



# SLX-D

## Command Strings

Third-party command strings for Shure SLX-D wireless system  
Version: 2 (2020-G)

---

# Table of Contents

<b>SLX-DCommand Strings</b>	<b>3</b>	<b>Channel Command Strings</b>	<b>6</b>
<b>Device Command Strings</b>	<b>3</b>	<b>Metering Command Strings</b>	<b>9</b>
		<b>Side Channel Command Strings</b>	<b>12</b>

# SLX-D

## Command Strings

The SLX-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:

<b>GET</b>	Finds the status of a property. After the AMX/Crestron sends a GET command, the system responds with a REPORT string
<b>SET</b>	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.
<b>REP</b>	When the system receives a GET or SET command, it replies with a REPORT command to indicate the status of the property.  <b>Important:</b> 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.
<b>SAMPLE</b>	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.

<b>0</b>	All channels
<b>1, 2</b>	Individual channels

## Device Command Strings

### ALL

<b>Description</b>	Discovery of device properties.
--------------------	---------------------------------

<b>Commands</b>	< GET x ALL > < REP ... >
<b>Variables</b>	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.
<b>Notes</b>	None.

## FLASH

<b>Description</b>	Controls the flash to identify a device or channel.
<b>Commands</b>	< SET FLASH ON > < REP FLASH ON >  Device initiates an identify then stops flashing:  < REP FLASH OFF >  <b>Note:</b> When used with no channel index the command initiates a Device Identify. When used with a channel index the command initiates a Channel Identify.  < SET x FLASH ON > < REP x FLASH ON >
<b>Variables</b>	When used, x is the channel number.
<b>Notes</b>	None.

## MODEL

<b>Description</b>	Discovery of the model name of the device.
<b>Commands</b>	< GET MODEL > < REP MODEL {SLXD4yyyyyyyyyyyyyyyyyyyyyyyyyyyy} >
<b>Variables</b>	Where the repeating y represents the spaces returned by the device to pad the model name to 32 characters.
<b>Notes</b>	The device always responds with a 32-character model name.

## DEVICE\_ID

<b>Description</b>	Controls the Device ID.
--------------------	-------------------------

<b>Commands</b>	<pre>&lt; GET DEVICE_ID &gt; &lt; REP DEVICE_ID {SLXD4yyy} &gt;  &lt; SET DEVICE_ID {Name1} &gt; &lt; REP DEVICE_ID {Name1yyy} &gt;</pre>
<b>Variables</b>	Where the repeating <b>y</b> represents the spaces returned by the device to pad the Device ID to 8 characters.
<b>Notes</b>	<p>The device always responds with 8-character ID.</p> <p>SET accepts 1-8 Characters from the set: A-Z,a-z,0-9,!"#\$%&amp;'()*+,-./:;&lt;=&gt;@[\\]^_`~ and space.</p>

## RF\_BAND

<b>Description</b>	Discovery of the RF band.
<b>Commands</b>	<pre>&lt; GET RF_BAND &gt; &lt; REP RF_BAND {G55yyyyy} &gt;</pre>
<b>Variables</b>	Where the repeating <b>y</b> represents the spaces returned by the device to pad the response to 8 characters.
<b>Notes</b>	None.

## LOCK\_STATUS

<b>Description</b>	Discovery of the transmitter Lock.
<b>Commands</b>	<pre>&lt; GET LOCK_STATUS &gt; &lt; REP LOCK_STATUS ALL &gt;</pre>
<b>Variables</b>	None.
<b>Notes</b>	<p>Report responses:</p> <p>OFF MENU ALL</p>

## FW\_VER

<b>Description</b>	Discovery of the firmware version.
<b>Commands</b>	Self test passed:

	<pre>&lt; GET FW_VER &gt; &lt; REP FW_VER {2.0.15.2yyyyyyyyyyyyyyyy} &gt;  Self test failed:  &lt; GET FW_VER &gt; &lt; REP FW_VER {2.0.15.2*yyyyyyyyyyyyyyyy} &gt;</pre>
<b>Variables</b>	Where the repeating <b>y</b> represents the spaces returned by the device to pad the response to 24 characters.
<b>Notes</b>	Package version number reported as Maj.Min.Pack.Build.

