



# Axient Digital

## Command Strings

Third-party command string information for the Shure Axient Digital wireless system.  
Version: 4 (2019-L)

---

# Table of Contents

<b>Axient DigitalCommand Strings</b>	<b>3</b>	<b>Metering Command Strings</b>	<b>11</b>
<b>Device Command Strings</b>	<b>3</b>	<b>Side Channel Command Strings</b>	<b>18</b>
<b>Channel Command Strings</b>	<b>6</b>	<b>Transmitter Slots Command Strings</b>	<b>25</b>

# Axient Digital Command Strings

The Axient Digital 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 through 4</b>	Individual channels

## Axient Digital Naming

- AD4Q - Is a 4 channel device (Q - Quad)
- AD4D - is a 2 channel device (D - Dual)
- AD4 channels have 8 transmitter slots
- AD4 channels typically have 1 RF section and 2 Antenna inputs.

# Device Command Strings

## ALL

<b>Description</b>	Discovery of device properties.
<b>Commands</b>	< GET x ALL > < REP ... >
<b>Variables</b>	When <b>x</b> is zero, the device responds with REP for all device-specific properties and ALL channel-related properties including all metered properties.  When <b>x</b> is a channel number, the device responds with REP for all device-specific properties and ALL channel <b>x</b> -related properties including all metered properties.
<b>Notes</b>	None.

## DEVICE\_ID

<b>Description</b>	Controls the Device ID.
<b>Commands</b>	< GET DEVICE_ID > < REP DEVICE_ID {AD4Q-Ayyyyyyyyyyyyyyyyyyyyyyyy} >  < SET DEVICE_ID {Name1} > < REP DEVICE_ID {Name1yyyyyyyyyyyyyyyyyyyyyyyy} >
<b>Variables</b>	Where the repeating <b>y</b> represents the spaces returned by the device to pad the Device ID to 31 characters.
<b>Notes</b>	The device always responds with 31-character ID.  SET accepts 1-8 Characters from the set: A-Z,a-z,0-9,!"#\$%&'()*+,-./:;<=>?[^\^_`~ and space.

## ENCRYPTION\_MODE

<b>Description</b>	Discovery of the Encryption Mode.
<b>Commands</b>	< GET ENCRYPTION_MODE > < REP ENCRYPTION_MODE ON > or < REP ENCRYPTION_MODE OFF >
<b>Variables</b>	None.
<b>Notes</b>	None.

## FLASH

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

## FW\_VER

<b>Description</b>	Discovery of the firmware version.
<b>Commands</b>	<p>Self test passed:</p> <p>&lt; GET FW_VER &gt; &lt; REP FW_VER {2.0.15.2yyyyyyyyyyyyyyyy} &gt;</p> <p>Self test failed:</p> <p>&lt; GET FW_VER &gt; &lt; REP FW_VER {2.0.15.2*yyyyyyyyyyyy} &gt;</p>
<b>Variables</b>	Where the repeating y 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.

## MODEL

<b>Description</b>	Discovery of the model name of the device.
<b>Commands</b>	<p>&lt; GET MODEL &gt; &lt; REP MODEL {AD4Q-Ayyyyyyyyyyyyyyyyyyyy} &gt;</p>

<b>Variables</b>	Where the repeating <b>y</b> 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.

## QUADVERSITY\_MODE

<b>Description</b>	Discovery of Quadversity mode.
<b>Commands</b>	< GET QUADVERSITY_MODE > < REP QUADVERSITY_MODE OFF > or < REP QUADVERSITY_MODE ON >
<b>Variables</b>	None.
<b>Notes</b>	Quadversity Mode - While only applicable to AD4Q, an attempt to GET on an AD4D shall report OFF.

## RF\_BAND

<b>Description</b>	Discovery of the RF band.
<b>Commands</b>	< GET RF_BAND > < REP RF_BAND {G55yyyyy} >
<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.

## TRANSMISSION\_MODE

<b>Description</b>	Discovery of the transmission mode.
<b>Commands</b>	< GET TRANSMISSION_MODE > < REP TRANSMISSION_MODE STANDARD > or < REP TRANSMISSION_MODE HIGH_DENSITY >
<b>Variables</b>	None.
<b>Notes</b>	None.



