or NO to the "`Are you sure [YES/NO]?`" prompt was entered or a response other than Y or N to the "`Continue display Y/N?`" prompt. was entered.

APPENDIX B
TELNET FEATURES

telnet -- user interface to a remote system using the telnet protocol

USAGE:

telnet [host [port]]

OPTIONS:

- host - Specify the IP address of the remote host, in Internet dotted-decimal notation.
- port - Specify which port number on the remote host to establish the connection to.

1. DESCRIPTION

telnet communicates with another host using the telnet protocol. If ***telnet*** is invoked without arguments, it enters command mode, indicated by its prompt (**`telnet>`**). In this mode, it accepts and executes the commands listed below. If it is invoked with arguments, it performs an ***open*** command (see below) with those arguments.

Once a connection has been opened, ***telnet*** enters "character-at-a-time" input mode. Text typed is immediately sent to the remote host for processing.

If the ***localchars*** toggle is TRUE, the user's **quit**, **intr**, and **flush** characters are trapped locally, and sent as telnet protocol sequences to the remote side. There are options (see toggle ***autoflush*** and toggle ***autosynch*** below) which cause this action to flush subsequent output to the terminal (until the remote host acknowledges the telnet sequence) and flush previous terminal input (in the case of **quit** and **intr**).

While connected to a remote host, ***telnet*** command mode may by entered by typing the telnet "escape character" (initially "^**]**", (control-right bracket)). When in command mode, the normal terminal editing conventions are available.

2. TELNET COMMANDS

The following commands are available. Only enough of each command to uniquely identify it need be typed (this is also true for arguments to the **mode**, **set**, **toggle**, and **display** commands).

open host [port]

Opens a connection to the named ***host***. If no ***port*** number is specified, ***telnet*** will attempt to contact a telnet server at the default port. The ***host*** specification must be an Internet address specified in dotted-decimal notation.

close

Closes a telnet session and returns to command mode.

quit

Closes any open telnet session and exits ***telnet***. An EOF (in command mode) also closes a session and exits.

status

Shows the current status of ***telnet***. This includes the peer that the user is connected to, as well as the current mode.

display [argument...]

Displays all, or some, of the set and toggle values (see below).

? [command]

Obtains ***help***. With no arguments, ***telnet*** prints a help summary. If a ***command*** is specified, ***telnet*** will print the help information for that command only.

send <arguments>

Sends one or more special character sequences to the remote host. The following are the <***arguments***> which may be specified (more than one argument may be specified at a time):

escape

Sends the current telnet escape character (initially "^]").

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).

brk

Sends the telnet BRK (Break) sequence, which may have significance to the remote system.

ip

Sends the telnet IP (Interrupt Process) sequence.

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.

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.

ga

Sends the telnet GA (Go Ahead) sequence, which likely has no significance to the remote system.

nop

Sends the telnet NOP (No Operation) sequence.

?

Prints out help information for the ***send*** command.

set <argument> <value>

Sets any one of a number of ***telnet*** variables to a specific ***value***. The special value "***off***" turns off the function associated with the variable. The values of variables may be interrogated with the display command. The ***argument*** variables which may be specified are as follows:

escape

This is the telnet escape character (initially "^[") which causes entry into telnet command mode (when connected to a remote system).

interrupt

If telnet is in localchars mode (see toggle **localchars** below) and the interrupt character is typed, a telnet IP sequence (see **send ip** above) is sent to the remote host. The initial value for the **interrupt** character is taken to be the terminal's **intr** character.

quit

If telnet is in **localchars** mode (see toggle **localchars** below) and the **quit** character is typed, a telnet BRK sequence (see **send brk** above) is sent to the remote host. The initial value for the **quit** character is taken to be the terminal's **quit** character.

flushoutput

If telnet is in **localchars** mode (see toggle **localchars** below) and the **flushoutput** character is typed, a telnet AO sequence (see **send ao** above) is sent to the remote host. The initial value for the **flush** character is taken to be the terminal's **flush** character.

erase

If telnet is in **localchars** mode (see toggle **localchars** below), then when this character is typed, a telnet EC sequence (see **send ec** above) is sent to the remote system. The initial value for the **erase** character is taken to be the terminal's **erase** character.

kill

If telnet is in **localchars** mode (see toggle **localchars** below), then when this character is typed, a telnet EL sequence (see **send el** above) is sent to the remote system. The initial value for the **kill** character is taken to be the terminal's **kill** character.

toggle <arguments> ...

