Savant® SmartAudio/SmartLink API

SSA-3220/SSA-3220D/SLN-84BT/SLN-86BT/SLN-88BT

Application Note

Minimum Release: da Vinci 5.2.3
Document Release Date: January 2014
Part Number: 009-1052-02

45 Perseverance Way, Hyannis, MA 02601 ▼ P: 508-683-2500 ▼ F: 508-683-2600 ▼ www.SavantSystems.com

Contents

To access the link to the topics in this document, click the topic text or topic page.

Overview

Savant® SmartAudio 20 (SSA-3220/SSA-3220D) and SmartLink (SLN-88BT, SLN-86BT, or SLN-84BT) products can be monitored and configured using a set of proprietary API messages sent. This application note describes the commands available to configure the various inputs and outputs located on the SmartAudio and SmartLink products. Commands are sent to the Savant device using either IP or an RS-232 serial port. The functionality described below is supported starting with the Release da Vinci 5.2.3.

Minimum Requirements

The following are required to monitor and configure the SmartAudio/SmartLink products through a Telnet session

Requirement	Description
PC or Mac® with either Telnet or RS-232 functionality.	Any home or business PC or Mac connected to the network that has Telnet or RS-232 functionality.

Telnet

Once the network has been connected and the PC is ready to communicate with the SmartAudio/SmartLink product, open a Telnet session and connect to port 8085 of the IP address of the switch. Refer to the example below.

telnet <IP address> 8085

IMPORTANT! A carriage return and linefeed (Hex: 0x0d0a) must be sent with the API command in order for the Savant® device to know that the command has been terminated. Most terminal applications will automatically include a carriage return and linefeed when the <enter> or <return> keys are pressed.

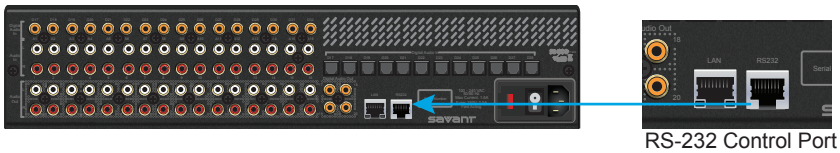
If you are using rpmTerminal in mixed mode, use \h0D\h0A as the carriage return and linefeed.
Example: switch-set1:1\h0D\h0A

RS-232 Port

RS-232 Port Location

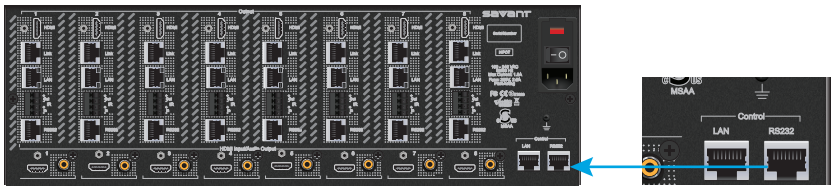
The RS-232 control port located in the lower right corner on the rear of the unit is used to communicate with the matrix via a computer or 3rd party control system.

SmartAudio



RS-232 Control Port

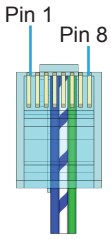
SmartLink



RS-232 Control Port

RS-232 Port Wiring

Wiring for this port follows the Savant standard where pin 5 is RXD (Receive), pin 6 is TXD (Transmit), and pin 4 is GND (Ground). The diagram below shows the wiring for the Savant side, the opposite side pins will vary based on the device being connected to.



RJ45 Connector
(Gold Pins Facing Up)

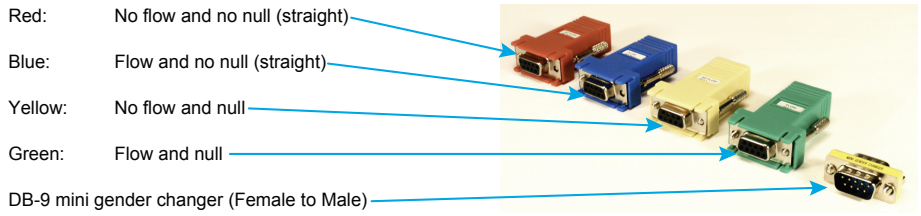
Pin 1:	(Not Used)	**Pin 5:**	**RXD (Receive)**
Pin 2:	(Not Used)	**Pin 6:**	**TXD (Transmit)**
Pin 3:	(Not Used)	Pin 7:	(Not Used)
Pin 4:	**GND**	Pin 8:	(Not Used)

