Table of Contents

Overview....2
Conventions....3
Port numbers....3
Results....3
Upper vs. Lower case....3
Commands....4
Connection commands....4
Connect....4
Disconnect....4
Status....5
XGET....5
XPUT....5
System Information commands....6
XVERSION....6
XINSTALLED....6
XDCSTYPE....6
XMAXPORTS....6
XMAXCARDS....7
XLASTEVENT....7
XALARM....7
XRESET....8
XCRON and XCROFF....8
XPORTCONFIG....8
XQUIT....8
XHELP....9
VxRouter Port Numbering....10
Error and Status Codes....17
Commands Removed from Version 4....17
Examples....18
Changing from V3 to V4 API commands....19
API Manual Revision History....19

Illustration Index

Illustration 1: Vx40 Port Numbering....11
Illustration 2: Vx40 Connections....12
Illustration 3: Vx160 Port Numbering....13
Illustration 4: Vx160 Connections....14
Illustration 5: Vx320 Port Numbering....15
Illustration 6: Vx320 Port Connections....16

This symbol is used to mark important information that will be needed to effectively use this API with the VxRouter matrix switch.

Overview

This document describes the command set used to control LSI's VxRouter series of matrix switches. The commands are all ASCII based and are terminated with either a linefeed character or a carriage-return / linefeed pair. Port numbers are all 4 digits in length and filled with leading zeros (Example: port 12 is encoded as 0012).

The VxRouter commands are based on the command set used by the DCS product line. Some DCS commands were removed because they did not apply to the VxRouter system. New commands were added to accommodate features in the VxRouters that were not in the DCS.

This release of the API marks a radical departure in the procedure used to make or break connections. Previously, the VxRouter was treated as having 40, 160, or 320 bidirectional ports. When a connection command was received, in most cases two connection paths were created. One path allowed information to flow from a transmitter to a receiver, and the second path allowed information to flow from the receiver to the transmitter. The second path was sometimes referred to as the 'low speed channel' or the 'back-channel'.

One advantage of this automatic path creation was to require only one connection command to create two connection paths. The software took care of maintaining the second path. The drawback from this feature is that none of the ports used by the second path were available for any other use. This could lead to half the ports on a VxRouter being unusable.

The API commands have been modified in Version 4 of the interface. These commands give the user complete control over every input and output port of the VxRouter switch matrix. The drawback is the need to issue two commands to connect or disconnect bidirectional data instead of one command. You still need only one command to connect or disconnect video data.

The ability to make any possible connection means that it is now possible to have one keyboard/mouse connected to multiple CPU’s. The user can be watching the video from, and interacting with, CPU A while simultaneously sending keystrokes to CPU’s B, C, D,...

Conventions

Port numbers

- All port numbers are four places long, start counting at 1 and have leading zeros. They are four places long. For example, port 15 is encoded as 0015.
- A port number of 4 zeros is used in response to a status command to signify that no port is in use. A zero port number is not valid in a command string.
- A port number of four 9s (9999) in used to signify ALL ports. It is not valid in a response. For example, if you wanted to encode all output ports, you would use: O9999
- In the following sections, xxxx is used as a generic input port place holder; yyyy is used as the output port place holder. Do not use xxxx or yyyy in a command, but replace each with the appropriate input or output port number.

Results

Results from commands are ASCII strings terminated with a newline (linefeed). The first character is an 'R', followed by a 4 digit, zero-filled length. The length includes the trailing newline. Following the length will be either 'OK', or 'ERnnnn'. OK signifies the command was successful, while ERnnnn is an error code. After the OK or ERnnnn, a comment may appear giving more detailed information.

In the case of a status command, the OK is followed by the status response.

You must wait for a result response before sending another command.

There are several options to control the output from the API. These options are described in detail in the document *Using-the-ASCII-Interface.pdf*. Two of the options are:

1. [--CR] Include a carriage return on each line output. (Useful for Windows)
2. [--verbose] Append a comment to each response with more information about an error code, or repeat the successful command.

Examples of verbose output

Command: `CI000400007`
normal: `R0003OK`
verbose: `R0015OK#CI000400007`

Command: `CI000400087`
normal: `R0007ER0007`
verbose:

```
R0072ER0007#Output port number 87 is out of range (1 thru 80): 'CI000400087'
```

Upper vs. Lower case

The upper case single letter commands affect the upper shelf of a Vx320, the upstream to downstream paths of a Vx160, and all the paths on a Vx40.

The lower case single letter commands affect the lower shelf of a Vx320, the downstream to upstream paths of a Vx160, and are NOT valid on a Vx40.

All other commands may be in either case.

Commands

Connection commands

Connect

Connect one input port to one or more output ports. Connections are additive.

For example, connecting input 5 to output 7 will result in 7 being added to any existing connections to input 5. The 'i' and 'o' may be of either case.

Format: `CixxxxOyyyyO....`
`cixxxxoyyyyo....`

- xxxx of all 9's is illegal.
- yyyy of all 9's will connect xxxx to all outputs.
- Connections will be made in the order specified.

