```
/FXUtilities/hds-fx/config/FXstartup.sh
```

The FXstartup.sh file might contain (using C shell syntax for this example):

```
setenv XENVIRONMENT /FXUtilities/hds-fx/config/FXresources
```

The FXresources file would contain resources for the ViewStation local clients, such as:

```
# These are HDSterm resources
HDSterm*screen.printDevice:  /dev/tty2
HDSterm*screen.Background: ivory
# These are FX local Mwm resources
Mwm*keyboardFocusPolicy:  pointer
Mwm*configFile:  /usr/lib/X11/app-defaults/.mwmrc
# These are FX local OLWM resources
OpenWindows.SetInput:  followmouse
OpenWindows.WorkspaceColor:  White
```

1.2.8.1 Server-reset Command File

The server-reset file is run by the ViewStation whenever the server is reset. This would include starting up, manual reset commands, Xdm resets, and so on. This is a good place to start local processes. This is generally preferred over the startup command file.

You use the server-reset file by listing the file location in the configuration file with the following resource:

```
HDSsetup.server_reset.command.file: <file name>
```

You can also set this file name and path in the Startup Options menu of Setup Mode.

Processes in this file will be started on the ViewStation before XDMCP starts, so it is a good place to use ViewStation clients like xmodmap, so you could remap your keyboard before your login. This is especially useful for users with multinational keyboards and need special keys for their username and password.

1.2.8.2 Startup Command File

The ViewStation’s startup command file is run when the ViewStation starts up. This file specific for a ViewStation, not a user. User resources, if specified, will override these settings when the user logs in through this ViewStation. For this reason, it is used for specific ViewStation settings, such as for a print device, or color settings.

You use the startup command file by listing the file location in the configuration file with the following resource:

```
HDSsetup.startup.command.file: <file name>
```

You can also set this file name and path in the Startup Options menu of Setup Mode.

For example, suppose you have a ViewStation gray-scale monitor and you find that the HDSterm client’s background color (lightsteelblue) doesn’t give sufficient contrast with the text in the window. This Startup Command file would be a good place to specify a new color resource for the HDSterm. This would be specific to the gray-scale monitor but would not affect other ViewStations or other clients. Similarly, it would not be linked to your username and wouldn’t set this color change when you logged in from a ViewStation with a color monitor. This resource is:

```
HDSterm*screen.Background:    ivory
```

In another example, the Startup file might specify a printDevice resource to indicate the printer connections on this ViewStation.

```
HDSterm*screen.printDevice: /dev/tty2
```

A sample startup command file typically contains a reference to an Xenvironment file, which in turn contains the actual ViewStation client resources.

The Startup Command file name should be:

```
  close (pvout[0]);
  close (pvout[1]);
  execlp (cmd, cmd, rem_host, opendev, rem_device, (char *) 0);
  exit(1);        /* exec() failed */
  }
close (pvin[1]);
close (pvout[0]);

/* Now can use pvin[0] for reading, pvout[1] for writing */
...
cc = read (pvin[0], buffer, buf_size);
...
cc = write (pvout[1], buffer, count);
...
```

1.2.7 ViewStation Flash File System

The HDS ViewStation flashPROM board or PCMCIA devices (2.5 or 5.0 MB memory cards or 20 or 42 MB hard disk drives) can be used as a local Flash File system. This can be a convenient feature in many situations, such as operating the ViewStation at a remote location where loading fonts or files from the network might be difficult.

1.2.7.1 Operation with the flashPROM Board

In normal operation, the ViewStation’s server code is stored in the flashPROM memory. Depending on the server code file used, there may be some memory on the flashPROM board that is not used. This residual memory can be used for storage of other files, such as fonts, configuration files, local clients, and so on.

It is possible to use the entire flashPROM memory to store files of this type by specifying that the ViewStation load its server code from the network. This is generally contrary to the intent of using the flashPROM memory to avoid network downloading, but may be useful in some situations.

1.2.7.2 Operation with PCMCIA Devices

With the PCMCIA device installed, the Flash File system operates the same way utilizing, of course, the larger storage capacity.

The use and configuration of PCMCIA devices is described in detail in Section 1.10.

1.2.8 ViewStation Startup Files

The ViewStation has two files you can use to start processes on the ViewStation.

You could also use the **rsh** command in conjunction with a **stty** command to give transparent access to the port. For example, typing a command line directly into a modem might use:

```
stty raw –echo; rsh <fxname> opendev [ /dev/tty x ]
```

The **rsh** command may be used either in a shell script or in a compiled program. The shell syntax is given above, and would normally be activated as a sub-shell to allow redirection of the file descriptors created by the **rsh** command. Terminating the **rsh** command would terminate the process.