⚠ ***IMPORTANT!*** *When wiring to the RS232 port,* ***DO NOT*** *connect any wires within the cable that are not required for communication.*

⚠ ***IMPORTANT!*** *When using an RJ45 to DB9 adapter, the cable from the adapter to the SmartLink/SmartAudio should be a straight through cable. Any required crossover should be achieved by using a Null adapter.*

RJ45 to DB9 Adapters

Savant uses RJ45 connectors for RS-232, other manufacturers control systems may use the standard DB9. To make connection easy, Savant offers RJ45 to DB9 adapters in a variety of configurations that can be used to connect to SmartLink/SmartAudio for RS-232 control. Be sure and choose the adapter that provides a proper connection to the computers, or control systems RS-232 port.



⚠ ***IMPORTANT!*** If you are using RJ-45 to DB-9 adapters not supplied by Savant:

- Ensure that any wires required for communication/control are terminated within the adapter.
- Ensure that all wires ***NOT*** required for communication/control are ***NOT*** terminated in the connecter.
- Ensure that the unused wires in the connector are cut to prevent them shorting out, as they are still terminated in the RJ-45 connector on the controller side

RS-232 Port Settings

Terminal communications specifications	
Baud	19200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

Glossary

Term	Description
Connection	Physical wiring between two devices.
Connector	The end of the cable that connects into a port.
Display Resolution	Number of pixels that can be displayed in a vertical and horizontal dimension. Usually displayed as width by height, for example, 1920 x 1080.
EDID	Extended display identification data. An EDID is information that resides in a data display device. These EDID's are read from a device such as the SmartLink. Once read, the SmartLink can enhance the video and audio characteristics of the signal being sent.
Port	A port will have either holes or a slot that matches the connector being connected into the port.
Refresh Rate	Number of times in one second a display will redraw the data.
ainput	In the API commands, ainput refers to an audio input. Ainput refers to any port designated as an audio input.
aoutput	In the API commands, aoutput refers to an audio output. Aoutput refers to any port designated as an audio output.
vinput	In the API commands, vinput refers to a video input. Vinput refers to any port designated as a video input.
voutput	In the API commands, voutput refers to a video output. Voutput refers to any port designated as a video output.
processed	Data that has been manipulated.
passthru	A signal that passes through the switch without the ability to alter the volume level is termed a passthru signal. This term is used in the aoutput commands.
coaxial	A coaxial cable has an inner conductor surrounded by a shield. On the SmartAudio and SmartVideo products, the coaxial cables require audio cables capable of sending digital audio formats. These cables are terminated with an RCA type connector.
toslink	Toshiba Link which is a standard fiber optic system for transmitting audio signals.

API Command Set - SmartAudio

The SmartAudio (SSA-3220/SSA-3220D) support a variety of proprietary API commands. Refer to the tables for syntax and descriptions of each command.

Error Messages

The table below displays any error messages returned because of an error made during the entering or execution of a command.

Error	Description
err1	An unknown command was sent. Sending incorrect syntax would be an example of this.
err2	A bad parameter was sent. A variable out of range would be an example of a bad parameter.
err3	Switch is in maintenance mode (firmware upgrade in process). Error message is received if API is sent to switch during the upgrade process.
err4	When a command is sent and there is a bad response due to a hardware failure, err4 is returned.

Audio switch-set and switch-get

The table below displays information regarding the **switch-set** and **switch-get** commands for monitoring and configuring the SmartAudio switch. The switch-set command connects a port from the output side to a port on the input side. This connection will then allow audio to be streamed through the switch. The *switch-set* command can also be used to disconnect two previously connected ports. The *switch-get* command displays the connection status of each individual output port.

For the examples below:
- The variable **x** refers to an input port. (1 - 32).
- The variable **y** refers an output port (1 - 20).

