User's Reference Manual

EDACS®

Data Advantage™

ERICSSON ⪦

TABLE OF CONTENTS

OVERVIEW

This document describes the Ericsson EDACS® Data Advantage™ User Interface, its command syntax, and provides numerous examples. The User Interface itself is very 'UNIX like' in nature. Many of the commands will be familiar to those with experience in the UNIX environment.

The Data Advantage User Interface command shell syntax is terse, powerful, and potentially dangerous. Advantages of this approach are cross-product commonality, a rich command set, and most importantly, a high degree of flexibility for the Data Advantage end user. The disadvantage is random hacking at commands can result in significant degradation of the Data Advantage's ability to route data calls within an EDACS/Internet environment. The User Interface described herein has been designed to strike a reasonable compromise between potency and problematic potential.

The first section of this document provides general information required to make effective use of the User Interface. Topics include volume naming conventions, directory structures, and important system daemons.

The second section of this document covers the actual command set.   Some of these commands and their syntax are common to several VMEbus based EDACS products; such as the Data Advantage, Jessica, Billing Correlation Unit (BCU), and the Central Activity Logger (CAL). Other commands are unique to the Data Advantage.

Other sections include a discussion of File Transfer Protocol (FTP) and TELNET protocol.

GENERAL INFORMATION

The following section provides general information to assist the user in understanding some system topics which underlie the User Interface, and its effective operation.

VOLUME NAMES

Several Data Advantage internal devices, such as the system disk, are identified as a "volume". Volume naming conventions are similar to that used in the UNIX operating system. Specifically, a volume label is of the form *Major_Number.Minor_Number*. The Major_Number may be though of as identifying the type of a 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 Data Advantage's mass storage devices. The Major_Number for the Data Advantage's storage devices is **1**, which identifies them as being devices connected on the Data Advantage's SCSI bus. The Minor_Number identifies a particular devices address on the SCSI bus.

The following table defines the volume names for the Data Advantage's storage devices.

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).

Table 1 - Data Advantage Volume Names

SYSTEM DIRECTORIES

The following table describes the system directories mandatory for correct Data Advantage 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 Data Advantage system disk to suit their requirements. To assure compatibility will future software releases, it is recommended that users do not create any subdirectories, or store user data, beneath any of the directories specified in the following table.

Directory	Contents
/etc	System administration. (hosts, networks, passwd, etc. files.)
/tmp	Temporary files 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 DACAP.SX
/cnfg	Application configuration files, such as the Data Advantage's SYSTEM.TXT.
/backup	Backup files. When a new software version is installed, the previous version(s) will be moved to this directory.
/activity	Activity and diagnostic report files.

Table 2 - DA Root Directory Structure

The directories relating to NFS are reserved for future use. Specifically, NFS capabilities are not supported in the release of the Data Advantage.

SYSTEM FILES

The following table describes the three ordinary (non-directory) files contained in the root directory of the Data Advantage system disk. The files BITMAP.SYS and FLIST.SYS are system write access only, and are protected from accidental modification by the user. The file LOADER.SX is copied from floppy disk when the Data Advantage 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 Data Advantage from booting.

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	Data Advantage application loader file, mandatory for system bootstrap loading.

Table 3 - Data Advantage Root Directory Files

Several files may be generated by the Data Advantage in the /activity directory. These include activity logging and diagnostic information for the user. The user is free to access and/or remove files in this directory at their discretion.

In the event of a catastrophic system error (a.k.a. a software bug or internal hardware error), the Data Advantage 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. It is requested that the user copy this file to a floppy diskette, and forward it to Ericsson PRS Division for analysis. Any additional information, such as a copy of the current /cnfg/SYSTEM.TXT and traffic loading scenario, would be extremely helpful.

COPYING FILES TO AND FROM A FLOPPY

Occasionally files need to be copied to or from a floppy. These examples are for the convenience of users that are not familiar with UNIX. A later section contains detailed information for each command.

Floppies must be High Density and formatted for MS-DOS.

UNIX terminates lines in ASCII files with a Carriage Return, but not a Line Feed. MS-DOS terminates lines in ASCII files with both a Carriage Return and Line Feed. ASCII files copied from the Data Advantage are in UNIX style. If desired, they can be converted to MS-DOS style. The standard MS-DOS editor *Edit* will convert files automatically. *Unix2dos* is a common filter that will also convert files.

The sync command in the second example is required if the Data Advantage will be rebooted within one minute of copying the file onto the hard drive.

Hard Drive to Floppy Example:

pSH+> cd /activity
pSH+> pcmount 1.1
pSH+ > cp jan01.log 1.1/jan01.log
pSH+ > umount 1.1

Floppy to Hard Drive Example:

pSH+ > pcmount 1.1
pSH+ > cp 1.1/myfile 1.2/myfile
pSH+ > umount 1.1
pSH+ > sync

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, the status, priority, or even existence of these daemons is 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 culprit entity is being isolated. 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.

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.

Table 4 - Important system daemons

COMMAND DESCRIPTIONS

FREQUENTLY USED COMMANDS

The following section discusses commands most frequently used on the Data Advantage.

address

address - display the IP address assigned to an EDACS unit or group id

USAGE:
        address [-u edacs_unit_id] [-g edacs_group_id]

DESCRIPTION:

address will display the IP address that has been assigned to an EDACS unit or group id by the user via the [Device_Config_Table] in the configuration file /cnfg/SYSTEM.TXT.

OPTIONS:

-u	Display the IP address assigned to an EDACS unit id.
-g	Display the IP address assigned to an EDACS group id.

Table 5 - **address** options

EXAMPLE:

pSH+> address -u 507 -g 273
IP address of EDACS unit id 507: 190.03.00.07
IP address of EDACS group id 273: 190.03.01.03

pSH+> address -g 600 -u 2000
No IP address assigned to EDACS unit id 2000
No IP address assigned to EDACS group id 600

cat

cat - concatenate and display

USAGE:

cat [-benstv] filename...

DESCRIPTION:

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

OPTIONS:

-b	Number the lines, as -n, but omit the line numbers from blank lines.
-e	Display non-printable characters, as -v, and in addition display a $ character at the end of each line.
-n	Precede each line output with its line number.
-s	Substitute a single blank line for multiple adjacent blank lines.
-t	Display non-printable characters, as -v, and in addition display TAB characters as ^I (CTRL-I).
-v	Display non-printable characters (with the exception of TAB and NEWLINE characters) so that they are visible. Control characters print like ^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.

