



## SIERRA VIDEO

### **Routing Switcher Communication Protocol**

Details of the Generic Communication Protocol  
Common to most Sierra Video Routing Switchers.

#### **Introduction**

The protocol uses the 7-bit ASCII character set, usually sent over an RS232-C or RS422 serial link. Protocol commands can also be sent over the RJ-45 Ethernet connection.

Please refer to your routing switcher manual's Control via 9-pin Connectors or Ethernet Setup sections for proper configuration with the protocol.

#### **Generic Protocol**

Commands are sent to a routing switcher in a group called a *command string*. A command string can contain zero or more commands, limited only by the size of the receive buffer of the router, whose size depends on the particular router model.

A *command string* consists of a leader string of asterisk characters, zero or more commands, and a trailer string of exclamation marks. Larger routers require two leader (\*\*) and trailer (!!) characters, while small routers require only one, in order to make the protocol compact for those routers. The remainder of this document gives examples using doubled characters. Note that two leader/trailer characters may be sent to small routers even when only one is required, and they will still work fine.

If a leader character (\*\*) is encountered within a command string being processed by a router, the string up to that point is discarded and a new command string is expected. This ensures that a router will always act on a complete command string sent to it, even if the previous one was never completely received.

When a command string is received, it is not acted upon (but rather, is merely buffered up) until the final trailer character (!) character of the command string is received. At that time, the routing switcher begins to execute the commands within the string.

The protocol uses only 7-bit ASCII characters. The 8th bit of received characters is treated as if it is 0. Within the command string, certain ASCII characters may be present and are ignored: any ASCII character whose code is *less than* the SPACE character (includes all control characters and the SPACE character) and the DEL (ASCII 7F) character. Alphabetic characters within the command string may be in either upper-case or lower-case letters. The router always *sends* upper case characters, except for character strings such as input, output, and level names, which may have lower case characters in them.

When sending commands *to* the router, SPACE characters are optional, but if used may only appear before and after each individual command and NOT embedded within an individual command. Within command strings sent *from* the router, a single SPACE character appears before and after each individual command. SPACES may also appear in character strings, such as input, output, and level names.

Certain commands (R, Q, L, and G) have character strings that appear as arguments. The first three, "R", "Q", and "L", have character strings only in commands sent *from* the router, and these character strings are always terminated with a ~ (tilde) character. No special character marks the start of these strings, they simply begin at the appropriate point within the command. The "G" command, on the other hand, uses the ~ (tilde) character to mark both the *start and end* of a character string argument.

Just before the router begins executing a command string, it sends a string of leader characters (asterisks) to the host (the same number as are required in commands from the host). As it executes the commands, some of them may generate additional output back to the host. These command response characters are always preceded and followed by a space character, making the response string easily human-readable.

After the command string has been executed, the routing switcher returns the string " **OK** " (with a single space character before and after the word "OK"), followed by a string of trailer characters (exclamation marks) and a CR (carriage return, ASCII 0D) character, to the host. This indicates that the command has executed successfully. If an error occurs within any command

of a command string, the remainder of the command string is ignored and the router returns the string " **ERROR** ", followed by *an optional descriptive string* followed by a string of trailer characters and a CR character, to the host. An error can be caused by an unknown command name or bad arguments to a command.

The simplest possible command string would be:

```
** !!
```

which consists of the leader and trailer characters but no commands between them. This command string would generate the response:

```
** OK !!
```

This can be useful for verifying that the serial link to the router is operational. In routers requiring only one leader/trailer character, the simplest command string would be:

```
* !
```

which would generate the response:

```
* OK !
```

(To determine whether a particular router uses one or two leader/trailer characters, send it "!!\*\*" and check the response to see which of the above two responses it is. It won't hurt to always use two even if only one is required.)

The simplest error response is one with no optional descriptive string. For example, this command string:

```
** XXX !!
```

might generate this response from the router:

```
** ERROR !!
```

Beginning with version 5.01 of the Tahoe/Sierra/Yosemite router software, a descriptive text string was added following the word "ERROR", to help with diagnosing the error. For example, the above command string might generate this response from a router running this newer software:

\*\* ERROR Syntax:No Number:XX !!

The descriptive string always ends with a colon and up to three characters from the command string that caused the error. Generally, the error can be assumed to have occurred just before these characters.

## Commands to Router

Below is the complete set of commands that can be sent TO a routing switcher, a subset of which may be implemented in any given routing switcher. See command argument key below.

All routing switchers implement the **I, L, S, Q, and X** commands.

Syntax	Example	Response Command	What it Does
I	**I!!	I	Command capabilities.
Q	**Q!!	Q	Model name and version.
L	**L!!	L	Matrix size and level names.
S	**S!!	V, Y, or X commands	Matrix status.
O out	**O3!!	V, Y, or X commands	Output status.
N in	**N2!!	V, Y, or X commands	Input status.
RESET	**RESET!!	RESET	Reset the router.
RELOAD	**RELOAD!!	(none)	Reload the crosspoints.
CLEAR	**CLEAR!!	V, Y, or X commands	Set all outputs to input 1.
U { 0   1   2 }	**U0!!	(none)	Automatic output change reports on/off.
V out,in,in...	**V3,1,2,2!!	V, Y, or X commands	Connect specified inputs on each level.

W lvl,in,in...	**W1,4,19!!	V, Y, or X commands	Connect specified inputs to outputs 1...
X out,in,lvl	**X12,9,2!!	V, Y, or X commands	Connect specified input to output on level.
Y out,in	**Y1,7!!	V, Y, or X commands	Connect specified input to output on all levels.
Z in,in...	**Z13,12,8!!	V, Y, or X commands	Connect specified inputs to output 1.. on all levels.
B out,password,lock[,lvlmask]	**B17,8245,1!!	B	Lock or unlock an output.
F delayfields	**F2!!	(none)	Specify crosspoint output delay.
D numsyncs	**D30!!	(none)	Delay vertical sync intervals.
P reg connect connect ... ~	**PB Y1,7 Y2,3 ~!!	(none)	Preset a salvo connect sequence.
T reg	**TB!!	V, Y, or X commands	Trigger a salvo connect sequence.
HOST { 0   1 }	**HOST0!!	(RESET or terminal screen)	Switch port control mode to either generic serial protocol mode (1) or a user-friendly terminal mode (0).

M addr	**M203!!	M	Set remote address of sending device.
K password	**K1987!!	(none)	Set password for changing protected router settings.
G command [ arguments ] ~	**G LEVEL_NAME,4,Audio~!!	(various)	Query or set configuration data.
A{a_cmd_letter}			Extends command set by adding 26 two-letter commands.
AI	**AI!!	AI	Extended command capabilities.
AR in,password,secure[,lvlmask]	**AR27,294,1!!	AR	Secure or unsecure an input.

## Commands from Router

Below is the complete set of commands that can be sent *by* a routing switcher in response to receiving the above commands or due to other events. See command argument key below.

Either ERROR or OK is always sent as a response to every command string, appearing at the end of the response string.

Each command is described in detail in the later sections of this document.

Command	Syntax	Example	Response To	What Information It Provides
RESET	RESET string~	** RESET SVS Series 32 V1.00 (C) 1990~ OK !!	RESET	Router has finished resetting.
Q	Qmodel~version~	** QSeries 32~V1.00~ OK !!	Q	Model name and version.
I	I[A][B][C][D][F][G][H][K][L] [M][N][O][P][Q][R][S][T] [U][V][W][X][Y][Z]~	** IQLSCUX~ OK !!	I	Supported commands.

L	L[O[(#,#...)]][S[(#,#...)]] [P[(#[=#][M][L],...)] Nout,Nlv1,Nin, lv11~lv12~...lv1N~~	** LO(2)16,2,64, Video~Audio~~ OK L !!		Router size and attributes, level names.
V	V out,in,in...	** V3,1,2,2 !!	(changed crosspoint)	Reports inputs connected to output on each level.
X	X out,in,lv1	** X12,9,2 !!	(changed crosspoint)	Reports input connected to output on level.
Y	Y out,in	** Y1,7 !!	(changed crosspoint)	Reports input connected to output on all levels.
B	B out,password,lock[,lv1mask]	** B17,8245,1 OK !!	B or change report	Output lock status request or update as a result of a changed output lock.
M	M addr	** M203 OK !!	M	Current remote address.
G	G command [ arguments ] ~	** G LEVEL_NAME,4,0, ~Audio~~ !!	G or change report	Configuration query/set.
AI	AI[R]~	** AIR~ OK !!	AI	Supported extended commands.
AR	AR in,password,secure[,lv1mask]	** AR27,294,1 OK !!	AR or change report	Input security status request or update as a result of a changed input security.
OK	OK	** OK !!	(good command string)	Finished executing good command string.
ERROR	ERROR	** ERROR !!	(bad command string)	Error encountered while executing command string.
FULL	FULL	** FULL !!	P	Too many connects in a P command .
BUSY	BUSY	** BUSY !!	P	Someone else is currently changing the salvo, "P" command ignored.

## Command Arguments

Here is a description of the command arguments.

Arg Name	Description
in	Input number, numbering from 1, 0 for no-change, unknown or non-existent, '-' for disconnect.
out	Output number, numbering from 1, 0 for unknown or non-existent, '-' for disconnect.
lvl	Level number, numbering from 1, 0 for all-levels, i.e. AFV.
lvlmask	Level mask, 1 to 4 hexadecimal digits, 0 to FFFF, with bit 0 (least significant bit) being for level 1, etc, 0 or FFFF for all-levels.
numsyncs	Number of vertical sync intervals (i.e. video fields) to delay command processing.
delayfields	Number of video fields of delay from time command is received before crosspoint is switched.
reg	Salvo register, a letter from A to Z or a number from 1 to 256.
connect	A "V", "W", "X", "Y", or "Z" connect command.
Nout, Nin, Nlvl	Number of outputs, levels, and inputs, respectively.
lvl1,...lvlN	Level names.
command	A "G" command sub-command name.
arguments	Various "G" command arguments.
string	A character string that identifies a router's name and software version number.
model	A character string that identifies a router's model.
version	A character string that identifies a router's software version number.
addr	Remote address, generally in the range 136-255.
password	Router destination password or protected settings override password, a number from 0 to 9999.
lock	A value of 1 for output lock, or 0 for output unlock.
secure	A value of 1 for input secure, or 0 for input unsecure.

### Command Details "RESET": Routing Switcher Reset

From	Syntax	Description
Host	RESET	Ask router to reset.
Router	RESET string~	Router is now ready to receive commands.

When a routing switcher first powers up, either after power is applied or after a reset of the processor occurs, it sends a command string response containing the "RESET" command to the host. For example:

```
** RESET SVS Tahoe Vx.xx (C) 2000~ !!
```

This informs the host that the router has been reset, and gives the router name and software version number. A tilde character terminates the character string. Update mode, if implemented, is now ON (or in the initial state set by a DIP switch). "x.xx" is the software revision number, e.g. "V1.00".

It is also possible to force the routing switcher to reset, by sending it the command "**RESET**". If an administrator password has been set (using the "G ADMIN\_PASSWORD" command described in another document), then the password must be sent in a "K" message prior to sending this command, else this command will generate an error.

This command can take several seconds to execute. During the reset process, the router performs diagnostic tests of its EPROM and RAM memory, whose results can be obtained with the "G" command. In order to help ensure that this command isn't accidentally executed, it requires four additional characters following the "R" character, to spell out the word "RESET" in full. Note: the reset command was added to the protocol in Aug. 2000, and is not supported in router software versions released before that date.