API Format	Example API	Returned API	Description
switch-set**y**.**x**	switch-set**17**.**15**	switch17.15	Connect output 17 to input 15. Audio streaming into input 15 will be sent to output 17. The returned API verifies the connection has been made.
switch-set**y**.**disconnect**	switch-set**17.disconnect**	switch17.0	Disconnect output 17 from the input it is currently connected to. The returned API verifies there is no connection to output 17.
switch-get**y**	switch-get**17**	switch17.0	Display the connection status of output 17. In this example, output 17 is not connected to any of the inputs. This is displayed in the returned API.

Input and Output Configuration Setup

The table below displays information regarding the **aoutput-conf-set** and **ainput-conf-set** commands for configuring the type of input and/or output connection that will be employed. The *aoutput-conf-set* command configures specific parameters to each of the supported output connections. The *ainput-conf-set* command configures specific parameters to each of the supported digital input connections.

For the examples below:
- The variable **x** refers to the digital input ports 17 - 28.
- The variable **y** refers to any output port 1 - 20.

API Format	Example API	Returned API	Description
ainput-conf-set**x**:**coaxial**	ainput-conf-set**20**:**coaxial**	ainput-conf20:coaxial	Set input 20 to accept signaling from a coaxial cable. The returned API confirms that input 20 is set to the coaxial port.
ainput-conf-set**x**:**toslink**	ainput-conf-set**22**:**toslink**	ainput-conf22:toslink	Set input 22 to accept signaling from a toslink optical fiber cable. The returned API confirms input 22 is set to the toslink port.
aoutput-conf-set**y**:**processed**	aoutput-conf-set**18**:**processed**	aoutput-conf18:processed	Audio streaming to output 18 will be processed when passing audio from any configured input to output 18. Once an output port is set to ***processed***, the volume, mute and PEQ commands can be set. The returned API displays the status after being configured. **Note:** By default, all analog output ports are initially set to processed and all digital output ports are set to pass-thru prior to shipment.
aoutput-conf-set**y**:**passthru**	aoutput-conf-set**19**:**passthru**	aoutput-conf19:passthru	Audio streaming to output 19 will be unaltered when passing through the switch (volume level = 0 dB). The returned API displays the status after being configured. See note above.

Input and Output Configuration Status

The table below displays information regarding the **aoutput-conf-get** and **ainput-conf-get** commands for obtaining the configuration state of a specific input or output port. The *aoutput-conf-get* displays the configuration state of any specified output port, whereas the *ainput-conf-get* commands only display the configuration state of certain digital input ports.

For the examples below:
- The variable **x** refers to the digital input ports 17 - 28 (S/PDIF inputs).
- The variable **y** refers to any output port 1 - 20.

API Format	Example API	Returned API	Description
ainput-conf-get**x**	ainput-conf-get**28**	ainput-conf28:coaxial	Display the configuration state of input 28. In this example, input 28 is configured for a coaxial connection. This is displayed in the returned API.
ainput-conf-get**x**	ainput-conf-get**22**	ainput-conf22:toslink	Display the configuration state of input 22. In this example, input 22 is configured for a toslink connection. This is displayed in the returned API.
aoutput-conf-get**y**	aoutput-conf-get**20**	aoutput-conf20:passthru	Display the configuration state of output 20. In this example, output 20 is configured for a passthru connection. This is displayed in the returned API.

Audio *Input and Output Control Commands*

The tables below display information regarding the API commands required to set and get audio levels on both the input and output ports. Refer to table below.

For the examples below:
- The variable **x** refers to any input port (1 - 32)(S/PDIF inputs).
- The variable **y** refers to any output port (1 - 20).
- The variable **z** refers to the digital input ports (17-32).

