MXA310

Command Strings

MXA310 command strings for third-party control systems.
Version: 3.1 (2021-J)

Table of Contents

MXA310 Command Strings

Using a Third-Party Control System

The microphone can send an external logic control signal to any networked devices that receive logic signals through an Ethernet connection. This allows the microphone mute switch to mute a DSP audio signal, instead of (or in addition to) muting the microphone at the source. The microphone also receives logic commands over the network. Many parameters controlled through the web application can be controlled through a third party control system, using the appropriate command string.

Common applications:

- Mute
- LED color and behavior
- Loading presets
- Adjusting levels

A complete list of command strings is available in the device help or from www.shure.com.

To send a logic signal out when the mute button is pressed:

1. In the web application, select Configuration > Button Control.
2. Under the Button Properties menu, change the Mute Button Control setting to Logic out.

MXA310 Microflex® Advance™ 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 MXA310 responds with a REPORT string

SET

Changes the status of a parameter. After the AMX/Crestron sends a SET command, the MXA310 will respond with a REPORT string to indicate the new value of the parameter.

REP

When the MXA310 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 MXA310 when a parameter is changed on the MXA310 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 MXA310 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 MXA310 and can be ASCII numbers 0 through 5 as in the following table.

0	All channels
1 through 4	Individual channels
5	Automix output

Command Strings (Common)

Get All	
Command String: `< GET x ALL >`	Where x is ASCII channel number: 0 through 5. Use this command on first power on to update the status of all parameters.
MXA310 Response: `< REP ... >`	The MXA310 responds with individual Report strings for all parameters.
Get Model Number	
Command String: `< GET MODEL >`	
MXA310 Response: `< REP MODEL {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 32 characters of the model number. The MXA310 always responds with a 32 character model number.
Get Serial Number	
Command String: `< GET SERIAL_NUM >`	
MXA310 Response: `< REP SERIAL_NUM {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 32 characters of the serial number. The MXA310 always responds with a 32 character serial number.

Get Firmware Version	
Command String: `< GET FW_VER >`	
MXA310 Response: `< REP FW_VER {yyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyy is 18 characters. The MXA310 always responds with 18 characters.
Get Audio IP Address	
Command String: `< GET IP_ADDR_NET_AUDIO_PRIMARY >`	
MXA310 Response: `< REP IP_ADDR_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyy is a 15 digit IP address.
Get Audio Subnet Address	
Command String: `< GET IP_SUBNET_NET_AUDIO_PRIMARY >`	
MXA310 Response: `< REP IP_SUBNET_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyy is a 15 digit subnet address.
Get Audio Gateway Address	
Command String: `< GET IP_GATEWAY_NET_AUDIO_PRIMARY >`	
MXA310 Response: `< REP IP_GATEWAY_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyy is a 15 digit gateway address.
Get Channel Name	
Command String: `< GET x CHAN_NAME >`	Where x is ASCII channel number: 0 through 5.
MXA310 Response: `< REP x CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the channel name. The MXA310 always responds with a 31 character name.
Get Device ID	

Command String: `< GET DEVICE_ID >`	The Device ID command does not contain the x channel character, as it is for the entire device.
MXA310 Response: `< REP DEVICE_ID {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the device ID. The microphone always responds with a 31 character device ID.
Get Audio Gain	
Command String: `< GET x AUDIO_GAIN_HI_RES >`	Where x is ASCII channel number: 1 through 5. Channel number 0 (all channels) is not valid for this command.
MXA310 Response: `< 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: `< 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.
MXA310 Response: `< 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: `< 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).
MXA310 Response: `< 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: `< 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).
MXA310 Response: `< REP x AUDIO_GAIN_HI_RES yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400.
Get Post-Gate Audio Gain (firmware > v3.0)	

