# ANI4OUT

## Command Strings

Shure ANI4OUT command strings for third-party control systems, such as Crestron or Extron. Includes all programming commands for block and XLR versions.

Version: 3 (2019-G)

# Table of Contents

# ANI4OUT Command Strings

## ANI4OUT Command Strings

The device is connected via Ethernet to a control system, such as AMX, Crestron or Extron.

**Connection:** Ethernet (TCP/IP; select “Client” in the AMX/Crestron program)
**Port:** 2202

### Conventions

The device has 4 types of strings:

**GET**

Finds the status of a parameter. After the AMX/Crestron sends a GET command, the ANI4OUT responds with a REPORT string

**SET**

Changes the status of a parameter. After the AMX/Crestron sends a SET command, the ANI4OUT will respond with a REPORT string to indicate the new value of the parameter.

**REP**

When the ANI4OUT receives a GET or SET command, it will reply with a REPORT command to indicate the status of the parameter. REPORT is also sent by the ANI4OUT when a parameter is changed on the ANI4OUT or through the GUI.

**SAMPLE**

Used for metering audio levels.

All messages sent and received are ASCII. Note that the level indicators and gain indicators are also in ASCII

Most parameters will send a REPORT command when they change. Thus, it is not necessary to constantly query parameters. The ANI4OUT will send a REPORT command when any of these parameters change.

The character

“x”

in all of the following strings represents the channel of the ANI4OUT and can be ASCII numbers 0 through 4 as in the following table

| | |
|---|---|
| **0** | All channels |
| **1 through 4** | Individual channels |

### Command Strings (Common)

| |
|---|
| **Get All** |

| | |
|---|---|
| **Command String:**<br>`< GET x ALL >` | *Where x is ASCII channel number: 0 through 4. Use this command on first power on to update the status of all parameters.* |
| **ANI4OUT Response:**<br>`< REP ... >` | *The ANI4OUT responds with individual Report strings for all parameters.* |
| **Get Model Number** | |
| **Command String:**<br>`< GET MODEL >` | |
| **ANI4OUT Response:**<br>`< REP MODEL {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 32 characters of the model number. The ANI4OUT always responds with a 32 character model number.* |
| **Get Serial Number** | |
| **Command String:**<br>`< GET SERIAL_NUM >` | |
| **ANI4OUT Response:**<br>`< REP SERIAL_NUM {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 32 characters of the serial number. The ANI4OUT always responds with a 32 character serial number.* |
| **Get Firmware Version** | |
| **Command String:**<br>`< GET FW_VER >` | |
| **ANI4OUT Response:**<br>`< REP FW_VER {yyyyyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyyyyy is 18 characters. The ANI4OUT always responds with 18 characters.* |
| **Get Audio IP Address** | |
| **Command String:**<br>`< GET IP_ADDR_NET_AUDIO_PRIMARY >` | |
| **ANI4OUT Response:**<br>`< REP IP_ADDR_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyy is a 15 digit IP address.* |
| **Get Audio Subnet Address** | |

| | |
|---|---|
| **Command String:**<br>`< GET IP_SUBNET_NET_AUDIO_PRIMARY >` | |
| **ANI4OUT Response:**<br>`< REP IP_SUBNET_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyy is a 15 digit subnet address.* |
| **Get Audio Gateway Address** | |
| **Command String:**<br>`< GET IP_GATEWAY_NET_AUDIO_PRIMARY >` | |
| **ANI4OUT Response:**<br>`< REP IP_GATEWAY_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyy is a 15 digit gateway address.* |
| **Get Channel Name** | |
| **Command String:**<br>`< GET x CHAN_NAME` | *Where x is ASCII channel number: 0 through 4.* |
| **ANI4OUT Response:**<br>`< REP x CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the user name. The ANI4OUT always responds with a 31 channel name.* |
| **Get Device ID** | |
| **Command String:**<br>`< GET DEVICE_ID >` | *The Device ID command does not contain the x channel character, as it is for the entire ANI4OUT.* |
| **ANI4OUT Response:**<br>`< REP DEVICE_ID {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >` | *Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the device ID. The ANI4OUT always responds with a 31 character device ID.* |
| **Get Preset** | |
| **Command String:**<br>`< GET PRESET >` | |
| **ANI4OUT Response:**<br>`< REP PRESET nn >` | *Where nn is the preset number 01-10.* |
| **Set Preset** | |