API Format	Example API	Returned API	Description
aoutput-vol-set**y:dB**	aoutput-vol-set**10**:**-10**	aoutput-vol10:-10dB	Set the volume level on a specified output port. To take affect, the output must be configured as processed and the volume range is -117dB to +10dB. In the example, the volume on output 10 is set to -10dB.
aoutput-vol-get**y**	aoutput-vol-get**10**	aoutput-vol10:0dB	Displays the volume level state on a specific output port. In the example, the volume state of output port 10 is line-level 0 dB.
aoutput-mute-set**y:on/off**	aoutput-mute-set**15**:**on**	aoutput-mute15:on	Configures an output to either mute or unmute the audio signal being transmitted out a specific output. In the example, the audio being transmitted out port 15 is configured to be muted.
aoutput-mute-get**y**	aoutput-mute-**get15**	aoutput-mute15:on	Displays whether a specific port is set to mute or unmute. In the example, the audio signal being transmitted through output port 15 is muted.
aoutput-mono-set**y:on/off**	aoutput-mono-set**16**:**on**	aoutput-mono16:on	Configures an output to either transmit the received audio signal as mono or stereo. In the example, output port 16 is being configured to transmit the received signal in mono. **Note:** *off* would indicate signal is configured to be transmitted in stereo.
aoutput-mono-get**y**	aoutput-mono-get**17**	aoutput-mono17:on	Displays whether a specific port is set to transmit a received audio signal in mono or stereo. In the example, output port 17 is set to transmit out the received audio signal in mono (on). If signal is being transmitted in stereo, (off) would be displayed.
ainput-trim-set**x:dB**	ainput-trim-set**5**:**-10**	ainput-trim5:-10dB	Set the trim on the input port to even out the input signals. In the example, the trim has been set to -10 dB below line level (0dB).
ainput-trim-get**x**	ainput-trim-get**2**	ainput-trim2:-10dB	Displays the configured trim value in dB that the input is set for. In the example, input 2 is configured to trim the received incoming audio signal by 10 dB.
ainput-type-get**z**	ainput-type-get**18**	ainput-type18:encoded ainput-valid18:yes	Displays whether the received audio signal is either encoded or non-encoded PCM data. In this example, the received audio signal on input 18 has encoded PCM data. If the signal received was a non-encoded signal, then simply PCM would be returned. If cable is connected, ainput-valid18:yes is returned.
SSA-3220D Channel Output Delay (These commands are only available on the SSA-3220D)			
aoutput-delayleft-set**y:ms**	aoutput-delayleft-set**10:80**	aoutput-delayleft**10:80ms**	Sets the delay of a left analog output between 0 and 85ms. Example shows setting the delay of the left channel on output 10 to 80ms.

aoutput-delayright-set**y:ms**	aoutput-delayright-set**10:60**	aoutput-delayright**10:60ms**	Sets the delay of a right analog output between 0 and 85ms. Example shows setting the delay of the right analog channel on output 10 to 80ms.
aoutput-delayboth-set**y:ms**	aoutput-delayboth-set**10:10**	aoutput-delayleft**10:10ms** aoutput-delayright**10:10ms**	Sets the delay of a right analog channel between 0 and 85ms. Example shows setting the delay of t both left and right analog output 10 to 10ms.

Audio PEQ Processor API

The PEQ processor is a seven-band equalizer (EQ) which can be configured to set the frequency, the level, and the q of each band. Refer to the table below to configure the PEQ Processor.

Notes:

- Before setting the parameters on the PEQ Processor, the output ports must first be configured to **processed** which is achieved through the aoutput-conf-set command.
- The peqset command (peqset-outputy-bandz:freq125-level5-q1.414) must first be sent to each individual band (1-7) before the peqfreqset, peqlevelset, or peqqset command can be sent. For example, the peqset command must be sent to band 3 prior to sending the peqfreqset, peqlevelset, or peqqset to band 3.

Each of the API messages below configures a set of parameters to the PEQ processor for each output.

- The variable **y** refers any output port (1-20).
- The variable **z** refers to one of 7 bands. The bands are numbered 1-7.
- The **frequency** is entered as an integer. Each band is pre-configured to a frequency.
- The **level** variable is entered in dB. A range from -12 to +12 dB is supported.
- The **q** or bandwidth is the roll off of the affected frequency. A high q affects a very small range. A low q affects a wider range. The q control on the SmartAudio system has a range from .4041 to 7.2077. In terms of bandwidth, the supported range is from .2 to 3 octaves.