## Channel Command Strings

### CHAN\_NAME

<b>Description</b>	Control for the channel name.
<b>Commands</b>	<pre>&lt; GET x CHAN_NAME &gt; &lt; REP x CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} &gt;  &lt; SET x CHAN_NAME {Lead Vox} &gt; &lt; REP x CHAN_NAME {Lead Voxyyyyyyyyyyyyyyyyyyyy} &gt;</pre>
<b>Variables</b>	<p>Where <b>x</b> is the channel number.</p> <p>Where the repeating <b>y</b> represents or pads the 31-character string from the set: A-Z,a-z, 0-9,!"#\$\$%&amp;'()*+,-./:;&lt;=&gt;?@[^_`~ and space, that is, {1234567890123456789012345678901}.</p>
<b>Notes</b>	<p>SET only supports 8 characters.</p> <p>The device always responds with a 31-character name.</p>

### AUDIO\_GAIN

<b>Description</b>	Control for the channel audio gain.
<b>Commands</b>	<pre>&lt; GET x AUDIO_GAIN &gt; &lt; REP x AUDIO_GAIN 030 &gt;</pre> <p>There is an offset of 18 so the actual value = 30 - 18 = 12 dB.</p> <p>To set to 22 dB:</p>

	<pre>&lt; SET x AUDIO_GAIN 40 &gt; &lt; REP x AUDIO_GAIN 040 &gt;</pre> <p>To decrement the value down 5 dB:</p> <pre>&lt; SET x AUDIO_GAIN DEC 5 &gt; &lt; REP x AUDIO_GAIN 035 &gt;</pre> <p>To increment the value up 10 dB:</p> <pre>&lt; SET x AUDIO_GAIN INC 10 &gt; &lt; REP x AUDIO_GAIN 045 &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>Numeric  3 Characters  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</p>

## AUDIO\_OUT\_LVL\_SWITCH

<b>Description</b>	Discovery of the current Mic/Line switch status.
<b>Commands</b>	<pre>&lt; GET x AUDIO_OUT_LVL_SWITCH &gt; &lt; REP x AUDIO_OUT_LVL_SWITCH MIC &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>"MIC" - Audio output is at microphone level (default)  "LINE" - Audio output is at line level</p>

## GROUP\_CHANNEL

<b>Description</b>	Controls the group channel mappings.
<b>Commands</b>	<pre>&lt; GET x GROUP_CHANNEL &gt; &lt; REP x GROUP_CHANNEL {1,1yy} &gt;</pre> <pre>&lt; SET x GROUP_CHANNEL {6,100} &gt; &lt; REP x FREQUENCY 0652875 &gt; &lt; REP x GROUP_CHANNEL {6,100} &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.

	Where the repeating <b>y</b> represents the spaces returned by the device to pad the response to 5 characters.
<b>Notes</b>	<p>Device always returns a 5-character string  Refer to the Group/Channel mappings corresponding to the RF Band and Transmission Mode of the device  You must parse the "," from within the reported value  Characters before the "," are the Group ID Characters after the "," are the Channel ID  The value: "--,--" is the wildcard indicating no GROUP_CHANNEL value is set  You cannot SET to "--,--"</p> <p><b>Note:</b> GROUP_CHANNEL and FREQUENCY are related as described in FREQUENCY.</p>