| | |
|---|---|
| **Command String:**<br>`< SET PRESET nn >` | *Where nn is the preset number 1-10. (Leading zero is optional when using the SET command).* |
| **ANI4OUT Response:**<br>`< REP PRESET nn >` | *Where nn is the preset number 01-10.* |
| **Get Preset Name** | |
| **Command String:**<br>`< GET PRESET1 >`<br>`< GET PRESET2 >`<br>`< GET PRESET3 >`<br>**etc** | *Send one of these commands to the ANI4OUT* |
| **ANI4OUT Response:**<br>`< REP PRESET1 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`<br>`< REP PRESET2 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`<br>`< REP PRESET3 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`<br>**etc** | *Whereyyyyyyyyyyyyyyyyyyyyyyyyy is 25 characters of the preset name. The ANI4OUT always responds with a 25 character preset name* |
| **Get Audio Gain** | |
| **Command String:**<br>`< GET x AUDIO_GAIN_HI_RES >` | *Where x is ASCII channel number: 1 through 4.* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_GAIN_HI_RES yyyy >` | *Where yyyy takes on the ASCII values of 0000 to 1400. yyyy is in steps of one-tenth of a dB.* |
| **Set Audio Gain** | |
| **Command String:**<br>`< SET x AUDIO_GAIN_HI_RES yyyy >` | *Where yyyy takes on the ASCII values of 0000 to 1400. yyyy is in steps of one-tenth of a dB.* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_GAIN_HI_RES yyyy >` | *Where yyyy takes on the ASCII values of 0000 to 1400.* |
| **Increase Audio Gain by n dB** | |
| **Command String:**<br>`< SET x AUDIO_GAIN_HI_RES INC nn >` | *Where nn is the amount in one-tenth of a dB to increase the gain. nn can be* |

| | |
|---|---|
| | *single digit ( n ), double digit ( nn ), triple digit ( nnn ).* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_GAIN_HI_RES yyyy >` | *Where yyyy takes on the ASCII values of 0000 to 1400.* |
| **Decrease Audio Gain by n dB** | |
| **Command String:**<br>`< SET x AUDIO_GAIN_HI_RES DEC nn >` | *Where nn is the amount in one-tenth of a dB to decrease the gain. nn can be single digit ( n ), double digit ( nn ), triple digit ( nnn ).* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_GAIN_HI_RES yyyy >` | *Where yyyy takes on the ASCII values of 0000 to 1280.* |
| **Get Channel Audio Mute** | |
| **Command String:**<br>`< GET x AUDIO_MUTE >` | *Where x is ASCII channel number: 0 through 4.* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_MUTE ON >`<br>`< REP x AUDIO_MUTE OFF >` | *The ANI4OUT will respond with one of these strings.* |
| **Mute Channel Audio** | |
| **Command String:**<br>`< SET x AUDIO_MUTE ON >` | |
| **ANI4OUT Response:**<br>`< REP x AUDIO_MUTE ON >` | |
| **Unmute Channel Audio** | |
| **Command String:**<br>`< SET x AUDIO_MUTE OFF >` | |
| **ANI4OUT Response:**<br>`< REP x AUDIO_MUTE OFF >` | |
| **Toggle Channel Audio Mute** | |
| **Command String:** | |

