ULX-D

ULX-D Command Strings

Third-party command string information for the Shure ULX-D wireless system.

Version: 1.0 (2022-L)

Table of Contents

ULX-D
ULX-D Command Strings

The ULX-D device is connected via Ethernet to a control system, such as

- AMX, Crestron, or Extron
- Symetrix, Biamp, other digital signal processors (DSP)
- Specialized custom programs

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

Conventions

There are 4 types of strings:

GET	Finds the status of a property. After the AMX/Crestron sends a GET command, the system responds with a REPORT string
SET	Changes the status of a property. After the AMX/Crestron sends a SET command, the system responds with a REPORT string to indicate the new value of the property.
REP	When the system receives a GET or SET command, it replies with a REPORT command to indicate the status of the property. REP ERR occurs when a command is improperly formatted or when the values are out of range. **Important:** With the exception of the metered properties, the device sends a REPORT when a value changes. Thus, it is not necessary to constantly query most device properties.
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

The character "x" in all of the following strings represents the channel and can be ASCII numbers 0 through 4 as in the following table.

0	All channels
1 through 4	Individual channels

ULX-D Naming

- ULXD4 is a 1-channel device
- ULXD4D is a 2-channel device (D - Dual)
- ULXD4Q is a 4-channel device (Q - Quad)

Device Command Strings

ALL

Description	Discovery of all report strings for the selected channel(s).
Commands	< GET x ALL > < REP ... >
Variables	When **x** is zero, the device responds with REP for all device-specific properties and ALL channel-related properties including all metered properties. When **x** is a channel number, the device responds with REP for all device-specific properties and ALL channel **x**-related properties including all metered properties.
Notes	This group is only sent once on receipt of the GET ALL command. Examples: < REP x METER_RATE #### > < REP x CHAN_NAME {string} > < REP x ENCRYPTION v > < REP x AUDIO_GAIN ## > < REP x GROUP_CHAN gg cc > < REP x FREQUENCY ###### >

AUDIO_SUMMING_MODE

Description	Controls device-based audio summing mode.
Commands	< GET AUDIO_SUMMING_MODE > < SET AUDIO_SUMMING_MODE value > < REP AUDIO_SUMMING_MODE value >
Variables	None.
Notes	Values can be: OFF - Audio summing mode is disabled. 1+2 - Audio summing mode is enabled for channel pair 1/2. 3+4 - Audio summing mode is enabled for channel pair 3/4. 1+2/3+4 - Only applies to quad Rx. Audio summing mode is enabled for channel pairs 1/2 and 3/4. Ignored by device if not valid. Device reports current status without applying any change. 1+2+3+4 - Only applies to quad Rx. Ignored by device if not valid. Device reports current status without applying any change.

DEVICE_ID

Description	Controls the device ID.
Commands	< GET DEVICE_ID > < REP DEVICE_ID {yyyyyyyy} > < SET DEVICE_ID {yyyyyyyy} > < REP DEVICE_ID {yyyyyyyy} >
Variables	Where the repeating **y** represents or pads the 8-character string
Notes	A blank string represents an unknown device ID.

ENCRYPTION

Description	Controls the encryption mode.
Commands	< GET ENCRYPTION > < REP ENCRYPTION setting > < SET ENCRYPTION setting > < REP ENCRYPTION setting >
Variables	None.
Notes	Valid values for setting are: OFF - Encryption mode is disabled AUTO - Encryption mode is set to automatic control. MANUAL - Encryption mode is set to manual control. Manual must be used to regenerate the encryption key.