For example, the command:

```
**RESET!!
```

would initiate a reset (if the password has been entered, if required), and when the router finished the reset operation, something like this would be received, just as if the router had powered up:

```
** RESET SVS Tahoe Vx.xx (C) 2000~ !!
```

### "RELOAD": Routing Switcher Reload

From	Syntax	Description
Host	RELOAD	Ask router to reload the crosspoint connections into the matrix hardware from NVRAM.
Router	(None)	

Larger routers can have the crosspoint hardware located in a different frame from the processor. In the case where these boards are not powered down or reset at the same time, this command will restore the crosspoint matrix.

For example, the command:

```
**RELOAD!!
```

would immediately force the router to output all crosspoint data to the crosspoint hardware.

## "I": Capabilities Inquiry

From	Syntax	Description
Host	I	Ask router to send a list of commands that are available.
Router	I[B][C][D][F][G][H][K][L][M][N][O][P][Q][R][S][T][U][V][W][X][Y][Z]~	Available commands. (Stuff inside [] is optional depending on router)

The command "I" requests that command capability information be returned to the host. The information is sent as a string of characters. The first characters are a space followed by "I", the next characters are the letters of the commands that are implemented and available in this router, and the last character is "~" (tilde). Do not count on the characters being in any specific order. Search all characters for a particular one.

All routing switchers implement the **I, L, S, Q, and X** commands.

For example, the command:

```
**I!!
```

might return the following string:

```
** ILSX~ OK !!
```

indicating that the router supports the **I, L, S, and X** commands from the host.

## "Q": Model Name and Software Version Inquiry

From	Syntax	Description
Host	Q	Ask router to send the model name string and software version number string.
Router	Qmodel~version~	Model name string and software version number string.

The command "Q" requests that the router model name and software version number string be returned to the host. The information is sent as a string of characters. The first characters are a space followed by "Q", the next characters are the router model name, terminated by a "~" (tilde). Following this are the characters of the software version number string, again terminated by a "~" (tilde).

For example, the command:

```
**Q!!
```

might return the following string:

```
** QSmall~V2.1~ OK !!
```

indicating that the router model name is "Small" and the software version number is "V2.1".

### "L": Matrix Size and Level Names Inquiry

From	Syntax	Description
Host	L	Ask router to send router matrix size, level names, and level information.
Router	L[O[(#,#...)]][S[(#,#...)]][P[(#[=#][M][L],...)]Nout,Nlv1,Nin,lv11~lv12~...lv1N ~~	Matrix size, level names, and level information.

The command "L" requests that matrix size (Nout, Nlv1, Nin) and level name information (lv11, lv12, etc.) be returned to the host. The information is sent as a string of characters. The first characters are a space followed by "L", some optional values described below, then the number of outputs (Nout), a comma, the number of levels (Nlv1), a comma, the number of inputs (Nin), a comma, and then the level names, each terminated by a "~" (tilde), and the last followed by two tildes.

Routers can provide names for inputs, outputs, and levels. The number of characters in a name can vary depending on the router model. The "L" command provides access only to *level names*, and furthermore, it truncates any level name that is longer than 6 characters to only 6 characters, in order to retain compatibility with older routers and controlling devices that limited level names to 6 characters. Refer to the "G" command for accessing input and output names and full-length untruncated level names.

The number of level names in the "L" command will be the same as the number of levels that was given in the response (Nlvl). Valid characters for level names are any printable (i.e. non-control) ASCII character (including SPACE) except "\*" (asterisk), "~" (tilde), and "!" (exclamation mark). The number of characters reported by the "L" command in level names may be no more than 6 (longer names are truncated when reporting them with "L"). In systems that do not support naming of levels, the level names will be fixed number strings, e.g. "1", "2", etc.

For example, the command:

```
**L!!
```

might return the following string:

```
** L64,3,32,VIDEO~AudioL~AudioR~~ OK !!
```

indicating that the router has 64 outputs, 3 levels, and 32 inputs, and the levels are named "VIDEO", "AudioL", and "AudioR".

Some routers can have different sized matrices on each level. However, the "L" command always reports what is referred to as the "basic router size". This is the size of the largest levels in the router. For example, a 2-level router with level 1 being 16x32 and level 2 being 32x16 would report a basic router size of 32x32 in the "L" command. In these kinds of routers, it is not considered an error to send a crosspoint command that specifies an all-levels *take* using an input or output number that is beyond the range of some of the levels, as long as those numbers lie within the basic router size. Levels which are smaller than the specified numbers will not be affected by such a *take* command. On the other hand, it *is an error* to send a crosspoint command that specifies a *take on a specific level* with an input or output number that is beyond the range of that level. To find out the actual physical size of each level, use the "G LEVEL\_INFO" command. Routers that do not support the "G" command or "G

LEVEL\_INFO" subcommand always have the same-size crosspoint matrix on each level.

Some routers support virtual-to-physical mapping. These virtual-mapped routers use the same virtual matrix size for each level, and it will be that size that is reported in the "L" command for Nout and Nin. The physical crosspoint matrices can still be different sizes, and the "G LEVEL\_INFO" command can be used to find out what these sizes are, but they are of less importance in a virtual-mapped router, because *take* commands are specified using virtual source and destination numbers rather than physical input and output numbers.

### **'O' Option: Open-Connection Outputs**

If the letter "O" (oh) follows "L", it indicates that each output (or port) may also be an open connection, i.e. connected to NO input, by specifying an input number of '-' (dash). If the letter "O" *does not* follow "L", the router does not allow inputs to be open, they must be connected to some output. As a command example, for a router allowing open connections, the above response string would look like this:

```
** L064,3,32,VIDEO~AudioL~AudioR~~ OK !!
```

Furthermore, the letter "O" may be followed by one or more comma-separated numbers enclosed in parentheses, to specify that only those levels allow an open connection. For example:

```
** LO(2,3)64,3,32,VIDEO~AudioL~AudioR~~ OK !!
```

This specifies that levels 2 and 3 allow open connections but level 1 does not.

### **'S' Option: Single-Connection Outputs**

If the letter "S" follows "L", it indicates that the router allows each input (or port, see below) to be connected to no more than a single output (otherwise, an input can go to any number of outputs). Such a router will automatically disconnect existing connections if necessary in order to make a newly requested connection. As a command example, if the router allows inputs to go to at most one output, the above response string would look like this:

```
** LOS64,3,32,VIDEO~AudioL~AudioR~~ OK !!
```

Furthermore, the letter "S" may be followed by one or more comma-separated numbers enclosed in parentheses, to specify that only those levels require an input to be connected to no more than a single output. For example:

```
** LO(2,3)S(2,3)64,3,32,VIDEO~AudioL~AudioR~~ OK !!
```

This specifies that levels 2 and 3 inputs can only be disconnected or connected to one output, while level 1 inputs cannot be disconnected but can be connected to any number of outputs.

### **'P' Option: Port-Oriented Routing**

If the letter "P" follows "L", it indicates that the router is a *port unit*, i.e. rather than having inputs and outputs, it has ports which may be connected to each other. A port unit still uses input and output numbers in the connect commands, but these numbers refer (either directly or indirectly via mapping tables) to ports rather than inputs and outputs. In some routers, input and output numbers are mapped to ports, while in others (1980's-vintage port routers) the input and output numbers are directly interpreted as port numbers. Users of the older-vintage port units should substitute the word "port" for "input" and "output" when reading this document.

In the older-vintage routers, it is unimportant which port number of a pair to be connected is specified in place of the output number, and which is specified in place of the input number. In newer routers, it does matter: the router hardware automatically configures the port connector polarity depending on whether the port is connected an input (or source) number or an output (or destination) number. In these routers, when a port is specified via an input or source number, the port polarity is configured as if the port is a VTR IN PLAY MODE, and if a port is specified via an output or destination number, the port polarity is configured as if the port is a VTR IN RECORD MODE.

Note that while a matrix unit allows input N to be connected to output N, port units do not allow port N to be connected to port N, because this would imply a connection of a port to itself.

As a command example, for an older-vintage port unit router the above response string would look like this:

```
** LO SP64,3,64,VIDEO~AudioL~AudioR~~ OK !!
```

Note that the number of inputs and outputs is equal, which is the case in pure port routers, since the input and output numbers actually refer to ports.

The letter "P" may be followed by one or more comma-separated sequences, the entire group enclosed in a pair of parentheses:

```
P(#[=#][M][L],...)
```

The first '#' is a level number, indicating that that level is a *port* level (some routers may be non-port routers on some levels, and port-routers on other levels). This may optionally be followed by an '=' and another number giving the number of ports present on this level. If this second number isn't present, then the number of ports can be assumed to be equal to the number of inputs or number of outputs, which should both be equal. The following sections describe the meaning of the letters "M" and "L".

### **'M' Port-Option: Mapped Ports**

An optional letter 'M' may occur within the optional 'P(#...)' sequence. It indicates that the port level is *mapped* to inputs and outputs. A mapped port level has a table that the user defines that assigns zero or one port to each router input and output. In a mapped port level, input and output numbers are still used as usual; they *do not* refer to *ports* as they do in a pure unmapped port level. When a connection is requested between a particular input and output, the mapping table is consulted to determine the corresponding ports, which are then connected. If no port is mapped to the input and/or output, no connection is made and no warning is produced. (If the connection request is part of an all-level (AFV) connect command, connections on the remaining levels are still made.) As with unmapped port levels, it is illegal to connect a port to itself, so the input and output number of a connect command may not be mapped to the same port.

Port levels may or may not be mapped port levels in a non-virtual-mapped router. In a virtual-mapped router, port levels are *always mapped port levels*. In virtual-mapped routers, the virtual-to-physical mapping tables also serve as the port mapping tables. On non-port levels of a virtual-mapped router, the virtual mapping tables map a virtual source or destination to a physical input or output connector, while on port levels they map a virtual source or destination to a *physical port connector*. Therefore, virtual-mapped routers will always include the "M" option within the "L" command response, for all port level(s).

## 'L' Port-Option: Port-Level Lockouts

An optional letter 'L' may occur within the optional 'P' (#...) sequence. It indicates that the port level has user-configurable lockouts. For non-mapped port levels, lockouts consist of a table that, for each *port*, gives a list of other *ports* that that port may be connected to. For mapped port levels, lockouts consist of a table that, for each *output or destination*, gives a list of *inputs or sources* that it may be connected to. When a connection is requested on a port level that has lockouts, the lockout table is consulted. If the connection is disallowed by the lockout table, no connection is made and no warning is produced.

Port levels may or may not have port-level lockouts in a non-virtual-mapped router. In a virtual-mapped router, lockouts of an input from an output apply to all levels, port and non-port. Virtual-mapped routers will *not* include the "L" option within the "L" command response.

### 7.4.3.3 Example of 'P' Option with "(#...)" Sequence

Here's an example of a complex "L" command with multiple port routing levels:

```
** LO(2,3)S(2,3)P(2=16L,3=128ML)64,3,32,VIDEO~AudioL~AudioR~~ OK
!!
```

This indicates that the router has 64 inputs, 3 levels, and 32 outputs. Levels 2 and 3 are port levels that allow open connections and only a single connection to each port. Port 2 is a pure port level with 16 ports and it includes a lockout table (2=16L). Port 3 is a mapped port level with 128 ports and a lockout table (3=128ML). The level names are "VIDEO", "AudioL", and "AudioR".

## "O": Output Status Inquiry

From	Syntax	Description
Host	O out	Ask router to send source status for all levels of specified output.
Router	Y out,in or V out,in,in... or X out,in,lvl	Source status of the specified output.

The command "O" requests that matrix status information for a single output be returned to the host. The status information is sent as a "Y" command or a "V" command or as a sequence of L "X" commands, where L=number of levels. Whether a "Y", "V", or "X" command is sent depends on several factors:

- Very old routers only supported the "X" command.
- Routers with pre-version-10 software supported all three commands, "Y", "V", and "X", but usually provided a DIP switch that could be set to force the router to only send "X" commands.
- Routers with version 10 or later software use the setting of the "U" command (described below) to determine whether to send only "X" commands, or "Y", "V", and "X" commands.