API Format	Example API	Returned API	Description
peqset-output**y**-band**z**:freq**125**-level**5**-q**1.414**	peqset-output**2**-band**2**:freq**125**-level**5**-q**1.414**	peqset-output2-band2:freq125Hz-level5db-q1.414	In the Example API, set band 2 of the equalizer on output port 2 to a frequency of 125 Hz, dB level of 5, and q value of 1.414.
peqfreqset-output**y**-band**z**:**125**	peqfreqset-output**2**-band**2**:**125**	peqfreqset-output2-band2:125Hz	In the Example API, set band 2 of the equalizer on output port 2 to 125 Hz.
peqlevelset-output**y**-band**z**:**-10**	peqlevelset-output**2**-band**2**:**-10**	peqlevelset-output2-band2:-10dB	In the Example API, set band 2 of the equalizer on output port 2 to -10 dB.
peqqset-output**y**-band**z**:**1**	peqqset-output**2**-band**2**:**1**	peqqset-output2-band2:1	In the Example API, set band 2 of the equalizer on output port 2 to a q value of 1.
peqget-output**y**-band**z**	peqget-output**2**-band**2**	peqget-output**y**-band**z**:freq125Hz-level-10db-q1	In the Example API, get the PEQ status of band 2 of the equalizer on output port 2.

Maintenance commands

Maintenance commands are used to upgrade firmware, display certain information, or reboot system. Refer to the commands and descriptions below for information on the maintenance commands.

IMPORTANT! Prior to da Vinci 5.2.3 matrix profiles in Blueprint did not have a UID entry field. Any chassis ,and host running da Vinci 5.2.3, or higher with a UID assigned in Blueprint do not require this procedure as firmware and FPGA is updated automatically.

API Format	Example API	Returned API	Description
ipaddress-get	ipaddress-get	ipaddress:<xxx.xxx.xxx.xxx>	Displays the ip address of the device.
reboot	reboot	N/A	Reboots device. No API is returned since the TCP connection is lost when entering the reboot command
fwrev	fwrev	fwrevPrimary;<firmware revision>	Displays the current revision of active running firmware. **Note:** The response fwrevBackup;<firmware revision> is only displayed if there is an issue running the primary version.
upgrade:**<ip address>;<filename.bin>**	upgrade:**10.10.10.10;filename.bin**	upgrade:20%;Continue upgrade:100%;Success	The upgrade command is used to up-rev the current revision of firmware. When upgrading firmware, add the ip address of the TFTP server and bin file being upgraded.
fpga-rev		fpga-rev<fpga revision>	Displays the current revision of FPGA code loaded.
fpga-upgrade:**<ip address>;<filename.mcs>**	fpga-upgrade:**10.10.10.10;filename.mcs**	fpga-upgrade:20%;Continue fpga-upgrade:100%;Success	The fpga-upgrade command is used to up-rev the current revision of FPGA code. When upgrading FPGA, add the ip address of the TFTP server and file being upgraded.

API Command Set - SmartLink

Monitoring and configuring the SmartLink products using a set of proprietary API commands is described below. The API commands in the tables below support the SmartLink products. Refer to the tables for syntax and descriptions of each command.

Error Messages

The table below displays any error messages that will appear because of an error made during the entering or execution of a command. Refer to the table below.

Error	Description
err1	An unknown command was sent. Sending incorrect syntax would be an example of this.
err2	A bad parameter was sent. A variable out of range would be an example of a bad parameter.
err3	Switch is in maintenance mode (FPGA upgrade in process). Error message is received if API is sent to switch during the upgrade process.
err4	When a command is sent and there is a bad response due to a hardware failure, err4 is returned.

Video switch-set and switch-get

The table below displays information regarding the **switch-set** and **switch-get** commands for monitoring and configuring the SmartLink products. The *switch-set* command connects a port from the output side to a port on the input side. This connection will then allow video to be streamed through the switch. The *switch-set* command can also be used to disconnect two previously connected ports. The *switch-get* command displays the connection status of each individual output port. Refer to table below.

For the examples below:
- The variable **x** refers to the input HDMI ports 1 - 8 located at the bottom of the module.
- The variable **y** refers to the HDBaseT or HDMI output ports 1 - 8 on the SLN-88BT, ports 1-6 on the SLN-86BT, and ports 1-4 on the SLN-84BT.
- Output ports are the Link or HDMI ports at the top of the module.