ENCRYPTION_REGENERATE_KEY

Description	Controls the regeneration of the encryption key.
Commands	< SET ENCRYPTION_REGENERATE_KEY value > < REP ENCRYPTION_REGENERATE_KEY value >
Variables	None.
Notes	This command is only applicable when the encryption is set to MANUAL. Reminder: Sync devices after updating encryption. Values are: REQUESTED, COMPLETED, INACTIVE Examples: < SET ENCRYPTION_REGENERATE_KEY REQUESTED >

	SET generates three responses. When the command is accepted: < REP ENCRYPTION_REGENERATE_KEY REQUESTED > When the new key is created: < REP ENCRYPTION_REGENERATE_KEY COMPLETED > When the receiver is ready to accept a new request: < REP ENCRYPTION_REGENERATE_KEY INACTIVE >

FLASH

Description	Controls the flash to identify a device or channel.
Commands	< SET FLASH ON > < REP FLASH ON > Device initiates an identify then stops flashing: < REP FLASH OFF > < SET FLASH OFF > < REP FLASH OFF > When used with a channel index, the command initiates a Channel Identify. < SET x FLASH ON > < REP x FLASH ON >
Variables	When used, **x** is the channel number.
Notes	No REP once the flashing has completed. Flash rate and pattern is fixed and device specific.

FREQUENCY_DIVERSITY_MODE

Description	Controls device-based frequency diversity mode.
Commands	< SET FREQUENCY_DIVERSITY_MODE status > < GET FREQUENCY_DIVERSITY_MODE > < REP FREQUENCY_DIVERSITY_MODE status >
Variables	None.
Notes	Values for the status can be: OFF - Frequency diversity mode is disabled. 1+2 - Frequency diversity mode is enabled for channel pair 1/2.

	3+4 - Frequency diversity mode is enabled for channel pair 3/4. Only applies to quad Rx. Ignored by device if not valid. Device reports current status without applying any change. 1+2/3+4 - Frequency diversity mode is enabled for channel pairs 1/2 and 3/4. Only applies to quad Rx. Ignored by device if not valid. Device reports current status without applying any change. Reports when channel designator is invalid for the device.

FW_VER

Description	Discovers the firmware version.
Commands	Self test passed: < GET FW_VER > < REP FW_VER {ver} >
Variables	None.
Notes	Version is a 24-character string. Package version number reported as Maj.Min.Pack.Build with values from 00000-65535. Example: 65535.65535.65535.65535* If the version is followed by an * the previous firmware update was incomplete. Please use Shure Update Utility to re-update the device.

HIGH_DENSITY

Description	Controls high-density mode.
Commands	< SET HIGH_DENSITY value > < GET HIGH_DENSITY > < REP HIGH_DENSITY value >
Variables	None.
Notes	Values can be: OFF - High-density mode is disabled ON - High-density mode is enabled

MODEL

Description	Discovery of the device model name.

Commands	< GET MODEL > < REP MODEL {string} >
Variables	None.
Notes	Format: 32-character string.

NA_DEVICE_NAME

Description	Discovers the Dante device name on dual and quad devices.
Commands	< GET NA_DEVICE_NAME > < REP NA_DEVICE_NAME {string} >
Variables	None.
Notes	The device always responds with 31-character ID. In these examples, the repeating y pads these strings, representing white space. Getting the Dante Device Name from a ULXD4Q that uses Yamaha naming conventions: < GET NA_DEVICE_NAME > < REP NA_DEVICE_NAME {Y001-Shure-ULXD4Q-435577yyyyyyy} > Getting the name from a device changed by Dante Controller to "GuitarRig3:" < GET NA_DEVICE_NAME {GuitarRig3} > < REP NA_DEVICE_NAME {GuitarRig3yyyyyyyyyyyyyyyyyyyyy} >

SCAN_LOCK

Description	Controls the scan lock.
Commands	< GET SCAN_LOCK > < REP SCAN_LOCK OFF > < SET SCAN_LOCK ON > < REP SCAN_LOCK ON >
Variables	None.
Notes	When SCAN_LOCK is ON: The front panel scan button is ignored. This device prevents any changes if another device deploys a group scan.

SYNC_LOCK

Description	Controls the sync lock.
Commands	< GET SYNC_LOCK > < REP SYNC_LOCK OFF > < SET SYNC_LOCK ON > < REP SYNC_LOCK ON >
Variables	None.
Notes	When SYNC_LOCK is ON: The front panel sync buttons are ignored. This device prevents IR Sync functions.

