Using the VxRouter ASCII Interface

The VxRouter can be controlled via an ASCII interface. This interface is accessible via a serial RS-232 port or over the network via a TCP port. Both ports use the same syntax. The command syntax is defined in the document:
VxRouter-ASCII-API.pdf.

The serial port is configured for 9600 baud, 1 stop bit, no parity, and software flow control. Each line contains only one command and must end with a carriage return (CR) and line feed (LF), or just a line feed (LF). The characters are not echoed.

The network interface listens on TCP port **17567**. It accepts the same commands as the serial interface. You may use telnet to manually open a connection and control the VxRouter.

VxRouter Configuration V4.0-9 and earlier

The VxRouter interface in controlled by entries in the file */etc/inittab*. Listed below are the relevant portions of the file that control the interfaces:

```
#run the ASCII interface program
::respawn:/usr/local/sbin/vxrapi --verbose
::respawn:/usr/local/sbin/vxrapi --serial
```

The line with '`--verbose`' starts the network connection. By default the network interface is started unless the '`--serial`' option is present.

If you make changes to the file */etc/inittab*, you must run the commands:

```
kill -hup 1
killall vxrapi
```

for the changes to take effect.

VxRouter Configuration V4.0-10 and later

Starting in version 4.0-10 and later, the interface program no longer requires the --serial option. Both the serial and network interfaces are started with the one command. Listed below are the relevant portions of the file that control the interfaces:

```
#run the ASCII interface program
::respawn:/usr/local/sbin/vxrapi --verbose
```

The version of the api program may be determined by any of the following:
1) looking in the system log file: */var/log/api* for the api signon message
2) running the command: `vxrapi -v` on the vxRouter
3) sending the command 'xversion' to the api command port

VxRouter Configuration

Each model VxRouter has it's own api program. They are: **vx40api**, **vx160api**, and **vx320api**. The generic name that is used in place of the actual name is: **vxrapi**.

The interface program has several options to control its operation. These options can be listed by running the api program with the option '--help'. Here is the output:

```
~ # vxrapi -help
Version: V4.0-11
Usage: vxrapi [options]
                          ----  network options  ----
-L|--listen[=]port      listen on this port, all addresses      (default: 17567)
   --vx[=]IP address    address of VxRouter                     (default: 127.0.0.1)
                             we will send commands to the above IP address, socket: 27567
                        ----  serial options  ----
-S|--serial[=]device    use this serial device                  (default: /dev/ttyS2)
-B|--baud[=]speed       sets serial baud rate                   (default: 9600)
                        ----  generic options  ----
   --CR                 output CRLF instead of just LF
-f|--facility[=]name    syslog facility reporting level         (default is local4)
                        valid names: auth, daemon, user, local0 through local7
                          see the man page for syslog.conf for more information
   --debug              write debug messages to the log file
   --api                write API messages received to the log file (level: INFO)
   --avr                write AVR commands to the log file (level: INFO)
   --clog               write connection status changes to the log file (level: NOTICE)
   --delay[=]delay      in ms, how long an upStream output must remain off, default is 300 ms.
                            minimum is 50ms, maximum is 1000ms, 0 will disable the delay
-b|--bcast[=]period     in seconds, how long between port status broadcasts, 0 will disable
                           broadcast
-V|--verbose            enable error text
-h|--help               display this help and exit
-v|--version            output version information and exit

 Default (no options) is to listen on socket 17567 at all IP addresses, listen on /dev/ttyS2
                              send to socket 27567 at 127.0.0.1

   signal SIGUSR1 will toggle api debug logging  (--api)
   signal SIGUSR2 will toggle avr debug logging  (--avr)
```

To change the serial port baud rate to 115200 and send CRLF at the end of each line, the command syntax is:

```
::respawn:/usr/local/sbin/vx40api -CR --baud=115200
```

Please remember that changes you make to the VxRouter firmware will be lost if you install a new SD card from Thinklogical™.

