SHURE®

Using a Third-Party Control System

The microphone 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

MXA910 Microflex®Advance™ Command Strings

This document can also be found at: http://shure.custhelp.com/app/answers/detail/a_id/6058

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 MXA910 responds with a REPORT string

SET

Changes the status of a parameter. After the AMX/Crestron sends a SET command, the MXA910 will respond with a REPORT string to indicate the new value of the parameter.

REP

When the MXA910 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 device when a parameter is changed on the MXA910 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 MXA910 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 MXA910 and can be ASCII numbers 0 through 9 as in the following table.

0	All channels
1 through 8	Individual channels
9	Automix output

Command Strings (Common)

Get All		
	Command String: `< GET x ALL >`	Where x is ASCII channel number: 0 through 9. Use this command on first power on to update the status of all parameters.
	MXA910 Response: `< REP ... >`	The MXA910 responds with individual Report strings for all parameters.
Get Channel Name		
	Command String: `< GET x CHAN_NAME >`	Where x is ASCII channel number: 0 through 9.
	MXA910 Response: `< REP x CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the user name. The MXA910 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.
	MXA910 Response: < REP DEVICE_ID {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >	Where **yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy** is 31 characters of the device ID. The MXA910 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 9. Channel number 0 (all channels) is not valid for this command.
	MXA910 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.
	MXA910 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).
	MXA910 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).
	MXA910 Response: < REP x AUDIO_GAIN_HI_RES 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 9. Channel Audio Mute is pre-meter
	MXA910 Response: < REP x AUDIO_MUTE ON > < REP x AUDIO_MUTE OFF >	The MXA910 will respond with one of these strings.
Mute Channel Audio		
	Command String: < SET x AUDIO_MUTE ON >	
	MXA910 Response: < REP x AUDIO_MUTE ON >	
Unmute Channel Audio		
	Command String: < SET x AUDIO_MUTE OFF >	
	MXA910 Response: < REP x AUDIO_MUTE OFF >	
Toggle Channel Audio Mute		
	Command String: < SET x AUDIO_MUTE TOGGLE >	
	MXA910 Response: < REP x AUDIO_MUTE ON > < REP x AUDIO_MUTE OFF >	The MXA910 will respond with one of these strings.
Get Device Audio Mute		
	Command String: < GET DEVICE_AUDIO_MUTE >	Device Audio Mute is post-meter.
	MXA910 Response: < REP DEVICE_AUDIO_MUTE ON > < REP DEVICE_AUDIO_MUTE OFF >	The MXA910 will respond with one of these strings.
Mute Device Audio		
	Command String: < SET DEVICE_AUDIO_MUTE ON >	
	MXA910 Response: < REP DEVICE_AUDIO_MUTE ON >	

Unmute Device Audio		
	Command String: < SET DEVICE_AUDIO_MUTE OFF >	
	MXA910 Response: < REP DEVICE_AUDIO_MUTE OFF >	
Toggle Device Audio Mute		
	Command String: < SET DEVICE_AUDIO_MUTE TOGGLE >	
	MXA910 Response: < REP DEVICE_AUDIO_MUTE ON > < REP DEVICE_AUDIO_MUTE OFF >	The MXA910 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: 0 through 9. It is not necessary to continually send this command. The MXA910 will send a REPORT message whenever the status changes.
	MXA910 Response: < REP x AUDIO_OUT_CLIP_INDICATOR ON > < REP x AUDIO_OUT_CLIP_INDICATOR OFF >	The MXA910 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 MXA910. The flash automatically turns off after 30 seconds.
	MXA910 Response: < REP FLASH ON > < REP FLASH OFF >	The MXA910 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.
	MXA910 Response: < REP METER_RATE sssss > < SAMPLE aaa bbb ccc ddd eee fff ggg hhh iii >	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 fff = output 6 ggg = output 7 hhh = output 8 iii = output 9
Stop Metering		
	Command String: < SET METER_RATE 0 >	A value of 00000 is also acceptable.
	MXA910 Response: < REP METER_RATE 00000 >	
Get Audio Peak Level		
	Command String: < GET x AUDIO_IN_PEAK_LVL >	
	MXA910 Response: < REP x AUDIO_IN_PEAK_LVL nn >	Where nn is the audio level and is 00-60.
Get Audio RMS Level		
	Command String: < GET x AUDIO_IN_RMS_LVL >	
	MXA910 Response: < REP x AUDIO_IN_RMS_LVL nn >	Where nn is the audio level and is 00-60.
Get Preset		
	Command String: < GET PRESET >	
	MXA910 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).
	MXA910 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 strings to the MXA910.
	MXA910 Response: < REP PRESET1 {yyyyyyyyyyyyyyyyyyyyyyyyy} > < REP PRESET2 {yyyyyyyyyyyyyyyyyyyyyyyyy} > < REP PRESET3 {yyyyyyyyyyyyyyyyyyyyyyyyy} > etc	Where**yyyyyyyyyyyyyyyyyyyyyyyyy** is 25 characters of the device ID. The MXA910 always responds with a 25 character device ID
Get Gate Out Status		
	Command String: < GET x AUTOMIX_GATE_OUT_EXT_SIG >	Where x is ASCII channel number: 0 through 8. It is not necessary to continually send this command. The MXA910 will send a REPORT message whenever the status changes.
	MXA910 Response: < REP x AUTOMIX_GATE_OUT_EXT_SIG ON > < REP x AUTOMIX_GATE_OUT_EXT_SIG OFF >	The MXA910 will respond with one of these strings.
Set LED State		
	Command String: < SET DEV_LED_IN_STATE ON > < SET DEV_LED_IN_STATE OFF >	Send one of these commands to the MXA910.
	MXA910 Response: < REP DEV_LED_IN_STATE ON > < REP DEV_LED_IN_STATE OFF >	The MXA910 will respond with one of these strings.
Get LED Brightness		
	Command String: < GET LED_BRIGHTNESS >	
	MXA910 Response: < REP LED_BRIGHTNESS n >	Where n can take on the following values: 0 = LED disabled 1 = LED dim 2 = LED default
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
	MXA910 Response: < REP LED_BRIGHTNESS n >	
Get LED Mute Color		
	Command String: < GET LED_COLOR_MUTED >	
	MXA910 Response: < REP LED_COLOR_MUTED nnnn >	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE
Set LED Mute Color		
	Command String: < SET LED_COLOR_MUTED nnnn >	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE
	MXA910 Response: < REP LED_COLOR_MUTED nnnn >	
Get LED Unmute Color		
	Command String: < GET LED_COLOR_UNMUTED >	
	MXA910 Response: < REP LED_COLOR_UNMUTED nnnn >	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE

Set LED Unmute Color		
	Command String: `< SET LED_COLOR_UNMUTED nnnn >`	Where nnnn can be RED, GREEN, BLUE, PINK, PURPLE, YELLOW, ORANGE, or WHITE
	MXA910 Response: `< REP LED_COLOR_UNMUTED nnnn >`	
Get LED Mute Flashing		
	Command String: `< GET LED_STATE_MUTED >`	
	MXA910 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
	MXA910 Response: `< REP LED_STATE_MUTED nnn >`	
Get LED Unmute Flashing		
	Command String: `< GET LED_STATE_UNMUTED >`	
	MXA910 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
	MXA910 Response: `< REP LED_STATE_UNMUTED nnn >`	
Get X-Axis Beam (Lobe) Steering		
	Command String: `< GET x BEAM_X >`	Where the X-Axis is parallel with the Shure logo.
	MXA910 Response: `< REP x BEAM_X nnnn >`	Where nnnn is 0000-3048 in centimeters. The value 1524 is the centerline of the MXA910.
Set X-Axis Beam (Lobe) Steering		
	Command String: `< SET x BEAM_X nnnn >`	Where nnnn is 0000-3048 in centimeters. The value 1524 is the centerline of the MXA910.
	MXA910 Response: `< REP x BEAM_X nnnn >`	
Get Y-Axis Beam (Lobe) Steering		
	Command String: `< GET x BEAM_Y >`	Where the Y-Axis is perpendicular to the X-Axis.
	MXA910 Response: `< REP x BEAM_Y nnnn >`	Where nnnn is 0000-3048 in centimeters. The value 1524 is the centerline of the MXA910.
Set Y-Axis Beam (Lobe) Steering		
	Command String: `< SET x BEAM_Y nnnn >`	Where nnnn is 0000-3048 in centimeters. The value 1524 is the centerline of the MXA910.
	MXA910 Response: `< REP x BEAM_Y nnnn >`	
Get Beam (Lobe) Height		
	Command String: `< GET x BEAM_Z >`	Where height is the distance down from the MXA910.
	MXA910 Response: `< REP x BEAM_Z nnn >`	Where nnn is 000-914 in centimeters.
Set Beam (Lobe) Height		
	Command String: `< SET x BEAM_Z nnn >`	Where nnn is 000-914 in centimeters.
	MXA910 Response: `< REP x BEAM_Z nnn >`	

Get Beam (Lobe) Width		
	Command String: `< GET x BEAM_W >`	
	MXA910 Response: `< REP x BEAM_W nnnn >`	Where nnnn can be WIDE, MEDIUM, or NARROW
Set Beam (Lobe) Width		
	Command String: `< SET x BEAM_W nnnn >`	Where nnnn can be WIDE, MEDIUM, or NARROW
	MXA910 Response: `< REP x BEAM_W nnnn >`	