API Format	Example API	Returned API	Description
switch-set**y**.**x**	switch-set**1**.**2**	*switch1.2*	Connect output 1 to input 2. Video streaming into input 1 will be sent to output 2. The returned API verifies the connection has been made.
switch-set**y**.**disconnect**	switch-set**1**.**disconnect**	*switch1.0*	Disconnect output 1 from the input it is currently connected to. The returned API verifies there is no connection to output 1.
switch-get**y**	switch-get**4**	*switch4.0*	Display the connection status of output 4. In this example, output 4 is not connected to any of the inputs as displayed in the returned API.

Output configuration and status command

The table below displays information on the **voutput-get**, **voutput-power-set**, and **voutput-power-get** commands. The *voutput-get* command displays the mode, cable length, and signal health of the port. The *voutput-power-set* command can add or remove the DC power being sent over the HDBaseT signal. The *voutput-power-get* command will display whether the HDBaseT signaling is configured with or without the DC power.

For the examples below:
- The variable **y** refers to the *HDBaseT* output ports 1 - 8 on the SLN-88BT, ports 1-6 on the SLN-86BT, and on ports 1-4 on the SLN-84BT.

API Format	Example API	Returned API	Description
voutput-get**y**	voutput-get**5**	voutput5-mode:HDBaseT voutput5-cablelen:30m voutput5-qual:-10dB -11dB	Display the output mode, cable length (meters), and link strength in dB. **Note:** Any length of cable below 30 meters is displayed as 0
voutput-power-set**y:on/off**	voutput-power-set**5:on**	voutput-power5:on	Configure whether DC power will be added or removed from the HDBaseT signal on the output of the port configured. In the example, DC power is included in the HDBaseT signal. **Note:** By default, voutput-power is set to on.
voutput-power-get**y**	voutput-power-get**5**	voutput-power5:on	Displays if DC power is configured on the output. In the example, DC power is configured to ride on the HDBaseT signal.

Input status commands

The table below displays information on the **vinput-get**, and **ainput-get** commands. These commands get the status of the input ports. Refer to table below.

For the examples below:
- The variable **x** refers to the *HDBaseT* input ports 1 - 8.

API Format	Example API	Returned API	Description
vinput-get**x**	vinput-get**5**	vinput5:1920x1080p@60Hz vinput5: Not Locked	Displays the resolution and frequency of the video signal received at a specified input port. If the video signal is not synced, **Not Locked** will be displayed.
ainput-get**x**	ainput-get**6**	ainput6:None ainput6:PCM ainput6:Encoded ainput6:HBR	Displays the specifications of the audio signaling received at a specified input port. The *Returned API* column displays each of the possible values.

Route command

The **route** command displays the connection status of each output port. When the route command is sent, the returned API message will display which zones or inputs are connected to the output.

For the examples below, the following is true:

- The command **route0.0.0.0.0.0.0.0** reads the connection status of each output port. The sent API reads the current connection status of ports 1-8. The returned API displays the input port connected to the queried output port. Refer to examples below.
- Adding the variable **d** to the API command sent, changes the command from a status message to a control message and disconnects the specified output port.

API Format	Example API	Returned API	Description
route**0.0.0.0.0.0.0.0**	route**0.0.0.0.0.0.0.0**	route0.1.0.0.0.0.2.0	The route0.0.0.0.0.0.0.0 command queries the status of each of the output ports. In this example, the returned value displays that input port 1 is connected to output port 2 and input port 2 is connected to output port 7.
route**0.0.0.0.0.0.0.0**	route**0.0.0.0.0.0.0.0**	route1.0.0.0.0.0.0.0	In the example to the left, input 1 is connected to output 1. The rest of the inputs are not connected.
route**0.0.0.0.0.0.0.0**	route**0.0.0.0.0.0.0.0**	route1.1.1.1.1.1.1.1	In the example to the left, input 1 is connected to all 8 outputs. The information on input 1 is transferred to all outputs.
route**0.0.0.0.0.0.0.d**	route**0.0.0.0.0.0.0.d**	route1.1.1.1.1.1.1.0	In the example to the left, the route command is used as a control message rather than just a status message. Initially all 8 output ports are connected to input 1. The route command sent, disconnects output port 8 from input port 1. The returned API displays that all output ports are still connected to input port 1 except port 8.