Table 6 - **cat** options

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 go into an indeterminate state.

WARNING: Once started, *cat* cannot be aborted. It may, however, be suspended and resumed using flow control characters ^S and ^Q. Because of this, it is recommended that the **more** command be used on large files to avoid losing control of the console for an extended period of time. Also, using the **cat** command on binary files should be avoided, as it may cause lockup of the display, requiring a power cycle of the console to continue operation.

cd

cd - change working directory

USAGE:
> cd [directory]

DESCRIPTION:

directory becomes the new working directory.

clear

clear - clear the terminal screen

USAGE:
>   clear

DESCRIPTION:

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

See also the command *clr*.

OPTIONS:
None.

clr

clr - clear 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.

cp

cp - copy 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 within *directory2* (as a subdirectory), along with its files and subdirectories.

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 -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.

Table 7 - **cp** options

NOTES:
cp refuses to copy a file onto itself.
The wildcard character "*" is not supported.

WARNING

If the system is rebooted less than 60 seconds after any copy operations, file corruption may result. To avoid this situation, use the 'sync' command after a 'cp' operation if the system is to be immediately rebooted, reset, or powered down.

---

df

df - display the size statistics for all mounted file systems

USAGE:
>   df

DESCRIPTION:

df displays, for each mounted file system, the total size, the amount in use, the amount available, and where it is mounted. The values are given in kilobytes.

OPTIONS:
None.

EXAMPLE:
```
pSH+> df
Filesystem   kbytes     used      avail    capacity  Mounted on
01.02        83121      11498     71623    13.833%   /
```

du

du - display 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. Entries are generated only for each directory in the absence of options.

OPTIONS:

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

Table 8 - **du** options

exit

exit - exit the shell

USAGE:

exit

DESCRIPTION:

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

OPTIONS:
None.

head

head - display 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 look like:

    ==> filename <==

Thus, a common way to display a set of short files, identifying each one, is:
pSH+> head -1 filename1 filename2 ...

EXAMPLE:

The following fictitious example would produce the following output;

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...

help

help - get 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
address  df       logmask    mac       network   purge     restart   scsi      stats      version
clr      log      lp         more      passwd    reboot    route     shutdn    timedat
```

```
pSH+> help address
address - display the IP address for an EDACS id (not reentrant, not locked)
```