Toggle (between TRUE and FALSE) various flags that control how telnet responds to events. More than one argument may be specified. The state of these flags may be interrogated with the display command. Valid arguments are as follows:

localchars

If this is TRUE, then the **flush**, **interrupt**, **quit**, **erase**, and **kill** characters (see **set** above) are recognized locally, and transformed into appropriate telnet control sequences (**ao**, **ip**, **brk**, **ec**, and **el,** respectively; see **send** above). The initial value for this toggle is FALSE.

autoflush

If **autoflush** and **localchars** are both TRUE, then when the **ao**, **intr**, or **quit** characters are recognized (and transformed into telnet sequences; see **set** above for details), telnet refuses to display any data on the user's terminal until the remote system acknowledges (via a telnet Timing Mark option) that it has processed those telnet sequences.

autosynch

If **autosynch** and **localchars** are both TRUE, then when either the **intr** or **quit** characters are typed (see **set** above for descriptions of the **intr** and **quit** characters), the resulting telnet sequence sent is followed by the telnet SYNCH sequence. This procedure should cause the remote system to begin deleting all previously typed input until both of the telnet sequences have been read and acted upon. The initial value of this toggle is FALSE.

crmod

Toggles the RETURN mode. When this mode is enabled, most RETURN characters received from the remote host will be mapped into a RETURN followed by a LINEFEED. This mode does not affect those characters typed by the user, it affects only those received from the remote host. This mode is not very useful unless the remote host only sends RETURN, but never LINEFEED. The initial value for this toggle is FALSE.

options

Toggles the display of some internal telnet protocol processing (having to do with telnet options). The initial value for this toggle is FALSE.

netdata

Toggles the display of all network data (in hexadecimal format). The initial value for this toggle is FALSE.

?

Displays the legal toggle commands.

3. NOTES

After exiting telnet, the first character typed is always lost.

The telnet implementation provided under pSOSystem does not support the "line-by-line" mode.

There is no adequate way for dealing with flow control.

APPENDIX C
FILE TRANSFER PROTOCOL FEATURES

ftp - file transfer program

USAGE:

ftp [host_address]

1. DESCRIPTION

ARPANET standard File Transfer Protocol (FTP) can be used to transfer files between the BCU/CAL and a remote network site. This is done with the command ftp [host_address], where host_address refers to the Internet Protocol (IP) address of the remote host in dotted-decimal notation. This appendix describes the FTP client services available under pSOSystem, which are a subset of ARPANET FTP.

When the client host with which ***ftp*** is to communicate is specified on the command line, ***ftp*** immediately attempts to establish a connection to an FTP server on that host; otherwise, ***ftp*** enters its command interpreter and awaits instructions from the user. When ***ftp*** is awaiting commands from the user, it displays the prompt "**ftp>"**.

OPTIONS:

- host_address - Internet address of the remote host in dotted-decimal notation.

2. FTP COMMANDS

! [command]

Runs command as a shell command on the local machine.

account [passwd]

Supplies a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no ***passwd*** argument is included, the user will be prompted for an account password in a non-echoing input mode.

append local-file [remote-file]

Appends a local file to a file on the remote machine. If ***remote-file*** is left unspecified, the ***local-file*** name is used in naming the remote file. File transfer uses the current settings for "representation type," "file structure," and "transfer mode."

ascii

Sets the "representation type" to "network ASCII." This is the default type.

bell

Sounds a bell after each file transfer command is completed.

binary

Sets the "representation type" to "image."

bye

Terminates the FTP session with the remote server and exits ***ftp***. An EOF will also terminate the session and exit.

cd remote-directory

Changes the working directory on the remote machine to ***remote-directory***.

cdup

Changes the remote machine working directory to the parent of the current remote machine working directory.

close

Terminates the FTP session with the remote server, and returns to the command interpreter. Any defined macros are erased.