Results:

- `R0003OK or R0007ERnnnn`

When switching keyboard/mouse channels it is now possible to connect one keyboard/mouse to multiple computers. *Unless the appropriate video connections are also made, you will be sending commands to computers that you are not currently viewing.* **This could have disastrous results!**

Disconnect

Disconnect one or more input or output ports. Disconnecting outputs only affects that port, but disconnecting inputs will affect every output connected to that input. The 'i' and 'o' may be of either case.

Format: `Dixxxx...` or `Doyyyy.....`
`dixxxx...` or `dOyyyy.....`

- This will disconnect all outputs connected to input port xxxx or will disconnect output port yyyy only.
- DI9999 or DO9999 will disconnect ALL ports.
- It is not an error to disconnect a port that is not connected.

Results:

- `R0003OK or R0007ERnnnn`

Status

Return the connection status of an input or output port.

Format: Sixxxx
sixxxx

- Return the list of output ports that are connected to input port xxxx.
- If nothing is connected to port xxxx, 0000 will be returned.

Format: Soyyyy
soyyyy

- Return the input port number that is connected to output port yyyy.
- If nothing is connected to port xxxx, 0000 will be returned.

Results (SI):

- RnnnnOKIxxxxOyyyyOyyyyOyyyy... or R0007ERnnnn
  - xxxx is connected to output(s) yyyy.
  - 0000 means not connected.

Results (SO):

- RnnnnOKIxxxxOyyyy or R0007ERnnnn
  - xxxx is connected to output yyyy.
  - 0000 means not connected.

XGET

Return the entire switch connection state. The output from this command is used by the XPUT command.

Format: XGET

- This will return a (very) long string listing every connection. The connection list consists of a series of input port numbers, followed by output port numbers.
- The XPUT command requires all the characters in the response that come after the 'OK'.

Example Results:

- R0037OKI00010003I00020004i00100010001100l2
- R0029OKI00015003I00105010501150l2
- R0012OKI51230145
- R0003OK or R0007ERnnnn

XPUT

Restore the entire switch connection state. The output from XGET is used by this command.

Format: XPUTstr

- The XPUT command will turn off all outputs and then make the connections that are listed in 'str'. 'str' is the response that was returned from a XGET command.
- XPUT without any ports is valid and is interpreted as a command to disconnect all ports.
- Example: XPUTI00010003I00020004i00100010001100l2

Results:

- R0003OK or R0007ERnnnn

System Information commands

XVERSION

Returns the API program version number.

Format: XVERSION

Results:

- R0009OKV4.0-3

XINSTALLED

Returns an inventory of installed I/O cards. A zero means the card is not installed and a one means the card is installed. The number of cards determines the number of zeros and ones returned and the number of cards varies for each VxRouter model.

Format: XINSTALLED

Example Results:

- (Vx40) R0019OK1111000011110000
- (Vx160) R0019OK1100000011000000
- (Vx320) R0043OK00000000000000000000000000000000100000000

XDCSTYPE

Returns a string containing the VxRouter model. This command is a (DCS) legacy command and the name reflects its usage as a DCS command though it is still valid for VxRouters.

Format: XDCSTYPE

Example Results:

- (Vx40) R0013OKVx40Router
- (Vx160) R0014OKVx160Router
- (Vx320) R0014OKVx320Router

XMAXPORTS

Returns the maximum number of non-blocking ports in the switch. This may also be interpreted as the maximum port number allowed in a command.

Format: XMAXPORTS

Example Results:

- (Vx40) R0007OK0080
- (Vx160) R0007OK0160
- (Vx320) R0007OK0320

XMAXCARDS

Returns the maximum number of either upstream or downstream cards in the switch. The total number of cards in the system is **twice** the value returned. The Vx160 has 8 upstream and 8 downstream cards for a total of 16 cards. To maintain consistency, the VxRouters that are uni-directional (the Vx40 and Vx320-video) also return half the number of cards.

	UpStream Cards	DownStream Cards	Total cards	Returned value
Vx40	16	0	16	8
Vx160	8	8	16	8
Vx320	20	20	40	20
Vx320-Video	20	0	20	10

The Vx160 example shows the message returned when the 'verbose' API command line option is enabled. In this case, the API command is returned at the end of the response message.

Format: XMAXCARDS

Example Results:

- `(Vx40)  R0007OK0008`
- `(Vx160)  R0017OK0008#XMAXCARDS`
- `(Vx320)  R0007OK0020`

XLASTEVENT

Returns a timestamp string that was set the last time a connection was made or broken. This can be used to determine if the switch status has changed since the last XLASTEVENT command was issued.

Format: XLASTEVENT

Example Results:

- `R0027OKThu Jul  1 11:23:52 2010`

XALARM

Returns the VxRouter hardware alarm status.

The return value is a decimal number that represents a bit-map of the actual alarm bits. For example, if 19 is returned, the binary format will be: 10011. This shows that 3 alarms are active (3 bits are '1'). The leftmost bit in the example is bit 4, followed by bits 3, 2, 1 and 0 (the rightmost bit). Bit 0 corresponds to alarm 1, bit 1 to alarm 2, etc.