```
pSH+> help df
df    - display file system usage (not reentrant, not locked)
```

NOTES:

help on an individual command provides two additional pieces of information; if the command is reentrant, and if 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 reentrant is only available to one user at a time. Lock status indicates if a non reentrant command is current 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.

log

log - specify/display activity logging status

USAGE:

    log [-cds] [-p on|off] [-f on|off] [-m warn|all|none]

DESCRIPTION:

log is used in managing the Data Advantage activity logging, which consists of error, warning, and information text messages related to system operation. The messages can be directed to three separate devices: a text file on the hard disk, the line printer, and the console.

The text file, also known as the activity log file, is located in the /activity directory of the hard disk. The format of the file name is 'mmmdds.log', where 'mmm' is a three letter abbreviation of the current month, 'dd' is the current day of the month, and 's' is an optional suffix character. The maximum size of the file may be 1,400,000 bytes, which allows it to be copied to a floppy.

A new activity log file is always opened when the system day changes, with no suffix character in the file name. This file will continue to grow until the maximum size is reached, or the system day changes. If the maximum amount of log data for a given day exceeds the maximum, a new file is created with the same month and day characters, but with the suffix character 'a'. If this file also grows to the maximum size, a new file is created with the suffix character 'b'. This continues until the character 'z' is reached, at which time the file may grow until either the month/day changes, or the hard disk is filled to capacity.

The current activity log file may have its contents deleted using the **-c** option. Other log files have to be deleted using the rm command.

The activity log messages may be output directly to the console as they are generated using the **-d** option. No other console commands may be executed while logging to the console is enabled. To abort logging to the console, enter 'q' from the console keyboard.

NOTE

Using **log -d** from a Telnet session will disable the automatic session termination. The **exit** command will still log the user out of the Data Advantage. However, the telnet session must be terminated by entering **CTRL** and **]** to get a telnet command prompt and then **q**.

If a printer is attached to the system, the **-p** option may be used to direct all activity log data to it.

The **-s** option will output the status of the activity logging, including whether output is directed to the file or printer, the current activity log file name, and the time of the last activity file log clear since the system was first booted.

The **-m** option controls the type of logging messages which will be produced. **-m warn** specifies that any warning messages generated by the system will be logged, and any informational messages will be suppressed. **-m all** specifies that both warning and information messages will be logged. **-m none** specifies that all warning and information messages will be suppressed. Note that system error messages will always be sent to the current logging device(s), regardless of the **-m** option selection.

OPTIONS:

-c	Clear activity log file.
-d	Display all activity log messages directly to the console.
-s	Display current status of activity logging.
-p	Enable/Disable output to the printer.
-f	Enable/Disable output to the log file.
-m	Control the amount of log information being produced.

Table 9 - **log** options

--- **NOTE** ---

The values for the **-m**, **-p**, and **-f** options are non-volatile, i.e. the status of these settings are saved even if the system is powered down or reset.

EXAMPLES:

pSH+> log -p off -f off
  Activity log output to printer is disabled.
  Activity log output to file is disabled.

pSH+> log -m warn -d
  Warning logging enabled on all boards.
Activity display enabled - press 'q' and <RETURN> to quit

q
psh+>

lp

lp - print a text file

USAGE:
> lp file_name

DESCRIPTION:

lp queues the text file specified by *file_name* to be output on the Data Advantage printer. If no printer is connected, or another condition exists which prevents proper printout, an appropriate error message will be displayed on the console.

OPTIONS:
None.

ls

ls - list the contents of a directory

USAGE:

      ls [-aACdfFgilqrRs1] filename ...

DESCRIPTION:

For each *filename* which is a directory, *ls* lists the contents of the directory; for each *filename* which 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	List all entries. In the absence of this option, entries whose names begin with a `.' are not listed
-A	Same as -a, except that `.' and `..' are not listed.
-C	Force multi-column output, with entries sorted down the columns; for ls, this is the default when output is to a terminal.
-d	If argument is a directory, list only its name (not its contents); often used with -l to get the status of a directory.
-f	Force each argument to be interpreted as a directory and list 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	Mark directories with a trailing slash (`/'), executable files with a trailing asterisk (`*').
-g	For ls, show the group ownership of the file in a long output.
-i	For each file, print the i-number in the first column of the report.
-l	List in long format, giving 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	Display non-graphic characters in filenames as the character ?; for ls, this is the default when output is to a terminal.
-r	Reverse the order of sort to get reverse alphabetic or oldest first as appropriate.
-R	Recursively list subdirectories encountered.
-s	Give size of each file, including any indirect blocks used to map the file, in kilobytes.
-1	Force single-column output.

Table 10 - **ls** options

more

more - browse or page 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 to browsing the file, enter the CR character (i.e. Enter or Return). To terminate more, enter the character "q", followed by CR.

OPTIONS:

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

Table 11 - **more** options

EXAMPLE:

pSH+> more -/activity/jan06.log

Fri Jan 6 09:17:02 1995:Board=1, Source File=mcc_subs.c, Line=621
Subsystem startup complete.

pSH+>

mv

mv - move or rename 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. Override 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 you type a line starting with y, mv moves the specified file or directory, otherwise mv does nothing with that file or directory.

Table 12 - **mv** options

NOTES:

mv refuses to move a file or directory onto itself.
mv will not move a directory from one file system to another.

mkdir

mkdir - make a directory

USAGE:
      mkdir [-p] dir_name ...

DESCRIPTION:

mkdir creates the directory *dir_name*.

OPTIONS:

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

Table 13 - **mkdir** options

passwd

passwd - password management.

USAGE:
      passwd [login_name]

DESCRIPTION:

passwd changes (or installs) a password associated with the user 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 forestall 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:
The passwd command may only be invoked from the local console. Attempts to use it via a TELNET session will be denied.

pcmount

pcmount - mount 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 Data Advantage floppy disk drive is **1.1**. A diskette should be inserted in the floppy drive prior to using *pcmount*. If no diskette is present, an appropriate error message will be displayed on the console.

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

— **NOTE** —

Diskettes must be Double Sided High Density (1.44MB).

EXAMPLE:

The following example would mount a diskette in the Data Advantage floppy disk drive.

pSH+> pcmount 1.1

ping

ping - send 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 an encoded timestamp, and then an arbitrary number of bytes to pad out the packet. If *host_address* responds, *ping* will print out a message indicating the host is alive, then exit. Otherwise, after *timeout* seconds, it will print out a message indicating 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 the local network interface is running.

host_address must be specified in Internet dotted-decimal notation.

OPTIONS:

-s	Send one ping per second to host_address.
timeout	Maximum time to wait, in seconds, for a response from host_address.

Table 14 - **ping** options

EXAMPLE:

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

purge

purge - remove (unlink) files matching the given specifications

USAGE:

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

DESCRIPTION:

purge removes (directory entries for) one or more files.   Unlike the rm command, purge supports the wild card character '*'.

OPTIONS:

-q	Force files to be removed without displaying permissions, asking questions or reporting errors (quiet operation).
-d	Only remove files matching the file_spec if the file time and date is less than the time and date entered with this option.

Table 15 - **purge** options

NOTES:

It is forbidden to remove the file `..' (the parent directory) to avoid the consequences of inadvertently doing something like `purge .*'.

EXAMPLE:

pSH+> purge apr25*.log
purge: remove ./apr25b.log ? n
purge: remove ./apr25i.log ? n
Purge complete.

pSH+> purge -d 1994 04 24 02 *.log
purge: ./may09.log is not old enough to delete
purge: remove ./apr24.log ? y
 -> ./apr24.log deleted
purge: remove ./apr24a.log ? y
 -> ./apr24a.log deleted
purge: ./apr24c.log is not old enough to delete
purge: ./apr24d.log is not old enough to delete
Purge complete.

pwd

pwd - display the pathname of the current working volume

USAGE:
>   pwd

DESCRIPTION:

pwd prints the pathname of the working (current) directory.

OPTIONS:
None.

EXAMPLE:

pSH+> cd /activity
pSH+> pwd
1.2/activity

reboot

reboot - reboot the system

USAGE:
>   reboot [-h]

DESCRIPTION:

reboot performs an immediate and complete system reboot

OPTIONS:

-h	Perform immediate full system reboot (hard).

Table 16 - **reboot** options

NOTES:

It is recommended that the *sync* command be issued prior to reboot, to assure that all user data is flushed to the hard disk.

restart

restart - perform orderly Data Advantage system restart

USAGE:
>   restart

DESCRIPTION:

restart is used only when a *shutdn* command was previously invoked. It commands all the boards to allow incoming data calls to be accepted by, and routed through, the Data Advantage again.

An error message will be logged to the activity log device(s) and a message will be sent directly to the local console if the Data Advantage is unable to restart within 30 seconds of issuing this command. If, after 30 seconds, the Data Advantage is finally able to restart, an appropriate information message will be issued to the console and the activity log device(s).   If, however, the Data Advantage successfully restarts within the 30 second time limit, no message will be issued directly to the console, but one will be output to the activity log device(s) if information/warning logging has been enabled (see log command for more details).

This command will not be executed if a system shutdown is currently in progress due to a previously issued *shutdn* command.

OPTIONS:
None.

EXAMPLES:

pSH+> restart
  System already started.

pSH+> restart
  Shutdown in progress - cannot execute command.

pSH+> restart
  Data Advantage system startup requested.

rm

rm - remove (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	Force files to be removed without displaying permissions, asking questions or reporting errors.
-i	Ask whether to delete each file, and, under -r, whether to examine each directory. Sometimes called the interactive option.
-r	Recursively delete the contents of a directory, its subdirectories, and the directory itself.

Table 17 - **rm** options

NOTES:

It is forbidden to remove the file `..' (the parent directory).

Wild card characters ('*' and '?') are not supported at this time. See the **purge** command.

rmdir

rmdir - remove (unlink) directories

USAGE:
> rmdir directory ...

DESCRIPTION:

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

OPTIONS:
None.

shutdn

shutdn - perform an orderly Data Advantage system shutdown

USAGE:
    shutdn

DESCRIPTION:

shutdn is the inverse of the **restart** command. It commands all boards to ignore any further incoming data calls received by the Data Advantage. Any incoming call(s) in progress when this command is issued will be allowed to finish. Any data fragments queued in the Data Advantage when this command is issued will continue to be transmitted until complete.

An error message will be logged to the activity log device(s) and a message will be sent directly to the local console if the Data Advantage is unable to shutdown within 30 seconds of issuing this command. If, after 30 seconds, the Data Advantage is finally able to shutdown, an appropriate information message will be issued to the console and the activity log device(s).   If, however, the Data Advantage successfully shuts down within the 30 second time limit, no message will be issued directly to the console, but one will be output to the activity log device(s) if information/warning logging has been enabled (see log command for more details).

This command will not be executed if a system startup is currently in progress due to a previously issued **restart** command.

OPTIONS:
None.

EXAMPLES:

pSH+> shutdn
  Data Advantage system shutdown requested.

pSH+> shutdn
  Shutdown already in progress.

pSH+> shutdn
   Startup in progress - cannot execute command.

stats

stats - request Data Advantage system statistics

USAGE:

      stats [-b | -e] [-n]

DESCRIPTION:

stats will request statistics from all boards in the system. The results are placed in the file /activity/statfile.txt.

The Data Advantage provides two levels of statistics, Basic statistics and Extended statistics. The Basic statistics provides a brief report on a board's behavior, and is suitable to most users. The Extended statistics includes both the Basic statistics, plus detailed information targeted towards Data Advantage design engineering.

Of special interest for the user is the fact that the Extended statistics also includes statistics from RF control stations. When Data Advantage receives a request for Extended statistics, it forwards the request to all RF control stations. The RF control stations will response with formatted statistics containing both RDI protocol specific and RF site specific information. It may take several seconds for Data Advantage to collect statistics depending on the traffic in the RF control stations. Upon receiving the responses from all RF control stations, the content of the statfile.txt will be automatically displayed on the console. The Data Advantage will prompt the user to press [Enter] key to see the following pages. Entering 'q' and pressing [Enter] terminates the command.

When the Data Advantage completes its boot sequence, statistics are cleared, and the collection start time is set to the current system time. When statistics are requested, they are returned with the collection start time and end time. The end time will be the time at which the statistics were requested. By default, the statistics are cleared each time after they are requested, and the collection start time is set to the current time. This action may be overridden if the **-n** option is specified in the stats command.

The following list summarizes, both format and content, of the information contained in the Basic statistics group.