1.2.6.3 Usepty Script

The **usepty** script to use a pseudo-tty, provided on the HDSware tape in the */hds-fx/contrib/usepty/src* directory, is also useful here. A typical command line for a pseudo-tty for port 1 might look like:

```
usepty /dev/ptyrb rsh <fxname> opendev /dev/tty1 &
```

The **usepty** process will repeatedly monitor the pseudo-tty port and wait for a process to open on the corresponding slave side. When the slave side is opened, **usepty** will execute the **rsh** command to open the port on the ViewStation. When the slave side is closed, the **usepty** process will resume monitoring the pseudo-tty port.

1.2.6.4 Implementing "opendev" in a C Program

To use this command in a C program, the typical technique would create two pipes, one for input and one for output, then fork a child process and let the child process execute the **rsh** command. The sample file below illustrates this technique (". . ." indicates omitted code segments of the user’s choice):

```
...
char *cmd = RSH_PATH;          /* "/usr/ucb/rsh" or "rsh" */
char *rem_host = "hds-fx";          /* example only */
char *opendev = "opendev";
char *rem_device = "/dev/tty1"; /* example only */
int  pvin[2];
int  pvout[2];
...
pipe (pvin);          /* pipe to read from remote device */
pipe (pvout);          /* pipe to write to remote device */
if (fork() == 0) {
  dup2 (pvin[1], 1); /* stdout goes to pipe pvin */
  close (pvin[0]);
  close (pvin[1]);
  dup2 (pvout[0], 0); /* stdin comes from pipe pvout */
```

Remote Mount Directory: /usr3/bin/Asterix/fonts

results in

Full Font Path: /usr3/bin/Asterix/fonts/HDS-fonts

Note that if a file open to a particular font path fails, that font path is removed from the current font path, and the remaining paths are used. The failed font path is used again only after a ViewStation restart or a font path rehash.

1.2.6 Access to ViewStation Ports

The ViewStation’s Ports, one parallel and two serial, are generally accessible by any device on the network with conventional naming techniques. They are designated:

parallel port = /dev/pp0

serial port 1 (RJ11) = /dev/tty1

serial port 2 (DB25) = /dev/tty2

Other ports are named the same way:

audio port = /dev/audio

1.2.6.1 The opendev Command

The "opendev" command available from the ViewStation’s Console window is the entry point for a generalized bi-directional read/write capability for the ViewStation’s serial and parallel ports (the parallel port is write-only). This capability is described more fully below.

1.2.6.2 Remote Access to ViewStation Ports

It is sometimes desireable to permit a process running on a host to have direct access to one of the ports on the ViewStation; in other words, letting the remote ViewStation port act as a local port on the host. One situation where this would be useful might be additional user input from a serial device, such as a digitizer tablet, modem, or barcode reader. Other devices, such as badge readers or a security key, could also be connected to the ViewStation serial ports.

The ViewStation server has a command called "opendev". This command opens a device on the ViewStation, specified as an argument to the command, for bi-directional communication. The argument to the **opendev** command may be any device driver installed at a /dev/xxx node.

The command syntax is as follows:

rsh <*fxname*> opendev [*/dev/ttyx*]

which executes a remote shell command to the ViewStation to execute the **opendev** command with the specified device. This means that standard input from the **rsh** TCP port is written to */dev/ttyx*, and data from */dev/ttyx* is written as output to the **rsh** TCP port. The ViewStation has two serial ports, designated */dev/tty1* (an RJ11, RS-232 port) and */dev/tty2* (a DB25, RS-232 port), and one parallel port designated */dev/pp0* (DB25 Centronics) for local printing.

1.2.5.5 Mount Directory Example

For example, if the Local Mount Directory for Host1 specifies "/fonts" (the leading "/" is necessary), then a Mount Point is created in the ViewStation. The fontpath is specified as "/fonts/100dpi,/fonts/75dpi,/fonts/misc". The Remote Mount Directory for Host1 is "/usr2/lib/X11/HDS/hds-fx/fonts". When a font is requested, the ViewStation server finds the /fonts mount point and the fontpath, strips off the mount point, prepends the Remote Mount Directory, and then opens a connection to Host1 to the directory "/usr2/lib/X11/HDS/hds-fx/fonts/100dpi", etc.

This file structure can provide extra functionality for using the ViewStation on your network. If you want to have a backup font host, you could specify Local Mount Directories "/fonts" on Host1 and also on Host2. The ViewStation's mount point would first go to Host1, and if that file open failed (for whatever reason), the ViewStation would go on to Host2, and so on. The Remote Mount Directory on Host1 could be different than the Remote Mount Directory on Host2, but as long as those were correctly specified on the Host1 and Host2 menus, they would be found.

Another possible application is the handling of special fonts, such as the OPEN LOOK fonts, or special fonts for a word-processing application, that you wouldn't want to have on every network host. The following table shows the entries needed for specifying three font directories on three different hosts.