## FREQUENCY

<b>Description</b>	Controls frequency settings.
<b>Commands</b>	<pre>&lt; GET x FREQUENCY &gt; &lt; REP x FREQUENCY 0578350 &gt;  &lt; SET x FREQUENCY 602125 &gt; &lt; REP x GROUP_CHANNEL {--,--yyyy} &gt; (Note 1) &lt; REP x FREQUENCY 0602125 &gt;</pre>
<b>Variables</b>	<p>Where <b>x</b> is the channel number.</p> <p>Where the repeating <b>y</b> represents the spaces returned by the device to pad the response to 7 characters.</p>
<b>Notes</b>	<p>Frequency and Group Channel</p> <p>FREQUENCY - Device always returns a 7-character, numeric string  GROUP_CHANNEL - Device always returns a 10-character string  Range and Step per the RF Band</p> <ol style="list-style-type: none"> <li>1. GROUP_CHANNEL and FREQUENCY are related: <ol style="list-style-type: none"> <li>a. Setting FREQUENCY results in the GROUP_CHANNEL value reverting to "--,--" if not already indicating "--,--" in addition to the FREQUENCY value.</li> <li>b. Setting GROUP_CHAN results in the corresponding FREQUENCY value being reported in addition to the GROUP_CHAN value.</li> </ol> </li> </ol> <p>Commands:</p> <p>Starting from a default condition:</p> <pre>&lt; GET x GROUP_CHANNEL &gt; &lt; REP x GROUP_CHANNEL {1,1yyyyyyy} &gt; &lt; GET x FREQUENCY &gt; &lt; REP x FREQUENCY 0606025 &gt;</pre> <p>SET the FREQUENCY to some new value:</p>

```
< SET x FREQUENCY 620000 >
< REP x GROUP_CHANNEL {--,--yyyy} >
< REP x FREQUENCY 0620000 >
```

Similarly, when setting GROUP\_CHANNEL, the corresponding FREQUENCY is reported:

```
< SET x GROUP_CHANNEL {6,6} >
< REP x FREQUENCY 0614650 >
< REP x GROUP_CHANNEL {6,6yyyyyy} >
```

- For channels operating in FD-C mode, there is a second FREQUENCY2 and GROUP\_CHANNEL2 that behave in the same manner as FREQUENCY.

```
< GET x FD_MODE >
< REP x FD_MODE FD-C >
```

```
< GET x FREQUENCY >
< REP x FREQUENCY 0578350 >
```

```
< GET x FREQUENCY2 >
< REP x FREQUENCY2 0578850 >
```

## Metering Command Strings

The majority of properties generate REP (Report) messages when their values change (for example, Frequency, Channel Name, and so on.)

For attributes such as audio meters, RF meters, channel quality 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.
- Combine the metered attributes into a single SAMPLE message per channel.

for example,

```
< SAMPLE chNum ALL qual audBitmap audPeak audRms rfAntStats rfBitmapA rfRssiA rfBitmapB rfRssiB >
```

returns

```
< SAMPLE 1 ALL 005 000 045 062 BB 31 099 31 085 >
```

where each field is documented and easy to parse.

- Generate periodic SAMPLE messages at the interval set via the METER\_RATE.

**Note:** To turn off sampling, use

```
< SET x METER_RATE 00000 >
```

where **x** is the channel number.

The following sections detail METER\_RATE and SAMPLE followed by the set of metered attributes.

## METER\_RATE

<b>Description</b>	Controls the meter rate.
<b>Commands</b>	<pre>&lt; GET x METER_RATE &gt; &lt; REP x METER_RATE 00000 &gt;  &lt; SET x METER_RATE 01000 &gt; &lt; REP x METER_RATE 01000 &gt;</pre> <p><b>Note:</b> This results in one SAMPLE every second.</p>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>Numeric, 5-character fixed output.  00000 - Metering OFF (default)  00100 to 65535 - The interval of each SAMPLE report in milliseconds.</p> <p>For example,</p> <p>00100 - Sample every 10 millisecond (10 samples per sec)</p> <p>01000 - Sample every second</p> <p>05000 - Sample every 5 seconds</p>