```
***************************************************************************
```
Basic Statistics for Board Number [board number], '[board label]'.
Statistics Start - [time] on [date]
Statistics End - [time] on [date]

Input	Output	Description
[X]	[X]	Total Number of fragments attempted
[X]	[X]	Total Number of failed fragments
[X]	[X]	Average Bytes/Fragment
[X]	[X]	Average fragment rate per minute
[X]		Input Peak fragment rate per minute
		Time of Peak - [time] on [date]
	[X]	Output Peak fragment rate per minute
		Time of Peak - [time] on [date]

Outbound Message Statistics.

	[X]	Active Outbound Messages.
	[X]	Peak Outbound Messages.
		Time of Peak - [time] on [date]

where

   [board number] = Physical position of board in card cage as viewed from left to right. The leftmost board (CAP board) is always board 1.

   [board label] = Text description of board as found in the SYSTEM.TXT file.

   [time] = Time in Hours:Minutes:seconds.

   [date] = Date in Month/Day/Year.

   [X] = Unsigned integer value.

OPTIONS:

-b	Request basic statistics from all Data Advantage boards.
-e	Request extended statistics from all Data Advantage boards.
-n	Do not clear boards' statistics after request.

Table 18 - **stats** options

EXAMPLES:

pSH+> stats -b
  Basic statistics requested from all boards.
  Data will be output to file 01.02/activity/statfile.txt.

sync

sync - force 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.

tail

tail - display 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 all jammed together, not specified separately with their own `-' signs.

+number	Begin 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	Begin 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.

Table 19 - **tail** options

timedat

timedat - control time on the CAP board Real Time Clock (RTC)

USAGE:
>   timedat [-t hh:mm:ss] [-d mm/dd/yyyy]

DESCRIPTION:

timedat is used to set the current system time and date.

The Data Advantage system time and the time stored in battery backed RAM, is set using the -t option.

The Data Advantage system date, and the date stored in battery backed RAM, is set using the -d option.

NOTE

The 'date' command also sets the system time and date, but it does not update the values stored in battery backed RAM. Therefore, if 'date' is used to set the time, and the system is powered off and on, this time value will be lost.

If no options are specified, the current system time and date are displayed.

OPTIONS:

-d	Update current Data Advantage date.

Table 20 - **timedat** options

EXAMPLES:

pSH+> timedat
  DATE: May 8, 1994 TIME: 3:01:24 pm

usage: timedat [-t hh:mm:ss] [-d mm/dd/yyyy]
  -t = set Data Advantage time
  -d = set Data Advantage date

pSH+> timedat -t 15:02:00
  Data Advantage system date/time set to 05/08/1994 15:02:00

pSH+> timedat -d 05/09/1994
  Data Advantage system date/time set to 05/09/1994 15:02:11

umount

umount - umount 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 Data Advantage 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
LOADER.SX SYSTEM.TXT
pSH+> cd 1.2/
pSH+> umount 1.1
pSH+> cd 1.1/
1.1/: no such file or directory

version

version - display software versions

USAGE:
> version

DESCRIPTION:

version displays the current software revisions of the Data Advantage application and operating system components.

OPTIONS:
None.

EXAMPLE:

pSH+> version

```
Data Advantage Application Software Versions:
File: LOADER.SX Version: R1A06 Path: 01.02/LOADER.SX
File: DACAP.SX  Version: R1A06 Path: 01.02/loads/DACAP.SX
File: WNI.SX    Version: R1A06 Path: 01.02/loads/WNI.SX