The Default Font Path on the User Preferences menu is "/fonts1/100dpi,/fonts2/Xol,/fonts3/HDS-fonts":

Host 1:

specifying

Local Mount Directory: /fonts1

Remote Mount Directory: /usr/lib/X11/hds-fx/fonts

results in

Full Font Path: /usr/lib/X11/hds-fx/fonts/100dpi

Host 2:

specifying

Local Mount Directory: /fonts2

Remote Mount Directory: /home/openwin/MITXol/fonts

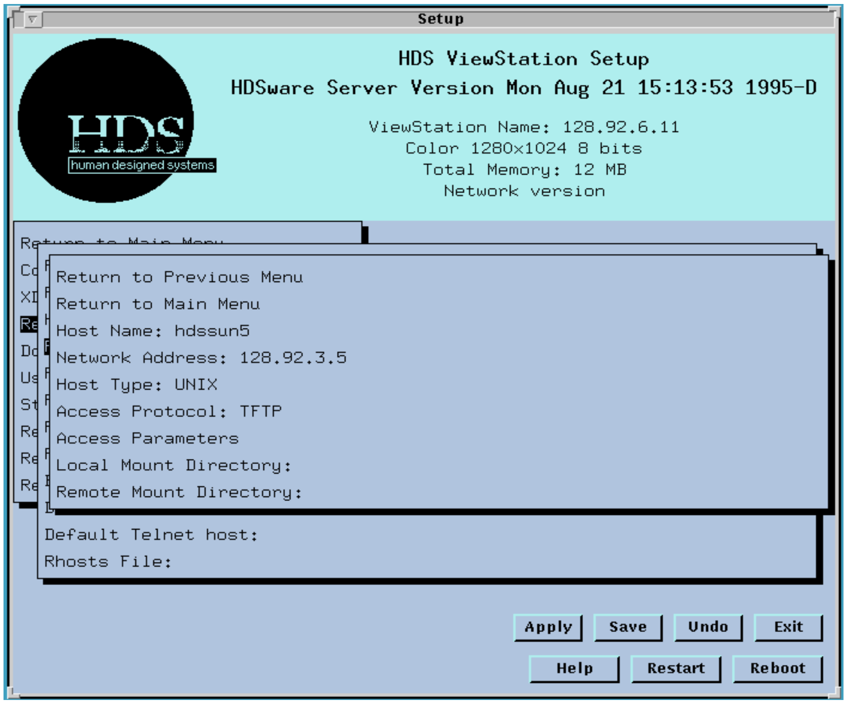
results in

Full Font Path: /home/openwin/MITXol/fonts/Xol

Host 3:

specifying

Local Mount Directory: /fonts3



HDS ViewStation Setup
HDSware Server Version Mon Aug 21 15:13:53 1995-D
ViewStation Name: 128.92.6.11
Color 1280x1024 8 bits
Total Memory: 12 MB
Network version

Return to Previous Menu
Return to Main Menu
Host Name: hdssun5
Network Address: 128.92.3.5
Host Type: UNIX
Access Protocol: TFTP
Access Parameters
Local Mount Directory:
Remote Mount Directory:

Default Telnet host:
Rhosts File:

The remote host address is given by the Network Address. If the Network Address field is empty, then the Host Name is used for the remote host address. Domain Name addresses are resolved by your Domain Name Server (or NIS).

1.2.5.3 Local Mount Directory

A Local Mount Directory defines an internal ViewStation location for the server to look for things. The file system maintains an internal list of these Local Mount Directories and searches that list when it opens a file. (The list contains all the specified internal file locations, like HDS$$ROM, Host Names, Local Mount Directories, and local ports, such as */dev/tty1*, etc.). If the Local Mount Directory is specified, then it is used as the mount point. If no Local Mount Directory is specified, the Host Name is prepended with leading slash and it becomes the mount point.

1.2.5.4 Remote Mount Directory

A Remote Mount Directory defines a location on the Remote Host as a starting place for the ViewStation to look for things. The Remote Mount Directory is typically a directory location where the HDSware tape was loaded or some special files are located. The Mount Point is the main directory, with subdirectories specified by the path itself. The Remote Mount Directory is prepended to the path (after the mount point is stripped off) to give a complete file location.

1.2.5 ViewStation Internal File System Structure

File names used on the ViewStation can include specification of a remote host and the directory structure on that host, as follows:

/<hostIP>/<path>/filename

For example:

/128.91.3.12/home/HDSware/hds-fx/image/V30-d.hds

which shows the host and path where the HDSware tape was loaded. Note the use of the leading slash. In this case

HDSware host = 128.91.3.12

HDSware path = /home/HDSware

Ordinary file management commands, such as cd, pwd, ls, etc. work in the ViewStation's Console window to show the ViewStation's internal file organization.

1.2.5.1 FXUtilities Abbreviation

There is a special case for host and path names. The HDSware host and HDSware path are abbreviated as "FXUtilities". This designation can be used instead of the full host and path name for that location.

From the example above,

FXUtilities = /128.91.3.12/home/HDSware

so you could enter:

/FXUtilities/hds-fx/image/V30-D.hds

or /HDSware/hds-fx/image/V30-D.hds

as an equivalent path and filename.