The alarm bits are defined in the VxRouter manual and vary depending on the model.

Format: XALARM

Example Results:

- (Vx320) `R0007OK0522`

  Decimal 522 is `1000001010` in binary (bits 9, 3 and 1 are 'on')

XRESET

Resets the internal switch hardware to its power on state.

- Format: XRESET
- Results:
  - R00030K or R0007ERnnnn

XCRON and XCROFF

Enables or disables sending CR's on each line. XCRON is typically used when a (Windows) telnet client connecting to the vxRouter requires each line to end wth a CR/LF pair.

- Format: XCRON or XCROFF
- Results:
  - R00030K

XPORTCONFIG

Returns a comma delimited list of three numbers: NU, NB, OFF.

The first number (NU) is the maximum number of uni-directional paths in the system. The second (NB) is the maximum number of bi-directional paths and the third is currently defined as zero.

Format: XPORTCONFIG

Example Results:

- R00170K0080,0000,0000 Vx40
- R00170K0000,0160,0000 Vx160
- R00170K0000,0320,0000 Vx320

XQUIT

End the network connection. When making or breaking connections, it is important to identify the 'direction' of the connection. In this context, connection direction refers to data flow from the extender transmitter to the extender receiver, or receiver to transmitter. Transmitter to receiver data flow is called an 'upstream to downstream' connection and a receiver to transmitter data flow is a 'downstream to upstream' connection.

XHELP

Print a list of valid commands.

This is a multi-line response that does NOT start with OK or ER or end with a length. It is intended as a debugging aid and not for use in a production environment. It does not follow the format rules for command responses.

Sample response:

```
C,D,S                   - upstream to downstream or upper shelf path
c,d,s                   - downstream to upstream or lower shelf path

[C|c]IxxxxOyyyyO...     - Connect Input         CIxxxxO9999 broadcasts port xxxx
[D|d]Ixxxx...           - Disconnect Input(s)   DI9999 disconnects all
[D|d]Oyyyy...           - Disconnect Output(s)  DO9999 disconnects all
[S|s]Ixxxx              - return list of output ports
[S|s]Oyyyy              - return input port
XGET                    - return switch status of all the ports
XPUT[I|i]nnnnOnnnnO...  - restore switch status (from XGET)
XVERSION                - return software version
XINSTALLED              - return list of installed cards
XMAXCARDS               - return max. num of cards
XMAXPORTS               - return max. num of ports
XDCSTYPE                - return Vx320Router
XLASTEVENT              - return timestamp of when VxRouter status last changed
XPORTCONFIG             - returns a string with 3 numeric values: NU,NB,OFF
                NU  = number of uni-directional paths
                NB  = number of bi-directional paths
                OFF = is not used and returns 0
                A VxRouter contains 1*NU or 2*NB paths
XRESET                  - resets the switch hardware to the initial poweron state
XALARM                  - returns an alarm bit mask (0 means no alarms), bits are system dependant
XTEST1                  - diagnostic test 1 - connect 1 to 1, 2 to 2, ... n to n
XQUIT                   - end this session
XHELP                   - this text
```

VxRouter Port Numbering

When making or breaking connections, it is important to identify the 'direction' of the connection. In this context, connection direction refers to data flow from the extender transmitter to the extender receiver, or receiver to transmitter. Transmitter to receiver data flow is called an 'upstream to downstream' connection and a receiver to transmitter data flow is a 'downstream to upstream' connection.

The Vx40 is actually an 80 port switch. It can be thought of as having 80 upstream ports (similar in design to the upper half of the Vx320).

The Vx160 has I/O cards labeled Upstream and Downstream. The Vx320 has two cards shelves. The upper shelf holds the upstream ports and the lower holds the downstream ports.

The Vx160 only allows connections between upstream and downstream ports. You may not connect an upstream port to an upstream port or a downstream port to a downstream port.

The Vx320 only allows connections within a card shelf. You may not connect an upper shelf port to a lower shelf port.

The illustrations below show the port number assignments.