A 6-level router using only "X" commands could generate as many as 6 "X" commands of status output. The first command is for level 1, the next for level 2, etc. until the last level is reached. (However, each "X" command contains the level number, so the sequence in which the data is sent is not really important.)

When a router is able to send "Y", "V", and "X" commands, a "Y" command will typically be sent if the router has only one level, and may also be sent if the output has all of its levels connected the same, although this is not required and some routers may instead send a "V" command or a series of "X" commands in that situation. A "V" command will be sent when an output has different connections on different levels, which is a more compact representation of the status than a series of "X" commands.

The length of a "Y" or "V" or "X" command depends on the size of the particular router involved. Larger routers use longer numbers for inputs, outputs, and levels.

A space character precedes each "Y" or "V" or "X" command (spaces should be ignored by the command parser, however). Each command begins with the command letter, a **Y** or **V** or **X**, followed by the output number and a comma character. Following that, the "Y" command has the input number connected to the output, the "V" command has L input numbers separated by commas (L=number of levels) in order by level number, and the "X" command has the level number at which the input-output pair is connected.

The input number may be a dash ('-') if no connection exists (for routers that are capable of having their inputs disconnected). The input number may be **0** if the output doesn't exist at that level in the router (or isn't mapped to a physical output, in virtual-mapped routers), or if the connection is not known by the controller, as may be the case for some control panels immediately after they are powered up.

For example, the command:

\*\*O5!!

to a 3-level router might have the following three commands as its Response:

```
** X65,23,1 X5,-,2 X5,0,3 !!
```

Note the dash, indicating that on level 2, output 65 is not connected to an input. Also note the 0, indicating that the connection on level 3 is either unknown or that output 65 doesn't exist or isn't mapped on level 3.

Or, a 3-level router might have the following single command as its Response:

```
** V65,23,-,0 !!
```

which has the same information as the three X commands in the previous example.

If the router has only one level, or if all levels are connected the same, the router might instead use the Ycommand. For example:

```
** Y65,23 !!
```

The number of digits used for each number depends on the router. Newer routers use the minimum number of digits necessary, i.e. there are no unnecessary leading zeroes in a number. Older routers use the maximum number of digits ever required for an input or output number on that particular router (e.g. a router with between 10 and 99 inputs would use two digits for the input number).

### "N": Input Status Inquiry

From	Syntax	Description
Host	N in	Ask router to send status of outputs connected on one or more levels to the specified input.
Router	Y out,in or X out,in,lvl	Status of all outputs connected on one or more levels to the specified input.

The command "N" requests that matrix status information for a single input be returned to the host. The status information is sent as one or more "Y" and/or "X" commands. This command works by identifying all outputs that connect to the specified input on any level, and then generating "Y" and "X" status commands to report the status of those outputs. The "Y" and "X" commands are formatted exactly as with the "O" command. For any given output connected to the input, a "Y" command is used if the output is connected to

the input on all levels, else "X" commands are used on all levels on which the input is connected to that output.

Prior to version 10 software, this command was only useful on those router levels that allow an input to be connected to at most one output, and status commands were only sent for such levels. Beginning with version 10 software, a response is generated for all outputs connected to the specified input on any level, regardless of how many outputs that might be, and regardless of whether the level allows only one output connection or many output connections.

For example, the command:

```
**N4!!
```

to a router might have the following commands as its Response:

```
** X12,4,1 X12,4,2 X12,4,3 X13,4,4 Y23,4 !!
```

showing that input 4 connects to outputs (12,12,12,13) on levels 1-4, and to output 23 on levels 1-4.

Note that a "V" command is never used in the response, unlike an "O" request described above. This is because the "V" command specifies inputs connected to an output, not outputs connected to an input.

### "S": Status Inquiry

From	Syntax	Description
Host	S	Ask router to send source status of all router outputs.
Router	Y out,in or V out,in,in... or X out,in,lvl	Status of all outputs connected on one or more levels to the specified input.

The command "S" requests that matrix status information be returned to the host. The status information is sent as a sequence of "Y" and/or "V" and/or "X" commands. The order of the commands is not significant, and different models of routers may send the status of their outputs in different orders. Each command string contains the output number and either a level number or data for all levels, so the order in which the status data is sent is not really important.

For example, the SVS *Tahoe Series* routers first send one or more commands that give the status for all levels of output 1, then send commands giving

status for all levels of output 2, etc. Other routers may send commands in a different order: first for all outputs of level 1, then for all outputs of level 2, etc.

The "Y", "V", and "X" commands are formatted exactly as with the "O" command.

Refer to the "O" command description for information about whether the router sends status using only "X" commands, or "Y", "V", and "X" commands.

A 2-level 16-output router could generate as many as  $2 \times 16 = 32$  "X" commands of status output.

For example, the command:

```
**S!!
```

might have the following four "X" commands at the beginning of its Response:

```
** x1,23,1 x1,3,2 x2,-,1 x2,0,2
```

This indicates that output 1 is connected to input 23 on level 1, output 1 is connected to input 3 on level 2, output 2 is *unconnected* on level 1 (as indicated by the dash for the input number), and output 2 either does not exist or is not available on level 2 or its connection is unknown (as indicated by the zero input number).

Or, the router might instead use the **V** command. For example:

```
** v1,23,3 v2,-,0
```

Indicating the same as the previous example.

If the router has only one level, or if all levels are connected the same, it might instead use the **Y** command. For example:

```
** Y1,23 Y2,-
```

This indicates that output 1 is connected to input 23 and output 2 is unconnected.

Routers that are *unmapped port units* will produce symmetrical status output, i.e. if there is a **Y01,23** command, there will also be a **Y23,01** command.

## "CLEAR": Clear Matrix

From	Syntax	Description
Host	CLEAR	Ask router to clear all router outputs.
Router	(None)	All outputs are unlocked, input secures are unsecured, and all outputs are either disconnected or connected to input 1.

The command "**CLEAR**" requests that the switch matrix be cleared so that all outputs are disconnected from inputs (in routers where this is possible) or else all outputs at all levels have input #1 as their source (when disconnecting is not possible). **If output locks are supported, all output locks are removed by this command. If input security is supported, all input secures are removed by this command.** This command can take several seconds to execute (depending on the size of the switch matrix), and therefore the OK response at the end of the command string could be quite late. In order to help ensure that this command isn't accidentally executed, it requires four additional characters following the "C" character, to spell out the word "CLEAR" in full.

For example, the command:

```
**CLEAR!!
```

would clear the matrix, and when finished, the following response would be generated:

```
** OK !!
```

## "U": Automatic Output Change Reporting On/Off

From	Syntax	Description
Host	U { 0   1   2   3   4 }	Set the state of automatic output change reporting.
Router	(None)	

The command "**U**" turns on or off the automatic sending of output change reports. Output change reports are automatic messages sent to the host whenever an output's crosspoint status (i.e. connected source) is changed. The command letter must be followed by either a number 0..4 to specify the new automatic change report state, as follows:

- 0: Automatic output change reporting is turned off.
- 1: Automatic output change reporting is turned on, "X" commands are used (but see discussion below for older routers). Crosspoint

change commands do not immediately report changed status, but instead, the report comes up to a few seconds after the crosspoint change command is received.

- 2: Like 1 except crosspoint change commands immediately report the changed status as part of the response to the command.
- 3: Like 1 except this enables "V" and "Y" commands also (version 10 and later).
- 4: Like 2 except this enables "V" and "Y" commands also (version 10 and later).

When automatic change reporting is on and an output's crosspoint status is changed, command(s) are automatically sent on all host protocol ports to report the crosspoint change. The commands that are sent are either "X" commands or "Y", "V", and "X" commands. Refer to the description of the "O" command above for information about which routers use only "X" commands and which ones use "Y", "V", and "X" commands.

When an automatic change report is sent, the command is preceded by a leader sequence and followed by a trailer sequence and a CR character.

There have been several additions to the original "U" command, so the version supported by a particular router depends on what software it is running:

The oldest software that supported the "U" command only supported argument values of "U0" and "U1". In those routers, a DIP switch set the initial "U" command reporting state at power-up.

The next generation of software added a second DIP switch that allowed the selection of either strictly "X" commands or the most compact mixture of "Y", "V", and "X" commands for status reporting. This was independent of the "U" command argument.

The next generation of software added support for an argument value of "U2", which enabled more rapid reporting of a crosspoint change made by sending a host protocol command. Crosspoint changes made via control panels were still reported at a slower background rate.

Beginning with version 10 router software, arguments "U3" and "U4" were added, allowing the use of "Y" and "V" commands to be controlled by the "U" command. Also, the "U" command argument setting is now stored in non-volatile memory so that once set, it will remain that way across router power-

ups, until changed. Also, the DIP switches no longer control whether automatic is on or off, or whether "Y" and "V" commands are enabled. Instead, the DIP switches now only have an effect when reinitializing the non-volatile memory, at which time they are used to determine the initial state of the "U" command argument.

The maximum value of the "U" command argument is given by the MAX\_U\_ARG() keyword in a "G ROUTER\_INFO" command.

The router can send change reports any time a response to a command is not being sent. The router should never mix the command characters of change reports with those of command responses.

A matrix clear command ("CLEAR") will cause change reports to be sent for each output/level combination that is changed.

When a host protocol command is sent to the router to change a crosspoint, the router response to that command depends on the "U" command setting and also on the version of software in the router:

The oldest software responded to host protocol crosspoint commands simply with "OK", and would *not* send any Y/V/X command in response, even if "U1" was turned on. Change reporting was only done (when "U1" was turned on) for crosspoints changed through other means, such as via a control panel, and in that case the router would send "X" commands (and "Y" and "V" commands, if these were enabled) in a low-priority background loop that might require several seconds after the control panel changed the crosspoint before the Y/V/X command was sent. It was assumed that the controlling device would not be interested in knowing about a change it had sent itself.

The next generation of software began reporting crosspoint changes made via host protocol crosspoint commands, along with the reports of changes made through other means such as control panels. This was still done in a low-priority background loop, so it might take several seconds after a controlling device changed a crosspoint before a Y/V/X command would be sent. The "OK" response was immediate, but the Y/V/X change report was delayed until some time after the "OK" was sent. This change allowed the controlling device to verify the exact state of the router matrix, including whether or not connect commands that it sent to the router were actually received and processed.

The next version of software added "U2". Turning on that mode would cause the router to respond to crosspoint commands *immediately with "X" (and "Y" and "V", if these were enabled) as part of the command response before "OK"*. The "U2" mode of operation is generally more convenient because it produces more immediate feedback to the controlling device that is sending a crosspoint change command. However, it slows down the execution of host commands because of the time required to generate the report immediately before continuing to execute additional commands that may follow.

Beginning with version 10 software, "U3" and "U4" were added. "U3" acts like "U1" with regard to reporting crosspoint changes made by the controlling device: they are reported in a low-priority background loop that can take several seconds to respond. "U4" acts like "U2": crosspoint changes made by the controlling device are reported to it immediately, making the reports more timely but slowing down host protocol command execution.

In the following examples, it is assumed that version 10 or later software is being used.

