# MXA901-R

## Command Strings

Shure MXA901-R command strings for third-party control systems, such as AMX, Crestron, or Extron. Includes all supported programming commands.

Version: 0.1 (2024-A)

# Table of Contents

# MXA901-R
# 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

MXA901 microphones use these channel numbers whenever a channel index is needed:

- Output: 09
- AEC reference input: 10

## Device Information

Use these commands to get information about the device, reboot, and restore default settings.

### 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 ><br><br>< 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 ><br><br>< 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:<br><br>The value is 3 versions separated by a period.<br>Each version shall be able to take on a value from 0 to 65535.<br>**ver** has an "*****" if the firmware is invalid.<br>**Example**: 65535.65535.65535 |
| **Example(s):** | < GET FW_VER ><br><br>< 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.<br>The length of IP address is 15 characters.<br>The value will be padded to ensure that 15 characters are always returned. |
| **Example(s):** | < GET IP_ADDR_NET_AUDIO_PRIMARY ><br><br>< REP IP_ADDR_NET_AUDIO_PRIMARY ip_addr ><br><br>< 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:<br>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.<br>The length of subnet is 15 characters.<br>The value will be padded to ensure that 15 characters are always returned. |
| **Example(s):** | < GET IP_SUBNET_NET_AUDIO_PRIMARY ><br><br>< REP IP_SUBNET_NET_AUDIO_PRIMARY subnet ><br><br>< 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:<br>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.<br>The length of subnet is 15 characters.<br>The value will be padded to ensure that 15 characters are always returned. |
| **Example(s):** | < GET IP_GATEWAY_NET_AUDIO_PRIMARY ><br><br>< REP IP_GATEWAY_NET_AUDIO_PRIMARY gateway ><br><br>< 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.<br><br>**Example:** 00:0E:DD:FF:F1:63 |

| | |
|---|---|
| **Example(s):** | < GET CONTROL_MAC_ADDR ><br><br>< REP CONTROL_MAC_ADDR addr ><br><br>< REP ERR > |

## Device ID

| | |
|---|---|
| **Parameter Name:** | DEVICE_ID |
| **Command Types Supported:** | GET, REP |
| **Indexing:** | n/a |
| **Value(s):** | Response is a text string.<br>Most devices allow device ID to be up to 31 characters.<br>Value is padded with spaces as needed to ensure that 31 characters are always reported |
| **Example(s):** | < GET DEVICE_ID ><br><br>< 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.<br>Most devices allow device ID to be up to 31 characters.<br>Value is padded with spaces to ensure that 31 characters are always reported. |
| **Example(s):** | < GET NA_DEVICE_NAME ><br><br>< 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.<br>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 ><br><br>< REP index CHAN_NAME string ><br>< 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.<br>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 ><br><br>< REP index NA_CHAN_NAME string ><br>< 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<br><br>ON<br>OFF |
| **Example(s):** | < GET FLASH ><br>< SET FLASH flash_state ><br>< REP FLASH flash_state ><br>< REP ERR > |

## 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 ><br>< SET PRESET ## ><br>< REP PRESET ## ><br>< 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 < >.<br><br>Note that if a preset is empty, **name** will say {empty} |
| **Example(s):** | < GET PRESET_NAME nn ><br>< REP PRESET_NAME nn name ><br>< 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:<br><br>ON<br>OFF |
| **Example(s):** | < GET ENCRYPTION ><br>< REP ENCRYPTION sts ><br>< 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 ><br>< REP DEFAULT_SETTINGS ## ><br>< 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 > |

# Channel and Coverage Area Commands

Use these commands to adjust gain and mute channels or coverage areas.