NET_SETTINGS

Description	Control for the network settings.
Commands	< GET NET_SETTINGS interface > < REP NET_SETTINGS interface ipMode ipAddr subnetMask gwAddr macAddr > < SET NET_SETTINGS interface ipMode ipAddr subnetMask gwAddr >
Variables	*interface*: SC, D1, D2 (Where SC = Shure Control, D1 = Dante Primary, D2 = Dante Secondary) *ipMode*: AUTO, MANUAL *ipAddr*: aaa.aaa.aaa.aaa *subnetMask*: bbb.bbb.bbb.bbb *gwAddr*: ccc.ccc.ccc.ccc (where 000.000.000.000 means no gateway is used) *macAddr*: xx:xx:xx:xx:xx:xx (in REP only)
Notes	When setting new Shure Control network settings you must reconnect to the device via its new IP address. When setting Dante network settings the device reboots on its own to apply the new settings. You must reconnect to the device. When setting network settings to “AUTO” mode you must use “na” for the IP address, Subnet Mask and Gateway address. Dante Secondary settings are only applicable when the device is operating with switch configuration: Split/Redundant. There are no asynchronous REPorts for the NET_SETTINGS. Use GET to retrieve the most up to date values. For the ULXD4, only “SC” is applicable.

Examples	Example of getting the current network settings for the Shure Control interface: < GET NET_SETTINGS SC > < REP NET_SETTINGS SC AUTO 192.168.001.025 255.255.255.000 000.000.000.000 00:0E:DD:45:60:EB > Example of setting the Shure Control network settings to MANUAL settings: < SET NET_SETTINGS SC MANUAL 192.168.1.123 255.255.255.0 192.168.1.1 > After you change the Shure Control network settings you must reconnect at the new IP address. Then you can get/confirm the new settings. < GET NET_SETTINGS SC > < REP NET_SETTINGS SC MANUAL 192.168.001.123 255.255.255.000 192.168.001.001 00:0E:DD:45:60:EB > Example of setting the Shure Control network settings to AUTO settings: < SET NET_SETTINGS SC AUTO na na na > After you change the Shure Control network settings you must reconnect at the new IP address. Then you can get/confirm the new settings. < GET NET_SETTINGS SC > < REP NET_SETTINGS SC AUTO 192.168.001.021 255.255.255.000 000.000.000.000 00:0E:DD:45:60:EB > Example of setting the Dante Primary interface to manual network settings: < SET NET_SETTINGS D1 MANUAL 10.10.1.15 255.255.255.0 10.10.1.1 > After the reboot you must reconnect, then you can get/confirm the new settings. < GET NET_SETTINGS D1 > < REP NET_SETTINGS D1 MANUAL 010.010.001.015 255.255.255.000 010.010.001.001 00:0E:DD:FC:B9:6F >

Channel Command Strings

AUDIO_GAIN

Description	Control for the channel audio gain.
Commands	< GET x AUDIO_GAIN > < REP x AUDIO_GAIN ### > < SET x AUDIO_GAIN # > < REP x AUDIO_GAIN ### > To decrement the value: < SET x AUDIO_GAIN DEC # > < REP x AUDIO_GAIN ### > To increment the value: < SET x AUDIO_GAIN INC # > < REP x AUDIO_GAIN ### >
Variables	Where **x** is the channel number and **#** represents dB.
Notes	000 to 060 in increments of 1 The values REPorted and SET are offset by 18 Actual range: -18 to 42 dB in 1 dB steps

AUDIO_MUTE

Description	Controls the channel audio mute output.
Commands	< GET x AUDIO_MUTE > < REP x AUDIO_MUTE OFF > < SET x AUDIO_MUTE ON > < REP x AUDIO_MUTE ON > < SET x AUDIO_MUTE OFF > < REP x AUDIO_MUTE OFF > < SET x AUDIO_MUTE TOGGLE > < REP x AUDIO_MUTE ON >
Variables	Where **x** is the channel number.
Notes	TOGGLE switches between ON and OFF.