Command String: `< GET x AUDIO_GAIN_POSTGATE >`	Where x is ASCII channel number: 1 through 4. Channel number 0 (all channels) is not valid for this command.
MXA310 Response: `< REP x AUDIO_GAIN_POSTGATE yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400. yyyy is in steps of one-tenth of a dB.
Set Post-Gate Audio Gain (firmware > v3.0)	
Command String: `< SET x AUDIO_GAIN_POSTGATE yyyy >`	Where x is ASCII channel number: 1 through 4. Where yyyy takes on the ASCII values of 0000 to 1400. yyyy is in steps of one-tenth of a dB.
MXA310 Response: `< REP x AUDIO_GAIN_POSTGATE yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400.
Get Channel Audio Mute	
Command String: `< GET x AUDIO_MUTE >`	Where x is ASCII channel number: 0 through 5. See table on page 1. Channel Audio Mute is pre-meter.

MXA310 Response:
`< REP x AUDIO_MUTE ON >`
`< REP x AUDIO_MUTE OFF >`
	The MXA310 will respond with one of these strings.

Mute Channel Audio	
Command String: `< SET x AUDIO_MUTE ON >`	
MXA310 Response: `< REP x AUDIO_MUTE ON >`	
Unmute Channel Audio	
Command String: `< SET x AUDIO_MUTE OFF >`	
MXA310 Response: `< REP x AUDIO_MUTE OFF >`	
Toggle Channel Audio Mute	
Command String:	

`< SET x AUDIO_MUTE TOGGLE >`	

MXA310 Response:
`< REP x AUDIO_MUTE ON >`
`< REP x AUDIO_MUTE OFF >`
	The MXA310 will respond with one of these strings.

Get Device Audio Mute	
Command String: `< GET DEVICE_AUDIO_MUTE >`	Device Audio Mute is equivalent to the physical mute button on the mic. Device Audio Mute is post-meter.

MXA310 Response:
`< REP DEVICE_AUDIO_MUTE ON >`
`< REP DEVICE_AUDIO_MUTE OFF >`
	The MXA310 will respond with one of these strings.

Mute Device Audio	
Command String: `< SET DEVICE_AUDIO_MUTE ON >`	
MXA310 Response: `< REP DEVICE_AUDIO_MUTE ON >`	
Unmute Device Audio	
Command String: `< SET DEVICE_AUDIO_MUTE OFF >`	
MXA310 Response: `< REP DEVICE_AUDIO_MUTE OFF >`	
Toggle Device Audio Mute	
Command String: `< SET DEVICE_AUDIO_MUTE TOGGLE >`	

MXA310 Response:
`< REP DEVICE_AUDIO_MUTE ON >`
`< REP DEVICE_AUDIO_MUTE OFF >`
	The MXA310 will respond with one of these strings.

Get Output Clip Status	

Command String: `< GET x AUDIO_OUT_CLIP_INDICATOR >`	Where x is ASCII channel number: 1 through 5. See table on page 1. It is not necessary to continually send this command. The microphone will send a REPORT message whenever the status changes.

MXA310 Response:
`< REP x AUDIO_OUT_CLIP_INDICATOR ON >`
`< REP x AUDIO_OUT_CLIP_INDICATOR OFF >`
	The MXA310 will respond with one of these strings.

Flash Lights on Microphone	

Command String:
`< SET FLASH ON >`
`< SET FLASH OFF >`
	Send one of these commands to the MXA310. The flash automatically turns off after 30 seconds.

MXA310 Response:
`< REP FLASH ON >`
`< REP FLASH OFF >`
	The MXA310 will respond with one of these strings.

Turn Metering On	
Command String: `< 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.

MXA310 Response:
`< REP METER_RATE sssss >`
`< SAMPLE aaa bbb ccc ddd eee >`
	Where
	aaa
	,
	bbb
	, etc is the value of the audio level received and is 000-060.
	aaa
	= output 1
	bbb
	= output 2
	ccc
	= output 3

	ddd = output 4 eee = output 5
Stop Metering	
Command String: `< SET METER_RATE 0 >`	A value of 00000 is also acceptable.
MXA310 Response: `< REP METER_RATE 00000 >`	
Get Automixer Gain Metering Rate (firmware > v3.0)	
Command String: `< GET METER_RATE_MXR_GAIN >`	

MXA310 Response:
`< REP METER_RATE_MXR_GAIN sssss >`
`< SAMPLE aaa bbb ccc ddd >`
	Where sssss is the metering rate in milliseconds. Setting sssss= 0 turns metering off.