## 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.<br>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:<br>1. OFF<br>2. ON |
| **Example(s):** | < GET index AUDIO_OUT_CLIP_INDICATOR ><br>< REP index AUDIO_OUT_CLIP_INDICATOR sts ><br>< REP ERR > |

## 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.<br>REP **index:** 2 digit representation of the index sent in the GET, all the appropriate channels if the **index** = 0.<br>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.<br>**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 ><br>< SET index AUDIO_GAIN_HI_RES gain ><br>< SET index AUDIO_GAIN_HI_RES inc step ><br>< SET index AUDIO_GAIN_HI_RES dec step ><br>< REP index AUDIO_GAIN_HI_RES gain ><br>< 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.<br>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 ><br>< REP x AUDIO_IN_RMS_LVL ### ><br>< 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.<br>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 ><br>< REP index AUDIO_IN_PEAK_LVL ### ><br>< 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:<br><br>ON<br>OFF<br>TOGGLE<br><br>**sts** is the current mute status for the designated channel and takes on values:<br><br>ON<br>OFF |
| **Example(s):** | < GET DEVICE_AUDIO_MUTE ><br>< SET DEVICE_AUDIO_MUTE cmd ><br>< 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:<br><br>ON<br>OFF<br>TOGGLE<br><br>**sts** is the current mute status for the designated channel and takes on values:<br><br>ON<br>OFF |
| **Example(s):** | < GET nn AUDIO_MUTE ><br>< SET nn AUDIO_MUTE cmd ><br>< REP nn AUDIO_MUTE sts > |

# Meter Rate Commands

Adjust different meter rates using these commands.

## 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.<br><br>0 = off<br>Values 1 to 99 are not valid and result in response.<br><br>**aaa :** 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 ><br>< SET METER_RATE rate ><br>< REP METER_RATE rate ><br>< REP ERR ><br>< SAMPLE aaa > |

## Metering Rate Pre-Compressor (RMS)

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

| | |
|---|---|
| | < REP ERR ><br>< SAMPLE_PRECOMP aaa > |

### 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.<br><br>0 = off<br>Values 1 to 99 are not valid and result in response.<br><br>**aaa:** 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_AECREF ><br>< SET METER_RATE_AECREF rate ><br>< REP METER_RATE_AECREF rate ><br>< REP ERR ><br>< SAMPLE aaa > |

## LED Status and Behavior

Control status LED color and behavior with these commands.

### 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:<br><br>ON = MUTED<br>OFF = UNMUTED |
| **Example(s):** | < GET DEV_MUTE_STATUS_LED_STATE ><br>< REP DEV_MUTE_STATUS_LED_STATE 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:<br><br>0: Disabled<br>1: 20%<br>2: 40%<br>3: 60%<br>4: 80%<br>5: 100% |
| **Example(s):** | < GET LED_BRIGHTNESS ><br>< SET LED_BRIGHTNESS level ><br>< REP LED_BRIGHTNESS level ><br>< REP ERR > |

## LED Color Unmuted

| | |
|---|---|
| **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 ><br>< SET LED_COLOR_UNMUTED color ><br>< REP LED_COLOR_UNMUTED color ><br>< 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 ><br>< SET LED_COLOR_MUTED color ><br>< REP LED_COLOR_MUTED color ><br>< 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 ><br>< SET LED_STATE_MUTED state ><br>< REP LED_STATE_MUTED state ><br>< 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 ><br>< SET LED_STATE_UNMUTED state ><br>< REP LED_STATE_UNMUTED state ><br>< 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:<br>1. OFF = Mute<br>2. ON = Unmute |
| **Example(s):** | < GET DEV_LED_IN_STATE ><br>< SET DEV_LED_IN_STATE sts ><br>< REP DEV_LED_IN_STATE sts ><br>< REP ERR > |

## Additional Resources

- Shure Knowledge Base FAQs
- Command strings for Shure devices
- Shure Enterprise Networking Troubleshooting Checklist
- Training from the Shure Audio Institute
- Shure Systems YouTube channel

### Download Shure Software

- Shure Designer
- Shure Update Utility
- Shure Web Device Discovery
- Software and firmware archive