MXA710

Command Strings

Shure MXA710 command strings for third-party control systems, such as AMX, Crestron, or Extron. Includes all supported programming commands.
Version: 3.4 (2023-C)

Table of Contents

MXA710 Command Strings

Using a Third-Party Control System

This device can be controlled using a third-party control system with the appropriate command string.

Common applications:

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

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

If using static IP addresses, set the Shure Control and the Audio Network settings to Manual in Designer. Use the Control IP address for TCP/IP communication with Shure devices.

See below for all supported command strings. This list is updated with each firmware release.

Using PuTTY and Other Telnet Clients

For all Telnet clients (including PuTTY), set Telnet negotiation to disabled or passive mode. Active Telnet negotiation is not supported by MXA devices.

If using PuTTY to enter commands for MXA devices, the first command you send may return an error. To fix, enter the command again and it should work normally.

Channel Number Assignments

MXA710 microphones use the following numbering to distinguish the channels for REP values.

MXA710-2FT

- Dante outputs: 1-4
- Automixer output: 5
- Post-gate channels: 1-4

MXA710-4FT

- Dante outputs: 1-8
- Automixer output: 9
- Post-gate channels: 1-8

Get All

Parameter Name:	ALL
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	Responds with `REP` for all device-specific properties and `ALL` channel-related properties.
Example(s):	< GET ALL >

Model

Parameter Name:	MODEL
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	**model** is a 32 character quoted string. The value is padded with spaces to ensure that 32 characters are reported.
Example(s):	< GET MODEL > < REP MODEL model >

Serial Number

Parameter Name:	SERIAL_NUM
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	**serial_num** is a 32 alphanumeric character string. Response is padded to ensure that 32 characters are always returned
Example(s):	< GET SERIAL_NUM > < REP SERIAL_NUM serial_num >

Firmware Version

Parameter Name:	FW_VER
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	Where **ver** is an 18 character literal string: The value is 3 versions separated by a period. Each version shall be able to take on a value from 0 to 65535. **ver** has an "*****" if the firmware is invalid. **Example**: 65535.65535.65535
Example(s):	< GET FW_VER > < REP FW_VER ver >

IP Address for Primary Audio Network

Parameter Name:	IP_ADDR_NET_AUDIO_PRIMARY
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	The value of IP address consist of 4 octets each separated by a period. The length of IP address is 15 characters. The value will be padded to ensure that 15 characters are always returned.
Example(s):	< GET IP_ADDR_NET_AUDIO_PRIMARY > < REP IP_ADDR_NET_AUDIO_PRIMARY ip_addr > < REP ERR >

Subnet Mask for the Primary Audio Network

Parameter Name:	IP_SUBNET_NET_AUDIO_PRIMARY
Command Types Supported:	GET, REP

Indexing:	n/a
Value(s):	**subnet** is subnet mask: 32 bit number represented in the Binary Coded Decimal notation in the form of A.B.C.D where each variable A or B or C or D are 8 bit octets each separated by a period. The length of subnet is 15 characters. The value will be padded to ensure that 15 characters are always returned.
Example(s):	< GET IP_SUBNET_NET_AUDIO_PRIMARY > < REP IP_SUBNET_NET_AUDIO_PRIMARY subnet > < REP ERR >

Network Gateway for Primary Audio Network Interface

Parameter Name:	IP_GATEWAY_NET_AUDIO_PRIMARY
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	**gateway** is network gateway: 32 bit number represented in the Binary Coded Decimal notation in the form of A.B.C.D where each variable A or B or C or D are 8 bit octets each separated by a period. The length of subnet is 15 characters. The value will be padded to ensure that 15 characters are always returned.
Example(s):	< GET IP_GATEWAY_NET_AUDIO_PRIMARY > < REP IP_GATEWAY_NET_AUDIO_PRIMARY gateway > < REP ERR >

Control MAC Address

Parameter Name:	CONTROL_MAC_ADDR
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	**addr** is a 17 character literal string formatted as 6 octets, each separated by a colon.