Set Automixer Gain Metering Rate (firmware > v3.0)	
Command String: `< SET METER_RATE_MXR_GAIN sssss >`	Where ssssssis a value from 0 to 99999 in milliseconds. 0 = Off 100 = Minimum value 99999 = Maximum value
MXA310 Response: `< SAMPLE aaa bbb ccc ddd >`	Where aaa, bbb, etc is the value of the audio level received and is 000-060. aaa = output 1 bbb = output 2 ccc = output 3 ddd

	= output 4
Get Audio Peak Level	
Command String: `< GET x AUDIO_IN_PEAK_LVL >`	
MXA310 Response: `< REP x AUDIO_IN_PEAK_LVL nnn >`	Where nnn is the audio level and is 000-060.
Get Audio RMS Level	
Command String: `< GET x AUDIO_IN_RMS_LVL >`	
MXA310 Response: `< REP x AUDIO_IN_RMS_LVL nnn >`	Where nnn is the audio level and is 000-060.
Get Preset	
Command String: `< GET PRESET >`	
MXA310 Response: `< REP PRESET nn >`	Where nn is the preset number 01-10.
Set Preset	
Command String: `< SET PRESET nn >`	Where nn is the preset number 1-10. (Leading zero is optional when using the SET command).
MXA310 Response: `< REP PRESET nn >`	Where nn is the preset number 01-10.
Get Preset Name	

Command String:
`< GET PRESET1 >`
`< GET PRESET2 >`
`< GET PRESET3>`
etc
	Send one of these commands to the MXA310.

MXA310 Response:
`< REP PRESET1 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`
`< REP PRESET2 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`
`< REP PRESET3 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`
etc
	Whereyyyyyyyyyyyyyyyyyyyyyyyyy is 25 characters of the preset name. The MXA310 always responds with a 25 character preset name

Get Gate Out Status	
Command String: `< GET x AUTOMIX_GATE_OUT_EXT_SIG >`	Where x is ASCII channel number: 0 through 4. It is not necessary to continually send this command. The MXA310 will send a REPORT message whenever the status changes.

MXA310 Response:
`< REP x AUTOMIX_GATE_OUT_EXT_SIG ON >`
`< REP x AUTOMIX_GATE_OUT_EXT_SIG OFF >`
	The MXA310 will respond with one of these strings.

External Switch Out	
Command String: `< GET EXT_SWITCH_OUT_STATE >`	It is not necessary to continually send this command. The MXA310 will send a REPORT message whenever the status changes.
MXA310 Response: `< REP EXT_SWITCH_OUT_STATE ON >` `< REP EXT_SWITCH_OUT_STATE OFF >`	The MXA310 will respond with one of these strings.
Mute Button Status	
Command String: `< GET MUTE_BUTTON_STATUS >`	It is not necessary to continually send this command. The MXA310 will send a REPORT message whenever the status changes.

MXA310 Response:
`< REP MUTE_BUTTON_STATUS ON >`
`< REP MUTE_BUTTON_STATUS OFF >`
	The MXA310 will respond with one of these strings.

Mute Button LED State	
Command String: `< GET MUTE_BUTTON_LED_STATE >`	
MXA310 Response:	The MXA310 will respond with one of these strings.

`< REP MUTE_BUTTON_LED_STATE ON >` `< REP MUTE_BUTTON_LED_STATE OFF >`	
Get Ring LED State (Use when GUI Lighting Style is set to RING)	
Command String: `< GET DEV_LED_IN_STATE >`	This command is only available when both “Mute Control Function” is set to “Logic Out” or "Disabled" AND Light Ring “Lighting Style” is set to “Ring” in the GUI.

