thinklogical

Configuring the ASCII Interface

Thinklogical, LLC ®
100 Washington Street
Milford, Connecticut 06460 U.S.A.
Telephone : 1-203-647-8700
Fax : 1-203-783-0049
www.thinklogical.com

Value Your Content
thinklogical®
Trust Our Proven Ingenuity

Revision: G

Preface

The router is 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: **Router-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.

(references to a VxRouter also apply to a HdxRouter and a MxRouter)

Router Configuration V4.0-9 and earlier

The router interface is 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.

After making changes to the file /etc/inittab, you must run the commands:

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

for the changes to take effect.

Router 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

API Command Line Options

Each model router has it's own API program. Examples are: ***vx40api***, ***vx160api***, ***mx48api*** 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: V4.5-1
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
-N|--connections[=]n      set the maximum number of open connections allowed    (default: 12)
   --mcast[=IP address]   replace broadcasts with multicast to this IP address  (default:
                                        239.255.13.9)
                          ----  serial options  ----
-S|--serial[=device]      use this serial device                        (default: /dev/ttyS2)
   --serial=none          disables the API serial port /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
-D|--debug                write debug messages to the log file (multiple options increase the debug
level
   --api                  write API messages received to the log file (level: NOTICE)
   --avr                  write AVR commands to the log file (level: NOTICE)
   --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 ms, how long between port status broadcasts, default is 4000 ms.
                             minimum is 500, maximum is 15000, 0 will disable the broadcast
                              if multicast is enabled, it will use this time setting
                              --bcast=0 will disable BOTH broadcasts and multicasts
-V|--verbose              enable error text in responses
-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
                              send to socket 27567 at 127.0.0.1
                              accept commands from the RS232 port
                              broadcast switch status every 4000 milliseconds

   signal SIGUSR1 will toggle api debug logging  (--api)
   signal SIGUSR2 will toggle avr debug logging  (--avr)
   signal SIGHUP will force the API to reread the P2P csv files located in /var/local/router/p2p/
```

To change the serial port baud rate to 115200 and send a CRLF at the end of each line, the command syntax for the line in /etc/inittab would be:

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

Please remember that any changes you make to the router 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 router internal SD card.

These are the command line options to enable the API 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. (see the example below for the syntax of the kill command)

Steps for versions 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** line that does NOT have the '--serial' option listed. In the example below, there are two such lines each marked with a '◄'.

```
 # ps -eF
UID        PID PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
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
```

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
```

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

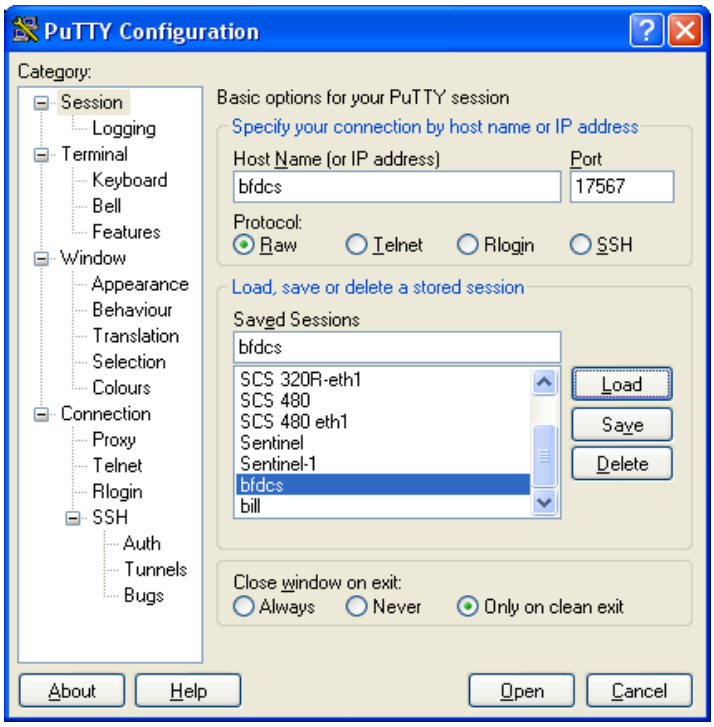
```
killall -USR1 vxrapi
killall -USR2 vxrapi
```