| | |
|---|---|
| `< SET x AUDIO_MUTE TOGGLE >` | |
| **ANI4OUT Response:**<br>`< REP x AUDIO_MUTE ON >`<br>`< REP x AUDIO_MUTE OFF >` | *The ANI4OUT will respond with one of these strings.* |
| **Get Analog Gain** | |
| **Command String:**<br>`< GET x AUDIO_OUT_LVL_SWITCH >` | *Where x is ASCII channel number: 0 through 4.* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_OUT_LVL_SWITCH LINE_LVL >`<br>`< REP x AUDIO_OUT_LVL_SWITCH AUX_LVL >`<br>`< REP x AUDIO_OUT_LVL_SWITCH MIC_LVL >` | *The ANI4OUT will respond with one of these strings.* |
| **Set Analog Gain** | |
| **Command String:**<br>`< SET x AUDIO_OUT_LVL_SWITCH LINE_LVL >`<br>`< SET x AUDIO_OUT_LVL_SWITCH AUX_LVL >`<br>`< SET x AUDIO_OUT_LVL_SWITCH MIC_LVL >` | *Where x is ASCII channel number: 0 through 4. Send one of these commands to the ANI4OUT.* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_OUT_LVL_SWITCH LINE_LVL >`<br>`< REP x AUDIO_OUT_LVL_SWITCH AUX_LVL >`<br>`< REP x AUDIO_OUT_LVL_SWITCH MIC_LVL >` | *The ANI4OUT will respond with one of these strings.* |
| **Get Sig/Clip LED** | |
| **Command String:**<br>`< GET x LED_COLOR_SIG_CLIP >` | *Where x is ASCII channel number: 0 through 4. It is not necessary to continually send this command. The ANI4OUT will send a REPORT message whenever the status changes.* |
| **ANI4OUT Response:**<br>`< REP x LED_COLOR_SIG_CLIP OFF >`<br>`< REP x LED_COLOR_SIG_CLIP GREEN >`<br>`< REP x LED_COLOR_SIG_CLIP AMBER >`<br>`< REP x LED_COLOR_SIG_CLIP RED >` | *The ANI4OUT will respond with one of these strings. This matches the sig/clip LEDs on the front of the ANI4OUT.* |

| | |
|---|---|
| **Flash Lights on ANI4OUT** | |
| **Command String:**<br>`< SET FLASH ON >`<br>`< SET FLASH OFF >` | *Send one of these commands to the ANI4OUT. The flash automatically turns off after 30 seconds.* |
| **ANI4OUT Response:**<br>`< REP FLASH ON >`<br>`< REP FLASH OFF >` | *The ANI4OUT will respond with one of these strings.* |
| **Turn Metering On** | |
| **Command String:**<br>`< SET METER_RATE sssss >` | *Where sssss is the metering speed in milliseconds. Setting sssss=0 turns metering off. Minimum setting is 100 milliseconds. Metering is off by default.* |
| **ANI4OUT Response:**<br>`< REP METER_RATE sssss >`<br>`< SAMPLE aaa bbb ccc ddd >` | *Where aaa, bbb, etc is the value of the audio level received and is 000-060.*<br>*aaa= output 1*<br>*bbb= output 2*<br>*ccc= output 3*<br>*ddd= output 4* |
| **Stop Metering** | |
| **Command String:**<br>`< SET METER_RATE 0 >` | *A value of 00000 is also acceptable.* |
| **ANI4OUT Response:**<br>`< REP METER_RATE 00000 >` | |
| **Get LED Brightness** | |
| **Command String:**<br>`< GET LED_BRIGHTNESS >` | |
| **ANI4OUT Response:**<br>`< REP LED_BRIGHTNESS n >` | *Where n can take on the following values:*<br>0 = LED disabled<br>1 = LED dim |