Operating System Software Version(s):
Node 1: ROM OS  Version: V03.02 Board Type: MVME-147
Node 2: ROM OS  Version: V01.03 Board Type: VCOM-24
Node 3: ROM OS  Version: V01.03 Board Type: VCOM-24
```

OCCASIONALLY USED COMMANDS

The following section discusses commands which are seldom, if ever, used on the Data Advantage.

cmp

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

Table 21 - **cmp** options

getid

getid - get user id and group id

USAGE:
> getid

DESCRIPTION:

getid displays the user and group id of the shell user.

OPTIONS:
None.

EXAMPLE:

pSH+> getid
uid: 20, gid: 100

getpri

getpri - get the priority of a task

USAGE:
> getpri task_name | -task_id

DESCRIPTION:

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

OPTIONS:
None.

EXAMPLE:

The following example would get the current priority of the Data Advantage's telnet server daemon.

pSH+> getpri tnpd
tnpd task priority = 50

network

network - display network statistics

USAGE:

       network topic [-node] [-as]

DESCRIPTION:

network displays a variety of statistical information regarding network activity. Refer to Appendix A for further discussion of the statistics group variables

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

if	Interface group statistics
icmp	ICMP group statistics
ip	IP group statistics
tcp	TCP group statistics
upd	UDP group statistics
fp	Ethernet (foreplane) statistics
bp	VMEbus (backplane) statistics

Table 22 - **network** topics

OPTIONS:

-node	Specify the Data Advantage board number from which the statistics are to be obtained; where node is the system board number (default is 1, i.e. the CAP board).
-a	Display all information available within the statistics group.
-s	Display only `special' inform available with the statistics group.

Table 23 - **network** options

EXAMPLES:

IF EXAMPLE:

The following example displays the Interface Group statistics on the CAP board. In this example, there are two network interfaces present; ifIndex 1 is the ethernet interface, and ifIndex 2 is the VMEbus Internet gateway. The ifIndex assignment is determined at system boot time, and reflects the order in which the network interfaces were installed. In most scenarios the ethernet interface is installed first (ifIndex = 1). In the case of subnetting, the interface with the most restrictive subnet mask will be installed first.

pSH+> network if

```
MIB Interface Table: ifIndex = 1
ifDesc = Ericsson: EDACS Data Gateway: 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


MIB Interface Table: ifIndex = 2
ifDesc = VMEbus Network Interface: Gateway: Node 1
ifAdminStatus   = 1           ifOperStatus    = 1
ifType          = 4           ifMtu           = 1500        ifSpeed = 10485760
ifInOctets      = 33096       ifInUcastPkts   = 618
ifInNUcastPkts  = 0           ifInDiscards    = 0
ifInErrors      = 0           ifInUnknownProtos = 0
ifOutOctets     = 82457       ifOutUcastPkts  = 1202
ifOutNUcastPkts = 0           ifOutDiscards   = 45
ifOutErrors     = 0           ifOutQLen       = 0
```

ICMP EXAMPLE:

The following example displays the ICMP group statistics on the CAP board.

```
pSH+> network 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 CAP board.

```
pSH+> network 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.

In this example, there is an ftp connection present between an network host and the Data Advantage CAP.

```
pSH+> network tcp
tcpRtoAlgorithm = [4]: Van Jocobson
tcpRtoMin      = 1000        tcpRtoMax       = 64000
tcpMaxConn     = 4294967295 tcpActiveOpens  = 4
tcpPassiveOpens = 9          tcpAttemptFails = 0
tcpEstabResets = 0          tcpCurrEstab    = 4
tcpInSegs      = 4705        tcpOutSegs      = 3996
tcpConnState   = 0          tcpRetransSegs  = 0
tcpInErrs      = 0          tcpOutRsts      = 0
```

In this example, there are no TCP connections currently present with the Data Advantage.

```
pSH+> network tcp
TCP Group statistics not currently available.
```

UDP EXAMPLE:

The following example displays the UDP Group statistics on the CAP board.

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

FP EXAMPLES:

The following example displays all (i.e. `-a') of the Ethernet foreplane statistics available on the CAP board. The first half of the display is the Interface Group statistics for the ethernet network interface. This data could have been viewed individually by entering `**network fp**'. The second half of the display is the "special" portion of the ethernet network statistics, which shows lower level information associated with the interface. This data could have been viewed individually by entering `**network fp -s**'.

```
pSH+> network fp -a

MIB Interface Table: ifIndex = -1
ifDesc = Ericsson: EDACS Data Gateway: MVME-147, AM7990
ifAdminStatus   = 1           ifOperStatus     = 1
ifType          = 6           ifMtu            = 1500        ifSpeed = 10485760
ifInOctets      = 801453      ifInUcastPkts    = 2204
ifInNUcastPkts  = 9945        ifInDiscards     = 0
ifInErrors      = 0           ifInUnknownProtos = 6510
ifOutOctets     = 99534       ifOutUcastPkts   = 1485
ifOutNUcastPkts = 386         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
```

BP EXAMPLES:

The VMEbus backplane of the Data Advantage may be viewed as a seperate Internet network within the Data Advantage. Each board is a unqiue host, with its own Internet address assignment.

The following example displays all (i.e. `-a') of the VMEbus backplane statistics available on the CAP board. The first half of the display is the Interface Group statistics for the VMEbus network interface. This data could have been viewed individually by entering  `**network bp**'. The second half of the display is the "special" portion of the VMEbus network statistics. This data could have been viewed individually by entering `**network bp -s**'. The "special" portion of the VMEbus backplane interface provides statistics relating to packet routing with the Data Advantage's Routing Address.

pSH+> network bp -a

```
MIB Interface Table: ifIndex = 2
ifDesc = VMEbus Network Interface: Gateway: Node 1
ifAdminStatus   = 1             ifOperStatus      = 1
ifType          = 4             ifMtu             = 1500        ifSpeed = 10485760
ifInOctets      = 39256         ifInUcastPkts     = 728
ifInNUcastPkts  = 0             ifInDiscards      = 0
ifInErrors      = 0             ifInUnknownProtos = 0
ifOutOctets     = 99767         ifOutUcastPkts    = 1422
ifOutNUcastPkts = 0             ifOutDiscards     = 45
ifOutErrors     = 0             ifOutQLen         = 0

Nism Special Function Statistics:
nismInUcastPkts  = 586         nismInOctets      = 57194       nismInErrors  = 0
nismOutUcastPkts = 537         nismOutOctets     = 30072       nismOutErrors = 0
nismGetSeg       = 586         nismRetSeg        = 0
nismGetBuf       = 537         nismRetBuf        = 0
nismGetSegErrors = 0           nismRetSegErrors  = 0
nismGetBufErrors = 0           nismRetBufErrors  = 0
nismQueueSends   = 586         nismQueueErrors   = 0
nismAddrErrors   = 0
```

REMOTE NODE EXAMPLE:

The following example displays the Interface Group statistics on node 2. With the exception of the `*fp*' command, any of the previous examples apply to all nodes within the Data Advantage.

pSH+> network if -2