For example, the command:

```
**U1!!
```

turns on automatic output change reporting using only "X" commands. If a control panel sets output 5 to input 17 on level 3, the following command might be received:

```
** X5,17,3 !!
```

Or, if the control panel sets output 5 to input 17 on all levels 1, 2, and 3, the following commands might be received:

```
** X5,17,1 X5,17,2 X5,17,3 !!
```

The command::

```
** U3 !!
```

is similar to "U1" except that it enables "Y" and "V" commands also. In that case, if the control panel sets output 5 to input 17 on all levels, the following command might be received:

```
** Y5,17 !!
```

Or, if the control panel sets output 5 to input 8 on levels 1 and 2, and to input 17 on level 3, the following commands might be received:

```
** V5,8,8,17 !!
```

To turn off output change reporting completely, use the command:

```
** U0 !!
```

The difference between an argument value of "U1" versus "U2" (or "U3" versus "U4") has to do with the response generated by the router when it receives a crosspoint connect command ("V", "W", "X", "Y", or "Z" command). For example, suppose the router sends the following crosspoint connect command sequence:

```
** Y1,18 Y8,34 !!
```

If U1 or U3 is in effect, the response to this command will be:

```
** OK !!
```

and then sometime later, perhaps up to several seconds later, the crosspoint change reports will be sent as separate command sequences for each output:

```
** Y1,18 !!
```

```
** Y8,34 !!
```

(The presence of Y commands indicates that the mode is U3. If it were U1, the Y commands would be replaced by one or more X commands). On the other hand, if U2 or U4 is in effect, the response to the original crosspoint change command sequence will be:

```
** Y1,18 Y8,34 OK !!
```

(The presence of Y commands indicates that the mode is U4. If it were U3, the Y commands would be replaced by one or more X commands). In this case, no additional crosspoint change reports will be sent because they already HAVE been sent.

A router which permits inputs to be connected to at most one output may generate multiple "Y" or "V" or "X" command responses, one for each connection that must be disconnected, and one for the new connection. For example, suppose output 8 connects to input 6 and output 3 connects to input

4. A command is received to connect output 8 to input 4. This would generate two update commands:

```
** Y3,- !!
```

```
** Y8,4 !!
```

The first command is necessary to disconnect output 3 from input 4, and the second to connect output 8 to input 4. Implied in this second connection is the disconnection of input 6 from output 8.

In *port units*, three update commands would be generated:

```
** Y6,- !!
```

```
** Y3,- !!
```

```
** Y8,4 !!
```

The first one is necessary to disconnect port 6 from port 8, the second to disconnect port 3 from port 4, and the third to connect ports 4 and 8. Some routers will generate symmetrical messages, e.g. "Y4,8" would be generated also.

When multiple crosspoint change commands are generated from a single connect command as shown above, the disconnect messages will not necessarily be sent before the connect messages. They could be sent afterwards.

### "K": Set Password to Change Protected Settings

From	Syntax	Description
Host	K password	Set password that is compared against an internal password to see if protected router settings can be changed.
Router	(None)	

The command "K" is followed by a password, which may be 0 to *cancel* the previous password, thus causing no password to be specified, or, it may be a value between 1 and 9999 to supply a password, which remains in effect until another password (or password 0) is supplied in another "K" command. The password is stored by the router under the remote address specified using an "M" command, or under the *control port's* remote address if an "M" command was not specified before the "K" command in the same command string as the

"K" command. The password that is set remains in effect until another "K" command is received from the same remote address.

This command is supported on larger routers. It would be used to establish a password for a remote address, to be used for such things as locking and unlocking outputs and modifying the router configuration. When the router receives a protected command (one which requires use of a password) from a remote address, it compares the current password for that remote address with the password required to execute the protected command. Protected commands consist of many of the "G" commands that change router configuration, as well as *take* and *salvo trigger* commands when the output to be taken is locked. In the former case, the administrator password is the one that must be sent in the "K" command to permit the router configuration to be changed, while in the latter case the password that was used to lock the output is the one that must be sent in the "K" command to permit the *take* to occur.

Refer to the "B" command and the "G ADMIN\_PASSWORD" command (in a separate document) for more information.

For example, the command:

```
** M197 K1777 !!
```

requests that password 1777 be stored as the current password for remote address 197. The command:

```
** M197 Y7,9 !!
```

requests that output 7 be connected to input 9, using the password stored for remote address 197 as the output lockout override password. The *take* will succeed if output 9 is either not locked or is locked with password 1777. Otherwise, the *take* will fail.

### "M": Set Remote Address

From	Syntax	Description
Host	M addr	Set addr as the remote address for the rest of the commands in the current command string.
Router	M addr	Echo the remote address.

The command " M" is followed by a remote address value, and it sets the remote address to be used by all remaining commands in the current

command string. This command is supported on larger routers. It would be used when a client/server software system on a router control port permits multiple remote users to send commands to the router through that control port. In such a system, the server software that talks to the control port can insert an "M" command at the beginning of each command string it sends to the router on behalf of its clients. The server would assign a different address to each of its clients. The range of available remote addresses can be obtained by using the "G CTRL\_PORTS" command, described in a separate document.

The router uses the address to control access to and modification of protected resources. The "M" command will typically work in conjunction with the "K" command, which allows a password to be sent to the router. Some commands, such as those that allow modification of router configuration settings, might require entry of an administrator password before allowing the router configuration to be modified. The "K" command is used to send the password, but the router must be able to record that password in association with a particular remote user. The router would save the password as the one established by the remote address specified in the "M" command. The examples below show how this would work.

If the "M" command is not present in a control string received on a serial port, the address assigned to the *serial port itself* is used as the address for all commands in that command string.

The "M" command works on an honor system. There is nothing to keep a remote device from forging a false address. This conforms with the Sierra philosophy of providing protection mechanisms that are not designed to be totally hack-proof, but rather, are designed under the assumption that controlling devices will honor the system. A dedicated hacker can always hack into the system if he chooses. Note, however, that the server in a client/server relationship can provide a great deal of added security for the system, forcing the correct "M" command to be sent each time, and filtering out bogus "M" commands received from clients.

When the router receives an "M" command in a command string, it echoes the same "M" command in its response. This allows the server connected to the serial port to parse the received response string and determine to which client it should route the response.

Normally the server will assign remote addresses to its clients. However, it should always provide a way to send these addresses to the client, because

the client needs to know its address in order to know how to interpret some responses. By simply passing each "M" response command received from the router back to the client, the server can let the client know what its address is.

For example, the command:

```
** M139 K9664 !!
```

establishes password 9664 as the password for remote address 139. The response to this command string would be:

```
** M139 OK !!
```

If remote address 13 later sends the command:

```
** M139 Y8,27 !!
```

to request that output 8 be connected to input 27, the router would check to see if output 8 has been locked. If so, the router compares the password of remote address 139 (which has previously been set to 96643) to the password that was used to lock output 8. If they match, the connection request succeeds, but if they don't match, output 8 is left unchanged.

### "V": Connect Levels

From	Syntax	Description
Host	V out,in,in...	Make a connection to an output on each level.
Router	(None or V/Y/X command, depending on which "U" command argument is in effect, see "U" command description.)	Optional status response.

The command "V" is used to request that a connection be made. It must be followed by an output number, a comma, and a comma-separated list of input numbers, one for each level, up to the number of levels in the router. Fewer than the number of levels may be specified if desired, and the remaining levels will be left unchanged.

For example, the command:

```
**V12,7,8,9!!
```

says that connections are to be made to output 12: from input 7 on level 1, input 8 on level 2, and input 9 on level 3.

An input number of 0 means the output connection is to be left *unchanged*.

An input number of '-' (dash) means the output is to be *disconnected*. If the router does not support disconnected outputs, the output connection will be left unchanged.

Unless otherwise noted in the documentation for a specific switcher, it can be assumed that the switcher software guarantees to send all level changes associated with the "V" command's output to the crosspoint matrix within the SAME vertical interval.

If the requested connection has an output number that does not exist on one or more levels, those levels are simply not changed. It is an error to request connection of an input that doesn't exist on that level, even if the input does exist on some other level. If the requested connection has an output or input number that isn't mapped to a physical connector (on virtual-mapped routers) on one or more levels, those levels are simply not changed.

If "U2" is in effect (see "U" command), the response will include one or more V, Y, or X commands to report the new status of the output. The response will be the same as if an "O" command were issued for the output immediately following the "V" command. No response is generated if this command is being used to define a salvo.

### "W": Connect Outputs

From	Syntax	Description
Host	W lvl,in,in...	Make a connection to each output on a level.
Router	(None or V/Y/X command, depending on which "U" command argument is in effect, see "U" command description.)	Optional status response.

The command "**W**" is used to request that a connection be made. It must be followed by a level number, a comma, and a comma-separated list of input numbers, one for each output, up to the number of outputs in the router. Fewer than the number of outputs may be specified if desired, and the remaining outputs will be left unchanged.

For example, the command:

```
**W1,17,3,9!!
```

says that connections are to be made on level 1: from input 17 to output 1, input 3 to output 2, and input 9 to output 3.

If the level number is specified as "0", this means that the connection is to be made on **all levels** (AFV).

For example, the command:

```
**W0,8,3,7!!
```

says that connections are to be made on all levels: from input 8 to output 1, input 3 to output 2, and input 7 to output 3.

An input number of 0 means the output connection is to be left *unchanged*.

An input number of '-' (dash) means the output is to be *disconnected*. If the router does not support disconnected outputs, the output connection will be left unchanged.

It is an error to request connection of an input or output that doesn't exist on the specified level, even if the input or output does exist on some other level. However, if the level number is "0", any input or output may be used as long as it exists on at least one level, and in that case no connection will be made on any level on which the input and output does not exist. If the requested connection has an output or input number that isn't mapped to a physical connector (on virtual-mapped routers) on one or more levels, those levels are simply not changed.

If "U2" is in effect (see "U" command), the response will include one or more V, Y, or X commands to report the new status of the outputs. The response will be the same as if "O" commands were issued for the outputs immediately following the "W" command. No response is generated if this command is being used to define a salvo.

### "X": Connect Crosspoint

From	Syntax	Description
Host	X out,in,lvl	Make a connection to an output on a level.
Router	(None or V/Y/X command, depending on which "U" command argument is in effect, see "U" command description.)	Optional status response.

The command "X" is used to request that a connection be made. It must be followed by an output number, a comma, an input number, a comma, and a level number.

For example, the command:

```
**X24,13,2!!
```

says that a connection is to be made between output 24 and input 13 on level 2. If the level number is specified as "0", this means that the connection is to be made on **all levels** (AFV).

For example, the command:

```
**X8,3,0!!
```

says that a connection is to be made between output 8 and input 3 on all levels.

An input number of 0 means the output connection is to be left *unchanged*, not very useful in an "X" command.

An input number of '-' (dash) means the output is to be *disconnected*. If the router does not support disconnected outputs, the output connection will be left unchanged.

When a level number of 0 is used with the "X" command, it can be assumed that the switcher software guarantees to send all level changes for the output to the crosspoint matrix within the **SAME** vertical interval, unless otherwise noted in the documentation for a specific switcher.

It is an error to request connection of an input or output that doesn't exist on the specified level, even if it does exist on some other level. However, if the level number is "0", any input or output number may be specified as long as it exists on at least one level, and in that case no connection will be made on any level on which the input or output does not exist. If the requested connection has an output or input number that isn't mapped to a physical connector (on virtual-mapped routers) on one or more levels, those levels are simply not changed.

If "U2" is in effect (see "U" command), the response will include one or more V, Y, or X commands to report the new status of the output. The response will be the same as if an "O" command were issued for the output immediately

following the "X" command. No response is generated if this command is being used to define a salvo.

### "Y": Connect AFV

From	Syntax	Description
Host	Y out,in	Make a connection to an output on all levels.
Router	(None or V/Y/X command, depending on which "U" command argument is in effect, see "U" command description.)	Optional status response.

The command "Y" is used to request that a connection be made. It must be followed by an output number, a comma, and an input number. The connection is made on all levels (AFV).

For example, the command:

```
**Y2,29!!
```

says that input 29 is to be connected to output 2 on all levels.

An input number of 0 means the output connection is to be left *unchanged*, not very useful in a "Y" command.