Vx40

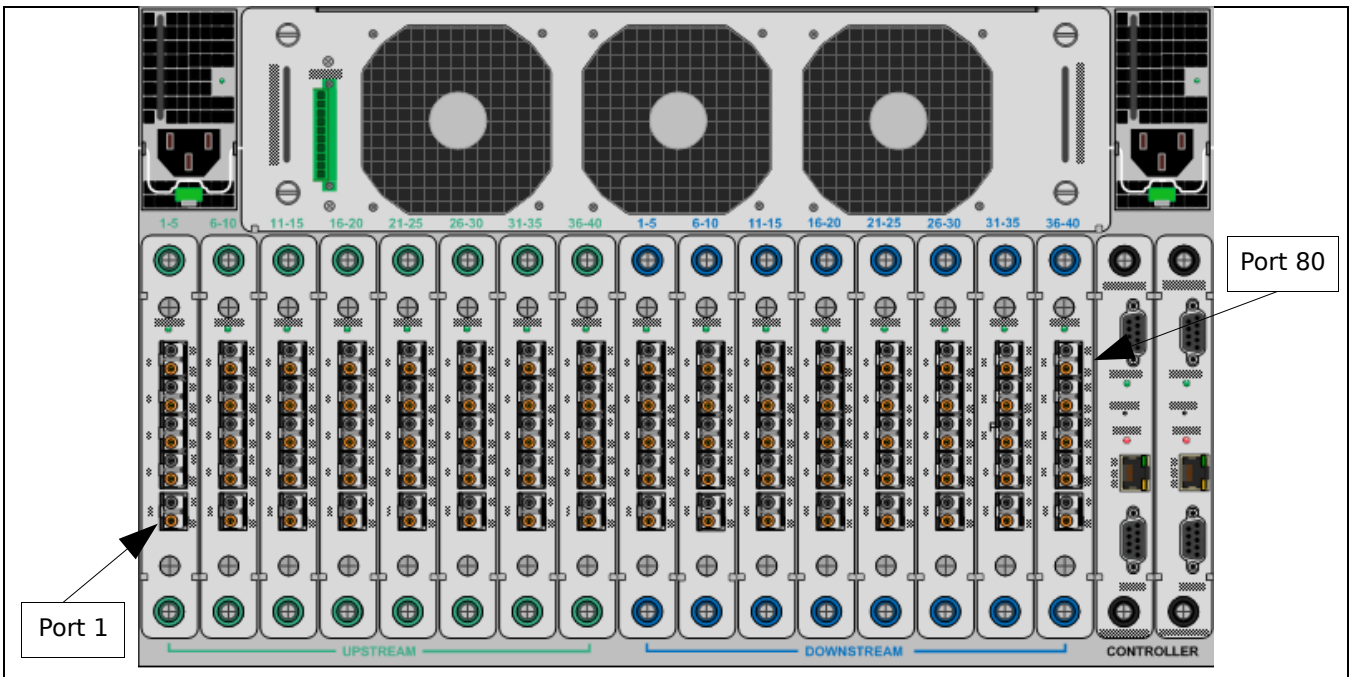
Connections may be made between any port on any card. The labels of 'upstream' or 'downstream' are not relevant. Vx40 ports are numbered from 1 to 80.

Vx160

Connections may only be made between upstream input ports and downstream output ports, or downstream input ports and upstream output ports.

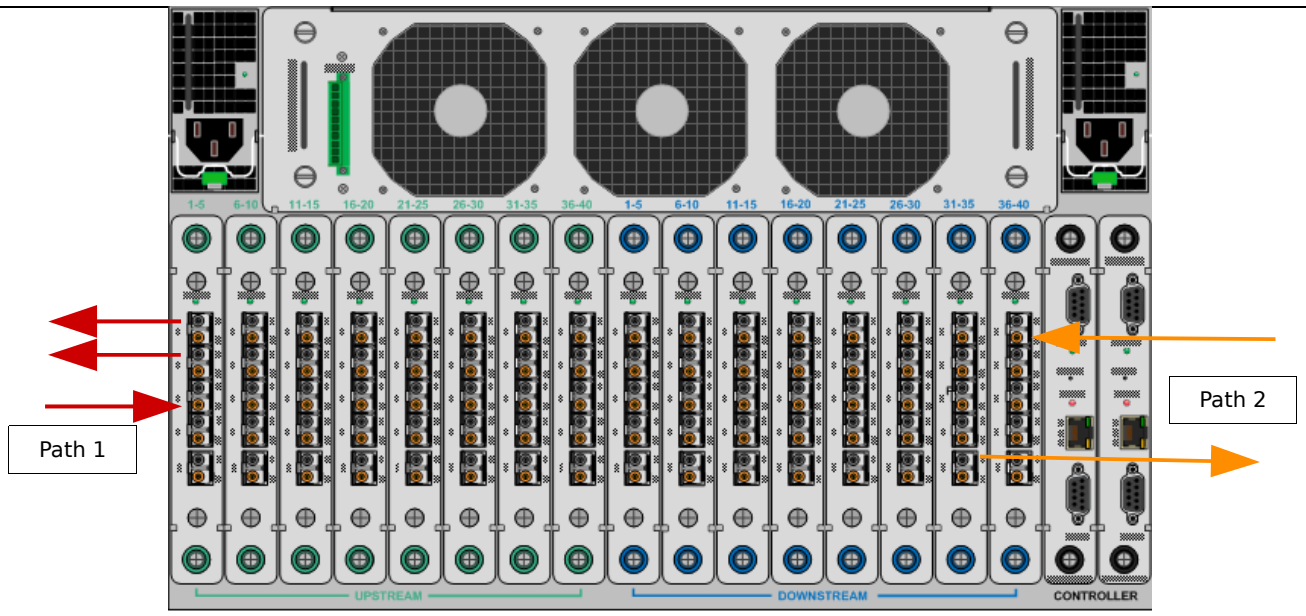
Vx320

Connections may only be made between input ports and output ports on either shelf. You may not make connections between shelves.