This technique is used in the ViewStation's configuration file for automatic selection of ViewStation paths once the HDSware information is specified.

With HDSware 3.1, a "BOOTPhost" abbreviation is also recognized if you are using BOOTP. This mount point uses the BOOTP host's IP address with the hd (home directory) entry in your bootptab file.

In the Console Window, the "ls" command shows the current mount points used by the ViewStation, including HDSware, FXUtilities, and BOOTPhost.

1.2.5.2 ViewStation Mount Points

The HDS ViewStation uses an internal file system that closely resembles the NFS file system, using mount points to establish local and remote paths. Local and Remote Mount Directories are entered in the Remote Hosts menu of Setup Mode for each remote host. Individual paths, such as for fonts or RGB color definition files, are accessed by means of the information the mount point provides.

```
*.info                          /dev/console
```

which sends messages from "info" level upward to the console window. If you want to set the level higher or lower, just use the desired level. If you want to send it to another destination, just change the destination entry. For example,

```
*.debug                         /dev/pp0
```

which sends all messages to the ViewStation’s parallel port, or

```
*.err                           @hostaddress
```

which sends messages **err** and above to the syslog daemon on the host specified.

Note that any changes must be entered into the syslog.conf file; the filename and path must be entered in Setup Mode; and Apply changes selected in order for the changes to be effective.

You could also use the ViewStation configuration file to make these changes. In this case you would modify the syslog entry appropriately, and the ViewStation restarted to re-read the configuration file. You could also use

```
kill –1 <syslogd pid>
```

to restart the syslog daemon without restarting the ViewStation.

1.2.4 Printing from the Console Window

There are a number of methods of printing the Console window contents, depending on your current setup. If you are running from your ViewStation display in an HDSterm window, you can use the HDSterm utilities, including HDSterm’s built-in print capabilities from its Commands menu. These print capabilities include Auto-print logging, printing the screen, etc. Refer to Section 2.2 for details.

Using the syslog.conf file, you can directly route error messages, at whatever level you specify, to one of the ViewStation’s serial ports, its parallel port, or to a remote host. This is discussed in detail below.

From an rlogin session you can capture the entire contents of the Console window screen to a file by using the **script** utility. Start an xterm or HDSterm window on a remote host, then run the **script** program (your UNIX may not have this utility or may call it something else) before you start the Console Window in that xterm window by starting an rlogin session to the ViewStation. In your xterm window, start the script with the command:

script <*filename*> for example, **script** tmp/hdsproblem.log

rlogin <*fxname*>

which will start saving everything that follows. Then issue the rlogin command to connect to the ViewStation Console Window client. All of the activity in the window is saved to this file. To exit the Console Window client, type “exit” which closes the rlogin session, and then close the script file with “exit”.

with the syslog.conf file. This file is located in the config directory in the HDSware path. By default, syslog.conf is NOT read. If you want to use this file, you must enter the syslog configuration file location in Setup Mode in the Startup Options menu. This file is read as soon as you Apply changes (and no longer read when you remove it). This makes it easy to change your error reporting if you are having difficulty. You can also enter a resource setting for it in the ViewStation's configuration file, as follows:

```
HDSsetup.syslog.config.file: /FXUtilities/hds-fx/config/syslog.conf
```

In the syslog.conf file you specify the level of message reporting you want and specify a destination for them. The ViewStation uses eight levels of messages, from debug level to emergency. Note that specifying a level of error message sends that level and all messages more urgent than that level to the destination.

Message reporting levels:

emerg	major problem, system is unusable
alert	problem with immediate action required
crit	critical condition
err	error condition
warning	warning condition
notice	normal, but significant information
info	information on system (default level)
debug	debug-level messages on operations

Message destinations:

/dev/console	This sends messages to all ViewStation console shells, including the Console clients, all "rlogin" remote sessions, and all serial port diagnostic sessions. This is the default destination.
/dev/pp0	This sends messages to the ViewStation parallel port
/dev/tty1	This sends messages to the ViewStation's serial port 1 (RJ11)
/dev/tty2	This sends messages to the ViewStation's serial port 2 (DB25)
@hostaddress	This sends messages to the syslog daemon on the specified host. If needed, that syslog daemon can redirect the messages.

The syslog.conf file has an entry with the following syntax (note the TAB character):

```
*.<message level> <tab character> <destination>
```

The default entry for messages is:

1.2.2.3 Privileged Commands in a Shell Script

If you want to use these privileged commands in a shell script, you must handle the password command carefully. For instance, if you want to use the “flash” command for all the ViewStations on a network, you could use one of these variations in your shell script:

Variation #1:

```
rsh <ViewStation name> sh –P << EOF
password
<the password>
flash files <path>
EOF
```

Variation #2:

Put in a file:

```
password
<the password>
flash files <path>
```

then use the “do” command with:

```
rsh <ViewStation name> sh –P < <
```

1.2.3 Console Window Messages

The messages available from the Console Window include things like the log file, network statistics, messages from HDSperx local clients, internal memory contents, status of the serial and parallel ports, and so on. They also inclulde messages from ViewStation clients. The messages include current activity messages, which might include the reading of a fontpath, loading a font, opening a client, etc. as well as any error messages from the host, or FTP, or the ViewStation itself. In many cases, the error messages indicate the nature of the problem; for instance, the error message “FTP Reply Error: 530 Login Incorrect" from FTP is sent when the FTP username has an invalid or empty password. Messages of this type are often deciphered for you by the ViewStation. The manual pages for each of these clients show the messages and point to the problems that generate them.