MXA310 Response:
`< REP DEV_LED_IN_STATE ON >`
`< REP DEV_LED_IN_STATE OFF >`
	The MXA310 will respond with one of these strings.

Set Ring LED State (Use when GUI Lighting Style is set to RING)	

Command String:
`< SET DEV_LED_IN_STATE ON >`
`< SET DEV_LED_IN_STATE OFF >`
	Send one of these commands to the MXA310. This command is only available when both “Mute Control Function” is set to “Logic Out” or "Disabled" AND Light Ring “Lighting Style” is set to “Ring” in the GUI.

MXA310 Response:
`< REP DEV_LED_IN_STATE ON >`
`< REP DEV_LED_IN_STATE OFF >`
	The MXA310 will respond with one of these strings.

Get Segments LED State (Use when GUI Lighting Style is set to SEGMENTS)	
Command String: `< GET x CHAN_LED_IN_STATE >`	This command is only available when both “Mute Control Function” is set to “Logic Out” or "Disabled" AND Light Ring “Lighting Style” is set to “Segments” in the GUI.

MXA310 Response:
`< REP x CHAN_LED_IN_STATE ON >`
`< REP x CHAN_LED_IN_STATE OFF >`
	The MXA310 will respond with one of these strings.

Set Segments LED State (Use when GUI Lighting Style is set to SEGMENTS)	

Command String:
`< SET x CHAN_LED_IN_STATE ON >`
`< SET x CHAN_LED_IN_STATE OFF >`
	Where x is ASCII channel number: 1 through 4. Send one of these commands to the MXA310. This command is only available when both “Mute Control Function” is set to “Logic Out” or

	"Disabled" AND Light Ring “Lighting Style” is set to “Segments” in the GUI.

MXA310 Response:
`< REP x CHAN_LED_IN_STATE ON >`
`< REP x CHAN_LED_IN_STATE OFF >`
	The MXA310 will respond with one of these strings.

Get LED Brightness	
Command String: `< GET LED_BRIGHTNESS >`	
MXA310 Response: `< REP LED_BRIGHTNESS n >`	Where n can take on the following values: 0 = LED disabled 1 = LED dim 2 = LED default **Firmware > v3.0:** 0 = LED disabled 1 = 20% 2 = 40% 3 = 60% 4 = 80% 5 = 100%
Set LED Brightness	
Command String: `< SET LED_BRIGHTNESS n >`	Where n can take on the following values: 0 = LED disabled 1 = LED dim 2 = LED default **Firmware > v3.0:** 0 = LED disabled 1 = 20% 2 = 40% 3 = 60%

	4 = 80% 5 = 100%
MXA310 Response: `< REP LED_BRIGHTNESS n >`	
Get LED Mute Color	
Command String: `< GET LED_COLOR_MUTED >`	
MXA310 Response: `< REP LED_COLOR_MUTED nnnn >`	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE. **Firmware > v3.0:** Where nnnn can be RED , GREEN , BLUE , PINK , PURPLE , YELLOW , ORANGE , WHITE , GOLD

	, YELLOWGREEN , TURQUOISE , POWDERBLUE , CYAN , SKYBLUE , LIGHTPURPLE , VIOLET , or ORCHID .
Set LED Mute Color	
Command String: `< SET LED_COLOR_MUTED nnnn >`	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE. **Firmware > v3.0:** Where nnnn can be RED , GREEN , BLUE , PINK

	, PURPLE , YELLOW , ORANGE , WHITE , GOLD , YELLOWGREEN , TURQUOISE , POWDERBLUE , CYAN , SKYBLUE , LIGHTPURPLE , VIOLET , or ORCHID .
MXA310 Response: `< REP LED_COLOR_MUTED nnnn >`	
Get LED Unmute Color	