EDID read

The EDID is structured in either 128 or 256 bytes of information. The RAW EDID information on either an output port or an input port can be displayed. The edid-readi command reads the EDID information on the input port entered. The **edid-reado** command reads the EDID information on the output port entered.

In the examples below:

- The variable **x** refers to the HDBaseT input ports 1 - 8 on the SLN-88BT, ports 1-6 on the SLN-86BT, and on ports 1-4 on the SLN-84BT.
- The variable **y** refers to the HDBaseT output ports 1 - 8 on the SLN-88BT, ports 1-6 on the SLN-86BT, and on ports 1-4 on the SLN-84BT.

API Format	Example API	Returned API	Description
edid-readi**x**	edid-readi**1**	edid-readibase:<EDID information> edid-readiext:<EDID information>	Returned API displays the EDID information on the input port.
edid-reado**y**	edid-reado**1**	edid-readobase:<EDID information> edid-readoext:<EDID information>	Returned API displays the EDID information on the output port.

NOTE: If there is no EDID information available, the responses will come back blank.

EDID mode

The EDID mode on the input ports can be configured as well as read. The two commands are the **edid-mode-setx:value** and **edid-mode-getx**. The table below describes what is required to either set or get the edid mode on each of the input ports.

For the examples below, the EDID mode can be configured to one of the values displayed below:

- The variable **x** refers to the HDBaseT input ports 1 - 8 on the SLN-88BT, ports 1-6 on the SLN-86BT, and on ports 1-4 on the SLN-84BT.
- The variable **mode** designates one of the following values displayed below.

  1 = Merge Mode: Merge the EDID's for all outputs connected to this input (Default Option).

  2 = Fixed Mode: Universal EDID - The EDID for this input reports all capabilities.

  3 = Fixed Mode: Universal-PCM EDID - Same as Fixed Mode: Universal but without encoded audio.

  4 = Fixed Mode: 50Hz EDID - Same as Fixed Mode: Universal but for 50 Hz only.

  5 = Fixed Mode: 50Hz-PCM EDID - Same as Fixed Mode: Universal for 50 Hz but without encoded audio.

  6 = Fixed Mode: 60Hz EDID - Same as Fixed Mode: Universal but for 60 Hz only.

  7 = Fixed Mode: 60Hz-PCM EDID - Same as Fixed Mode: Universal for 60 Hz but without encoded audio.

API Format	Example API	Returned API	Description
edid-mode-set**x:mode**	edid-mode-set**2:3**	edid-mode**2:3**	The edid-mode-set command configures an EDID mode to a specific input port. In the example, input port 2 is configured to Fixed Mode:Universal-PCM EDID.
edid-mode-get**x**	edid-mode-get**2**	edid-mode**2:3**	The edid-mode-get command reads and outputs the EDID mode configured on a specific input port. In the example, input port 2 is read and the Returned API displays the port is configured to Fixed Mode:Universal-PCM EDID

Maintenance commands

Maintenance commands are commands that can upgrade firmware, display certain information, or reboot system. Refer to the commands and descriptions below for information on the maintenance commands.

Firmware and FPGA revision and upgrade commands

The commands below describe how to display and upgrade both the firmware and FPGA code.

⚠ ***IMPORTANT!*** Prior to da Vinci 5.2.3 matrix profiles in Blueprint did not have a UID entry field. Any chassis ,and host running da Vinci 5.2.3, or higher with a UID assigned in Blueprint do not require this procedure as firmware and FPGA is updated automatically.