These messages are stored in a message log, which you can display using the "log" command. You can page through the logged messages with the "logpg" command. The messages are saved in a 64-line buffer so you can read them back, since they often come in large batches too fast to read. This log file buffer has a selectable size. You can change it with the resource:

HDSsetup.syslog.buffer.size

The message reporting level can be changed or these messages can be displayed to different devices, as discussed below.

1.2.3.1 ViewStation Error Message Reporting

The ViewStation has a great deal of flexibility in displaying error messages, both their level and the locations to which they are sent. These selections are made

rehash	Reread the Client Configuration Files specified in Setup
setenv	Modify environment variables `usage - setenv variable <value>`
srt	Display Token Ring Source Route table
token	Display Token Ring interface statistics
unsetenv	Remove environment variables `usage - unsetenv variable ...`
version	Display software versions

This list is subject to change. Many of these commands have command line arguments that affect their operation. Use help <command> for command syntax.

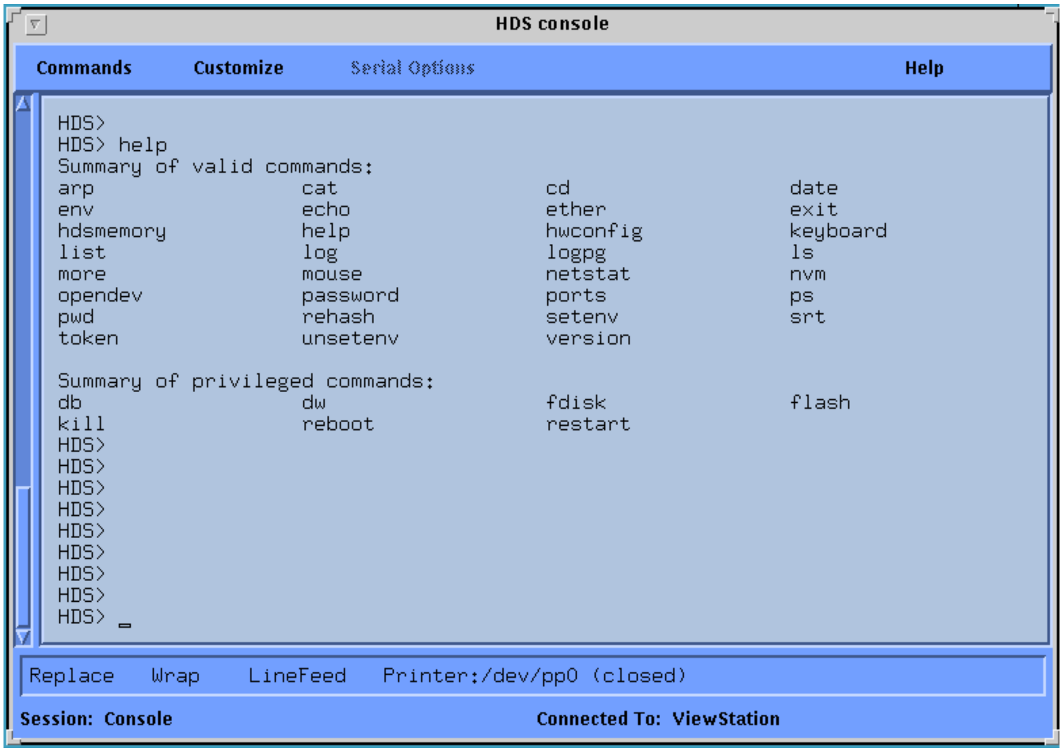
1.2.2.2 Summary of privileged commands

The following commands are privileged, that is, they cannot be used without supplying the Setup Mode password, if one has been assigned. All commands and clients can be defined as privileged, on an individual basis, by specifying the **passwdRequired** resource in the ViewStation's clients.config file.