## SAMPLE

<b>Description</b>	Specifies which of the Metering Commands you want to sample.																
<b>Commands</b>	<p>Sample input:</p> <p><b>Note:</b> The sample input that follows is actually 1 line of input spaced to show correlation.</p> <pre>&lt; SAMPLE chNum ALL audPeak audRms rfRssi &gt;</pre> <p>Sample response:</p> <pre>&lt; SAMPLE 1 ALL 102 102 086 &gt;</pre>																
<b>Variables</b>	<p>Key mapping:</p> <table border="1"> <thead> <tr> <th>Key</th> <th>Corresponding command string for value format</th> <th>Num Char</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td><b>audPeak</b></td> <td>AUDIO_LEVEL_PEAK</td> <td>3</td> <td></td> </tr> <tr> <td><b>audRms</b></td> <td>AUDIO_LEVEL_RMS</td> <td>3</td> <td></td> </tr> <tr> <td><b>rfRssi</b></td> <td>RSSI</td> <td>3</td> <td></td> </tr> </tbody> </table>	Key	Corresponding command string for value format	Num Char	Notes	<b>audPeak</b>	AUDIO_LEVEL_PEAK	3		<b>audRms</b>	AUDIO_LEVEL_RMS	3		<b>rfRssi</b>	RSSI	3	
Key	Corresponding command string for value format	Num Char	Notes														
<b>audPeak</b>	AUDIO_LEVEL_PEAK	3															
<b>audRms</b>	AUDIO_LEVEL_RMS	3															
<b>rfRssi</b>	RSSI	3															

<b>Notes</b>	None.
--------------	-------

## AUDIO\_LEVEL\_PEAK

<b>Description</b>	Discovery of peak audio level.
<b>Commands</b>	< GET x AUDIO_LEVEL_PEAK > < REP x AUDIO_LEVEL_PEAK 102 >
<b>Variables</b>	Where x is the channel number
<b>Notes</b>	This is a metered property using SAMPLE; it does not generate a report on a value change. Numeric, 3-character fixed output Units: dBFS The actual value = the reported value - 120 Value range as reported over command strings: 000 - 120 Value range after conversion to the actual value: -120 - 0 dBFS  The SLX-D values fall in the range of about -100 to 0 dBFS.

## AUDIO\_LEVEL\_RMS

<b>Description</b>	Discovery of the RMS audio level.
<b>Commands</b>	< GET x AUDIO_LEVEL_RMS > < REP x AUDIO_LEVEL_RMS 102 >
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	This is a metered property using SAMPLE; it does not generate a report on a value change. Format: Numeric, 3-character fixed output Units: dBFS The actual value = the reported value - 120 Value range as reported over command strings: 000 - 120 Value range after conversion to the actual value: -120 - 0 dBFS  The SLX-D values fall in the range of about -100 to 0 dBFS.

## RSSI

<b>Description</b>	Discovery of the RSSI.
<b>Commands</b>	< GET x RSSI 0 > < REP x RSSI 1 083 > < REP x RSSI 2 064 >
<b>Variables</b>	Where x is the channel number.

<b>Notes</b>	<p>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 - 120          Value range as reported over command strings: 000 - 120          Value range after conversion to the actual value: -120 - 0 dBm</p>
--------------	---

## Side Channel Command Strings

### TX\_MODEL

<b>Description</b>	Discovery of the transmitter model.
<b>Commands</b>	<p>&lt; GET x TX_MODEL &gt;          &lt; REP x TX_MODEL UNKNOWN &gt;</p> <p>Report of a transmitter being received:          &lt; REP x TX_MODEL SLXD2 &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Report responses:</p> <p>SLXD1          SLXD2          UNKNOWN</p>

### TX\_BATT\_MINS

<b>Description</b>	Discovery of the transmitter battery runtime minutes.
<b>Commands</b>	<p>&lt; GET x TX_BATT_MINS &gt;          &lt; REP x TX_BATT_MINS 65535 &gt;</p> <p>Report when data becomes known (example is 2 hours 5 minutes):          &lt; REP x TX_BATT_MINS 00125 &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Numeric, 5-character fixed output</p> <p>00000 to 65532 - Number of minutes of runtime</p> <p>65533 - Battery communication warning</p>

	65534 - Battery time calculating
	65535 - Unknown, or not applicable

## TX\_BATT\_BARS

<b>Description</b>	Discovery of the transmitter Battery Bars.
<b>Commands</b>	<p>&lt; GET x TX_BATT_BARS &gt;</p> <p>&lt; REP x TX_BATT_BARS 255 &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_BATT_BARS 004 &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 to 005</p> <p>255 - Unknown</p>