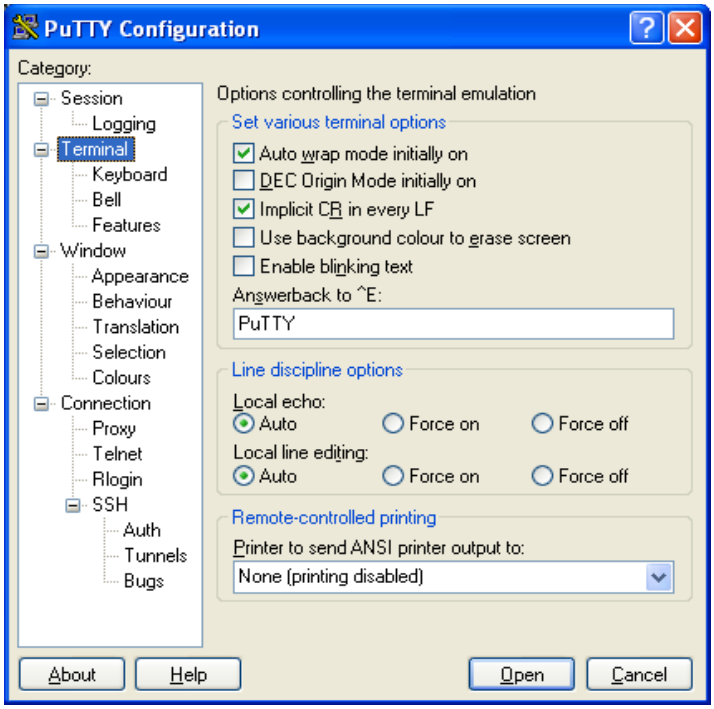
```
 # ps -eF
UID        PID PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
root       462    1  0   108   244   0 14:35 ?       00:00:00 /usr/local/sbin/vxrapi --verbose
root       463  453  0   206   344   0 14:35 pts/0   00:00:00 ps -eF
```

Note that there is only one copy of vxrapi running.

Using putty to communicate to the API

When using putty (a Windows communication program) to communicate to a router, 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 router.

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 were written to the file: /var/log/messages.)

Listed below are five 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
```

These 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 erased during a reset or loss of power to the VxRouter. Linux does offer the means to send these log file entries over the network to another system. That discussion is outside the scope of this document. Some information may be found in the document: **Router_Interfaces.pdf**

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.

Using the Router ASCII Interface V3.5

A modified version of API version 3, referred to as V3.5 now exists. It combines features of API V3 (automatic back channel connections) and the optimization of API V4. The API commands are described in the document: **Router-ASCII-API-V3.5.pdf**.

API V3.5 is only available on the Vx40, Vx160 and Vx320 routers. The files names are: ***vx40api.v35***, ***vx160api.v35***, ***vx320api.v35***

If you need to switch API versions, log in to the vxRouter and run these commands: (use the correct API file name for your router)

From V4 to V3.5

```
cd /usr/local/sbin
rm vxrapi
ln -s vx40api.v35 vxrapi
kill -hup 1
```

From V3.5 to V4

```
cd /usr/local/sbin
rm vxrapi
ln -s vx40api vxrapi
kill -hup 1
```

Disabling Broadcast on Router Input Ports

Version 4.4 (and later) of the API will allow input ports to be restricted to just one output port at a time. If input X is routed to output A and then connected to output B, X will be disconnected from A and then moved to B. This mode is referred to as Point-to-Point mode, or P2P for short.

The P2P definition files are stored on the Controller Card in the following files:

- ***/var/local/router/restrict/upstream.csv***
- ***/var/local/router/restrict/downstream.csv***

P2P mode is disabled when the definition files do not exist. By default, when there are no files, input ports may connect simultaneously to any number of output ports. All VX Routers are shipped without any P2P files stored on the Controller card.

These files are in the form of a comma separated value (csv) file. Each entry in the file is the input port that you want to restrict to P2P mode. You may have one or more entries per line in the file. The only characters allowed in the file are the digits 0 thru 9, commas, and spaces. Blank lines are allowed.

An example that set ports 1,2,3,4,9,11,15 to P2P mode is:

```
1,2,3,4
   9
11, 15
```

As a shortcut, you may use the value 9999 to indicate ALL input ports.

The ***upstream.csv*** file is read by all router models. The ***downstream.csv*** file is only read by the VX160 and VX320 models. Downstream on the VX160 controls connections between the Blue card inputs and Green card outputs; on the VX320 it controls the cards on the lower shelf.

If you make changes to either of these files, you may force the API to reread them by issuing the command: **`killall -HUP vxrapi`**

If P2P files are found at startup, the API will log this to the file: ***/var/log/api***. An example is shown below.

```
Mar 11 17:51:59 vxrouter vxrapi[508]: starting Vx80Router ASCII API   Version: V4.4-0  (build: 20)
Mar 11 17:51:59 vxrouter vxrapi[508]: parsing P2P csv file /var/local/router/p2p/upstream.csv
Mar 11 17:51:59 vxrouter vxrapi[508]: parse_api_csv_file.c@133: line 1 of csv file
             /var/local/router/p2p/upstream.csv  '1,2,3,4 '
Mar 11 17:51:59 vxrouter vxrapi[508]: point-to-point mode enabled
```

Using Multicast to send Router Status instead of Network Broadcast

Version 4.5 (and later) of the API has a command line option to use a network multicast address instead of a broadcast address to deliver router status messages. The default mode of the API is to broadcast over the network its' status. The use of a multicast address allows these messages to be routed to other network segments. If all this is foreign to you, don't worry, just forget about it and use the default settings.

The command line option to enable multicast is '**--mcast**'. Without any additional options, it will enable multicast using the default IP address of 239.255.13.9. When multicasting is enabled, the API will create a log entry showing the IP address it is using.

```
May 20 16:18:06 vxrouter vxrapi[2230]: starting Mx48Router ASCII API   Version: V4.5-1  (build: 27)
May 20 16:18:06 vxrouter vxrapi[2230]: Using multicast IP address 239.255.13.9
May 20 16:18:06 vxrouter vxrapi[2230]: multicast mode enabled
```

If broadcast is in effect, the log file will record that information.

```
May 20 16:35:15 vxrouter vxrapi[2268]: starting Mx48Router ASCII API   Version: V4.5-1  (build: 27)
May 20 16:35:15 vxrouter vxrapi[2268]: Using broadcast IP address 192.168.13.255
```

If you don't want to use the default multicast address, it can specify the address you require on the command line by adding an equal sign '=' and the IP address to the '**--mcast**' option. For example: '**--mcast=239.255.13.105**'

The selection of a multicast address is beyond the scope of this document. You will need to contact your IT / Networking department. The address you use may have some effect on your network infrastructure, so contact the appropriate personal.

The API will attempt to use the multicast IP address you specify. If it can not connect to that address, the API will faill back to using broadcast. This is also logged.

```
May 20 16:17:09 vxrouter vxrapi[2225]: starting Mx48Router ASCII API   Version: V4.5-1  (build: 27)
May 20 16:17:09 vxrouter vxrapi[2225]: broadcast_setup() vxASCIIapi.c@5193: Error: (Cannot assign
       requested address) in bind [IP address = 1.2.3.4], switching to broadcast
May 20 16:17:09 vxrouter vxrapi[2225]: Using broadcast IP address 192.168.13.255
```

The example above shows an attempt to use the IP address of 1.2.3.4. The API was not able to connect to that address, so it went back to using broadcast.

Both broadcast and multicast will use UDP port 17564 to transmit the status.