From the prompt line of the Console Window, use the **password** command to enter the assigned password, then use the privileged command you want. When the password is entered successfully, the Console Window prompt changes from hds> to HDS> to indicate privileged mode operation. If your password entry fails, you cannot use these privileged commands.

db	Display memory using byte format	
dw	Display memory using word format	
fdisk	Formats the ViewStation's hard disk	
flash	Program Flash memory `usage - flash server <filename>`	
	for example:	flash boot <*filename*>
	boot	Program Flash boot PROM
	files	[-rs] [-n] [filename]
		-rs Remove Flash server code image
		-n Display number of bytes available for Flash filesystem
	server	Program server code image into Flash
kill	Send a signal to a process, or terminate a process `usage - kill [ -signal ] pid ...`	
reboot	Reboot ViewStation - reloads server code image	
restart	Restart ViewStation from current RAM server code image	

help	Display help for given command `usage -   help [ command ]` **h** and **?** also display the help messages	
hwconfig	Displays the ViewStation’s hardware configuration	
keyboard	Reset the keyboard	
list	List supported HDSshell commands including downloadable clients	
log	Display all log messages	
logpg	Display all log messages, one page at a time	
ls	List the contents of a directory `usage -   ls [ -1ACFLRdflqrs ] [ filename ... ]`	
more	Displays a file one page at a time	
mouse	Reset the mouse or trackball	
netstat	Display network protocol statistics `usage - netstat [-r] [-m] [-s] [-tcp] [-udp] [-ip] [-icmp] [-all]`	
	-r	display routes
	-m	display network buffer statistics
	-s	display protocol statistics
	-tcp	display TCP protocol statistics
	-udp	display UDP protocol statistics
	-ip	display IP protocol statistics
	-icmp	display ICMP protocol statistics
	-all	same as specifying every option
nvm	NVM commands `usage - nvm dump  or nvm save`	
	dump	Display NVM Contents
	save	Save settings in NVM
opendev	Provide direct access to a device. This is discussed in Section 1.2.6.2 in this chapter. `usage -   opendev <device_pathname>` where <*device_pathname*> is either /dev/tty1, /dev/tty2, or /dev/pp0	
password	Enter Setup Configuration password	
ports	Display serial and parallel port statistics	
ps	Display local task status	
pwd	Display current working directory	



```
HDS>
HDS> help
Summary of valid commands:
arp                 cat                 cd                  date
env                 echo                ether               exit
hdsmemory           help                hwconfig            keyboard
list                log                 logpg               ls
more                mouse               netstat             nvm
opendev             password            ports               ps
pwd                 rehash              setenv              srt
token               unsetenv            version

Summary of privileged commands:
db                  dw                  fdisk               flash
kill                reboot              restart
HDS>
HDS>
HDS>
HDS>
HDS>
HDS>
HDS>
HDS>
HDS>
```

These commands are listed below and discussed briefly. Using the **help** command or entering the command followed by a "?" displays a brief help message showing the arguments the command accepts.

A more complete list, including local client applications, is available with the “list” command. This list includes the downloadable ViewStation clients as well as the shell clients.

1.2.2.1 Summary of Console commands:

arp	Display the Address Resolution Protocol table
cat	Displays files to the standard output
cd	Change current working directory (of the Flash File System)
date	Displays or sets the date and time
env	Display the HDSshell environment. This command is useful in analyzing the environment accessible to the ViewStation for a **rsh** command.
echo	Echo arguments to the standard output
ether	Display ethernet interface statistics
exit	Exit the Console Interface
hdsmemory	Starts the memory local client

in the configuration file. If no file is specified or the file is missing, rhosts security is not used. If the file is present but empty, no access to the ViewStation is allowed.

Passwords for Console Window Commands

Some Console shell commands are labeled "Privileged", such as burning new code into the FlashPROM, displaying the contents of memory, or killing local processes. These commands are treated differently and can only be used with password access. When you use these commands, you must enter the Setup Mode configuration password, if a configuration password has been set.

In addition to these local shell commands, all of the other ViewStation local clients and can have password protection extended to them on an individual basis by modifying the **passwdRequired** resource in the appropriate clients.config file. In this way all Console Window commands, even ordinary commands like **ls** and **pwd**, can be password protected. For example, the **opendev** command is password protected by entering this resource in the clients.config file as follows:

```
hds.client.opendev.passwdRequired:  true
```

Alternately, the configuration file resource:

```
HDSsetup.password_is_global: yes
```

sets all Setup Mode values and all local clients to be password protected.

Note that the Help message screen which displays the available commands in partitions showing privileged and ordinary commands does NOT reflect changes you make in the password protection status. Refer to the listing of client attributes in the discussion of the clients.config file in Section 2.1 for details.

The **nvm dump** command does not display passwords, but simply reports <not displayed>.

1.2.2 Console Window Commands

The Console Window is really a shell which has access to the ViewStation's internal operation. It accepts a number of shell commands, shown on the help screen.