CHAN_NAME

Description	Control for the channel name.
Commands	< GET x CHAN_NAME > < REP x CHAN_NAME {yyyyyyyy} > < SET x CHAN_NAME {yyyyyyyy} > < REP x CHAN_NAME {yyyyyyyy} >
Variables	Where **x** is the channel number. Where the repeating **y** represents or pads the 8-character string from the set: A-Z,a-z, 0-9,!"#$%&'()*+,-./:;<=>?@[\]^_`~ and space.
Notes	The device always responds with a 8-character name.

ENCRYPTION_WARNING

Description	Discovers an encryption mismatch status.
Commands	< GET x ENCRYPTION_WARNING > < REP x ENCRYPTION_WARNING status >
Variables	Where **x** is the channel number.
Notes	Values for status are: OFF - No mismatch detected ON - Mismatch detected.

FREQUENCY

Description	Controls frequency settings.
Commands	< SET x FREQUENCY ###### > < GET x FREQUENCY ###### > < REP x FREQUENCY ###### > < REP x GROUP_CHAN gg,cc >
Variables	Where **x** is the channel, ###### is the frequency in KHz, **gg** is group, **cc** is channel.
Notes	Invalid group or channel are reported as "--". Group and channel report only sent on change.

GROUP_CHAN

Description	Controls the group and channel.
Commands	< SET x GROUP_CHAN gg,cc > < GET x GROUP_CHAN > < REP x FREQUENCY ###### > < REP x GROUP_CHAN gg,cc >
Variables	Where: **x** is the channel number. ###### is the frequency in KHz (6 digits long) **gg** is a two-character group **cc** is the two-character channel.
Notes	GETs RF Group & Channel setting for channel x. REPorts group and channel and frequency for channel x. Frequency only reported on change. Invalid group or channel is reported as "--"

NA_CHAN_NAME

Description	Discovers Network Audio (Dante) channel name for dual and quad devices.
Commands	< GET x NA_CHAN_NAME > < REP x NA_CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >
Variables	Where **x** is the channel number. Where the repeating **y** pads the 31-character channel name with spaces.
Notes	None.

RF_INT_DET

Description	Reports when interference is detected.
Commands	< GET x RF_INT_DET > < REP x RF_INT_DET status >
Variables	Where **x** is the channel number.
Notes	Valid values for status are:

	NONE - No interference detected CRITICAL - Interference detected

Metering Command Strings

Most properties generate REP (Report) messages when their values change (for example, Frequency or Channel Name) For audio meters, RF meters, and the like, a REP on each change is inefficient and can flood many simple control systems.

The Shure approach is to use metering to periodically sample your channels and devices:

- You can still use GET to discover a value when necessary.
- The SAMPLE command packs multiple attributes into a single message.
- You can control the interval of the messages by setting the METER_RATE.

The following sections detail METER_RATE and SAMPLE followed by the set of metered attributes.

METER_RATE / SAMPLE

Description	Controls the meter rate and reports sampled data.
Commands	< GET x METER_RATE > < SET x METER_RATE ##### > < REP x METER_RATE ##### > < SAMPLE x ALL nn aaa eee >
Variables	Where **x** is the channel number. Where ##### is the rate in milliseconds. Where **nn** indicates the diversity LEDs from the receiver. (See RF_ANTENNA) Where **aaa** is the value of the RF level received on antenna in-use. (See RX_RF_LVL) Where **eee** is the audio level and is 000-255. (See AUDIO_LVL)
Notes	00100 to 99999 - The interval of each SAMPLE report in milliseconds. 00100 - Sample every 100 millisecond (10 samples per sec) 01000 - Sample every second 05000 - Sample every 5 seconds To turn off metering, use < SET x METER_RATE 00000 >

RF_ANTENNA

Description	Discovers RF antenna status and receiver squelch status.