	Example: 00:0E:DD:FF:F1:63
Example(s):	< GET CONTROL_MAC_ADDR > < REP CONTROL_MAC_ADDR addr > < REP ERR >

Device ID

Parameter Name:	DEVICE_ID
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	Response is a text string. Most devices allow device ID to be up to 31 characters. Value is padded with spaces as needed to ensure that 31 characters are always reported
Example(s):	< GET DEVICE_ID > < REP DEVICE_ID string >

Network Audio (Dante) Device Name

Parameter Name:	NA_DEVICE_NAME
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	Response is a text string. Most devices allow device ID to be up to 31 characters. Value is padded with spaces to ensure that 31 characters are always reported.
Example(s):	< GET NA_DEVICE_NAME > < REP NA_DEVICE_NAME string >

Channel Name

Parameter Name:	CHAN_NAME
Command Types Supported:	GET, REP
Indexing:	GET **index:** See Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index:** 2 digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**string** is 31 character channel name. Value is padded with spaces as needed to ensure that 31 characters are always reported.
Example(s):	< GET index CHAN_NAME > < REP index CHAN_NAME string > < REP ERR >

Network Audio (Dante) Channel Name

Parameter Name:	NA_CHAN_NAME
Command Types Supported:	GET, REP
Indexing:	GET **index:** See Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index:** 2 digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**string** is 31 character channel name. Value is padded with spaces as needed to ensure that 31 characters are always reported.
Example(s):	< GET index NA_CHAN_NAME > < REP index NA_CHAN_NAME string > < REP ERR >

Identify Device (Flash LED)

Parameter Name:	FLASH
Command Types Supported:	GET, SET, REP

Indexing:	n/a
Value(s):	**flash_state** takes on values ON OFF
Example(s):	< GET FLASH > < SET FLASH flash_state > < REP FLASH flash_state > < REP ERR >

Audio Clip Indicator

Parameter Name:	AUDIO_OUT_CLIP_INDICATOR
Command Types Supported:	GET, REP
Indexing:	GET **index** : See Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index** : 2 digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**sts** is current status for the channel: 1. OFF 2. ON
Example(s):	< GET index AUDIO_OUT_CLIP_INDICATOR > < REP index AUDIO_OUT_CLIP_INDICATOR sts > < REP ERR >

Metering Rate (RMS)

Parameter Name:	METER_RATE
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**rate** is a value from 100 to 99999 representing meter rate in milliseconds. 0 = off Values 1 to 99 are not valid and result in response.

	aaa bbb ccc ddd - Audio Levels take on values 000-060, which represent actual audio levels of -60 to 0 dBFS. Represent channels in order defined in Channel Number Assignment.
Example(s):	< GET METER_RATE > < SET METER_RATE rate > < REP METER_RATE rate > < REP ERR > < SAMPLE aaa bbb ccc ddd >

Post-Gate Metering Rate (RMS)

Parameter Name:	METER_RATE_POSTGATE
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**rate** is a value from 100 to 99999 representing meter rate in milliseconds. 0 = off Values 1 to 99 are not valid and result in response. **aaa bbb ccc ddd -** Audio Levels take on values 000-060, which represent actual audio levels of -60 to 0 dBFS. Represent mic input channels in order defined in Channel Number Assignment.
Example(s):	< GET METER_RATE_POSTGATE > < SET METER_RATE_POSTGATE rate > < REP METER_RATE_POSTGATE rate > < REP ERR > < SAMPLE aaa bbb ccc ddd >

Automixer Gain Metering Rate (RMS)

Parameter Name:	METER_RATE_MXR_GAIN
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**rate** is a value from 100 to 99999 representing meter rate in milliseconds. 0 = off Values 1 to 99 are not valid and result in response. **aaa bbb ccc ddd -** Audio Levels take on values 000-060, which represent actual audio levels of -60 to 0 dBFS. Represent mic input channels in order defined in Channel Number Assignment.