```
MIB Interface Table: ifIndex = 1
ifDesc = Ericsson VMEbus Backplane Network Interface: System Node 2
ifAdminStatus   = 1             ifOperStatus      = 1
ifType          = 4             ifMtu             = 1500        ifSpeed = 10485760
ifInOctets      = 12501         ifInUcastPkts     = 299
ifInNUcastPkts  = 0             ifInDiscards      = 0
ifInErrors      = 0             ifInUnknownProtos = 0
ifOutOctets     = 9184          ifOutUcastPkts    = 191
ifOutNUcastPkts = 0             ifOutDiscards     = 0
ifOutErrors     = 0             ifOutQLen         = 0
```

pcmkfs

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

USAGE:

> pcmkfs [-i] volume_name format

DESCRIPTION:

pcmkfs will perform initialization (i.e. format) of 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	Call device driver initialization procedure.

Table 24 - **pcmkfs** options

NOTES:

The Data Advantage only supports *format* specification **4**; 1.4 Mbyte (3 1/2" high density)

EXAMPLE:

The following example would correctly format a diskette installed in the Data Advantage 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

popd

popd - pop 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 activity
pSH+> pwd
1.2/activity
pSH+> popd
pSH+> pwd
1.2/

pushd

pushd - push 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

resume

resume - resume 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 Data Advantage telnet server daemon, had it been previously suspended.

pSH+> resume tnpd

route

route - examine/modify network routes

USAGE:
>   route [-sh] add|delete [host|net] destination gateway

DESCRIPTION:

route manually manipulates the network routing tables normally maintained by the system routing daemon, or through default routes and redirect messages from routers, and the Data Advantage's /cnfg/SYSTEM.TXT file. *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 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 can not 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:

-s	Display routes in Internet dotted decimal notation.
-h	Display routes in hexadecimal notation.
host	Specifies destination is an IP host address (default).
net	Specifies destination is an IP network address.

Table 25 - **route** options

EXAMPLES:

The following example shows the current routing in dotted decimal notation.

```
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      147.117.37.57    1        DIRECT         0xFFFFF200
147.117.34.0      147.117.39.17    2        DIRECT         0xFFFFF200
190.3.0.0         147.117.39.24    2        INDIRECT       0xFFFFF200
```

The following example shows the current routing in hexadecimal notation.

```
pSH+> route -h

Destination:      Next hop:        IF:      Type:          Subnet Mask:
0x7F000001        0x7F000001       7        DIRECT         0xFF000000
0x93752000        0x93752539       1        DIRECT         0xFFFFF200
0x93752200        0x93752711       2        DIRECT         0xFFFFF200
0xBE030000        0x93752718       2        INDIRECT       0xFFFFF200
```

The following examples show the addition and deletion of routes.

pSH+> rout add net 190.4.0.0 147.117.39.17
pSH+> route add host 191.1.2.3 147.117.39.17
pSH+> route add 191.1.2.4 147.117.39.17

pSH+> route delete net 190.4.0.0 147.117.39.17
pSH+> route delete host 191.1.2.3 147.117.39.17
pSH+> route delete 191.1.2.4 147.117.39.17

scsi

scsi - query 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.

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  No information available.
SCSI ID: 4  No information available.
SCSI ID: 5  No information available.
SCSI ID: 6  No information available.
SCSI ID: 7  No information available.
```

The following example queries a specific SCSI ID for an active device, in this case, the Data Advantage system hard disk.

```
pSH+> scsi -2
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)
```

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.

setenv

setenv - set environment variables

USAGE:
>   setenv variable_name value

DESCRIPTION:

setenv will change a shell's variables to a new value. If used without any arguments, *setenv* will print a list of the shell variables and there 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.

setid

setid - set 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

setpri

setpri - set task priority

USAGE:
> setpri task_name | -task_id new_priority

DESCRIPTION:

setpri will set 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 Data Advantage's telnet server daemon.

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

suspend

suspend - suspend 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 Data Advantage telnet server daemon.

pSH+> suspend tnpd

touch

touch - update 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	Do not create filename if it does not exist.
-f	Attempt to force the touch in spite of read and write permissions on filename.

Table 26 - **touch** options

COMMANDS TO BE AVOIDED

The following section discusses commands which, in general, should not be used on the Data Advantage. The reasons they should be avoided may be summarized as follows;

1.   Similar functionality is provided by Data Advantage specific commands, which are uniquely tailored to the Data Advantage's functional requirements.

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

3.   Their functionality, such as the kill command, may result in degraded or catastrophic system behavior, if used incorrectly.

console

console - redirect the console output to a telnet session

USAGE:
        console [-r] [task_name]

DESCRIPTION:

console will redirect output that is going to the Data Advantage'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.

Table 27 - **console** options

EXAMPLE:

Telnet into the Data Advantage and enter: pSH+> console -r DBGT

All output from the task 'DBGT' 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.

date

date - display or set 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.

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 second. The year may be omitted; the current value is supplied as default.

See also Data Advantage specific command *timedat*.

EXAMPLE:

This example would set the date to Oct. 8, 12:45 AM.

pSH+> date 1008045

echo

echo - echo 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	Do not add the NEWLINE to the output.

Table 28 - echo options

kill

kill - terminate a task

USAGE:
> kill task_name | -task_id

DESCRIPTION:

kill will terminate 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 do its own resource clean up then terminate.

OPTIONS:
None.

EXAMPLE:

The following example would kill the ftp server daemon on the Data Advantage.

pSH+> kill ftpd

logmask

logmask - filter data to be stored in activity log

USAGE:

     logmask board_number level_mask class_mask specific_mask

DESCRIPTION:

logmask is used only for system debugging purposes. It is only accessible by super-user (root), and intended solely for use by qualified service personnel. It is a powerful command, which, if used incorrectly, can cause severe system performance degradation. The use of *logmask* is beyond the scope of this document, and it is included only for completeness. Qualifiers to *logmask* are as follows;

board_number	Decimal node number of the Data Advantage board. 0 sets masks on all Data Advantage system boards.
level_mask	32-bit hex value of activity logging Level mask.
class_mask	32-bit hex value of activity logging Class mask.
specific_mask	32-bit hex value of activity logging Specific mask.

Table 29 - **logmask** qualifiers

OPTIONS:
None.