(note that the window prompt changes from hds> to HDS> - which indicates privileged mode)

```
HDS> <privileged command>
```

A system administrator might want to burn new firmware versions into all the ViewStations on the network from his remote location. This is possible using the remote shell commands and a list of host names and passwords. If you want to use the **rsh** command from a script to execute privileged commands in the Console window, you can follow this example:

```
rsh <fxname> sh -P < <password.script>
```

where <password.script> contains:

```
password
<fxname password>
<privileged command>
 (such as: flash /hdssun5/bootfile/V30-D.hds)
exit
```

1.2.1.4 Diagnostic Messages from a Remote Terminal

You can connect a character-based terminal (a VT100 type) to one of the ViewStation's serial ports. The ports are normally configured for a printer, so you may need to use the terminal's printer port or use a null modem cable; check the pin assignments in Section 1.1.10. Set the communications settings as your terminal requires. Then select the "Console" entry for the Serial Port Use field in Setup Mode. This setting automatically opens the console shell of the ViewStation for the serial terminal. This method is necessary if you are checking the start-up behavior of the ViewStation; sometimes it can also be used to get information about the ViewStation if the display is hung or the ViewStation seems to be dead. As it starts up, the ViewStation displays various initialization messages, such as the keyboard and mouse state, the nameserver suffix, etc.

1.2.1.5 Console Window Security

The Console Window is a powerful tool which has the capability to make important modifications to the ViewStation's internal operations. There are two methods of controlling security on the Console Window, by limiting access to the Console Window with the ViewStation's rhosts security (see Section 1.11.2) and by controlling access to individual commands with password control.

ViewStation rhost Security

The ViewStation uses an internal rhosts file to configure the rsh and rlogin daemons and control access to the ViewStation. By default, this security is not used. This feature is available with HDSware version 3.2 and later.

The ViewStation's rhost security is set up by entering an rhosts file name in Setup Mode on the Remote Hosts menu or with the resource:

```
HDSsetup.rhosts.file: <filename>
```

Depending on your use of the Console Window, you can use one or more of these types of display to run commands on the ViewStation or analyze the ViewStation's behavior. For instance, if you want to see whether the fontpath is loading correctly, you can do it from the Console Window. These three types of display are discussed below.

1.2.1.2 Console Window as a Local Display

When you click on the Console Window box in Setup Mode, a window appears. This is the console shell of the ViewStation and accepts shell commands and displays the ViewStation's activity. It is an ordinary HDSterm window and can be repositioned or iconified with your window manager. You can display a list of supported commands by using "h" or "help".

1.2.1.3 Console Window from a Remote Host

You can open the Console shell from another network device by using an **rlogin** command to the ViewStation hostname. This login capability, either rsh or rlogin, are controlled by the ViewStation's rhosts security (available with HDSware 3.2), which is discussed in Section 1.11.2. The ViewStation reports all attempts to gain remote access.

On a remote device, either a host computer or another ViewStation, (or even on your own ViewStation) just open an xterm window on your host, then issue the command:

rlogin <ViewStation hostname>   (or **rsh** or other syntax)

This puts the ViewStation console shell into the xterm window. This xterm window may be on your own ViewStation or from another device.

You can use **rsh** commands to the Console shell also, such as:

rsh hdsfx19 **nvm dump >** /tmp/nvmoutput.txt

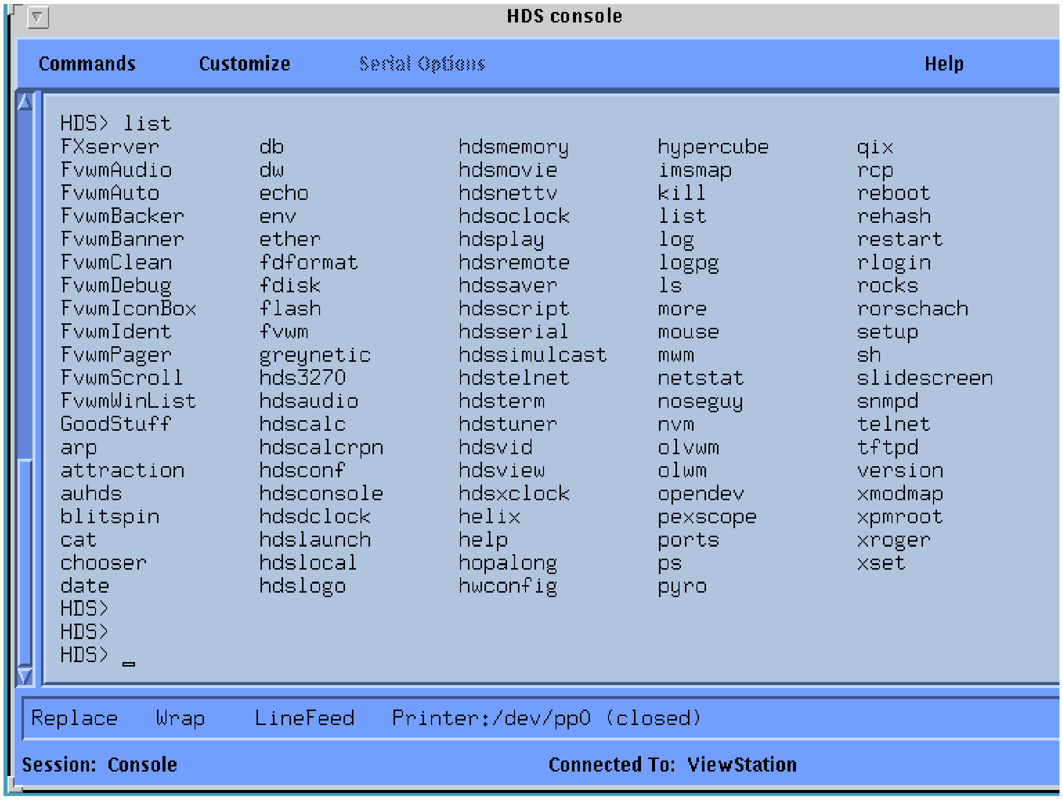
This is a convenient way to generate a system configuration file. You can configure one ViewStation the way you want (font locations, nameserver settings, download files, etc.), and then simply dump that ViewStation's NVM to a file, which can then be used for all other ViewStations.