Commands	< GET x RF_ANTENNA > < REP x RF_ANTENNA nn >
Variables	Where **x** is the channel number. Where **nn** indicates the RF LEDs from the receiver.
Notes	This is a metered property. It does not generate a report on a value change. Receiver squelch status: AX - Antenna A is on, Antenna B is off XB - Antenna A is off, Antenna B is on AB - Antenna A is on, Antenna B is on (not possible on ULXD4) XX - Antenna A is off, Antenna B is off

RX_RF_LVL

Description	Discovers the RF received level.
Commands	< GET x RX_RF_LVL > < REP x RX_RF_LVL rssi >
Variables	Where **x** is the channel.
Notes	This is a metered property using SAMPLE; it does not generate a report on a value change. Numeric, 3-character fixed output per antenna Units: dBm The actual value = the reported value - 128 Value range as reported over command strings: 000 - 255 Value range after conversion to the actual value: -128 - 127 dBm

AUDIO_LVL

Description	Discovery of the receiver RMS audio level.
Commands	< GET x AUDIO_LVL > < REP x AUDIO_LVL ### >
Variables	Where **x** is the channel number and ### is a 3-digit number between 000 and 255.
Notes	This is a metered property using SAMPLE; it does not generate a report on a value change. When metering is enabled, this value is sent at the metered sample rate. Units: dBFS The actual value = the reported value - 50

Side Channel Command Strings

BATT_BARS

Description	Discovers the battery bars.
Commands	< GET x BATT_BARS > < REP x BATT_BARS ### >
Variables	Where **x** is the channel number and ### are the bars.
Notes	Numeric, 3-character fixed output 000 to 005 - remaining charge 255 - Unknown

BATT_CHARGE

Description	Discovers the charge left in rechargeable batteries.
Commands	< GET x BATT_CHARGE > < REP x BATT_CHARGE ### >
Variables	Where **x** is the channel number and ### represents the percent of charge remaining.
Notes	Numeric, 3-character fixed output 000 to 100 - Percent 255 - Unknown

BATT_CYCLE

Description	Discovers the number of battery cycles as reported via side channel.
Commands	< GET x BATT_CYCLE > < REP x BATT_CYCLE ##### >
Variables	Where **x** is the channel number and ##### is the cycle count of full charges.
Notes	Numeric, 5-character fixed output 00000 to 65534 - Number of cycles

	65535 - Unknown

BATT_HEALTH

Description	Discovers the battery health of rechargeable batteries.
Commands	< GET x BATT_HEALTH > < REP x BATT_HEALTH ### >
Variables	Where **x** is channel number and **###** is the battery health in percent.
Notes	Numeric, 3-character fixed output 000 to 100 - Percent 255 - Unknown

BATT_RUN_TIME

Description	Discovers the minutes of run time remaining in a rechargeable microphone battery.
Commands	< GET x BATT_RUN_TIME > < REP x BATT_RUN_TIME ##### >
Variables	Where **x** is the channel number and **#####** is the remaining run time.
Notes	Valid only for microphone units in use, in-range, and out of charger. Numeric, 5-character fixed output 00000 to 65532 - Number of minutes of runtime 65533 - Battery communication error 65534 - Battery time calculating 65535 - Unknown, or not applicable Reports when data becomes known. For example, 2 hours 5 minutes would be: < REP x BATT_RUN_TIME 00125 >

BATT_TEMP_C

Description	Discovers the battery temperature in Celsius.
Commands	< GET x BATT_TEMP_C > < REP x BATT_TEMP_C ### >

Variables	Where **x** is the channel number and ### is the temperature offset by 40.
Notes	Numeric, 3-character fixed output 000 to 254 - Temperature 255 - Unknown, or not applicable The actual value = the reported value - 40.

BATT_TEMP_F

Description	Discovers the battery temperature in Fahrenheit.
Commands	< GET x BATT_TEMP_F > < REP x BATT_TEMP_F ### >
Variables	Where **x** is the channel number and ### is the temperature offset by 40.
Notes	Numeric, 3-character fixed output 000 to 254 - Temperature 255 - Unknown, or not applicable The actual value = the reported value - 40.

BATT_TYPE

Description	Discovers the battery type.
Commands	< GET x BATT_TYPE > < REP x BATT_TYPE batt >
Variables	Where **x** is the channel number.
Notes	GET supports determination of battery remaining minutes. Values for batt can be: LION ALKA NIMH LITH WARN - Tx is charging on dock or via USB. Only applies if battery is not LION. UNKN - Either no transmitter or not supported by transmitter

TX_DEVICE_ID

Description	Discovers the transmitter's Device ID.