An input number of '-' (dash) means the output is to be *disconnected*. If the router does not support disconnected outputs, the output connection will be left unchanged.

Unless otherwise noted in the documentation for a specific switcher, it can be assumed that the switcher software guarantees to send all level changes for the "Y" command's output to the crosspoint matrix within the SAME vertical interval.

Any input or output number may be specified as long as it exists on at least one level. No connection will be made on any level on which the input or output does not exist. If the requested connection has an output or input number that isn't mapped to a physical connector (on virtual-mapped routers) on one or more levels, those levels are simply not changed.

If "U2" is in effect (see "U" command), the response will include one or more V, Y, or X commands to report the new status of the output. The response will be the same as if an "O" command were issued for the output immediately following the "Y" command. No response is generated if this command is being used to define a salvo.

## "Z": Connect Outputs AFV

From	Syntax	Description
Host	Z in,in...	Make a connection to each output on all levels.
Router	(None or V/Y/X command, depending on which "U" command argument is in effect, see "U" command description.)	Optional status response.

The command "Z" is used to request that a connection be made. It must be followed by a comma-separated list of input numbers, one for each output, up to the number of outputs in the router. Fewer than the number of outputs may be specified if desired, and the remaining outputs will be left unchanged. The connection is made on all levels (AFV).

For example, the command:

```
**Z4,18,7!!
```

says that input 4 is to be connected to output 1 on all levels, input 18 to output 2 on all levels, and input 7 to output 3 on all levels.

An input number of 0 means the output connection is to be left *unchanged*.

An input number of '-' (dash) means the output is to be *disconnected*. If the router does not support disconnected outputs, the output connection will be left unchanged.

Any input or (implied) output number may be specified as long as it exists on at least one level. No connection will be made on any level on which an input or (implied) output number does not exist. If the requested connection has an output or input number that isn't mapped to a physical connector (on virtual-mapped routers) on one or more levels, those levels are simply not changed.

If "U2" is in effect (see "U" command), the response will include one or more V, Y, or X commands to report the new status of the outputs. The response will be the same as if "O" commands were issued for the outputs immediately following the "Z" command. No response is generated if this command is being used to define a salvo.

## "B": Output Lock Inquiry or Change

From	Syntax	Description
Host	B out,password,lock or B out,0,2	Change output lock status of all levels of 'out'.

	or <b>B</b> 0,0,lock,out1,password1,out2,password2,out3,password3,... or <b>B</b> 0,0,2,out1,out2,out3,...	Ask for output lock status of all levels of 'out' (argument '2' can also be '0' instead).  Change output lock status of each level individually.  Ask for output lock status of each level individually.
Router	<b>B</b> out,password,lock or <b>B</b> 0,0,3,out1,password1,out2,password2,out3,password3,...	Lock status of 'out' on all levels. or Lock status of the specified outputs on each level.

The command "**B**" requests that lock information for the specified outputs be returned to the host, and optionally that the lock status of those outputs be changed. Whenever the router receives a "B" command, it sends one back. Also, any change to the lock state of an output is reported automatically via a "B" command, regardless of the source of the change, and this automatic reporting cannot be turned off.

In routers that support the "B" command, the router reports changes to output lock status using the "B" command, not the "G OUTPUT\_LOCK" command.

When an output is locked, it cannot be routed to a new input unless the password sent using the "K" command matches the password used to lock the output.

In virtual-mapped routers, this command applies lockouts to *virtual destinations* rather than *physical outputs*, so the word "output" should be replaced with "destination" in this command description. Of course, once a virtual destination is locked, the physical output to which it is mapped is also effectively locked.

There are four forms of this command that can be sent *to the router*, and two forms that can be sent *from the router*.

### **"B" Commands Sent to the Router Command: B out,password,lock**

This form of the command is used to lock or unlock an output. It applies to *all levels* and only one output can be specified. It contains three arguments which are defined as follows:

1. out: The output number whose lock status is being changed.
2. password: The password for locking or unlocking the output, non-zero.
3. lock: A flag that is 0 to unlock the output, 1 to lock the output.

After changing the lock status, the router sends a "B" command to report the new lock status of the specified output, see below.

**Command: B out,0,2**

This form of the command is used to query for the lock status of an output. It queries for the lock status of *all levels* and only one output can be specified. The first argument is the output number whose lock status is being queried, the second argument is always 0 (zero), and the third argument is always 2 (for compatibility with pre-version-10 software, the third argument may instead be 0).

When it receives this command, the router sends a "B" command to report the current lock status of the specified output, see below.

**Command: B**

**0,0,lock,out1,password1,out2,password2,out3,password3,...**

This form of the command is used to lock or unlock individual outputs on each level. Its first two arguments are 0, the third argument is the 'lock' flag with the same meaning as described above for the first form of the command, and the remaining arguments are (out, password) pairs, one for each level, that provide the output number and lock password on that level. An output number of 0 means that no change is to be made on that level. On each level, the specified output is either locked (when 'lock' is 1) or unlocked (when 'lock' is 0) with the specified password.

After changing the lock status, the router sends a "B" command to report the new lock status of the specified outputs, see below.

**Command: B 0,0,2,out1,out2,out3,...**

This form of the command is used to query for the lock status of individual outputs on each level. Its first two arguments are 0, the third argument is 2, and the remaining arguments are output numbers, one for each level. An output number of 0 means that lock status is not needed on that level.

When it receives this command, the router sends a "B" command to report the current lock status of the specified outputs, see below.

### **"B" Commands Sent *from* the Router Command: B out,password,lock**

This form of the command is used to report the lock status of an output that has the same lock password on all levels. It contains three arguments which are defined as follows:

1. out: The output number whose lock status is being reported.
2. password: The password that locks the output, or 0 if the output is not locked.
3. lock: A flag that is 0 if the output is not locked (password is 0), or 1 if it is locked (password is not 0).

### **Command: B 0,0,3,out1,password1,out2,password2,out3,password3,...**

This form of the command is used to report the lock status of individual outputs on each level. Its first two arguments are 0, the third argument is 3, and the remaining arguments are (out, password) pairs, one for each level, that provide the output number and lock password on that level. An output number of 0 means that no report is being made on that level.

## **Lock Discussion**

An attempt to lock an output that is already locked (using the same password it is already locked with), or to unlock an output that is already unlocked (using any password), succeeds without doing anything. (Older versions of router software, prior to version 10, would generate an error response for this).

An attempt to lock an output that is already locked, using a different password than the one it is already locked with, fails, as does an attempt to unlock an output using a password that is different from the password that the output was locked with and is not the administrator password. Any use of a password larger than 9999 also fails. In any of those cases, the output lock status remains unchanged and if no change was made to any level, an error response is generated. The "B" command response from the router will indicate that the output still has the same lock state as before.

Beginning with router software version 10, an output lock applies to outputs on a *per-level basis* as indicated by the above command forms that apply locks to individual levels. Prior to that time, an output lock applied to an output *on all*

*levels*, and only the forms of the commands with exactly three arguments were supported. The "PER\_LEVEL\_LOCKS" keyword of the "G ROUTER\_INFO" command will be present in all routers that support locks on individual levels. Note that in pre-version 10 virtual-mapped routers, a level may be left out of a lockout by making sure the level is unmapped for the destination being locked.

The "B" command, which was added in its three-argument form at router software version V5.06, is preferred over the "G OUTPUT\_LOCK" command. Note that in pre-version 10 routers, the "B" command provided the same functionality as the "G OUTPUT\_LOCK" command, but beginning at version 10, the "B" command but *not the "G OUTPUT\_LOCK" command* provided the additional functionality of allowing locks to be specified on a per-level basis.

If the output is locked on different levels with different passwords, an attempt to unlock it using a password that matches the lock password of some but not all levels causes the matching levels to be unlocked and the non-matching levels to remain locked as they were before. If the output is locked on some but not all levels, an attempt to lock it causes the unlocked levels to be locked with the specified password and the locked levels to remain locked with the same password as before.

When locking an unlocked output, the specified password is recorded by the router as the lock password for that output (on the specified levels). When unlocking a locked output, the specified password is compared by the router to the lock password for the output on each of the specified levels. If they match, or if the specified password is the administrator password, the output is unlocked, else it remains locked. Attempting to lock an already-locked output, or unlock an already-unlocked output, has no effect on the lock state of that output.

Whenever a *take* is done, the router checks to see if the specified output for the *take* has been locked. If so, the current password of the control port that sent the *take* request (as set with the "K" command) is compared to the output's lock password. If they match, the *take* is allowed, but if not, the *take* has no effect. The *administrator password may NOT be used in lieu of the output's lock password to do a take*, so if a controlling device sends the administrator password in a "K" command, that will not permit the device to reroute locked outputs (but it can use the administrator password to unlock locked outputs using the "B" command).

As with all router configuration parameters, the output lockout data is stored in non-volatile storage and thus is retained across router power-ups. When a router is first initialized at the factory, all outputs are set to be unlocked.

When the router sends a "B" command response, it uses the first form of the response command if possible (i.e. if the output number in question, and its lock password, is the same on all levels). Otherwise, it uses the second form of the response command. When the second form of the "B" response command is sent to report changed lock passwords, only the lock status of the *changed levels* is sent, and the other levels have the (out,password) argument set to (0,0).

Note that the output lock version number is not present in this command. Since output lock status changes frequently, it is not really useful to cache output lock status for outputs, so the output lock version number is not really useful.

## **Types of Locks**

There are many possible ways to implement output lockouts in a controlling device using the "B" command. Svs has standardized on three types, which are called:

These are not mutually exclusive. All three could exist in one system. However, most users will choose just one of these schemes, or perhaps one of the two lock methods plus the protect method. The schemes are chosen on a controller-by-controller basis. That is, each controlling device can use whichever method(s) of output locking it chooses to use.

## **Simple Lock**

When a "Simple Lock" button is pushed on a controlling device, the current destination becomes locked and a LOCKED indicator would light up on the controlling device. Nobody can route that destination. If they try, they will get an error message or a LOCKED indicator will flash or something like that.

To unlock a simple-locked destination, you simply press the "Simple Lock" button again. *Anybody can unlock a simple-locked destination, not just the panel that locked it!!!!*

So, a simple-locked destination is a sort of global lock – anybody can lock or unlock it, but it serves as a reminder to people that someone has locked it and they should leave it alone. This will be the type of lock that most people will want to use.

A simple lock is actually a password lock, as all locks must be. A simple lock uses a global password of 9999 to lock or unlock the destination. Therefore, password 9999 is special. It could be entered by a user also, with password locking, but that would result in a simple-locked destination.

### **Password Lock**

When a "Password Lock" button is pushed on a controlling device, the user is prompted to enter a 4-digit password that is not allowed to be 0. Then, the current destination is locked with that password and a LOCKED indicator would light up. Nobody can route that destination. If they try, they will again get an error message or a LOCKED indicator will flash or something like that.

To unlock a password-locked destination, you press *any* lock or protect button (any one of the three types). You are then prompted to enter the password for the locked destination. You can enter either the correct password that the destination was originally locked with, or the administrator password. If you enter either one correctly, the destination is then unlocked.

So, a password-locked destination is far more solid of a lock than a simple locked destination. You have to know the password to unlock it. Nobody can make routes to the destination while it is locked. This type of lock is not expected to be used very often.

The password lock is the most direct use of the "B" command. When the user enters the password, that password is sent in a "B" command to lock or unlock the destination.

### **Protect**

Any controlling device with a "Protect" button is required to have a user or controller password that can be programmed by the user or by an administrator in some way. A controller setup menu will permit the password

to be set or changed. A user might be required to enter his username and password to log on to the controlling device before he can use it to control the router.