Example(s):	< GET METER_RATE_MXR_GAIN > < SET METER_RATE_MXR_GAIN rate > < REP METER_RATE_MXR_GAIN rate > < REP ERR > < SAMPLE aaa bbb ccc ddd >

AEC Reference In Metering Rate (RMS)

Parameter Name:	METER_RATE_AECREF
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**rate** is a value from 100 to 99999 representing meter rate in milliseconds. 0 = off Values 1 to 99 are not valid and result in response. **aaa bbb ccc ddd -** Audio levels take on values 000-060, which represent actual audio levels of −60 to 0 dBFS. Represent mic input channels in order defined in Channel Number Assignment.
Example(s):	< GET METER_RATE_AECREF > < SET METER_RATE_AECREF rate > < REP METER_RATE_AECREF rate > < REP ERR > < SAMPLE aaa >

Audio Gain (Digital)

Parameter Name:	AUDIO_GAIN_HI_RES
Command Types Supported:	GET, SET (INC, DEC), REP
Indexing:	GET **index:** See Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index:** 2 digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0. Setting gain on all channels at once is not supported.
Value(s):	**gain** is in units of one-tenth of a dB. The value is multiplied by 10 and then scaled by 1100. The resulting value has a range of 0 to 1400 representing gain from -110.0 dB to 30.0 dB.

	step is in units of one-tenth of a dB. The resulting gain when the **step** is applied must be in the range allowed in the SET.
Example(s):	< GET index AUDIO_GAIN_HI_RES > < SET index AUDIO_GAIN_HI_RES gain > < SET index AUDIO_GAIN_HI_RES inc step > < SET index AUDIO_GAIN_HI_RES dec step > < REP index AUDIO_GAIN_HI_RES gain > < REP ERR >

Audio Gain Postgate

Parameter Name:	AUDIO_GAIN_POSTGATE
Command Types Supported:	GET, SET, REP
Indexing:	GET **index**: See Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index**: Single digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**gain** is in units of one-tenth of a dB and then scaled by 1100 and takes on values 0 to 1400 representing gain from -109.9 dB to 30.0 dB **step** is in units of one-tenth of a dB. The resulting gain when the **step** is applied must be in the range allowed in the SET.
Example(s):	< GET index AUDIO_GAIN_POSTGATE > < SET index AUDIO_GAIN_POSTGATE gain > < SET index AUDIO_GAIN_POSTGATE inc step > < SET index AUDIO_GAIN_POSTGATE dec step > < REP index AUDIO_GAIN_POSTGATE gain > < REP ERR >

Audio Level (RMS)

Parameter Name:	AUDIO_IN_RMS_LVL
Command Types Supported:	GET, REP
Indexing:	GET **index**: Channel Number Assignment for product-specific channel assignments. 0 = all channels.

	REP **index**: Single digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**###** is audio level in the range of 00-60.
Example(s):	< GET x AUDIO_IN_RMS_LVL > < REP x AUDIO_IN_RMS_LVL ### > < REP ERR >

Audio Level (Peak)

Parameter Name:	AUDIO_IN_PEAK_LVL
Command Types Supported:	GET, REP
Indexing:	GET **index**: Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index**: Single digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**###** is audio level in the range of 000-060.
Example(s):	< GET index AUDIO_IN_PEAK_LVL > < REP index AUDIO_IN_PEAK_LVL ### > < REP ERR >

Device Mute

Parameter Name:	DEVICE_AUDIO_MUTE
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**cmd** is desired mute status and takes on values: ON OFF TOGGLE **sts** is the current mute status for the designated channel and takes on values: ON OFF

Example(s):	< GET DEVICE_AUDIO_MUTE > < SET DEVICE_AUDIO_MUTE cmd > < REP DEVICE_AUDIO_MUTE sts >

Channel Mute

Parameter Name:	AUDIO_MUTE
Command Types Supported:	GET, SET, REP
Indexing:	Where **nn** is the channel and takes on values defined in channel number assignments.
Value(s):	**cmd** is desired mute status and takes on values: ON OFF TOGGLE **sts** is the current mute status for the designated channel and takes on values: ON OFF
Example(s):	< GET nn AUDIO_MUTE > < SET nn AUDIO_MUTE cmd > < REP nn AUDIO_MUTE sts >

Presets

Parameter Name:	PRESET
Command Types Supported:	GET, SET, REP
Indexing:	**##** is the preset number and takes on values 1-10.
Value(s):	n/a
Example(s):	< GET PRESET > < SET PRESET ## > < REP PRESET ## > < REP ERR >