API Format	Example API	Returned API	Description
fwrev	fwrev	fwrevPrimary; <firmware revision> fwrevBackup; <firmware revision>	Displays the current revision of firmware loaded. Note: The response fwrevBackup;<firmware revision> is only displayed if the system is set to debug mode.
upgrade:<**ip address**>;<f**ilename.bin**>	upgrade: **10.10.10.10;firmware.bin**	upgrade:20%;Continue upgrade:100%;Success	The upgrade command is used to up-rev the current revision of firmware. When upgrading firmware, add the ip address of the TFTP server and bin file being upgraded.

fpga-rev	fpga-rev	fpga-rev<fpga revision>	Displays the current revision of FPGA code loaded.
fpga-upgrade:<**ip address**>;<**filename.mcs**>	upgrade: **10.10.10.10;fpga.mcs**	fpga-upgrade: 20%;Continue .... fpga-upgrade: 100%;Success	The fpga-upgrade command is used to up-rev the current revision of FPGA code. When upgrading FPGA, add the ip address of the TFTP server and file being upgraded.

HDBaseT Revision

The HDBaseT revision and upgrade commands below display the HDBaseT firmware revisions as well as the upgrade commands required to up-rev the HDBaseT firmware code. As displayed in the first row below, the HDBaseT is displayed for all eight video ports. The upgrade command up-revs each individual HDBaseT port through use of a TFTP server. Refer to the table below for information on each.

In the examples below:

- The variable y refers to the HDBaseT output ports 1 - 8 on the SLN-88BT, ports 1-6 on the SLN-86BT, and on ports 1-4 on the SLN-84BT.

API Format	Example API	Returned API	Description
hdbaset-rev	hdbaset-rev	hdbaset-rev1.30.52.010; 1.30.52.010; 1.30.52.010; 1.30.52.010; 1.30.52.010; 1.30.52.010; 1.30.52.010	Displays the revision of each individual video output port separated by a semi-colon. These ports are labeled as *link*. A failed port will report the revision as **Unkown**.
hdbaset-upgrade**y**:<**ip address**>;<**filename.bin**>	hdbaset-upgrade**y**:<**10.10.10.10**>;<**firmware.bin**>	hdbaset-upgrade: 20%;Continue .... hdbaset-upgrade: 100%;Success	The hdbaset-upgrade command is used to up-rev the firmware for the HDBaseT chipset. The command requires the port number, ip address of the TFTP server, and name of the bin file.
hdbaset-remote-upgrade**y**:<**ip address**>;<**filename.bin**>	hdbaset-remote-upgrade**y**: **10.10.10.10;<firmwame.bin>**	hdbaset-remote-upgrade: 20%;Continue .... hdbaset-remote-upgrade: 100%;Success	The hdbaset-remote-upgrade command is used to up-rev the HDBaseT receiver firmware. The command requires the port number, ip address of the TFTP server, and name of the bin file.

Status

The status command displays firmware and hardware information.

API Format	Example API	Returned API	Description
status	status	statusAPI1.0; Standalone-Video-Switch; rev0; pn068-1111-00; ready=yes	Displays the API revision; board type; part number; ready status (yes or no); and optional failure string. (Failure string is displayed in row below)
		statusAPI1.0; Standalone-Video-Switch; rev0; pn068-1111-00; ready=no status-failure15 IDEEPROM len error	In example, a failure occurred in the id of the EEPROM.

Important Notice

Disclaimer

The contents of this document are subject to change without notice; therefore, the information presented herein shall not be construed as a commitment or warranty.

Savant Systems, LLC shall not be liable for any technical or editorial errors or omissions contained herein or for incidental or consequential damages resulting from the performance, furnishing, reliance on, or use of this material.

Patents

Certain equipment and software described in this document is protected by issued and pending U.S. and foreign patents.

All products and services are trademarks or registered trademarks of their respective manufacturer.

Copyright

Trademarks

Technical Support

Savant Systems, LLC is dedicated to providing prompt and effective support.

To do this in a timely and efficient manner:

1. Access the Savant Portal: *dealers.SavantSystems.com*
2. Enter **User Name** and **Password** and click **Login**.
3. The Support Center page appears.
4. Select **Support Request** box.
5. The **Submit a Ticket** window appears.
6. Complete the information and click **Finish**.

Sales

Savant Systems, LLC is dedicated to providing prompt and effective sales support.

Residential: www.SavantSystems.com/residential_sales.aspx
Commercial: www.SavantSystems.com/commercial_contact.aspx
Fax: 1-508-683-2600
Email: sales@SavantSystems.com