cr

Toggles RETURN stripping during "network ASCII" type file retrieval. Records are denoted by a RETURN/ LINEFEED sequence during "network ASCII" type file transfer. When ***cr*** is on (the default), RETURN characters are stripped from this sequence to conform with the UNIX system single LINEFEED record delimiter. Records on non-UNIX-system remote hosts may contain single LINEFEED characters; when a "network ASCII" type transfer is made, these LINEFEED characters may be distinguished from a record delimiter only when ***cr*** is off.

delete remote-file

Deletes the file ***remote-file*** on the remote machine.

dir [remote-directory] [local-file]

Prints a listing of the directory contents in the directory, ***remote-directory***, and, optionally, places the output in ***local-file***. If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or ***local-file*** is "-", output is sent to the terminal.

disconnect

Terminates the FTP session with the remote server, and returns to the command interpreter. Any defined macros are erased. (Synonym for **close** above.)

get remote-file [local-file]

Retrieves the file ***remote-file*** and stores it on the local machine. If the local file name (***local-file***) is not specified, it is given the same name it has on the remote machine, subject to alteration by the current case, **ntrans**, and **nmap** settings. The current settings for "representation type," "file structure," and "transfer mode" are used while transferring the file.

glob

Toggles filename expansion, or "globbing," for **mdelete**, **mget**, and **mput**. If globbing is turned off, filenames are taken literally.

Globbing for **mput** is performed as in csh(1). For **mdelete** and **mget**, each remote filename is expanded separately on the remote machine, and the lists are not merged.

Expansion of a directory name is likely to be radically different from expansion of the name of an ordinary file: The exact result depends on the remote operating system and FTP server, and can be previewed by typing "**mls remote-files -**".

mget and **mput** are not meant to transfer entire directory subtrees of files. Entire directory subtrees may be transferred by performing a tar(1) archive of the subtree (using a "representation type" as "image" as set by the binary command).

hash

Toggles hash-sign (#) printing for each data block transferred.

help [command]

Prints an informative message about the meaning of a command. If no ***command*** argument is given, ***ftp*** prints a list of the known commands.

lcd [directory]

Changes the working directory to ***directory*** on the local machine. If ***directory*** is not specified, the user's local home directory is used.

ls [remote-directory] [local-file]

Prints an abbreviated listing of the contents of a directory (***remote-directory)*** on the remote machine and, optionally, places the output in ***local-file***. If ***remote-directory*** is left unspecified, the current working directory is used. If no local file is specified (or if ***local-file*** is "-"), the output is sent to the terminal.

mdelete [remote-files]

Deletes the ***remote-files*** on the remote machine.

mdir remote-files local-file

Similar to ***dir***, except multiple remote files may be specified. If interactive prompting is on, ***ftp*** will prompt the user to verify that the last argument is indeed the target ***local-file*** for receiving ***mdir*** output.

mget remote-files

Expands the ***remote-files*** on the remote machine and performs a ***get*** for each file name thus produced. See **glob** for details on the filename expansion. Resulting filenames will then be processed according to **case**, **ntrans**, and **nmap** settings. Files are transferred into the local working directory, which can be changed with "**lcd directory**." New local directories can be created with "**! mkdir directory**."

mkdir directory-name

Makes a directory(***directory-name***)on the remote machine.

mls remote-files local-file

Similar to **ls**(1V), except multiple remote files may be specified. If interactive prompting is on, ***ftp*** will prompt the user to verify that the last argument is indeed the target local file for receiving ***mls*** output.

mode [mode-name]

Sets the "transfer mode" to ***mode-name***. The only valid ***mode-name*** is ***stream***, which corresponds to the default "stream" mode.

mput directory-name

Expands wildcards in the list of local files given as arguments and performs a ***put*** for each file in the resulting list. See **glob** for details of filename expansion.

nlist [remote-directory][local-file]

Prints an abbreviated listing of the contents of a directory on the remote machine. If ***remote-directory*** is unspecified, the current working directory is used. If no local file is specified, (or if ***local-file*** is "-"), the output is sent to the terminal.

open host [port]

Establishes a connection to the specified host FTP server. An optional ***port*** number may be supplied, in which case, ***ftp*** will attempt to contact an FTP server at that port. If the ***auto-login*** option is on (default), ***ftp*** will also attempt to automatically log the user in to the FTP server (see below).

prompt

Toggles interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. By default, prompting is turned on. If prompting is turned off, any ***mget*** or ***mput*** will transfer all files, and any ***mdelete*** will delete all files.

put local-file [remote-file]

Stores a local file(***local-file)***, on the remote machine. If ***remote-file*** is left unspecified, the local file name is used in naming the remote file. File transfer uses the current settings for "representation type," "file structure," and "transfer mode."

pwd

Prints the name of the current working directory on the remote machine.

quit

Terminates the FTP session with the remote server and exits ***ftp***. An EOF will also terminate the session and exit. (Synonym for **bye**.)

quote arg1 arg2 ...