Command String: `< GET LED_COLOR_UNMUTED >`	
MXA310 Response: `< REP LED_COLOR_UNMUTED nnnn >`	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE. **Firmware > v3.0:** Where nnnn can be RED , GREEN , BLUE , PINK , PURPLE , YELLOW , ORANGE , WHITE , GOLD , YELLOWGREEN , TURQUOISE , POWDERBLUE

	, CYAN , SKYBLUE , LIGHTPURPLE , VIOLET , or ORCHID .
Set LED Unmute Color	
Command String: `< SET LED_COLOR_UNMUTED nnnn >`	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE. **Firmware > v3.0:** Where nnnn can be RED , GREEN , BLUE , PINK , PURPLE , YELLOW , ORANGE

	, WHITE , GOLD , YELLOWGREEN , TURQUOISE , POWDERBLUE , CYAN , SKYBLUE , LIGHTPURPLE , VIOLET , or ORCHID .
MXA310 Response: `< REP LED_COLOR_UNMUTED nnnn >`	
Get LED Mute Flashing	
Command String: `< GET LED_STATE_MUTED >`	
MXA310 Response: `< REP LED_STATE_MUTED nnn >`	Where nnn can be ON, OFF, or FLASHING.
Set LED Mute Flashing	

Command String: `< SET LED_STATE_MUTED nnn >`	Where nnn can be ON, OFF, or FLASHING.
MXA310 Response: `< REP LED_STATE_MUTED nnn >`	
Get LED Unmute Flashing	
Command String: `< GET LED_STATE_UNMUTED >`	
MXA310 Response: `< REP LED_STATE_UNMUTED nnn >`	Where nnn can be ON, OFF, or FLASHING.
Set LED Unmute Flashing	
Command String: `< SET LED_STATE_UNMUTED nnn >`	Where nnn can be ON, OFF, or FLASHING.
MXA310 Response: `< REP LED_STATE_UNMUTED nnn >`	
Reboot MXA310 (firmware > v2.0)	
Command String: `< SET REBOOT >`	
MXA310 Response:	The MXA310 does not send a response for this command
Get Error Events (firmware > v2.0)	
Command String: `< GET LAST_ERROR_EVENT >`	
MXA310 Response: `< REP LAST_ERROR_EVENT {yyyyy} >`	Where yyyy can be up to 128 characters.
Get Low Cut Filter (firmware > v2.0)	
Command String: `< GET LOW_CUT_FILTER >`	

MXA310 Response:
`< REP LOW_CUT_FILTER ON >`
`< REP LOW_CUT_FILTER OFF >`
	The MXA310 will respond with one of these strings.

Set Low Cut Filter (firmware > v2.0)	

Command String:
`< SET LOW_CUT_FILTER ON >`
`< SET LOW_CUT_FILTER OFF >`
`< SET LOW_CUT_FILTER TOGGLE >`
	Send on of these commands to the MXA310

MXA310 Response:
`< REP LOW_CUT_FILTER ON >`
`< REP LOW_CUT_FILTER OFF >`
	The MXA310 will respond with one of these strings.

Get PEQ Filter Enable (firmware > v3.0)	
Command String: `< GET xx PEQ yy >`	Where xx is the PEQ block 01-04 on mic channel. 5 is the PEQ on the automix out channel. Where yy is the PEQ filter 01-04 within the selected block. 00 can be used for all blocks or all filters.

MXA310 Response:
`< REP xx PEQ yy ON >`
`< REP xx PEQ yy OFF >`	

Set PEQ Filter Enable (firmware > v3.0)	

Command String:
`< SET xx PEQ yy ON >`
`< SET xx PEQ yy OFF >`
	Send one of these commands to the MXA310.

MXA310 Response:
`< REP xx PEQ yy ON >`
`< REP xx PEQ yy OFF >`
	Where xx is the PEQ block 01-04. 5 is the PEQ on the automix out channel. Where yy is the PEQ filter 01-04 within the selected block. 00 can be used for all blocks or all filters.