| | |
|---|---|
| | 2 = LED default |
| **Set LED Brightness** | |
| **Command String:**<br>`< SET LED_BRIGHTNESS n >` | *Where n can take on the following values:*<br>0 = LED disabled<br>1 = LED dim<br>2 = LED default |
| **ANI4OUT Response:**<br>`< REP LED_BRIGHTNESS n >` | |
| **Reboot ANI4OUT (firmware > v2.0)** | |
| **Command String:**<br>`< SET REBOOT >` | |
| **ANI4OUT Response:** | *The ANI4OUT does not send a response for this command* |
| **Get Error Events (firmware > v2.0)** | |
| **Command String:**<br>`< GET LAST_ERROR_EVENT >` | |
| **ANI4OUT Response:**<br>`< REP LAST_ERROR_EVENT {yyyyy} >` | *Where yyyy can be up to 128 characters.* |
| **Get Output Meter Mode (firmware > v2.0)** | |
| **Command String:**<br>`< GET OUTPUT_METER_MODE >` | |
| **ANI4OUT Response:**<br>`< REP OUTPUT_METER_MODE PRE_FADER >`<br>`< REP OUTPUT_METER_MODE POST_FADER >` | *The ANI4OUT will respond with one of these strings.* |
| **Set Output Meter Mode (firmware > v2.0)** | |
| **Command String:**<br>`< SET OUTPUT_METER_MODE PRE_FADER >` | *Send one of these commands to the ANI4OUT.* |

| | |
|---|---|
| `< SET OUTPUT_METER_MODE POST_FADER >` | |
| **ANI4OUT Response:**<br>`< REP OUTPUT_METER_MODE PRE_FADER >`<br>`< REP OUTPUT_METER_MODE POST_FADER >` | *The ANI4OUT will respond with one of these strings.* |
| **Get Limiter Engaged (firmware > v2.0)** | |
| **Command String:**<br>`< GET x LIMITER_ENGAGED >` | *Where x is ASCII channel number: 1 or 3. The limiter is only engaged when using summing mode* |
| **ANI4OUT Response:**<br>`< REP x LIMITER_ENGAGED ON >`<br>`< REP x LIMITER_ENGAGED OFF >` | *The ANI4OUT will respond with one of these strings.* |
| **Get Audio Summing Mode (firmware > v2.0)** | |
| **Command String:**<br>`< GET AUDIO_SUMMING_MODE >` | |
| **ANI4OUT Response:**<br>`< REP AUDIO_SUMMING_MODE OFF >`<br>`< REP AUDIO_SUMMING_MODE 1+2 >`<br>`< REP AUDIO_SUMMING_MODE 3+4 >`<br>`< REP AUDIO_SUMMING_MODE 1+2/3+4 >`<br>`< REP AUDIO_SUMMING_MODE 1+2+3+4 >` | *The ANI4OUT will respond with one of these strings.* |
| **Set Audio Summing Mode (firmware > v2.0)** | |
| **Command String:**<br>`< SET AUDIO_SUMMING_MODE OFF >`<br>`< SET AUDIO_SUMMING_MODE 1+2 >`<br>`< SET AUDIO_SUMMING_MODE 3+4 >`<br>`< SET AUDIO_SUMMING_MODE 1+2/3+4 >`<br>`< SET AUDIO_SUMMING_MODE 1+2+3+4 >` | *Send one of these commands to the ANI4OUT.* |
| **ANI4OUT Response:**<br>`< REP AUDIO_SUMMING_MODE OFF >`<br>`< REP AUDIO_SUMMING_MODE 1+2 >`<br>`< REP AUDIO_SUMMING_MODE 3+4 >` | *The ANI4OUT will respond with one of these strings.* |