Sends the arguments specified, verbatim, to the remote FTP server. A single FTP reply code is expected in return.

recv remote-file [local-file]

Synonym for ***get***.

remotehelp [command-name]

Requests ***help*** from the remote FTP server. If a ***command-name*** is specified it is supplied to the server as well.

rename from to

Renames the file ***from*** on the remote machine to have the name ***to***.

reset

Clears the reply queue. This command re-synchronizes command/reply sequencing with the remote FTP server. Resynchronization may be necessary following a violation of the FTP protocol by the remote server.

rmdir directory-name

Deletes a directory (***directory-name***) on the remote machine.

runique

Toggles storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a **get** or ***mget*** command, a ".1" is appended to the name. If the resulting name matches another existing file, a ".2" is appended to the original name. If this process continues up to ".99," an error message is printed, and the transfer does not take place. The generated unique filename will be reported. Note: ***runique*** will not affect local files generated from a shell command (see below). The default value is off.

send local-file [remote-file]

Synonym for ***put***.

sendport

Toggles the use of PORT commands. By default, ***ftp*** will attempt to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails, ***ftp*** will use the default data port. When the use of PORT commands is disabled, no attempt will be made to use PORT commands for each data transfer. This is useful when connected to certain FTP implementations that ignore PORT commands but incorrectly indicate they have been accepted.

status

Shows the current status of ***ftp***.

sunique

Toggles storing of files on remote machine under unique file names. The remote FTP server must support the STOU command for successful completion. The remote server will report the unique name. Default value is off.

tenex

Sets the "representation type" to that needed to talk to TENEX machines.

type [type-name]

Sets the "representation type" to t***ype-name***. The valid ***type-name***(s) are ascii for "network ASCII," binary or image for "image," and tenex for "local byte size" with a byte size of 8 (used to talk to TENEX machines). If no type is specified, the current type is printed. The default type is "network ASCII."

user user-name [password] [account]

Identifies user to the remote FTP server. If the ***password*** is not specified and the server requires it, ***ftp*** will prompt the user for it (after disabling local echo). If an ***account*** field is not specified, and the FTP server requires it, the user will be prompted for it. If an account field is specified, an account command will be relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless ***ftp*** is invoked with "auto-login" disabled, this process is performed automatically on initial connection to the FTP server.

verbose

Toggles verbose mode. In ***verbose*** mode, all responses from the FTP server are displayed to the user. In addition, if verbose mode is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose mode is on if **ftp's** commands are coming from a terminal, and off otherwise.

? [command]

A synonym for ***help***.

Command arguments which have embedded spaces may be quoted with quotation (") marks.

If any command argument which is not indicated as being optional is not specified, ***ftp*** will prompt for that argument.

3. ABORTING A FILE TRANSFER

The normal abort sequence, CTRL-C, will not work during a transfer. There is no command for aborting a file transfer.

4. FILE NAMING CONVENTIONS

Local files specified as arguments to **ftp** commands are processed according to the following rules:

1. If the file name “-” is specified, the standard input (for reading) or standard output (for writing) is used.

2. Failing the checks above, if "globbing" is enabled, local filenames are expanded according to the rules used in the csh(1); see the **glob** command. If the **ftp** command expects a single local file (for example, put), only the first filename generated by the "globbing" operation is used.

3. For **mget** commands and **get** commands with unspecified local file names, the local filename is the remote filename, which may be altered by a **case**, **ntrans**, or **nmap** setting. The resulting filename may then be altered if **runique** is on.