Get Bypass All EQ (firmware > v3.0)	
Command String: `< GET BYPASS_ALL_EQ >`	

MXA310 Response: `< REP BYPASS_ALL_EQ sts >`	Where sts can be: ON OFF
Set Bypass All EQ (firmware > v3.0)	
Command String: `< SET BYPASS_ALL_EQ sts >`	Where sts can be: ON OFF TOGGLE
MXA310 Response: `< REP BYPASS_ALL_EQ sts >`	Where sts can be: ON OFF
Get Polar Pattern (firmware > v2.0)	
Command String: `< GET x POLAR_PATTERN >`	

MXA310 Response:
`< REP x POLAR_PATTERN TOROID >`
`< REP x POLAR_PATTERN OMNI >`
`< REP x POLAR_PATTERN CARDIOID >`
`< REP x POLAR_PATTERN SUPER >`
`< REP x POLAR_PATTERN HYPER >`
`< REP x POLAR_PATTERN BIDIRECTION >`
	The MXA310 will respond with one of these strings.

Set Polar Pattern (firmware > v2.0)	

Command String:
`< SET x POLAR_PATTERN TOROID >`
`< SET x POLAR_PATTERN OMNI >`
`< SET x POLAR_PATTERN CARDIOID >`
`< SET x POLAR_PATTERN SUPER >`
`< SET x POLAR_PATTERN HYPER >`
`< SET x POLAR_PATTERN BIDIRECTION >`
	Send one of these strings to the MXA310.

MXA310 Response:	The MXA310 will respond with one of these strings.

`< REP x POLAR_PATTERN TOROID >`
`< REP x POLAR_PATTERN OMNI >`
`< REP x POLAR_PATTERN CARDIOID >`
`< REP x POLAR_PATTERN SUPER >`
`< REP x POLAR_PATTERN HYPER >`
`< REP x POLAR_PATTERN BIDIRECTION >`	

Get Lobe Angle (firmware > v2.0)	
Command String: `< GET x LOBE_ANGLE >`	
MXA310 Response: `< REP x LOBE_ANGLE nnn >`	Where nnn is 015, 030, 045, 060, 075, 090, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 285, 300, 315, 330, or 345.
Increment/Decrement Lobe Angle (firmware > v2.0)	

Command String:
`< SET x LOBE_ANGLE INC nn >`
`< SET x LOBE_ANGLE DEC nnn >`
	Send one of these strings to the MXA310. Where nn is 15, 30, 45, 60, etc.

MXA310 Response: `< REP x LOBE_ANGLE nnn >`	Where nnn is 015, 030, 045, 060, 075, 090, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 285, 300, 315, 330, or 345.
Set Lobe Angle (firmware > v2.0)	
Command String: `< SET x LOBE_ANGLE nn >`	
MXA310 Response: `< REP x LOBE_ANGLE nnn >`	Where nnn is 015, 030, 045, 060, 075, 090, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 285, 300, 315, 330, or 345.
Get Mute Control Function (firmware > v2.0)	
Command String: `< GET MUTE_CONTROL_FUNC >`	
MXA310 Response: `< REP MUTE_CONTROL_FUNC LOCAL >`	The MXA310 will respond with one of these strings.

`< REP MUTE_CONTROL_FUNC LOGIC >` `< REP MUTE_CONTROL_FUNC DISABLED >`	
Set Mute Control Function (firmware > v2.0)	

Command String:
`< SET MUTE_CONTROL_FUNC LOCAL >`
`< SET MUTE_CONTROL_FUNC LOGIC >`
`< SET MUTE_CONTROL_FUNC DISABLED >`
	Send on of these commands to the MXA310

MXA310 Response:
`< REP MUTE_CONTROL_FUNC LOCAL >`
`< REP MUTE_CONTROL_FUNC LOGIC >`
`< REP MUTE_CONTROL_FUNC DISABLED >`
	The MXA310 will respond with one of these strings.