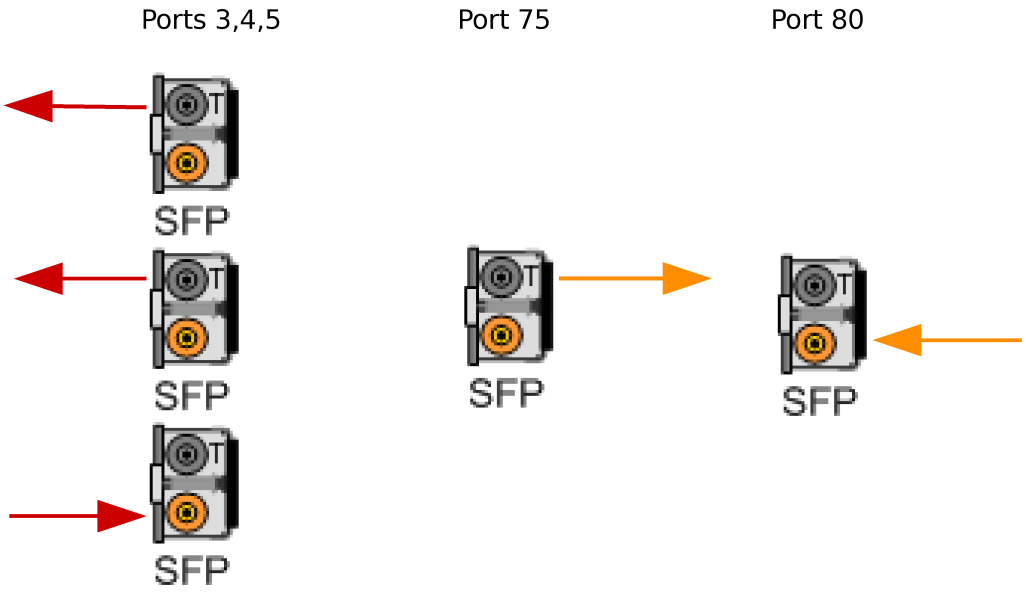
1 to 80
Uppercase Connect, Disconnect, Status are valid
(C,D,S)

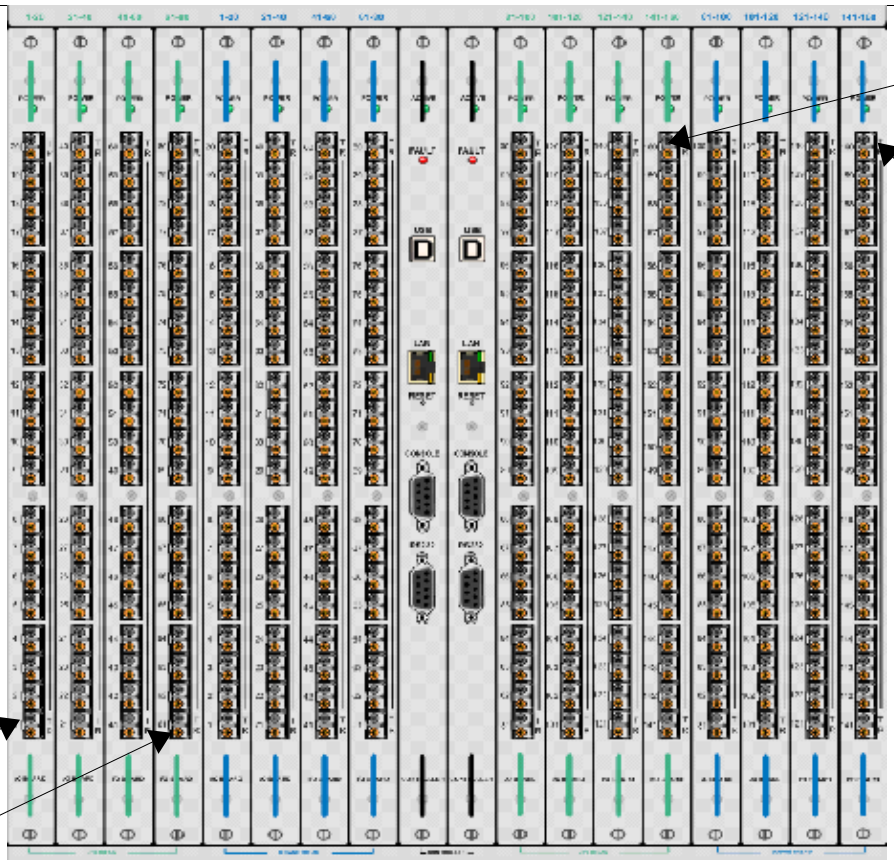
Illustration 1: Vx40 Port Numbering



1. `Ci0003o0004o0005` (Red) Connect input 3 to output 4 and 5
2. `Ci0080o0075` (Orange) Connect input 80 to output 75