4. For **mput** commands and **put** commands with unspecified remote file names, the remote filename is the local filename, which may be altered by an **ntrans** or **nmap** setting. The resulting filename may then be altered by the remote server if **sunique** is on.

FILE TRANSFER PARAMETERS

The FTP specification defines many parameters which may affect a file transfer.

The "representation type" may be one of "network ASCII," "EBCDIC," "image," or "local byte size" with a specified byte size (for PDP-10s and PDP-20s mostly). The "network ASCII" and "EBCDIC" types have a further subtype which specifies whether vertical format controls (NEWLINE characters, form feeds, etc.) are to be passed through ("non-print"), provided in telnet format ("telnet format controls"), or provided in ASA (FORTRAN) ("carriage control (ASA)") format. **ftp** supports the "network ASCII" (subtype "non-print" only) and "image" types, plus "local byte size" with a byte size of 8 for communicating with TENEX machines.

The "file structure" may be one of "file" (no record structure), "record," or "page." **ftp** supports only the default value, which is "file."

The "transfer mode" may be one of "stream," "block," or "compressed." **ftp** supports only the default value, which is "stream."

5. NOTES

Correct execution of many commands depends upon proper behavior by the remote server.

An error in the treatment of carriage returns in the 4.2 BSD code handling transfers with a "representation type" of "network ASCII" has been corrected. This correction may result in incorrect transfers of binary files to and from 4.2 BSD servers using a "representation type" of "network ASCII". Avoid this problem by using the "image" type.

APPENDIX D
NETWORK STATISTICS

The following appendix defines the Management Information Base (MIB) II variables which may be examined with the BCU/CAL ***netstat*** command.

1. INTERFACE GROUP VARIABLES

The interface group variables may be examined using the ***netstat if*** command.

1.1. ifIndex

A unique value for each interface. Its value ranges between 1 and the maximum number of network interfaces (**ifNumber**). The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

1.2. ifDescr

A textual string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface.

1.3. ifType

The type of interface, distinguished according to the physical/link protocol(s) immediately "below" the network layer in the protocol stack. The following types are applicable to the BCU/CAL:

Table D.1 - BCU/CAL Interface Types

ifType	Description
6 = ethernet-csmacd	Ethernet interface

1.4. ifMtu

The size of the largest datagram which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.

1.5. ifSpeed

An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.

1.6. ifPhysAddress

The interface's address at the protocol layer immediately "below" the network layer in the protocol stack. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length.

1.7. ifAdminStatus

The desired state of the interface. The testing state (3 in the table below) indicates that no operational packets can be passed.

Table D.2. - ifAdminStatus States

ifAdminStatus	Description
1 = up	Ready to pass packets.
2 = down	Disabled.
3 = testing	In a test mode.

1.8. ifOperStatus

The current operational state of the interface. The testing state (3 in the table below) indicates that no operational packets can be passed.

Table D.3. - ifOperStatus States

ifAdminStatus	Description
1 = up	Ready to pass packets.
2 = down	Disabled.
3 = testing	In a test mode.

1.9. ifLastChange

The value of **sysUpTime** at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.

1.10. ifInOctets

The total number of octets received on the interface, including framing characters.

1.11. ifInUcastPkts

The number of subnetwork-unicast packets delivered to a higher-layer protocol.

1.12. ifInNUcastPkts

The number of non-unicast (i.e., subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.

1.13. ifInDiscards

The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

1.14. ifInErrors

The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

1.15. ifInUnknownProtos

The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

1.16. ifOutOctets

The total number of octets transmitted out of the interface, including framing characters.

1.17. ifOutUcastPkts

The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.

1.18. ifOutNUcastPkts

The total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.

1.19. ifOutDiscards

The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

1.20. ifOutErrors

The number of outbound packets that could not be transmitted because of errors.

1.21. ifOutQLen

The length of the output packet queue (in packets).

2. IP GROUP VARIABLES

The IP group variables may be examined using the ***netstat ip*** command.

2.1. ipForwarding

The indication of whether this entity is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP gateways forward datagrams. IP hosts do not forward datagrams (except those source-routed via the host).

Note that for some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a "badValue" response if a management station attempts to change this object to an inappropriate value.