The **rlogin** and **rsh** commands are protected by normal system security procedures. If a configuration password is set, sensitive commands (such as burning flashPROM and displaying memory contents) to the Console window with the **rsh** process may be run only with password access. All commands may be password protected (see Section 1.11.1.3).

For example, if you open a Console window (locally or on a remote device), you can execute any of the unprivileged commands. But if you want to use a privileged command, such as burning a FlashPROM, you must first enter the password command, and enter the configuration password:

```
hds> password
hds> enter Setup Configuration password:
```

The contents of this diagnostic screen, and the commands supported, are specific to different firmware releases. Your screen may be slightly different. Console Window commands are added with each software release. You can get a full listing of available commands and HDSperx clients with the “list” command.



```
HDS> list
FXserver        db              hdsmemory       hypercube       qix
FvwmAudio       dw              hdsmovie        imsmap          rcp
FvwmAuto        echo            hdsnettv        kill            reboot
FvwmBacker      env             hdsoclock       list            rehash
FvwmBanner      ether           hdsplay         log             restart
FvwmClean       fdformat        hdsremote       logpg           rlogin
FvwmDebug       fdisk           hdssaver        ls              rocks
FvwmIconBox     flash           hdsscript       more            rorschach
FvwmIdent       fvwm            hdsserial       mouse           setup
FvwmPager       greynetic       hdssimulcast    mwm             sh
FvwmScroll      hds3270         hdstelnet       netstat         slidescreen
FvwmWinList     hdsaudio        hdsterm         noseguy         snmpd
GoodStuff       hdscalc         hdstuner        nvm             telnet
arp             hdscalcrpn      hdsvid          olvwm           tftpd
attraction      hdsconf         hdsview         olwm            version
auhds           hdsconsole      hdsxclock       opendev         xmodmap
blitspin        hdsdclock       helix           pexscope        xpmroot
cat             hdslaunch       help            ports           xroger
chooser         hdslocal        hopalong        ps              xset
date            hdslogo         hwconfig        pyro
HDS>
HDS>
HDS> _
```

Replace   Wrap   LineFeed   Printer:/dev/pp0 (closed)

Session: Console   Connected To: ViewStation

1.2.1.1 Access to the Console Window

The Console Window client can be displayed in three different ways; more than one window may be active at any time, so you can have simultaneous sessions. The ViewStation user might be looking at the message log while the system administrator is examining the ViewStation’s network statistics from his remote ViewStation.

The three methods of Console Window access are:

- As a local display on the ViewStation, by clicking on the Console Window box in Setup Mode.
- From a remote location, by using **rlogin** or **rsh** (or **rcmd** or **remsh**, depending on your host’s syntax) from the remote host to the ViewStation. This access can be controlled with the ViewStation’s rhosts file, discussed in Section 1.11.2.
- From a remote ASCII terminal connected directly to the ViewStation’s serial port.

HDS ViewStation: Internal Architecture

This chapter describes the ViewStation's internal architecture, its internal file system, and console shell commands used by the ViewStation.

Section 1.0 - HDS ViewStation

1.2 HDS ViewStation Internal Architecture

The HDS ViewStation uses a UNIX micro-kernel architecture for its internal operation. This micro-kernel allows the ViewStation to run client applications locally, to use standard UNIX ports and communications techniques, and to provide the same UNIX interface for user control.

The internal UNIX micro-kernel is controlled by a shell window called the Console Window. Its operation is described below.

1.2.1 ViewStation Console Shell Window

The ViewStation has a local client window, called the Console Window, which uses an HDSterm window as its display. The Console window may be displayed locally or remotely, as described below.

The Console Window has two basic functions:

- Executing commands on the ViewStation (like loading a file into FlashPROM). The easiest way to become familiar with the Console Window is to bring it up (click on its button on the Main Menu of Setup Mode) and look at its commands. The commands available from the Console window are displayed by typing "h" or "help" in the Console window.
- Displaying data or messages about the current ViewStation status.

The diagnostic and error messages are displayed automatically to track events on the ViewStation. The level of error messages (and their destination) can be changed by modifying your syslog.conf file. This is discussed in greater detail later in this chapter.