Restore Default Settings

Parameter Name:	DEFAULT_SETTINGS
Command Types Supported:	SET, REP
Indexing:	n/a
Value(s):	## = 00 if restore is successful
Example(s):	< SET DEFAULT_SETTINGS > < REP DEFAULT_SETTINGS ## > < REP ERR >

View Preset Name

Parameter Name:	PRESET_NAME
Command Types Supported:	GET, REP
Indexing:	1-10: specific preset identifier
Value(s):	**name** is a literal string 25 alphanumeric characters long, special characters allowed except blank spaces, {} and < >. Note that if a preset is empty, **name** will say {empty}
Example(s):	< GET PRESET_NAME nn > < REP PRESET_NAME nn name > < REP ERR >

Device Encryption Status

Parameter Name:	ENCRYPTION
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	**sts** is the encryption status, which can have the following values: ON OFF

Example(s):	< GET ENCRYPTION > < REP ENCRYPTION sts > < REP ERR >

Reboot

Note: This command does not send acknowledgement.

Parameter Name:	REBOOT
Command Types Supported:	SET
Indexing:	n/a
Value(s):	n/a
Example(s):	< SET REBOOT >

Get Error Events

Parameter Name:	LAST_ERROR_EVENT
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	Sends the last error logged on the device, as represented by **{str}. {str}** is up to 128 characters long.
Example(s):	< GET LAST_ERROR_EVENT > < REP LAST_ERROR_EVENT {str} > < REP ERR >

PEQ Filter Enable

Parameter Name:	PEQ
Command Types Supported:	GET, SET, REP
Indexing:	GET **index**: See Channel Number Assignment for product-specific channel assignments. 0 = all channels. REP **index**: 2 digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.

	filter is the filter number in the selected PEQ block **index.** 0: all filters.
Value(s):	**sts** is the desired PEQ filter status: ON OFF TOGGLE
Example(s):	< GET index PEQ filter > < SET index PEQ filter sts > < REP index PEQ filter sts > < REP ERR >

Active Mic Channels

Parameter Name:	NUM_ACTIVE_MICS
Command Types Supported:	GET, REP
Indexing:	n/a
Value(s):	**active** represents the number of active channels.
Example(s):	< GET NUM_ACTIVE_MICS > < REP NUM_ACTIVE_MICS active >

Mute LED State

Parameter Name:	DEV_MUTE_STATUS_LED_STATE
Command Types Supported:	GET, REP
Indexing:	n/a

Value(s):	**sts** is current mute LED state that takes on these values:
	ON = MUTED
	OFF = UNMUTED

Example(s):	< GET DEV_MUTE_STATUS_LED_STATE > < REP DEV_MUTE_STATUS_LED_STATE sts > < REP ERR >

Solo Channel to Automix

Parameter Name:	CHAN_AUTOMIX_SOLO_EN
Command Types Supported:	GET, SET, REP
Indexing:	**index:** Mic input channels. See Channel Number Assignment for product-specific channel assignments. 0 = all channels.
Value(s):	**sts** determines the requested state of SOLO mode: 1. ENABLE 2. DISABLE
Example(s):	< GET index CHAN_AUTOMIX_SOLO_EN > < SET index CHAN_AUTOMIX_SOLO_EN sts > < REP index CHAN_AUTOMIX_SOLO_EN sts > < REP ERR >

LED Brightness

Parameter Name:	LED_BRIGHTNESS
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**level** is the desired brightness level and takes on values: 0: Disabled 1: 20% 2: 40% 3: 60% 4: 80% 5: 100%
Example(s):	< GET LED_BRIGHTNESS > < SET LED_BRIGHTNESS level > < REP LED_BRIGHTNESS level > < REP ERR >

LED Mute Indication

Parameter Name:	LED_COLOR_UNMUTED

Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**color:** RED, ORANGE, GOLD, YELLOW, YELLOWGREEN, GREEN, TURQUOISE, POWDERBLUE, CYAN, SKYBLUE, BLUE, PURPLE, LIGHTPURPLE, VIOLET, ORCHID, PINK, WHITE
Example(s):	< GET LED_COLOR_UNMUTED > < SET LED_COLOR_UNMUTED color > < REP LED_COLOR_UNMUTED color > < REP ERR >