Table D.4. - ipForwarding States

ipForwarding	Description
1 = forwarding	Acting as a gateway
2 = not-forwarding	NOT acting as a gateway

2.2. ipDefaultTTL

The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity, whenever a TTL value is not supplied by the transport layer protocol.

2.3. ipInReceives

The total number of input datagrams received from interfaces, including those received in error.

2.4. ipInHdrErrors

The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.

2.5. ipInAddrErrors

The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

2.6. ipForwDatagrams

The number of input datagrams for which this entity was not their final IP destination, and as a result, an attempt was made to locate a route to forward them to that final destination. In entities which do not act as IP gateways, this counter will include only those packets which were source-routed via this entity, and the source-route option processing was successful.

2.7. ipInUnknownProtos

The number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

2.8. ipInDiscards

The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include datagrams discarded while awaiting reassembly.

2.9. ipInDelivers

The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).

2.10. ipOutRequests

The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include datagrams counted in **ipForwDatagrams**.

2.11. ipOutDiscards

The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in **ipForwDatagrams** if any such packets met this (discretionary) discard criterion.

2.12. ipOutNoRoutes

The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in **ipForwDatagrams** which meet this “no-route” criterion. This includes any datagrams which a host cannot route because all of its default gateways are down.

2.13. ipReasmTimeout

The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.

2.14. ipReasmReqds

The number of IP fragments received which needed to be reassembled at this entity.

2.15. ipReasmOKs

The number of IP datagrams successfully reassembled.

2.16. ipReasmFails

The number of failures detected by the IP reassembly algorithm (for whatever reason: timed out, errors, etc). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

2.17. ipFragOKs

The number of IP datagrams that have been successfully fragmented at this entity.

2.18. ipFragFails

The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set.

2.19. ipFragCreates

The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.

2.20. ipRoutingDiscards

The number of routing entries which were chosen to be discarded even though they are valid. One possible reason for discarding such an entry could be to free-up buffer space for other routing entries.

2.21. NOTES

1. The IP Address Translation Table (ipAd*xxx* variables) cannot be queried with the ***netstat*** command.

2. The IP Routing Table (ipRoute*xxx* variables) cannot be queried with the ***netstat*** command.

3. The IP Network To Media Table (ipNetToMedia*xxx* variables) cannot be queried with the ***netstat*** command.

4. Refer to the ***route*** command for definition of IP routing-related objects which may be queried or modified by the BCU/CAL.

3. ICMP GROUP VARIABLES

The ICMP group variables may be examined using the ***netstat icmp*** command.

3.1. icmpInMsgs

The total number of ICMP messages which the entity received. Note that this counter includes all those counted by **icmpInErrors**.

3.2. icmpInErrors

The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).

3.3. icmpInDestUnreachs

The number of ICMP Destination Unreachable messages received.

3.4. icmpInTimeExcds

The number of ICMP Time Exceeded messages received.

3.5. icmpInParmProbs

The number of ICMP Parameter Problem messages received.

3.6. icmpInSrcQuenchs

The number of ICMP Source Quench messages received.

3.7. icmpInRedirects

The number of ICMP Redirect messages received.

3.8. icmpInEchos

The number of ICMP Echo (request) messages received.

3.9. icmpInEchoReps

The number of ICMP Echo Reply messages received.

3.10. icmpInTimestamps

The number of ICMP Timestamp (request) messages received.

3.11. icmpInTimestampReps

The number of ICMP Timestamp Reply messages received.

3.12. icmpInAddrMasks

The number of ICMP Address Mask Request messages received.

3.13. icmpInAddrMaskReps

The number of ICMP Address Mask Reply messages received.

3.14. icmpOutMsgs

The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by **icmpOutErrors**.

3.15. icmpOutErrors

The number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers. This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter's value.

3.16. icmpOutDestUnreachs

The number of ICMP Destination Unreachable messages sent.

3.17. icmpOutTimeExcds

The number of ICMP Time Exceeded messages sent.

3.18. icmpOutParmProbs

The number of ICMP Parameter Problem messages sent.

3.19. icmpOutSrcQuenchs

The number of ICMP Source Quench messages sent.

3.20. icmpOutRedirects

The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.

3.21. icmpOutEchos

The number of ICMP Echo (request) messages sent.

3.22. icmpOutEchoReps

The number of ICMP Echo Reply messages sent.