| | |
|---|---|
| `< REP AUDIO_SUMMING_MODE 1+2/3+4 >`<br>`< REP AUDIO_SUMMING_MODE 1+2+3+4 >` | |
| **Get RMS Audio Level (firmware > v2.0)** | |
| **Command String:**<br>`< GET x AUDIO_IN_RMS_LVL >` | *where xis channel number: 0: all channels ANI4IN: 1-4* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_IN_RMS_LVLnnn >` | *where x is channel number defined in GET command. where nnn is audio level in the range of 000-060* |
| **Get Peak Audio Level (firmware > v2.0)** | |
| **Command String:**<br>`< GET x AUDIO_IN_PEAK_LVL >` | *where xis channel number: 0: all channels ANI4IN: 1-4* |
| **ANI4OUT Response:**<br>`< REP x AUDIO_IN_PEAK_LVLnnn >` | *where x is channel number defined in GET command. where nnn is audio level in the range of 000-060* |
| **Get Network Audio Device Name** | |
| **Command String:**<br>`< GET NA_DEVICE_NAME >` | |
| **ANI4OUT Response:**<br>`< REP NA_DEVICE_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >` | *Where {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} is a text string. Most devices allow device id to be up to 31characters. Value is padded with spaces as needed to ensure that 31 char are always reported.* |
| **Get Network Audio Channel Name** | |
| **Command String:**<br>`< GET NA_CHAN_NAME >` | *Where xx is channel number All channels: 0 ANI4OUT: 1-4* |
| **ANI4OUT Response:**<br>`< REP xx NA_CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy}` | *Where xx is channel number. Where {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} is 31 char channel name. Value is padded with spaces as needed to ensure that 31 char are always reported. Returns errors per Standardization of Values for Reporting Error States under any of the following conditions: Invalid channel* |
| **Get Control Network MAC Address** | |

| | |
|---|---|
| **Command String:**<br>`< GET CONTROL_MAC_ADDR >` | |
| **ANI4OUT Response:**<br>`< REP CONTROL_ MAC_ADDR yy:yy:yy:yy:yy:yy >` | *Where yy:yy:yy:yy:yy:yy is a 17 char literal string formatted as 6 octets, each separated by a colon. Example: 00:0E:DD:FF:F1:63* |
| **Restore Default Settings (firmware > v2.0)** | |
| **Command String:**<br>`< SET DEFAULT_SETTINGS >` | *Request the device to set itself to default settings.* |
| **ANI4OUT Response:**<br>`< REP PRESET xx >` | *where xx = 00 if restore is successful* |
| **Get LED State** | |
| **Command String:**<br>`< GET x LED_STATE_SIG_CLIP >` | *where x is channel number that takes on values: 0: all channels 1-4: individual channel* |
| **ANI4OUT Response:**<br>`< REP x LED_STATE_SIG_CLIP yyy >` | *where x is channel number that takes on values: 1-4: individual channel ;Where yyy is current LED state. Valid yyy values are: On - Steady ,Flashing, Off* |
| **Get PEQ Filter Enable (firmware > v2.0)** | |
| **Command String:**<br>`< GET xx PEQ yy >` | *Where xx is the PEQ block 01-04. Where yy is the PEQ filter 01-04 within the block. 00 can be used for all blocks or all filters.* |
| **ANI4OUT Response:**<br>`< REP xx PEQ yy ON >`<br>`< REP xx PEQ yy OFF >` | |
| **Set PEQ Filter Enable (firmware > v2.0)** | |
| **Command String:**<br>`< SET xx PEQ yy ON >`<br>`< SET xx PEQ yy OFF >` | *Send one of these commands to the ANI4OUT.* |
| **ANI4OUT Response:** | *Where xx is the PEQ block 01-04. Where yy is the PEQ filter 01-04 within* |

| | |
|---|---|
| `< REP xx PEQ yy ON >`<br>`< REP xx PEQ yy OFF >` | *the block. 00 can be used for all blocks or all filters.* |
| **Get Encryption Status (firmware > v2.0)** | |
| **Command String:**<br>`< GET ENCRYPTION >` | *Get device level encryption status;* |
| **ANI4OUT Response:**<br>`< REP ENCRYPTION ON >`<br>`< REP ENCRYPTION OFF >` | *Send one of these commands to the ANI4OUT.* |