LED Color Muted

Parameter Name:	LED_COLOR_MUTED
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**color:** RED, ORANGE, GOLD, YELLOW, YELLOWGREEN, GREEN, TURQUOISE, POWDERBLUE, CYAN, SKYBLUE, BLUE, PURPLE, LIGHTPURPLE, VIOLET, ORCHID, PINK, WHITE
Example(s):	< GET LED_COLOR_MUTED > < SET LED_COLOR_MUTED color > < REP LED_COLOR_MUTED color > < REP ERR >

LED State Muted

Parameter Name:	LED_STATE_MUTED
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**state**: ON, FLASHING, OFF
Example(s):	< GET LED_STATE_MUTED > < SET LED_STATE_MUTED state > < REP LED_STATE_MUTED state > < REP ERR >

LED State Unmuted

Parameter Name:	LED_STATE_UNMUTED
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**state**: ON, FLASHING, OFF
Example(s):	< GET LED_STATE_UNMUTED > < SET LED_STATE_UNMUTED state > < REP LED_STATE_UNMUTED state > < REP ERR >

Device LED In State

Parameter Name:	DEV_LED_IN_STATE
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**sts** indicates device's LED-In state: 1. OFF = Mute 2. ON = Unmute
Example(s):	< GET DEV_LED_IN_STATE > < SET DEV_LED_IN_STATE sts > < REP DEV_LED_IN_STATE sts > < REP ERR >

Bypass All EQ

Parameter Name:	BYPASS_ALL_EQ
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**sts** takes on values: ON OFF TOGGLE

Example(s):	< GET BYPASS_ALL_EQ > < SET BYPASS_ALL_EQ sts > < REP BYPASS_ALL_EQ sts > < REP ERR >

Bypass IntelliMix

Allows you to disable (bypass) and enable all IntelliMix DSP blocks.

Parameter Name:	BYPASS_IMX
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**sts** takes on values: ON OFF TOGGLE
Example(s):	< GET BYPASS_IMX > < SET BYPASS_IMX sts > < REP BYPASS_IMX sts > < REP ERR >

EQ Contour

Parameter Name:	EQ_CONTOUR
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**sts** takes on values: OFF LOWSHELF
Example(s):	< GET EQ_CONTOUR > < SET EQ_CONTOUR sts > < REP EQ_CONTOUR sts > < REP ERR >

Automixer Gate Out Status for Lobes

Parameter Name:	AUTOMIX_GATE_OUT_EXT_SIG
Command Types Supported:	GET, REP
Indexing:	GET **index**: Dante mic input defined in Channel Number Assignment. 0 = all channels. REP **index**: Two digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.
Value(s):	**sts** is current gate out signal for the channel: ON OFF
Example(s):	< GET index AUTOMIX_GATE_OUT_EXT_SIG > < REP index AUTOMIX_GATE_OUT_EXT_SIG sts > < REP ERR >

Device Installation

Parameter Name:	DEVICE_INSTALLATION
Command Types Supported:	GET, SET, REP
Indexing:	n/a
Value(s):	**pos** takes on values: CEILING WALL_HORIZONTAL WALL_VERTICAL TABLE
Example(s):	< GET DEVICE_INSTALLATION > < SET DEVICE_INSTALLATION pos > < REP DEVICE_INSTALLATION pos > < REP ERR >

Lobe Beam Angle

Parameter Name:	BEAM_ANGLE
Command Types Supported:	GET, SET, REP

Indexing:	**index** takes on values 1-8 for 4FT and 1-4 for 2FT.
Value(s):	**position** is the value for the beam angle: -90 degrees to +90 degrees.
Example(s):	< GET index BEAM_ANGLE > < SET index BEAM_ANGLE position > < REP index BEAM_ANGLE position > < REP ERR >

Lobe Beam Width

Parameter Name:	BEAM_W
Command Types Supported:	GET, SET, REP
Indexing:	**index** takes on values 1-8 for 4FT and 1-4 for 2FT
Value(s):	**width** takes on values: NARROW, MEDIUM, and WIDE
Example(s):	< GET index BEAM_W > < SET index BEAM_W width > < REP index BEAM_W width > < REP ERR >