# Channel Command Strings

## 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\_MUTE

<b>Description</b>	Control for the channel audio mute.
<b>Commands</b>	<pre>&lt; GET x AUDIO_MUTE &gt; &lt; REP x AUDIO_MUTE OFF &gt;</pre> <pre>&lt; SET x AUDIO_MUTE ON &gt; &lt; REP x AUDIO_MUTE ON &gt;</pre> <pre>&lt; SET x AUDIO_MUTE OFF &gt; &lt; REP x AUDIO_MUTE OFF &gt;</pre> <pre>&lt; SET x AUDIO_MUTE TOGGLE &gt; &lt; REP x AUDIO_MUTE OFF &gt;</pre>

<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	TOGGLE switches between ON and OFF.

## 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 {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} &gt;  &lt; SET x CHAN_NAME {Lead Vox} &gt; &lt; REP x CHAN_NAME {Lead Voxyyyyyyyyyyyyyyyyyyyyyyyyyy} &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>

## ENCRYPTION\_STATUS

<b>Description</b>	Discovery of the encryption status.
<b>Commands</b>	<pre>&lt; GET x ENCRYPTION_STATUS &gt; &lt; REP x ENCRYPTION_STATUS OK &gt;</pre> <p>When a mismatched transmitter is detected:</p> <pre>&lt; REP x ENCRYPTION_STATUS ERROR &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	It is unnecessary to continuously query this parameter. The receiver automatically sends a REPort when interference is detected.

## 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;</pre>



	<pre>&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 {1,1yyyyyy} &gt; &lt; GET x FREQUENCY &gt; &lt; REP x FREQUENCY 0606025 &gt;</pre> <p>SET the FREQUENCY to some new value:</p> <pre>&lt; SET x FREQUENCY 620000 &gt; &lt; REP x GROUP_CHANNEL {--,--yyyy} &gt; &lt; REP x FREQUENCY 0620000 &gt;</pre> <p>Similarly, when setting GROUP_CHANNEL, the corresponding FREQUENCY is reported:</p> <pre>&lt; SET x GROUP_CHANNEL {6,6} &gt; &lt; REP x FREQUENCY 0614650 &gt; &lt; REP x GROUP_CHANNEL {6,6yyyyyy} &gt;</pre> <ol style="list-style-type: none"> <li>2. For channels operating in FD-C mode, there is a second FREQUENCY2 and GROUP_CHANNEL2 that behave in the same manner as FREQUENCY.</li> </ol> <pre>&lt; GET x FD_MODE &gt; &lt; REP x FD_MODE FD-C &gt;</pre> <pre>&lt; GET x FREQUENCY &gt; &lt; REP x FREQUENCY 0578350 &gt;</pre> <pre>&lt; GET x FREQUENCY2 &gt; &lt; REP x FREQUENCY2 0578850 &gt;</pre>

## FD\_MODE

<b>Description</b>	Discovery of the type of frequency diversity.
<b>Commands</b>	<pre>&lt; GET x FD_MODE &gt; &lt; REP x FD_MODE OFF &gt;  &lt; GET x FD_MODE &gt; &lt; REP x FD_MODE FD-C &gt;  &lt; GET x FD_MODE &gt; &lt; REP x FD_MODE FD-S &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>FD-C : Combining mode</p> <p>FD-S : Selection mode</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 {1,1yyyyyy} &gt;  &lt; SET x GROUP_CHANNEL {6,100} &gt; &lt; REP x FREQUENCY 0652875 &gt; &lt; REP x GROUP_CHANNEL {6,100yyyyy} &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 10 characters.</p>
<b>Notes</b>	<p>For channels operating in FD-C mode, GROUP_CHANNEL2 corresponds to the second Group Channel value</p> <p>Device always returns a 10-character string</p> <p>Refer to the Group/Channel mappings corresponding to the RF Band and Transmission Mode of the device</p> <p>You must parse the "," from within the reported value</p> <p>Characters before the "," are the Group ID Characters after the "," are the Channel ID</p> <p>The value: "--,--" is the wildcard indicating no GROUP_CHANNEL value is set</p> <p>You cannot SET to "--,--"</p> <p><i>Note: GROUP_CHANNEL and FREQUENCY are related as described in FREQUENCY.</i></p>