Illustration 2: Vx40 Connections





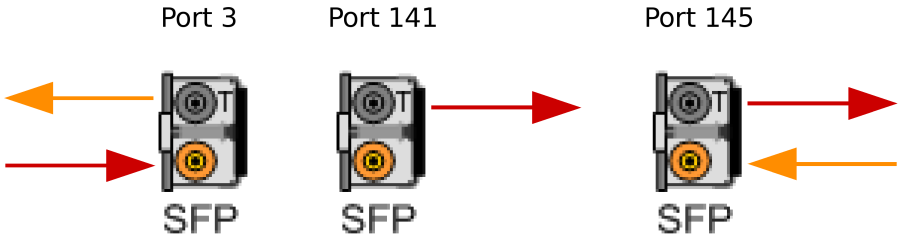
Upstream input to Downstream output use **UPPER** CASE commands

Downstream input to Upstream output use **lower** case commands

Illustration 3: Vx160 Port Numbering

1. `Ci0003o0141o0145` (Red) Connect input 3 to outputs 141 and 145
2. `ci0145o0003` (Orange)Connect input 145 to output 3

Illustration 4: Vx160 Connections



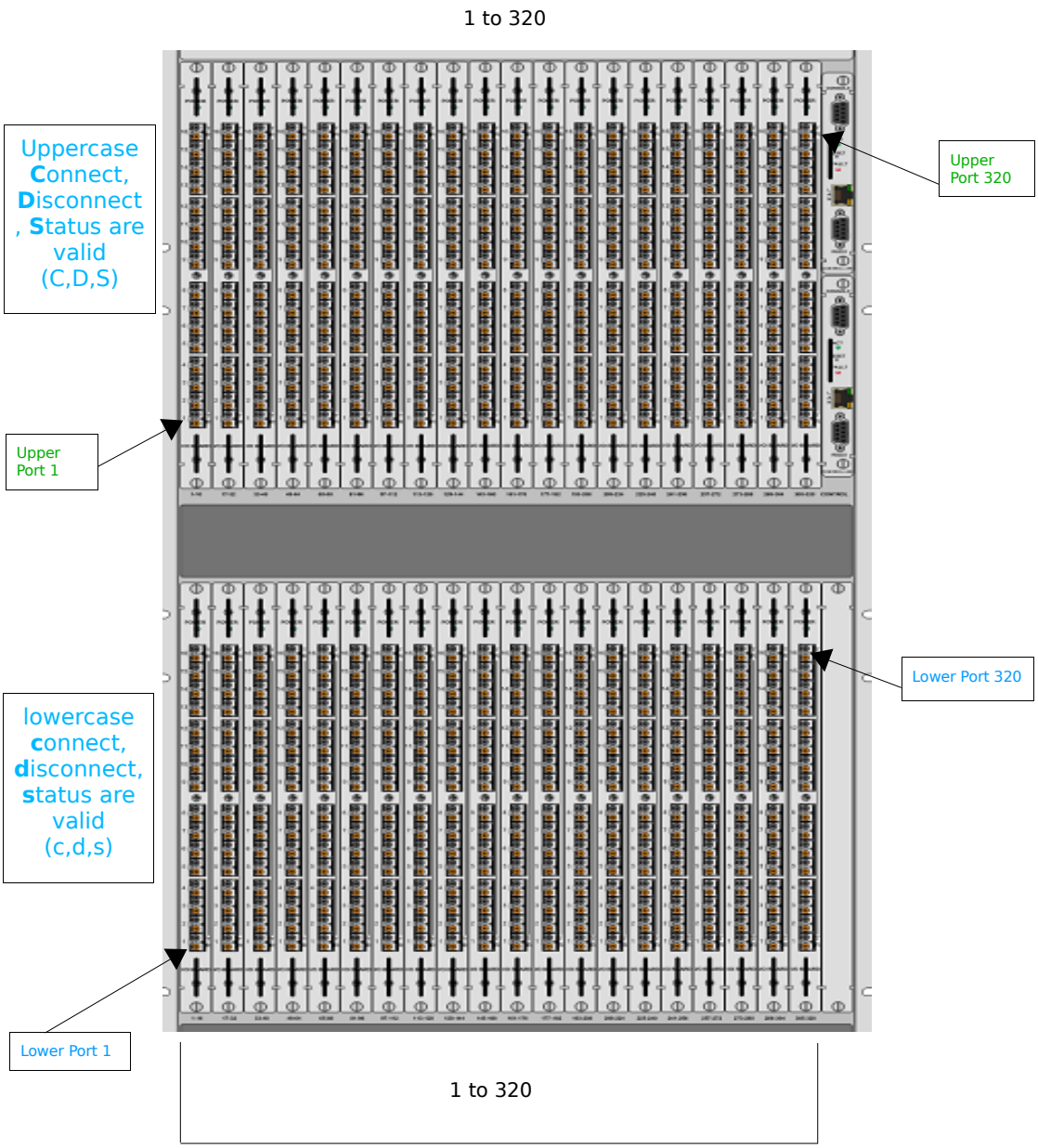
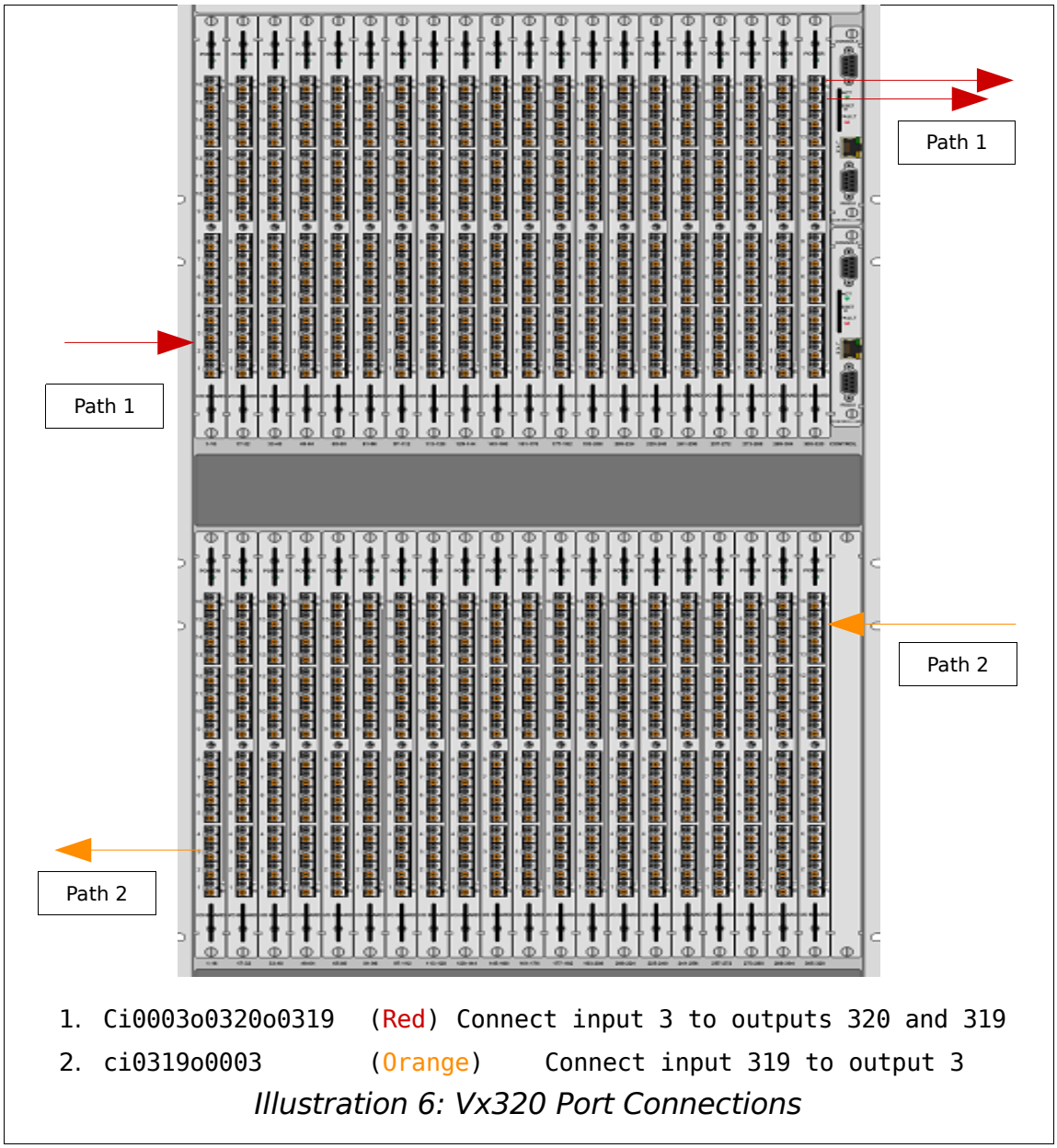
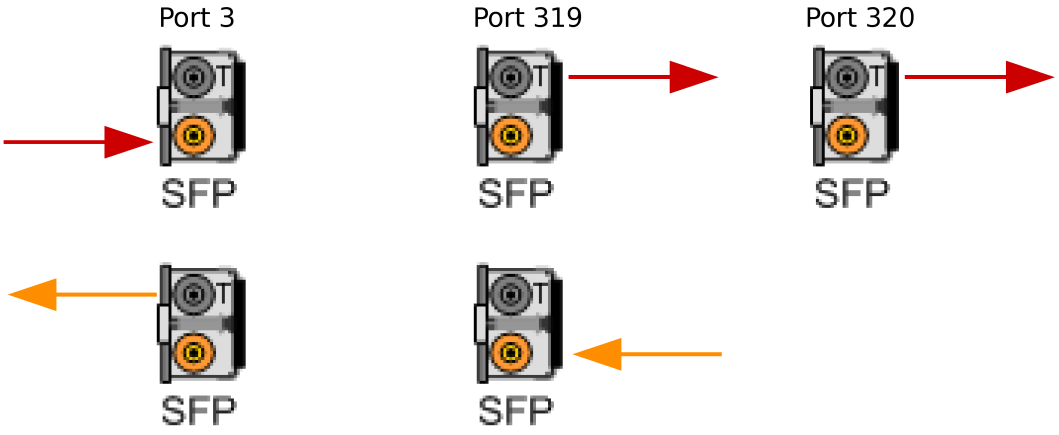