Debugging Aids

The api program has several options that may aid in setup and debugging. These include logging the incoming ASCII commands, logging the communications to the internal control process (aka AVR) and appending a comment to the response message that is returned after each command. This comment contains more details about an error.

All these messages are written to the system log file: */var/log/api* located on the VxRouter.

These are the command line options to enable debug aids:

- --debug log a lot of detailed data
- --api log the api commands received
- --avr log the commands to the internal control process
- --clog log connection make/break status
- --verbose append a comment to each command response

The '--api' and '--avr' settings may be toggled by sending a system signal to the running api program. As shown in the help text above, **SIGUSR1** will toggle the api setting, and **SIGUSR2** will toggle the avr setting.

Steps for V4.0-9 and earlier.

To send a signal to a program, you must know the Process ID (PID) of the program. PIDs may be listed with the command: '`ps -eF`'. A sample is shown below. The second column lists the PIDs and the last column lists the program command line. You need to find the **vxrapi** that does NOT have the '--serial' option listed. In the example below, there are two such lines each marked with a '◄'.

```
/usr/local/sbin # ps -eF
UID        PID PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
root       998    1  0   185   408   0 Jul07 ?        00:00:00 vxrcntl
root       999  998  0   185   408   0 Jul07 ?        00:00:00 vxrcntl
root      1002  999  0   185   408   0 Jul07 ?        00:01:02 vxrcntl
root      1003  999  0   185   408   0 Jul07 ?        00:06:51 vxrcntl
root      2427    1  0   109   236   0 13:39 ?        00:00:00 /usr/local/sbin/vxrapi --verbose ◄
root      2428    1  0   109   244   0 13:39 ?        00:00:00 /usr/local/sbin/vxrapi --serial
root      2434 2427  0   109   156   0 14:34 ?        00:00:00 /usr/local/sbin/vxrapi --verbose ◄
root      2437  772  0   206   344   0 14:34 pts/0    00:00:00 ps -eF
root      3098    1  0   269   412   0 Jul07 ?        00:00:03 /usr/sbin/ntpd
```

PID 2427 is the parent process for the network interface (it's PPID is 1). This is the PID that should receive the signal to toggle the debug mode.