Commands	< GET x TX_DEVICE_ID > < REP x TX_DEVICE_ID {yyyyyyyy} >
Variables	Where **x** is the channel number. Where the repeating **y** represents or pads the 8-character string
Notes	8-character string - Device ID of the transmitter being received Blank, all spaces - Unknown, not applicable, or there is no transmitter

TX_FW_VER

Description	Discovers the transmitter firmware version.
Commands	< GET x TX_FW_VER > < REP x TX_FW_VER {ver} >
Variables	Where **x** is the channel.
Notes	Package version number reported as Maj.Min.Pack.Build. If the version is followed by an * the previous firmware update was incomplete. Please use Shure Update Utility to re-update the device. The REP returns a blank, padded string if the firmware version is unknown or if there is no transmitter.

TX_MENU_LOCK

Description	Discovers the transmitter menu lock status.
Commands	< GET x TX_MENU_LOCK > < REP x TX_MENU_LOCK status >
Variables	Where **x** is the channel number.
Notes	Valid values for status are: ON - Transmitter menu is locked OFF - Transmitter menu is unlocked UNKN - Transmitter menu status is unknown, not applicable, or there is no transmitter

TX_MUTE_BUTTON_STATUS

Description	Discovers the transmitter mute button status of ULXD6 and ULXD8 transmitters.

Commands	< GET x TX_MUTE_BUTTON_STATUS > < REP x TX_MUTE_BUTTON_STATUS status >
Variables	Where **x** is the channel index.
Notes	Values for status are: PRESSED - Mute button is on, audio is muted RELEASED - Mute button is off, audio is unmuted UNKN - Button status is unknown, not applicable, or there is no transmitter

TX_MUTE_STATUS

Description	Discovers the transmitter mute status of the ULXD6 and ULXD8 transmitters.
Commands	< GET x TX_MUTE_STATUS > < REP x TX_MUTE_STATUS status >
Variables	Where **x** is the channel number.
Notes	Values for status are: ON - Tx is muted. OFF - Tx is not muted. UNKN - Unknown, not applicable, or there is no transmitter.

TX_OFFSET

Description	Discovery of the transmitter offset.
Commands	< GET x TX_OFFSET > < REP x TX_OFFSET ### >
Variables	Where **x** is the channel number.
Notes	Numeric, 3-character fixed Range - 000 to 033 255 - Unknown The actual value = the reported value - 12. This means that the actual range is -12 to +21 dB in 1 dB increments.

TX_POWER_SOURCE

Description	Discovers the power source of the ULXD6 and ULXD8 transmitters.

Commands	< GET x TX_POWER_SOURCE > < REP x TX_POWER_SOURCE value >
Variables	Where **x** is the channel number.
Notes	Values can be: BATTERY - Powered by batteries EXTERNAL - Power source is external UNKNOWN - Power Source is unknown, not applicable or there is no transmitter

TX_PWR_LOCK

Description	Discovers the transmitter power lock status.
Commands	< GET x TX_PWR_LOCK > < REP x TX_PWR_LOCK status >
Variables	Where **x** is the channel number.
Notes	Values for status can be: ON - Power lock is on and the power switch is disabled OFF - Power lock is off (default) UNKN - Power lock status is unknown

TX_RF_PWR

Description	Discovers the transmitter RF power.
Commands	< GET x TX_RF_PWR > < REP x TX_RF_PWR pwr >
Variables	Where **x** is the channel number.
Notes.	Values for pwr can be: LOW NORMAL HIGH UNKN

TX_TYPE

Description	Discovers the transmitter type.

Commands	< GET x TX_TYPE > < REP x TX_TYPE model >
Variables	Where **x** is the channel number.
Notes	Values for model can be: QLXD1 QLXD2 ULXD1 ULXD2 ULXD6 ULXD8 UNKN - Either no transmitter or not supported by transmitter