Illustration 5: Vx320 Port Numbering



1. `Ci0003o0320o0319` (Red) Connect input 3 to outputs 320 and 319
2. `ci0319o0003` (Orange) Connect input 319 to output 3

Illustration 6: Vx320 Port Connections



Error and Status Codes

1.1 **0001** – Syntax error.

1.2 **0002** – The command is missing an input port field.

1.3 **0003** – The command is missing an output port field.

1.4 **0004** – The command has multiple input fields, but only 1 is allowed.

1.5 **0005** – The command has multiple output fields, but only 1 is allowed.

1.6 **0006** – The input port value is out of range.

1.7 **0007** – The output port value is out of range.

1.8 **0008** – The command contains an invalid character.

1.9 **0009** – A port value of 9999 is not allowed in this command.

1.10 **0010** – A SNMP error occurred (only a DCS switch will generate this error).

1.11 **0011** – An error occurred while attempting an internal fork command.

1.12 **0012** – API received a command that is not allowed.

1.13 **0013** – Not enough memory to process the command

1.14 **0014** – File I/O error; an error occurred while reading or writing a file.

1.15 **0015** – The VxRouter control process is not responding.

Error 10:	This is only generated by a DCS switch.
Error 11, 13, 14, 15:	An internal program error: contact Thinklogical™ if you get this message.
Error 12:	API V4 does not use this error code.

Commands Removed from Version 4

The following commands were useful in the DCS switch, but are not appropriate for the VxRouter switches and have been eliminated in Version 4.

- XGOTOBLACK
- XFROMBLACK
- XFILESAVE
- XFILELOAD
- XFILEDIR
- XFILEDEL
- XPOWEROFF
- XSAVE
- XLOAD
- XMVKEYBD

Examples

`Ci0005o0010`LF	Connect Input 5 to Output 10
`ci0005o0010`LF	Connect Input 5 to Output 10
`Si0004`CR LF	Get the connection status of Input 4
`xputi00010003i00020004`LF	Turn off all outputs, connect Input 1 to Output 3, Input 2 to Output 4

Changing from V3 to V4 API commands

The basic V3 commands (connect, disconnect, status) still exist in V4 of the API. The V3 connect command automatically creates an active keyboard connection. This no longer happens in V4. The user must send the second connect command to create this connection. Because the user must explicitly create the active keyboard connection, the *XMVKYBD* command is no longer necessary. In its place, use the 'c' connect command.

The port numbers used below are for these examples only and they may not be valid for all VxRouter models. In the V4 API, the 'I' and 'O' in the commands may be either upper or lower case, **except** in the XPUT command. In XPUT, the case of the letter 'I' determines the direction of the connection. A lower case 'i' would correspond to a lower case 'c', and an upper case 'I' to an upper case 'C'.

Command(s)	V3	V4
connect video input 7 to output 12	`Ci0007o0012`	`Ci0007o0012`
connect data input 9 to output 15 and the active keyboard is on port 15	`Ci0009o0015`	`Ci0009o0015` `ci0015o0009`
broadcast video input 66	`Bi0066`	`Ci0066o9999`
broadcast data input 66 and have the active keyboard on port 13	`Bi0066o0013`	`Ci0066o9999` `ci0013o0066`
disconnect output 54	`Do0054`	`Do0054`
disconnect input 18	`Di0018`	`Di0018`
save switch settings	`XSAVE`	`xget`
load switch settings	`XLOAD.......`	`xput.......`

connect data input 3 to data outputs 4 & 5 (active keyboard is at output 4)
 	
then move the keyboard to output 5 (from the example on page 6)*
	`Ci0003o0004o0005`
	
	`XMVKYBDI0005O0003`
		`CI0003o0004o0005`
		`ci0004o0003`
		
		`di0004`
		`di0005`
		`ci0005o0003`

connect input 54 to outputs 1,2,3,4	`Ci0054o0001o0002o0003o0004`	`Ci0054o0001o0002o0003o0004`
disconnect all outputs	`Do9999`	`do9999` *(breaks lower paths)* `Do9999` *(breaks upper paths)* *or you may use* `xput`

* the 'di' commands will break any back-channel connections that might be in place from the keyboards at `0004` and `0005`. This is done to prevent the accidental connection of a keyboard to more than one CPU.

API Manual Revision History

- 4.0-0 Initial release
- 4.0-1 added the XCRON and XCROFF commands
- 4.0-2 added examples of verbose responses to commands