When a "Protect" button is pushed on a controlling device, the current destination becomes locked at *all other* controlling devices (a LOCKED indicator would light up) and *protected at this controlling device* (a PROTECTED indicator would light up). Nobody at other controlling devices can route the destination. If they try, they will get an error message or a LOCKED indicator will flash or something like that. At the controlling device that protected it, however, the destination *can be freely routed*, with no error messages, because the user password matches the destination lock password.

To unlock a protected destination, the user simply presses the "Protect" button again, on the controlling device that originally protected the destination. It can also be unlocked if you know the password of the user or controlling device that locked it, or if you know the administrator password. This is done exactly the same way as unlocking a password-locked destination, as described above.

So, a protected destination is like a password-locked destination at other controlling devices, but at the controlling device that protected it, it is like an unlocked destination. This type of destination protection is expected to be very popular.

A protected destination is actually password locked, of course. The controlling device sends the user password in the "B" command to lock or unlock the destination. The controlling device recognizes that, although the destination is locked, the lock password matches its user password, and therefore it should display the destination as PROTECTED rather than LOCKED. The controlling device sends its user password to the router using a "K" message, when it first connects with the router (or when a user logs on to the controlling device). The router records this password. Whenever that same controlling device sends a TAKE command of any kind to the router, the router recognizes that although the destination to be taken is locked, the lock password matches the controlling device password that was previously sent to it in the "K" message, and it goes ahead and performs the Take.

Note that it is relatively simple for a "hacker" to beat the system. By monitoring control panel traffic on a control panel bus, a hacker could obtain the passwords for locked outputs, and this would allow him to route them to any

inputs he pleased. Sierra philosophy is to provide a system of "elegant simplicity" that serves basic needs of typical users. To truly protect a locked output, it would be necessary to implement full encryption of commands between the router and controlling devices. This is far beyond the Sierra philosophy.

## Output Lock Examples

For example, to request whether or not output 21 is locked on all levels:

```
** B21,0,2 !!
```

The response might be:

```
** B21,0,0,0 OK !!
```

indicating that output 21 is NOT locked on any level. Or, the response might be:

```
** B21,6741,1 OK !!
```

indicating that output 21 is locked with password 6741 on all levels. Or, the response might be:

```
** B0,0,3,21,0,21,0,21,6741,21,6741 OK !!
```

indicating that output 21 is not locked on levels 1 and 2 (password is 0) and is locked with password 6741 on levels 3 and 4. To clear this lockout:

```
** B21,6741,0 !!
```

which unlocks both levels 3 and 4 because they are both locked with the same password, 6741.

If the lock status of output 21 changes, the router might send the following:

```
** B0,0,3,21,1234,21,0,21,6741,21,6741 OK !!
```

indicating that output 21 is locked with password 1234 on level 1, is not locked on level 2 (password is 0), and is locked with password 6741 on levels 3 and 4. To clear this lockout:

```
** B0,0,0,21,1234,0,0,21,6741,21,6741 !!
```

which specifies (0,0) for level 2 since it is already unlocked, and which provides the proper passwords for unlocking on levels 1, 3 and 4. If instead this command were sent:

```
** B21,6741,0 !!
```

then only levels 3 and 4 would be unlocked, and level 1 would remain locked with password 1234. The router would send the following to indicate this:

```
** B0,0,3,21,1234,21,0,21,0,21,0 OK !!
```

To request the lock status of output 21 on level 1 and output 44 on level 4:

```
** B0,0,2,21,0,0,44 !!
```

The router might reply with:

```
** B0,0,3,21,1234,0,0,0,0,44,9876 OK !!
```

indicating that output 21 on level 1 is locked with password 1234, and output 44 on level 4 is locked with password 9876. To unlock these outputs, the following could be sent, assuming that the administrator password is 7777:

```
** B0,0,0,21,7777,0,0,0,0,44,7777 !!
```

### "F": Field Delay for Crosspoint Output

From	Syntax	Description
Host	F delayfields	Set delay from command received to crosspoint output for future crosspoint commands.
Router	(None)	

The command "F" is used to specify the delay between the time a crosspoint change request is received by the router and the time the crosspoint switch actually occurs. It must be followed by a number giving the number of video fields of delay desired. If the number is smaller than the smallest delay that the router can handle, the smallest delay is used instead. If it is larger than the largest delay the router can handle, the largest delay is used instead. Note that this command does not cause a delay in command processing, as the "D" command does.

To understand this command more fully, consider the way that router software will typically handle a crosspoint command. The last character of the command string, the final "!" (exclamation) character, is received somewhere in a particular video field, call it video field 7. The router parses the command string and, for each crosspoint it contains, it puts the crosspoint in a buffer that is marked to be delivered to the crosspoint hardware on a particular video field. Suppose that previously, an "F5" command has been received. Then crosspoint commands whose final "!" command string character was received on video field 7 would be placed in a buffer that is marked to be delivered to the crosspoint hardware at video field 13 ( $7+5+1=13$ ).

To understand the reason for adding 1 in the previous sum, consider an "F0" command: it would ask for output at the very next video field, field 8 in our case. So, it is necessary to add the "F" argument plus 1 to the field number on which the crosspoint command is received to get the field number at which the crosspoint will be output.

A typical router will have a minimum delay that is between 1 and 2 fields. Suppose a crosspoint command is received just before a vertical field mark. The software may be able to prepare the crosspoint data and send it to the hardware when that vertical field mark occurs, but the hardware itself typically has a one-field delay in it, so the soonest that such a crosspoint would switch would be one field (plus a little) from when it was received. If the command were received towards the beginning of a field rather than the end of a field, the delay would be closer to two fields. Industry parlance is to call this a one-field delay, because only *full fields* of delay are counted.

The *delayfields* argument of this command takes into account the hardware delay. So, if a router has a minimum delay, including the hardware delay, of one full field, as described in the previous paragraph, then a *delayfields* value of 1 causes this minimum delay to be used. A *delayfields* value of 0 will also cause this minimum delay, because the router uses its minimum if a smaller value is specified. A *delayfields* value of 2, however, will add one more field to the minimum possible delay. Thus, *delayfields* specifies the *number of full fields* of delay between end-of-crosspoint-command-string-received and crosspoint-switch-occurs. Note that the actual minimum value of *delayfields* depends on the particular router model.

Routers typically have a limit to the number of crosspoint commands they can process in one field. First, there is an inherent delay in *sending* the command to the router, but beyond that, the router requires time to parse the command and buffer up the crosspoint data, plus it requires time to deliver the buffered

data to the hardware when the desired video field arrives. Each individual router has documentation to describe its limitations on how many crosspoints it can process in a given amount of time.

Larger values for *delayfields* give the router more time to process commands. Although the long-term average number of crosspoints that can be processed per unit of time is unchanged, a larger *delayfields* value can improve router performance during a short burst of many crosspoint commands. For example, suppose a large number of crosspoint commands is sent to the router in a single large command. If *delayfields* is small, the router typically won't have time to parse and process all these crosspoint commands and place the data in the crosspoint delivery buffer before the target video field arrives. By making *delayfields* larger, the user can give the router more time to process the crosspoint commands.

If too many crosspoint commands are received and the router is not able to process them fast enough, it will output the crosspoint connections as soon as it can. Unexpected delays in crosspoint output are a sign that the router is being pushed beyond its limits.

The *fielddelay* value applies to the *entire router*, not just to the control port on which the "F" command is received. It is therefore recommended that a single value be settled on for the *fielddelay* value, rather than changing the value constantly depending on needs. Once changed, the router records the value in non-volatile memory and uses it each time it is powered up, so it is only necessary to change it one time.

Even though a crosspoint isn't changed until the *fielddelay* time has elapsed, the router records the new crosspoint state *immediately upon receiving the crosspoint change request*, so a controlling device may receive a report of a crosspoint change before the change has actually taken effect, and this is more likely to happen the larger *fielddelay* is. Since SVS routers currently make no guarantees about when they will report a crosspoint change anyway, this behavior is usually of no concern. There is a case where this could cause problems. If the *fielddelay* value were to be changed while two different devices were changing the same output, it is possible for the router to report the incorrect input value for that output. This would happen if the earlier device that changed the output did so before the *fielddelay* value was changed, and the later device that changed the output did so after the *fielddelay* value was reduced but soon enough that its input value would be sent to the crosspoint hardware before that of the earlier device. A bit later, the earlier device's input value is sent to the crosspoint hardware, but the

router has recorded the later device's input value as being the one in effect. To prevent this scenario, we recommend that an appropriate *fielddelay* value be chosen, set, and left alone.

Here is an example of an "F" command:

```
** F5 Y1,5 X2,6,3 !!
```

This says that input 5 is to be connected to output 1 on all levels and input 6 is to be connected to output 2 on level 3, after a delay of 5 fields from the beginning of the field that follows receipt of the "!" character.

### "D": Delay Vertical Sync Intervals

From	Syntax	Description
Host	D numsyncs	Delay a number of vertical sync intervals.
Router	(None)	

The command "**D**" is used to delay before continuing execution of the commands that follow. It must be followed by a number giving the number of vertical sync intervals by which to delay. If the number is 1, the delay will be to the VERY NEXT vertical sync interval. If the number is 0, no delay occurs. The number must be no larger than 255. Note that this command will also delay the time at which the remaining command responses and the trailer character are returned to the host.

For example, the command:

```
** Y1,5 D200 D100 Y1,6 S !!
```

says that input 5 is to be connected to output 1 on all levels, then a delay of 300 (=200+100) sync intervals is to occur, then input 6 is to be connected to output 1 on all levels, then a status response is to be returned.

It is generally recommended that the host computer be responsible for timing the initiation of commands, rather than using this command to do the job. The host computer can simply send the appropriate commands at the appropriate times. The "P" and "T" commands described below can aid in ensuring that lengthy connect sequences aren't delayed due to the time it takes to send them to the router.

### "P": Preset a Salvo Connect Sequence

From	Syntax	Description
------	--------	-------------

Host	P reg connect connect ... ~	Store a list of connect commands in a salvo register.
Router	(None)	

The command "P" is used to set up a *salvo*, which is a series of connect commands for later execution with the "T" command (see below). It must be followed by a register letter from A to Z or a register number from 1 to 256 giving the *register* into which the connect sequence is to be stored, followed by zero or more *connect commands*(V, W, X, Y, or Z), followed by a "~" (tilde) character.

Registers A-Z are the same registers as 1-26. The letter designators are allowed to shorten up the command sequence slightly, so that "T" commands can be sent in compact form if one of the first 26 registers is used.

For example, the command:

```
** PB X2,5,0 Z7 ~ !!
```

says that two connect commands (output 2 to input 5 AFV, and output 1 to input 7 AFV) are to be stored into salvo register B (i.e. register 2). Another example:

```
** P239 Y5,7 ~ !!
```

says that one connect command (output 5 to input 7 AFV) is to be stored in salvo register 239.

Only *supported connect commands* (those connect commands whose command letters appear in the "I" command response) may follow the register letter up to the tilde character. The connect commands do not take effect until the register is *triggered* using the "T" command below.

The maximum allowed number of connect commands is determined by the particular router. In all cases where this command is implemented, there is guaranteed to be space available to store at least two complete switch matrices AT ONE LEVEL. If more connect commands are received than there is space available to store them, the error response string "FULL ERROR Salvo Space Full" is sent to the host, and only the first part of the salvo is stored. For example:

```
** FULL ERROR Salvo Space Full !!
```

says that the salvo request filled memory and the salvo could not be completely stored.

Refer to documentation for each router to determine the amount of memory available for storing salvo sequences. To be completely on the safe side regarding available memory, you could use only registers A and B, and reload them in between each trigger command if necessary. A register that is no longer needed can be cleared with the sequence "Px~", where 'x' represents the register letter or number, thus freeing up its memory.