3.23. icmpOutTimestamps

The number of ICMP Timestamp (request) messages sent.

3.24. icmpOutTimestampReps

The number of ICMP Timestamp Reply messages sent.

3.25. icmpOutAddrMasks

The number of ICMP Address Mask Request messages sent.

3.26. icmpOutAddrMaskReps

The number of ICMP Address Mask Reply messages sent.

4. TCP GROUP VARIABLES

The TCP group variables may be examined using the ***netstat tcp*** command.

Note that instances of object types that represent information about a particular TCP connection are transient; they persist only as long as the connection in question.

4.1. tcpRtoAlgorithm

The algorithm used to determine the timeout value used for retransmitting unacknowledged BCU/CAL octets.

Table D.5. - TCP Retransmit Timeout Algorithm

Value	Algorithm
1 (other)	none of the following
2 (constant)	constant RTO
3 (rsre)	MIL-STD-1778, Appendix B
4 (vanj)	Van Jacobson's algorithm [10]

4.2. tcpRtoMin

The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC-793.

4.3. tcpRtoMax

The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the UBOUND quantity described in RFC-793.

4.4. tcpMaxConn

The limit on the total number of TCP connections the entity can support. In entities where the maximum number of connections is dynamic, this object should contain the value -1.

4.5. tcpActiveOpens

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

4.6. tcpPassiveOpens

The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

4.7. tcpAttemptFails

The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.

4.8. tcpEstabResets

The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

4.9. tcpCurrEstab

The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

4.10. tcpInSegs

The total number of segments received, including those received in error. This count includes segments received on currently established connections.

4.11. tcpOutSegs

The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.

4.12. tcpRetransSegs

The total number of segments retransmitted; that is, the number of TCP segments transmitted containing one or more previously transmitted octets.

4.13. tcpConnState

The state of this TCP connection.

The only value which may be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to set this object to any other value.

If a management station sets this object to the value deleteTCB(12), then this has the effect of deleting the TCB (as defined in RFC-793) of the corresponding connection on the managed node, resulting in immediate termination of the connection.

As an implementation-specific option, an RST segment may be sent from the managed node to the other TCP endpoint (Note, however, that RST segments are not sent reliably).

Table D.6. - tcpConnState States

State	Description
1	closed
2	listen
3	synSent
4	synReceived
5	established
6	finWait1
7	finWait2
8	closeWait
9	lastAck
10	closing
11	timeWait
12	deleteTCB

4.14. tcpConnLocalAddress

The local IP address for this TCP connection. In the case of a connection in the listen state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used.

4.15. tcpConnLocalPort

The local port number for this TCP connection.

4.16. tcpConnRemAddress

The remote IP address for this TCP connection.

4.17. tcpConnRemPort

The remote port number for this TCP connection.

4.18. tcpInErrs

The total number of segments received in error (e.g., bad TCP checksums).

4.19. tcpOutRsts

The number of TCP segments sent containing the RST flag.

5. UDP GROUP VARIABLES

The UDP group variables may be examined using the ***netstat udp*** command.

5.1. udpInDatagrams

The total number of UDP datagrams delivered to UDP users.

5.2. udpNoPorts

The total number of received UDP datagrams for which there was no application at the destination port.

5.3. udpInErrors

The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

5.4. udpOutDatagrams

The total number of UDP datagrams sent from this entity.

5.5. NOTES

The UDP Listener table cannot be queried via the ***netstat*** command.

APPENDIX E
PNAD DAEMON ERROR CODES

This section contains error codes associated with the BCU/CAL TCP/IP networking daemon PNAD. In general, the BCU/CAL provides error messages with a text description of the error. However, some operations may result in a hexadecimal value being displayed.

These types of error codes are most likely to be experienced by the more advanced BCU/CAL system operator, and are thus provided for reference. For example, the ***ping*** command may return an error code of 0x5033, which indicates that the network destination is unreachable. In this example, the operator should be able to recognize that the network routing tables are not properly configured.

1. FATAL ERRORS

The following table contains fatal system errors codes associated with the BCU/CAL networking operation. These errors should never occur during normal system observation. If observed, they should be report to Ericsson GE for problem analysis.