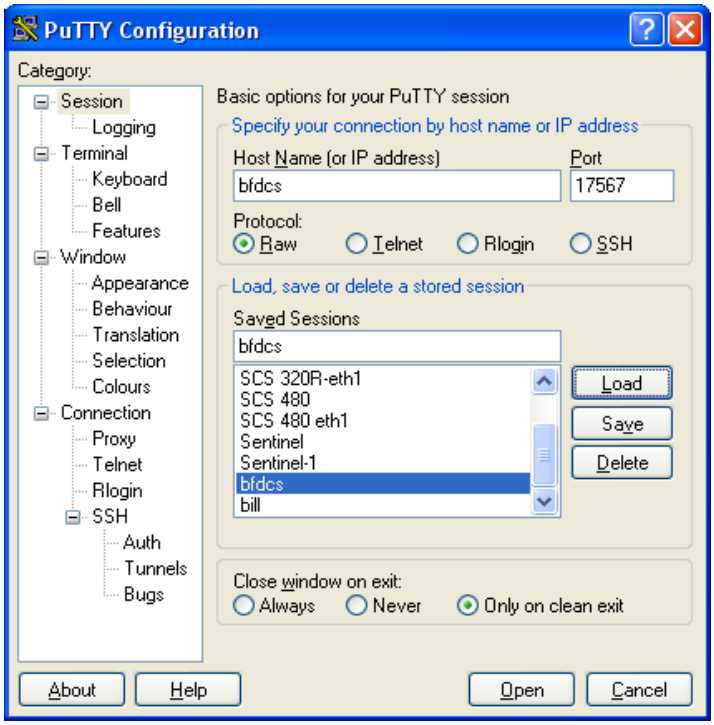
```
kill -USR1 2427
kill -USR2 2427
```

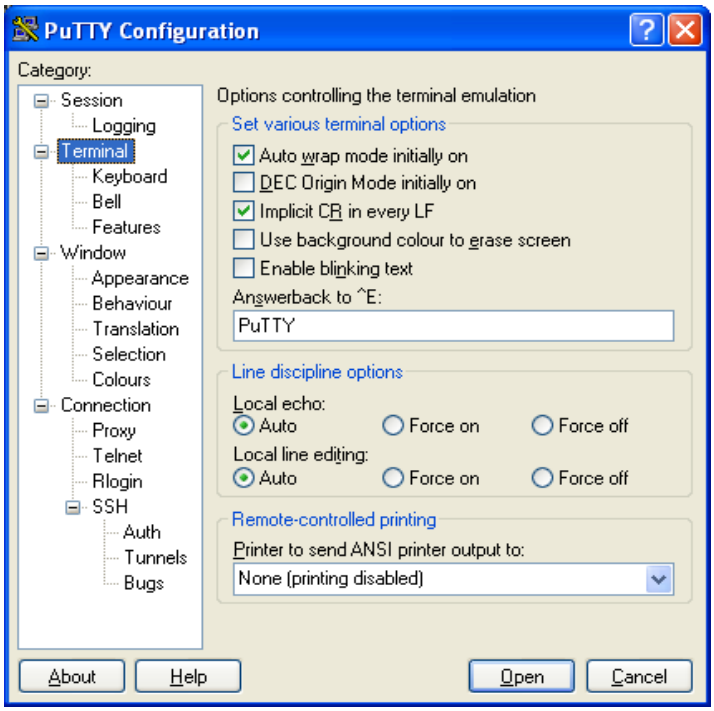
Version 4.0-10 and later can use a simpler method to send these signals.

```
killall -USR1 vxrapi
killall -USR1 vxrapi
```

Using putty to communicate to the API

When using putty (a Windows communication program) to communicate to a VxRouter, you must select 'Raw' mode and change the port to 17567. Also, under Terminal settings, you should check the 'Implicit CR in every LF' box.





You may also use telnet to access the ASCII control port. The command syntax is:

```
telnet  IPaddress  17567
```

Replace ***IPaddress*** with the actual IP address of the VxRouter.

Log file debugging entries

When debugging is turned on, several different types of messages are written to the system log file: */var/log/api*. (Early systems did not have a separate log file for the API, all log entries are located in the file: */var/log/messages*.)

Listed below are four commands that were sent to a Vx160. Below those commands is a piece of the log file showing the corresponding log entries. The first line in the log file records the event that enabled logging.

```
ci0150o0151
si0150
so0151
do0151
CI031200001
```

```
Jul 29 15:44:18 vxrouter vxrapi[391]: caught signal SIGUSR1 (10), turn on api debugging
Jul 29 15:44:55 vxrouter vxrapi[460]: command:  ci0150o0151   response: R0000OK#ci0150o0151
Jul 29 15:45:14 vxrouter vxrapi[460]: command:  si0150   response: R0000OKI0150O0151#si0150
Jul 29 15:45:18 vxrouter vxrapi[460]: command:  so0151   response: R0000OKI0150O0151#so0151
Jul 29 15:45:25 vxrouter vxrapi[460]: command:  do0151   response: R0000OK#do0151
Jul 29 15:58:32 vxrouter vxrapi[463]: command:  CI031200001   response: R0000ER0006#Input port
number 312 is out of range
```

The log entries list the command that was received and the response sent back to the sender.

The log files contain other system related messages and are a valuable aid in troubleshooting. Please note that these logs are maintained in RAM, and will be destroyed during a reset or loss of power to the VxRouter.

A note for telnet users

By default, vxrapi does not send a CR on each line. This is a problem for Windows telnet client software. The api (v4.0-4 and later) has a command to turn on CR's. This command is **XCRON**. If your telnet output from vxrapi is spread out over the screen, then send the XCRON command.