Salvo registers are accessible to the host and all control panels that support salvo programming. A specified register can only be modified by one source at a time. If the salvo register that a "P" command specifies is currently being modified by someone else (e.g. a control panel), the response string "BUSY ERROR Salvo # In Use" is sent to the host, and the "P" command is ignored. For example:

```
** BUSY ERROR Salvo 239 In Use !!
```

says that the salvo request cannot be honored because requested salvo register #239 is being modified by someone else.

Salvo sequences will typically be stored in non-volatile memory in the router, although this is not a requirement. Prior to version 2.00, the *Sierra Video Systems* 504001 router controller did not store salvo sequences in non-volatile memory, but beginning at that version it did.

It is recommended that salvo registers 100-163 be reserved for control-panel-unique salvos, with register 100 being for panel address 0, 101 for panel address 1, etc.

Salvos can also be stored using the "G PRESET, STORE" command, see below. Note: A "P" command can safely be done in the middle of a "G PRESET, STORE" command, as long as it is done to a *different* salvo register than the one that the "G PRESET, STORE" command is using.

### "T": Trigger a Salvo Connect Sequence

From	Syntax	Description
Host	T reg	Trigger a list of connect commands stored in a salvo register.
Router	(None)	

The command "T" is used to trigger a previously set up *salvo* (set using the "P" command above). It must be followed by a register letter from A to Z or a register number from 1 to 256 giving the *register* to be triggered.

For example, the command:

```
** TB D180 TC !!
```

says to trigger salvo register B (same as 2), delay 180 sync intervals, then trigger salvo register C (same as 3). When the register is triggered, this means that the connect commands stored in it take effect.

If a salvo is triggered and it attempts to connect a locked output or port, or a disallowed input/output pair, or a port to itself, the salvo trigger operation is aborted, no crosspoint changes are performed, and an error is reported: "ERROR Salvo Has Locked Xpts".

### "HOST": Select Host or Terminal Protocol

From	Syntax	Description
Host	HOST { 0   1   2 }	Select host or terminal or extron protocol.
Router	(None)	

The command "HOST" selects which protocol will run on the serial port. The argument specifies the protocol, as follows:

- 0: standard generic serial protocol mode (as described in this document)
- 1: user-friendly terminal protocol mode of interaction.
- 2: extron serial protocol mode (see extron protocol document).

In order to help ensure that this command isn't accidentally executed, it requires four additional characters following the "H" character, to spell out the word "HOST" in full, followed by a digit.

For example, the command:

```
**HOST0!!
```

would switch the serial port to a user-friendly terminal control protocol. Further interaction on the control port from this point on depends on the details of that terminal protocol. However, the terminal protocol will be designed to allow a switch *back* to the generic serial protocol mode by sending this sequence:

```
** HOST1 !!
```

Two asterisks and exclamations can be used if desired. After this, regular serial protocol commands should work once again.

## "A": Extended Commands

The command letter "A" is used to extend the basic command set by adding 26 possible 2-letter commands, each consisting of the letter "A" followed by another letter.

The 2-letter command "AA" is hereby reserved in case it becomes necessary to extend the command set to include 3-letter commands.

To find out if the "A" command is supported, use the "I" command and check the response to see if the letter "A" is included.

To find out which 2-letter "A" commands are supported, use the "AI" command and check the response to see which letters are allowed following "A".

## "AI": Extended Capabilities Inquiry

From	Syntax	Description
Host	AI	Ask router to send a list of extended commands that are available.
Router	AI[R]~	Available extended commands. (Stuff inside [] is optional depending on router)

The command "AI" requests that extended command capability information be returned to the host. The information is sent as a string of characters. The first characters are a space followed by "AI", the next characters are the letters of the extended commands (those consisting of the letter "A" followed by one of these letters) that are implemented and available in this router, and the last character is "~" (tilde). Do not count on the characters being in any specific order. Search all characters for a particular one.

To find out if the "A" command is supported, use the "I" command and check the response to see if the letter "A" is included.

For example, the command:

```
**AI!!
```

might return the following string:

\*\* AIR~ OK !!

indicating that the router supports the **AI** and **AR** commands from the host.

### "AR": Input Security Inquiry or Change

From	Syntax	Description
Host	AR in,password,secure or AR in,0,2 or AR 0,0,secure,in1,password1,in2,password2,in3,password3,...	Change input security status of all levels of 'in'.  Ask for input security status of all levels of 'in' (argument '2' can also be '0' instead).
	AR 0,0,2,in1,in2,in3,...	Change input security status of each level individually.  Ask for input security status of each level individually.
Router	AR in,password,secure or AR 0,0,3,in1,password1,in2,password2,in3,password3,...	Security status of 'in' on all levels. or Security status of the specified inputs on each level.

The command "**AR**" requests that security information for the specified inputs be returned to the host, and optionally that the security status of those inputs be changed. Whenever the router receives a "AR" command, it sends one back. Also, any change to the security state of an input is reported automatically via an "AR" command, regardless of the source of the change, and this automatic reporting cannot be turned off.

This command is exactly analogous to the "B" command for output locks, but applies to inputs instead, and the term "secure" is used instead of "locked" to distinguish between the two.

The "SRC\_SECURES" keyword of the "G ROUTER\_INFO" message will be present in all routers that support input security passwords.

When an input is secured, it cannot be routed to a new output unless the password sent using the "K" command matches the password used to secure the input.

In virtual-mapped routers, this command applies security to *virtual sources* rather than *physical inputs*, so the word "input" should be replaced

with "source" in this command description. Of course, once a virtual source is secured, the physical input to which it is mapped is also effectively secured.

There are four forms of this command that can be sent *to the router*, and two forms that can be sent *from the router*.

### **"AR" Commands Sent to the Router Command: AR in,password,secure**

This form of the command is used to secure or unsecure an input. It applies to *all levels* and only one input can be specified. It contains three arguments which are defined as follows:

1. in: The input number whose security status is being changed.
2. password: The password for securing or unsecuring the input, non-zero.
3. secure: A flag that is 0 to unsecure the input, 1 to secure the input.

After changing the security status, the router sends an "AR" command to report the new security status of the specified input, see below.

### **Command: AR in,0,2**

This form of the command is used to query for the security status of an input. It queries for the security status of *all levels* and only one input can be specified. The first argument is the input number whose security status is being queried, the second argument is always 0 (zero), and the third argument is always 2.

When it receives this command, the router sends an "AR" command to report the current security status of the specified input, see below.

### **Command: AR 0,0,secure,in1,password1,in2,password2,in3,password3,...**

This form of the command is used to secure or unsecure individual inputs on each level. Its first two arguments are 0, the third argument is the 'secure' flag with the same meaning as described above for the first form of the command, and the remaining arguments are (in, password) pairs, one for each level, that provide the input number and security password on that level. An input number of 0 means that no change is to be made on that level. On each level, the specified input is either secured (when 'secure' is 1) or unsecured (when 'secure' is 0) with the specified password.

After changing the security status, the router sends an "AR" command to report the new security status of the specified inputs, see below.

**Command: AR 0,0,2,in1,in2,in3,...**

This form of the command is used to query for the security status of individual inputs on each level. Its first two arguments are 0, the third argument is 2, and the remaining arguments are input numbers, one for each level. An input number of 0 means that security status is not needed on that level.

When it receives this command, the router sends an "AR" command to report the current security status of the specified inputs, see below.

**"AR" Commands Sent from the Router Command: AR in,password,secure**

This form of the command is used to report the security status of an input that has the same security password on all levels. It contains three arguments which are defined as follows:

1. in: The input number whose security status is being reported.
2. password: The password that secures the input, or 0 if the input is not secured.
3. secure: A flag that is 0 if the input is not secured (password is 0), or 1 if it is secured (password is not 0).

**Command: AR 0,0,3,in1,password1,in2,password2,in3,password3,...**

This form of the command is used to report the security status of individual inputs on each level. Its first two arguments are 0, the third argument is 3, and the remaining arguments are (in, password) pairs, one for each level, that provide the input number and security password on that level. An input number of 0 means that no report is being made on that level.

## **Input Security Discussion**

An attempt to secure an input that is already secured (using the same password it is already secured with), or to unsecure an input that is already unsecured (using any password), succeeds without doing anything. (Older versions of router software, prior to version 10, would generate an error response for this).

An attempt to secure an input that is already secured, using a different password than the one it is already secured with, fails, as does an attempt to unsecure an input using a password that is different from the password that the input was secured with and is not the administrator password. Any use of a password larger than 9999 also fails. In any of those cases, the input security status remains unchanged and if no change was made to any level, an error response is generated. The "AR" command response from the router will indicate that the input still has the same security state as before.

If the input is secured on different levels with different passwords, an attempt to unsecure it using a password that matches the security password of some but not all levels causes the matching levels to be unsecured and the non-matching levels to remain secured as they were before. If the input is secured on some but not all levels, an attempt to secure it causes the unsecured levels to be secured with the specified password and the secured levels to remain secured with the same password as before.

When securing an unsecured input, the specified password is recorded by the router as the security password for that input (on the specified levels). When unsecuring a secured input, the specified password is compared by the router to the security password for the input on each of the specified levels. If they match, or if the specified password is the administrator password, the input is unsecured, else it remains secured. Attempting to secure an already-secured input, or unsecure an already-unsecured input, has no effect on the security state of that input.

Whenever a *take* is done, the router checks to see if the specified input for the *take* has been secured. If so, the current password of the control port that sent the *take* request (as set with the "K" command) is compared to the input's security password. If they match, the *take* is allowed, but if not, the *take* has no effect. The *administrator password may NOT be used in lieu of the input's security password to do a take*, so if a controlling device sends the administrator password in a "K" command, that will not permit the device to reroute secured inputs (but it can use the administrator password to unsecure secured inputs using the "AR" command).

As with all router configuration parameters, the input security data is stored in non-volatile storage and thus is retained across router power-ups. When a router is first initialized at the factory, all inputs are set to be unsecured.

When the router sends a "AR" command response, it uses the first form of the response command if possible (i.e. if the input number in question, and its

security password, is the same on all levels). Otherwise, it uses the second form of the response command. When the second form of the "AR" response command is sent to report changed security passwords, only the security status of the *changed levels* is sent, and the other levels have the (in,password) argument set to (0,0).

## **Types of Input Security**

There are many possible ways to implement input security in a controlling device using the "AR" command. Svs has standardized on three types, which are called:

These are not mutually exclusive. All three could exist in one system. However, most users will choose just one of these schemes, or perhaps one of the two security methods plus the protect method. The schemes are chosen on a controller-by-controller basis. That is, each controlling device can use whichever method(s) of input security it chooses to use.

Note that this discussion parallels the similar discussion for output locks, but there are some substantial differences.

### **Simple Security**

Refer to the discussion of a "Simple Lock" in the discussion of types of output locks for a description of how simple locks work. Simple security works the same way, and uses the same global password of 9999.

*The Simple Security method is unlikely to be used, because when input security is needed, it is normally important to use a password that is restricted to certain users.*

### **Password Security**

When a "Password Secure" button is pushed on a controlling device, the user is prompted to enter a 4-digit password that is not allowed to be 0. Then, the current source is secured with that password and a SECURE indicator would light up. Nobody can route that source. If they try, they will again get an error message or a SECURE indicator will flash or something like that.

To unsecure a password-secured source, you press *any* secure or protect button (any one of the three types). You are then prompted to enter the password for the secured source. You can enter either the correct password that the source was originally secured with, or the administrator password. If you enter either one correctly, the source is then unsecured.

So, a password-secured source is far more solid security than a simple-secured source. You have to know the password to unsecure it. Nobody can make routes to the source while it is secured. *This type of security is expected to be used most often.*

Password-security is the most direct use of the "AR" command. When the user enters the password, that password is sent in a "AR" command to secure or unsecure the source.