EXAMPLES:

pSH+> logmask 0 03 0 f
  Mask values for all boards set to:
  Level=00000003h Class=00000000h Specific=0000000Fh

  WARNING: This overwrites the log command -i option status.

pSH+> logmask 1 f3 ffffffff 0000ff
  Mask values for board 1 set to:
  Level=000000F3h Class=FFFFFFFFh Specific=000000FFh

  WARNING: This overwrites the log command -i option status.

mac

mac - display/set 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 super-user (root) can modify the MAC address. Attempts by any users other than root will result in a "Permission denied" warning.

The board's MAC address is stored in battery-backed RAM, with the expected life of the battery being about five years. If the battery dies, the contents of this RAM will be lost, along with the MAC address. This command would be used to restore that address, which is printed on the VME board itself for reference. Any other use of this command to randomly change an address is definitely not recommended.

OPTIONS:

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

Table 30 - **mac** options

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.

mkfs

mkfs - make a file system (volume initialization)

USAGE:

mkfs [-i] volume_name label size num_of_fds

DESCRIPTION:

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

OPTIONS:

-i	Call device driver initialization procedure.

Table 31 - **mkfs** options

mount

mount - mount a file system volume

USAGE:

mount volume_name [sync_mode]

DESCRIPTION:

mount will mount a pHILE+ formatted 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).

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

OPTIONS:
None.

sleep

sleep - suspend execution for the specified interval

USAGE:
      sleep time

DESCRIPTION:

sleep suspends execution for *time* seconds.

OPTIONS:
None.

FILE TRANSFER PROTOCOL (FTP)

ftp - file transfer program

USAGE:

    ftp [host_address]

DESCRIPTION

ftp is the user interface to the ARPANET standard File Transfer Protocol (FTP). *ftp* transfers files to and from a remote network site. This section of the document describes the FTP client services available under pSOSystem, which are a subset of ARPANET FTP.

The client host with which *ftp* is to communicate may be specified on the command line. If this is done, *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.

WARNING

If the system is rebooted less than 60 seconds after any files have been transferred to the Data Advantage hard drive, file corruption may result. To avoid this situation, use the 'sync' command after an ftp operation if the system is to be immediately rebooted, reset, or powered down.

FTP COMMANDS

! [command]

Run command as a shell command on the local machine.

account [passwd]

Supply 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]

Append 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

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

bell

Sound a bell after each file transfer command is completed.

binary

Set the "representation type" to "image".

bye

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

cd remote-directory

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

cdup

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

close

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

cr

Toggle 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 an "network " ASCII" type transfer is made, these LINEFEED characters may be distinguished from a record delimiter only when *cr* is off.

delete remote-file

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

dir [remote-directory] [local-file]

Print a listing of the directory contents in the directory, *remote-directory*, and, optionally, placing 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

A synonym for close.

get remote-file [local-file]

Retrieve the file *remote-file* and store 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

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

Globbing for mput is done as in csh(1). For mdelete and mget, each remote file name 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 doing `**mls remote-files -**'.

mget and mput are not meant to transfer entire directory subtrees of files. You can do this by transferring a tar(1) archive of the subtree (using a "representation type" of "image" as set by the binary command).

hash

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

help [command]

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

lcd [directory]

Change 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]

Print an abbreviated listing of the contents of a directory, *remote-directory*, on the remote machine and, optionally, placing 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]

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

mdir remote-files local-file

Like *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

Expand the *remote-files* on the remote machine and do a *get* for each file name thus produced. See glob for details on the filename expansion. Resulting file names 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

Make a directory, *directory-name*, on the remote machine.

mls remote-files local-file

Like **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]

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

mput directory-name

Expand wild cards in the list of local files given as arguments and do a *put* for each file in the resulting list. See glob for details of filename expansion.

nlist [remote-directory] [local-file]

Print an abbreviated listing of the contents of a directory on the remote machine. 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.

open host [port]

Establish 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

Toggle 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]

Store 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

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

quit

A synonym for bye.

quote arg1 arg2 ...

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

recv remote-file [local-file]

A synonym for *get*.

remotehelp [command-name]

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

rename from to

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

reset

Clear 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

Delete a directory, *directory-name*, on the remote machine.

runique

Toggle 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]

A synonym for *put*.

sendport

Toggle 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

Show the current status of *ftp*.

sunique

Toggle 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

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

type [type-name]

Set the "representation type" to *type-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]

Identify yourself 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 done automatically on initial connection to the FTP server.

verbose

Toggle 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 quote (") marks.

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

ABORTING A FILE TRANSFER

The normal abort sequence, CTRL-C will not work during a transfer.

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 above checks, if "globbing" is enabled, local file names 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 a 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 specifies 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-10's and PDP-20's mostly). The "network" "ASCII" and "EBCDIC" types have a further subtype which specifies whether vertical format control (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".

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.

TELNET PROTOCOL

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.

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.

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]

Open 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

Close a TELNET session and return to command mode.

quit

Close any open TELNET session and exit *telnet*. An EOF (in command mode) will also close a session and exit.

status

Show the current status of *telnet*. This includes the peer one is connected to, as well as the current mode.

display [argument...]

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

? [command]

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

send <arguments>

Send 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**
> Send the current telnet escape character (initially `^]').

synch

Send 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

Send the TELNET BRK (Break) sequence, which may have significance to the remote system.

ip

Send 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>

Set 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:

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 local-chars 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:

localchars

If this is TRUE, then the flush, interrupt, quit, erase, and kill characters (see set above) are recognized locally, and transformed into (hopefully) appropriate TELNET control sequences (respectively ao, ip, brk, ec, and el; 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 is 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 throwing away 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

Toggle 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, 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

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

netdata

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

?

Display the legal toggle commands.

NOTES

Using **log -d** from a Telnet session will disable the automatic session termination. The **exit** command will still log the user out of the Data Advantage. However, the telnet session must be terminated by entering **CTRL** and **]** to get a telnet command prompt and then **q.**

This implementation of telnet does not support the "line by line" mode.

There is no adequate way for dealing with flow control.

After exiting telnet, the first character typed may be lost.

APPENDIX A: NETWORK STATISTICS

The following appendix defines the MIB-II variables which may be examined with the ***network*** command

INTERFACE GROUP VARIABLES

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

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.

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.

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 Data Advantage;

ifType	Description
6 = ethernet-csmacd	Ethernet interface
4 = ddn-x25	Direct Data Network (VMEbus)

Table 32 - Data Advantage Interface Types

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.

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.