Get Channel Mute LED State	
Command String: `< GET x CHAN_MUTE_STATUS_LED_STATE >`	where x is the channel requested: 0: all channels 1-4: individual channel

MXA310 Response:
`< REP x CHAN_MUTE_STATUS_LED_STATE ON >`
`< REP x CHAN_MUTE_STATUS_LED_STATE OFF >`
	where x is the channel number: 1-4: individual channel; ON = MUTED OFF = UNMUTED

Get Device Mute LED State	
Command String: `< GET DEV_MUTE_STATUS_LED_STATE >`	

MXA310 Response:
`< REP DEV_MUTE_STATUS_LED_STATE ON >`
`< REP DEV_MUTE_STATUS_LED_STATE OFF >`
	ON = MUTED OFF = UNMUTED

Get Network Audio Device Name	
Command String: `< GET NA_DEVICE_NAME >`	
MXA310 Response: `< 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: `< GET xx NA_CHAN_NAME >`	Where xx is channel number All channels: 0 MXA310: 1-5, 5 being automix channel
MXA310 Response: `< 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.
Get Control Network MAC Address	
Command String: `< GET CONTROL_MAC_ADDR >`	
MXA310 Response: `< 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: `< SET DEFAULT_SETTINGS >`	Request the device to set itself to default settings.
MXA310 Response: `< REP PRESET xx >`	where xx = 00 if restore is successful
Get PEQ Filters (firmware > v2.0)	
Command String: `< GET PEQ_FLTRxx >`	where xx is the filter number 01-04: individual filter

MXA310 Response:
`< REP PEQ_FLTRxx ON >`
`< REP PEQ_FLTRxx OFF >`
	where xx is PEQ filter number PEQ filter status: ON OFF

Set PEQ Filters (firmware > v2.0)	

Command String:
`< SET PEQ_FLTRxx ON >`
`< SET PEQ_FLTRxx OFF >`
`< SET PEQ_FLTRxx TOGGLE >`
	where xx is filter number PEQ filter status:. ON OFF TOGGLE

MXA310 Response:
`< REP PEQ_FLTRxx ON >`
`< REP PEQ_FLTRxx OFF >`
	where xx is PEQ filter number PEQ filter status: ON OFF

Get Active Mic Channels	
Command String: `< GET NUM_ACTIVE_MICS >`	
MXA310 Response: `< REP NUM_ACTIVE_MICS x >`	where n is number of active channels that takes on values: MXA310: channels 1-4
Get Automix Channel Solo Enable	
Command String: `< GET x CHAN_AUTOMIX_SOLO_EN >`	where x is channel number: 0 is not valid MXA910: channels 1-8

MXA310 Response:
`< REP x CHAN_AUTOMIX_SOLO_EN ENABLE >`
`< REP x CHAN_AUTOMIX_SOLO_EN DISABLE >`
	where x is channel number: 0 is not valid MXA910: channels 1-8; where sts indicates channel x's SOLO state: ENABLE DISABLE

Set Automix Channel Solo Enable	

Command String:
`< SET x CHAN_AUTOMIX_SOLO_EN ENABLE >`
`< SET x CHAN_AUTOMIX_SOLO_EN DISABLE >`
	where x is channel number: 0 is not valid MXA910: channels 1-8; where sts determines the requested state of SOLO mode: ENABLE DISABLE

MXA310 Response:
`< REP x CHAN_AUTOMIX_SOLO_EN ENABLE >`
`< REP x CHAN_AUTOMIX_SOLO_EN DISABLE >`
	where x is channel number: 0 is not valid MXA910: channels 1-8; where sts indicates channel x's SOLO state: ENABLE DISABLE

Get Encryption Status (firmware > v2.0)	
Command String: `< GET ENCRYPTION >`	Get device level encryption status;

MXA310 Response:
`< REP ENCRYPTION ON >`
`< REP ENCRYPTION OFF >`
	Send one of these commands to the MXA310.