### **Protect Security**

Refer to the discussion of a "Protect Lock" in the discussion of types of output locks for a description of how protected locks work. Protect security works the same way.

*Although a site may choose to implement protect-security, it is probably unlikely, because that makes it a bit too easy for the user of the protected panel to route the secure input. Most likely, sites will want to use full password security if they actually make use of input security.*

As with output locks, note that it is relatively simple for a "hacker" to beat the system. By monitoring control panel traffic on a control panel bus, a hacker could obtain the passwords for secured inputs, and this would allow him to route them to any inputs he pleased. Sierra philosophy is to provide a system of "elegant simplicity" that serves basic needs of typical users. To truly protect a secured input, it would be necessary to implement full encryption of commands between the router and controlling devices. This is far beyond the Sierra philosophy.

### **Input Security Examples**

For example, to request whether or not input 21 is secured on all levels:

```
** AR21,0,2 !!
```

The response might be:

```
** AR21,0,0,0 OK !!
```

indicating that input 21 is NOT secured on any level. Or, the response might be:

```
** AR21,6741,1 OK !!
```

indicating that input 21 is secured with password 6741 on all levels. Or, the response might be:

```
** AR0,0,3,21,0,21,0,21,6741,21,6741 OK !!
```

indicating that input 21 is not secured on levels 1 and 2 (password is 0) and is secured with password 6741 on levels 3 and 4. To clear this secure:

```
** AR21,6741,0 !!
```

which unsecures both levels 3 and 4 because they are both secured with the same password, 6741.

If the security status of input 21 changes, the router might send the following:

```
** AR0,0,3,21,1234,21,0,21,6741,21,6741 OK !!
```

indicating that input 21 is secured with password 1234 on level 1, is not secured on level 2 (password is 0), and is secured with password 6741 on levels 3 and 4. To clear this secure:

```
** AR0,0,0,21,1234,0,0,21,6741,21,6741 !!
```

which specifies (0,0) for level 2 since it is already unsecured, and which provides the proper passwords for unsecuring on levels 1, 3 and 4. If instead this command were sent:

```
** AR21,6741,0 !!
```

then only levels 3 and 4 would be unsecured, and level 1 would remain secured with password 1234. The router would send the following to indicate this:

```
** AR0,0,3,21,1234,21,0,21,0,21,0 OK !!
```

To request the security status of input 21 on level 1 and input 44 on level 4:

```
** AR0,0,2,21,0,0,44 !!
```

The router might reply with:

```
** AR0,0,3,21,1234,0,0,0,0,44,9876 OK !!
```

indicating that input 21 on level 1 is secured with password 1234, and input 44 on level 4 is secured with password 9876. To unsecure these inputs, the following could be sent, assuming that the administrator password is 7777:

```
** AR0,0,0,21,7777,0,0,0,0,44,7777 !!
```

## Command Execution Timing

Unless the documentation for a specific switcher specifies otherwise, it can be assumed that the switcher changes its crosspoints between line 6 and line 20 of the field, and furthermore, it is guaranteed that all levels of an output which is being changed by a single host crosspoint command will be changed during the SAME vertical interval.

However, it is not guaranteed that all outputs that are changed by a given command string will go out in the same vertical interval! Whether or not they do depends both on the timing of when the last "!" of the command was received relative to the start of the vertical interval, and upon the number of crosspoint commands in the command string. In the best case, all crosspoint commands are output during the next vertical interval following receipt of the final "!" character. In the worst case, 2 or more vertical intervals could be required to get all the crosspoint data out.

The "P" and "T" commands are similar to the regular crosspoint commands, in that all levels of each output setting in the "P" command are guaranteed to be sent to the crosspoint hardware in a single vertical interval, but the entire string of commands making up the "P" command may not all make it out within one vertical interval.

The "P" and "T" commands are useful because it may be desirable to make numerous connections all at one vertical sync interval, yet the command string for doing this may be too long to process in a single vertical interval. The "T" command is optimized for fast processing time, and is therefore preferred over individual crosspoint connect commands when time is of the essence.

The number of connect commands that a "P" command can have within it and still allow the "T" command to be processed within a single vertical interval

depends on the particular switcher, so refer to the switcher documentation for this information.

For best results, always send the final "!" so it will be received at the start of the vertical interval, to give the software the entire interval in which to process the command string. Another way of doing this is to precede the crosspoint connect command(s) with a "D" command, which will cause a delay to the start of some vertical interval before processing the rest of the command string.

### **Optional Commas**

The comma characters within the connect commands **V, W, X, Y, and Z** are optional and may be left out as long as the maximum required number of digits are supplied for each number. The "L" command is used to obtain this number. If the "L" command indicates that there are 120 outputs, then 3 digits must be used to specify the output number if the comma following the output number is to be left out. Likewise, if the number of inputs is 32, then 2 digits must be used to specify the input number if the comma following the input number is to be left out. Or, if the number of levels were 10, then 2 digits must be used to specify the level number if the comma following the level number is to be left out.

For example, if the "L" command response is:

```
L8,1,32,VIDEO~~
```

then the following "X" command could be used, with no commas:

```
**X6031!!
```

which is treated the same as "X6,3,1", requesting a connection from input 3 to output 6 on level 1.

When the number in question is the *last* number in the command (i.e. it is not followed by a comma ever), then it doesn't matter how many digits are used for it. For example, in the "Y" command the input number is last, so any number of digits, including just one digit, can be used for the input number.

### **Aggregated Final Command Example**

Most of the above examples showed only one command per command string, but multiple commands can be included in a single command string and spaces may be included for better readability.

For example:

```
** CLEAR X1,13,2 L S !!
```

This requests that the matrix be cleared, then a connection be made, and finally that matrix size and status information be returned to the host. The returned response string might look like this:

```
** L1,1,16,1~~ X1,13,1 OK !!
```

indicating that the router has one output, 16 inputs, one level named "1", and currently has its output connected to input 13.

### **AUDINP\_GAINS: Audio Input Gain Adjustment**

Vaname: AUDINP\_GAINS

Use: Modify and query input audio gain values.

Routers: SPII, VS

Syntax: G VAR1,AUDINP\_GAINS,input\_num [,value] ~

Values:

The 'index' field is the input number, which can range from 1 to 64.

The 'value' field is the audio gain adjustment. This number represents an index into an array of gain values. The index ranges from 0 which represents the most negative gain value to 41 which represents the most positive gain value. A value of 20 will set the input gain to unity.

The actual table of values reads as follows:

<b>Index</b>	<b>Gain Value</b>	<b>Increment</b>
0 to 19	-10.0 to -0.5	one half db steps
20	0 (Unity gain)	
21 to 41	+0.5 to +20.5	one db steps

Example:

```
** G VAR1, AUDINP_GAINS, 4, 20~!!
```

Set input 4 to unity gain.

### **AUDOUT\_GAINS: Audio Output Gain Adjustment**

Varname: AUDOUT\_GAINS

Use: Modify and query output audio gain values for SPII and VS type routers.

Syntax: G VAR1,AUDOUT\_GAINS,output\_num [,value] ~

Values:

The 'index' field is the output number, which can range from 1 to 64.

The 'value' field is the audio gain adjustment. This number represents an index into an array of gain values. The index ranges from 0 which represents the most positive gain value to 77 which represents the most negative gain value and 78 which represents "mute" or no audio. A value of 30 will set the output gain to unity.

The actual table of values reads as follows:

<b>Index</b>	<b>Gain Value</b>	<b>Increment</b>
0 to 29	+15.0 to +0.5	one half db steps
30	0 (Unity gain)	
31 to 64	-0.5 to -33.5	one db steps
65 to 77	-35.5 to -59.5	two db steps
78	"mute"	

Example:

```
** G VAR1, AUDOUT_GAINS, 8, 78~!!
```

Set output 8 to "mute".

```
** G VAR1, AUDOUT_GAINS, 1, 0~!!
```

Set output 1 to the highest positive level: +15db.

## **MUTE\_DELAYS: Mute Delay Adjustment**

Varname: MUTE\_DELAYS

Use: Modify and query output mute delay values.

Routers: SPII

Syntax: G VAR1,MUTE\_DELAYS,output\_num [,value] ~

Values:

The 'index' field is the output number, which can range from 1 to 64.

The 'value' field is the amount of mute delay time in one half second increments. Each half second increment is represented by a value that is actually 10 times the delay desired. Thus, a delay of 4.5 seconds is represented by the value 45.

Example:

```
** G VAR1, MUTE_DELAYS, 5, 100~!!
```

Set the amount of time to mute the signal before turning it on to the maximum value of 10 seconds.

```
** G VAR1, MUTE_DELAYS, 6, 35~!!
```

Set the delay time to 3.5 seconds.

Salvos via Serial Commands

### **“P”: Preset a Salvo**

The command "P" is used to set up a *salvo*, which is a series of connect commands for later execution with the "T" command. It must be followed by a register letter from A to Z or a register number from 1 to 256 giving the *register* into which the connect sequence is to be stored, followed by zero or more *connect commands* (V, W, X, Y, or Z), followed by a "~" (tilde) character.

Registers A-Z are the same registers as 1-26. The letter designators are allowed to shorten up the command sequence slightly, so that "T" commands can be sent in compact form if one of the first 26 registers is used.

For example, the command:

```
** PB X2,5,0 Z7 ~ !!
```

says that two connect commands (output 2 to input 5 AFV, and output 1 to input 7 AFV) are to be stored into salvo register B (i.e. register 2). Another example:

```
** P239 Y5,7 ~ !!
```

says that one connect command (output 5 to input 7 AFV) is to be stored in salvo register 239.

Only *supported connect commands* (those connect commands whose command letters appear in the "I" command response) may follow the register letter up to the tilde character. The connect commands do not take effect until the register is *triggered* using the "T" command below.

The maximum allowed number of connect commands is determined by the particular router. In all cases where this command is implemented, there is guaranteed to be space available to store at least two complete switch matrices AT ONE LEVEL. If more connect commands are received than there is space available to store them, the error response string "FULL ERROR Salvo Space Full" is sent to the host, and only the first part of the salvo is stored. For example:

```
** FULL ERROR Salvo Space Full !!
```

says that the salvo request filled memory and the salvo could not be completely stored.

### **PRESET\_NAMES: Modify or query preset (aka salvo) names**

Action	From	Syntax
Modify	Host	G PRESET_NAMES,<register>,<preset_name_version>,<name1>~,<name2>~ ~~
Query	Host	G PRESET_NAMES,<register>~
Query Response	Route r	(same as Modify)

Modify is Protected:	No	
Report Changes:	Yes	

The "G PRESET\_NAMES" command is used to modify or query for preset (aka salvo) names.

The Modify command has four arguments:

<register> indicates which preset register is to be named, 1-256 or A-Z.

<preset\_name\_version> is the preset name version number, or 0 to increment the existing version number.

<name1> is the first preset name.

<name2> is the second preset name.

The Query command has one argument: the number of the preset in question.

The Query Response command has the same form as the Modify command.

Refer to the "G INPUT\_NAMES" command for information about names in general.

Refer to the description of the "G VERSION\_NUMBERS" command for a discussion of the purpose and use of version numbers and the <preset\_name\_version> argument.

For example, to request the current names for preset 34:

```
** G PRESET_NAMES,34~ !!
```

If the preset name version number was 154 and the preset name were "STDO A", the response command would be:

```
** G PRESET_NAMES,34,154,~~,~STDO A~~ !!
```

To change the name of preset 34 to "StdioB" and increment the preset name version number:

```
** G PRESET_NAMES,34,0,~,~StdioB~~ !!
```

or this with version 8.03 or later of the router software:

```
** G PRESET_NAMES,34,0,~StdioB~~ !!
```