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.

ifAdminStatus

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

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

Table 33 - ifAdminStatus States

ifOperStatus

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

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

Table 34 - ifOperStatus States

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.

ifInOctets

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

ifInUcastPkts

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

ifInNUcastPkts

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

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.

ifInErrors

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

ifInUnknownProtos

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

ifOutOctets

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

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.

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.

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.

ifOutErrors

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

ifOutQLen

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

IP GROUP VARIABLES

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

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 (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.

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

Table 35 - ipForwarding States

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.

ipInReceives

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

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.

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.

ipForwDatagrams

The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find 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.

ipInUnknownProtos

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

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 any datagrams discarded while awaiting re-assembly.

ipInDelivers

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

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 any datagrams counted in ipForwDatagrams.

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.

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. Note that this includes any datagrams which a host cannot route because all of its default gateways are down.

ipReasmTimeout

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

ipReasmReqds

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

ipReasmOKs

The number of IP datagrams successfully re-assembled.

ipReasmFails

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

ipFragOKs

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

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.

ipFragCreates

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

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.

Notes

1.   The IP Address Translation Table (ipAd*xxx* variables) can not be queried with the *network* command.

2.   The IP Routing Table (ipRoute*xxx* variables) can not be queried with the *network* command.

3.   The IP Network To Media Table (ipNetToMedia*xxx* variables) can not be queried with the *network* command.

4.   Refer to the *route* command for definition of IP routing related objects which may be queried or modified by the Data Advantage.

ICMP GROUP VARIABLES

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

icmpInMsgs

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

icmpInErrors

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

icmpInDestUnreachs

> The number of ICMP Destination Unreachable messages received.

icmpInTimeExcds

> The number of ICMP Time Exceeded messages received.

icmpInParmProbs

> The number of ICMP Parameter Problem messages received.

icmpInSrcQuenchs

> The number of ICMP Source Quench messages received.

icmpInRedirects

> The number of ICMP Redirect messages received.

icmpInEchos

> The number of ICMP Echo (request) messages received.

icmpInEchoReps

> The number of ICMP Echo Reply messages received.

icmpInTimestamps

> The number of ICMP Timestamp (request) messages received.

icmpInTimestampReps

> The number of ICMP Timestamp Reply messages received.

icmpInAddrMasks

> The number of ICMP Address Mask Request messages received.

icmpInAddrMaskReps

> The number of ICMP Address Mask Reply messages received.

icmpOutMsgs

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

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.

icmpOutDestUnreachs

> The number of ICMP Destination Unreachable messages sent.

icmpOutTimeExcds

> The number of ICMP Time Exceeded messages sent.

icmpOutParmProbs

> The number of ICMP Parameter Problem messages sent.

icmpOutSrcQuenchs

> The number of ICMP Source Quench messages sent.

icmpOutRedirects

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

icmpOutEchos

> The number of ICMP Echo (request) messages sent.

icmpOutEchoReps

> The number of ICMP Echo Reply messages sent.

icmpOutTimestamps

> The number of ICMP Timestamp (request) messages sent.

icmpOutTimestampReps

> The number of ICMP Timestamp Reply messages sent.

icmpOutAddrMasks

> The number of ICMP Address Mask Request messages sent.

icmpOutAddrMaskReps

> The number of ICMP Address Mask Reply messages sent.

TCP GROUP VARIABLES

The TCP group variables may be examined using the ***network 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.

tcpRtoAlgorithm

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

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]

Table 36 - TCP Retransmit Timeout Algorithm

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.

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.

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.

tcpActiveOpens

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

tcpPassiveOpens

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

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.

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.

tcpCurrEstab

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

tcpInSegs

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

tcpOutSegs

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

tcpRetransSegs

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

tcpConnState

The state of this TCP connection.

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

Table 37 - tcpConnState States

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.

tcpConnLocalPort

>   The local port number for this TCP connection.

tcpConnRemAddress

>   The remote IP address for this TCP connection.

tcpConnRemPort

>   The remote port number for this TCP connection.

tcpInErrs

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

tcpOutRsts

>   The number of TCP segments sent containing the RST flag.

UDP GROUP VARIABLES

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

udpInDatagrams

The total number of UDP datagrams delivered to UDP users.

udpNoPorts

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

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.

udpOutDatagrams

The total number of UDP datagrams sent from this entity.

Notes

The UDP Listener table can not be queried via the *network* command.

NISM GROUP VARIABLES

This section defines statistical objects collected by the Network Interface-Shared Memory (nism) Layer 2 entity. These data are supplemental to the Interface Table MIB-II group variables, which are also supported by the nism. The nism group variables may be examined using the ***network bp -s*** command.

Within the context of this section, "application" is synonymous with the Data Advantage's internal IP router.

nismInUcastPkts

Total number of packets received by the nism special function handler which were successfully forwarded to the application.

nismInOctects

Total number of octets received by the nism special function handler which were successfully forwarded to the application.

nismInErrors

Total number of packets received by the nism special function handler which could not be forwarded to the application due to errors or lack of resources.

nismOutUcastPkts

Total number of packets transmitted by the nism on behalf of an application request to send.

nismOutOctets

Total number of octets transmitted by the nism on behalf of an application request to send.

nismOutErrors

Total number of application requests to send a nism packet buffer which could not be satisfied due to errors.

nismGetSeg

Total number of segments allocated by the nism from the application registered region.

nismRetSeg

Total number of segments returned by the nism to the application registered region.

Note that the segment containing the packet is normally returned to it's owning region by the application. The nism will only return a segment (and increment nismRetSeg) if it can not be forwarded to the application level.

nismGetBuf

Total number of packet buffers allocated by the application from this nodes nism queue.

nismRetBuf

Total number of packet buffers returned by the application to this nodes nism queue.

nismGetSegErrors

Total number of segments the nism requested to be allocated from the application registered region which failed due to errors.

nismRetSegErrors

Total number of segments the nism requested to be returned to the application registered region which failed due to errors.

nismGetBufErrors

Total number of application requests to allocate a nism packet buffer which could not be satisfied due to lack of resources or errors.

nismRetBufErrors

Total number of application requests to return a nism packet buffer which could not be satisfied due to errors.

nismQueueSends

Total number of nism packet mail messages which were successfully enqueued to the application.

nismQueueErrors

Total number of attempts by the nism to mail a packet message to the application which failed due to errors.

nismAddrErrors

Total number of times the nism attempted to return a packet buffer which failed due to (hardware) address boundary errors.

This page intentionally left blank