Table E.1. - PNAD Fatal Errors

Error Code	Pneumonic	Meaning
0x5F01	FAT_INSUFFMEM	Insufficient memory allocated by NC_DATASIZE or REGION 0 too small .
0x5F02	FAT_NRT	The number of initial routing table entries specified exceeds NC_NROUTE.
0x5F03	FAT_NNI	The number of initial NI table entries specified exceeds NC_NNI.
0x5F04	FAT_NIHSIZE	Invalid NI address.
0x5F05	FAT_NIMTU	Invalid MTU for NI.
0x5F06	FAT_PNAMEM	pNA+ memory error.
0x5F07	FAT_PNATASK	PNAD task creation error.
0x5F08	FAT_PNAINIT	pNA+ initialization error.
0x5F09	FAT_NIINIT	NI initialization error.
0x5F0A	FAT_RTINIT	Routing table initialization error.
0x5F0B	FAT_ARPINIT	ARP table initialization error.
0x5F0C	FAT_TIMERINIT	PNAD timer initialization error.
0x5F0D	FAT_EVENT	PNAD event error.
0x5F0E	FAT_CHKSUM	pNA+ Checksum error.

2. INFORMATIONAL ERRORS

The following table contains non-fatal error codes that may be observed on the BCU/CAL during network operations. A brief explanation of the error code's meaning is provided.

Table E.2. - PNAD Informational Errors

Error	Pneumonic	Meaning
0x5004	EINTR	Interrupted system call
0x5005	EIO	I/O error
0x5006	ENXIO	No such address
0x5009	EBADS	The socket descriptor is invalid
0x500C	ENOMEM	Not enough memory
0x500D	E_ACCESS	Permission denied
0x5011	EEXIST	Entry exists

Table E.2. - PNAD Informational Errors (Cont.)

Error	Pneumonic	Meaning
0x5016	EINVALID	An argument is invalid
0x5017	ENFILE	An internal table has run out of space
0x5020	EPIPE	The connection is broken
0x5023	EWOULDBLOCK	This operation would block and the socket is non-blocking
0x5024	EINPROGRESS	The socket is non-blocking and the connection cannot be completed immediately
0x5025	EALREADY	The socket is non-blocking and a previous connection attempt has not yet completed
0x5026	ENOTSOCK	Socket operation on a non-socket
0x5027	EDESTADDRREQ	The destination address is invalid
0x5028	EMSGSIZE	Message is too long
0x5029	EPROTOTYPE	Protocol is wrong type for socket
0x502A	ENOPROTOOPT	Protocol is not available
0x502B	EPROTONOSUPPORT	Protocol is not supported
0x502C	ESOCKTNOSUPPORT	Socket type is not supported
0x502D	EOPNOTSUPP	Operation is not supported on socket
0x502E	EPFNOSUPPORT	Protocol family is not supported
0x502F	EAFNOSUPPORT	Address family is not supported
0x5030	EADDRINUSE	Address is already in use
0x5031	EADDRNOTAVAIL	Address is not available
0x5032	ENETDOWN	Network is down
0x5033	ENETUNREACH	Network is unreachable
0x5034	ENETRESET	Network dropped connection on reset
0x5035	ECONNABORTED	The connection has been aborted by the peer
0x5036	ECONNRESET	The connection has been reset by the peer
0x5037	ENOBUFS	An internal buffer is required, but cannot be allocated
0x5038	EISCONN	The socket is already connected
0x5039	ENOTCONN	The socket is not connected
0x503A	ESHUTDOWN	Cannot send after socket shutdown
0x503B	ETOOMANYREFS	Too many references; cannot splice
0x503C	ETIMEDOUT	Connection has timed out
0x503D	ECONNREFUSED	The attempt to connect was refused
0x5040	EHOSTDOWN	Host is down
0x5041	EHOSTUNREACH	The destination host could not be reached from this node
0x5042	ENOTEMPTY	Directory is not empty
0x5046	ENIDOWN	NI_INIT returned -1
0x5047	ENMTU	The MTU is invalid
0x5048	ENHWL	The hardware length is invalid
0x5049	ENNOFIND	The route specified cannot be found
0x504A	ECOLL	Collision in select call; these conditions have already been selected by another task
0x504B	ETID	The task ID is invalid