## INTERFERENCE\_STATUS

<b>Description</b>	Discovery of interference.
<b>Commands</b>	<pre>&lt; GET x INTERFERENCE_STATUS &gt;</pre> <pre>&lt; REP x INTERFERENCE_STATUS NONE &gt;</pre> <p>When interference is detected:</p> <pre>&lt; REP x INTERFERENCE_STATUS DETECTED &gt;</pre> <p>For FD-C channels only:</p> <pre>&lt; GET x INTERFERENCE_STATUS2 &gt;</pre> <pre>&lt; REP x INTERFERENCE_STATUS2 NONE &gt;</pre> <p>When interference is detected:</p> <pre>&lt; REP x INTERFERENCE_STATUS2 DETECTED &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	It is unnecessary to continuously query this parameter. The receiver automatically sends a REPort when interference is detected.

## UNREGISTERED\_TX\_STATUS

<b>Description</b>	Discovery of an unregistered transmitter.
<b>Commands</b>	<pre>&lt; GET x UNREGISTERED_TX_STATUS &gt;</pre> <pre>&lt; REP x UNREGISTERED_TX_STATUS OK &gt;</pre> <p>A REP when the condition is occurring:</p> <pre>&lt; REP 1 UNREGISTERED_TX_STATUS ERROR &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	It is unnecessary to continuously query this parameter. The receiver automatically sends a REPort when an unregistered transmitter is detected.

## 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><i>Note: This results in one SAMPLE every second.</i></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 100 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>	Standard Channel: Quadversity = OFF, FD = OFF or FD-S:

```

< SAMPLE chNum ALL qual audBitmap audPeak audRms rfAntStats rfBitmapA rfRssiA rf-
BitmapB rfRssiB >
< SAMPLE 1 ALL 005 031 102 102 BB 31 086 31 065 >

Quadversity Channel: Quadversity = ON, FD = OFF

< SAMPLE chNum ALL qual audBitmap audPeak audRms rfAntStats rfBitmapA rfRssiA rf-
BitmapB rfRssiB rfBitmapC rfRssiC rfBitmapD rfRssiD >
< SAMPLE 1 ALL 005 031 102 102 BBBB 31 083 31 068 31 069 31 072 >

FD-C Channel: Quadversity = OFF, FD = FD-C

< SAMPLE chNum ALL qual audBitmap audPeak audRms rfAntStatsF1 rfBitmapF1A
rfRssiF1A rfBitmapF1B rfRssiF1B antStatsF2 rfBitmapF2A rfRssiF2A rfBitmapF2B rfRssiF2B
>
< SAMPLE 1 ALL 005 031 102 102 BB 31 082 31 060 BB 31 082 31 060 >

Quadversity and FD-C Channel: Quadversity = ON, FD = FD-C

Sample input:

Note: The sample input that follows is actually 1 line of input spaced to show correlation.

< SAMPLE chNum ALL qual audBitmap audPeak audRms
rfAntStatsF1 rfBitmapF1A rfRssiF1A rfBitmapF1B rfRssiF1B rfBitmapF1C rfRssiF1C
rfBitmapF1D rfRssiF1D
rfAntStatsF2 rfBitmapF2A rfRssiF2A rfBitmapF2B rfRssiF2B rfBitmapF2C rfRssiF2C
rfBitmapF2D rfRssiF2D >

Sample response:

< SAMPLE 1 ALL 005 031 102 102 BBBB 31 084 31 065 31 070 31 070 BBBB 31 084 31
065 31 070 31 070 >
    
```

**Variables**

Key mapping:

Key	Corresponding command string for value format	Num Char	Notes
qual	CHANNEL_QUALITY	3	
aud- Bitmap	AUDIO_LED_BITMAP	3	Use for simple UIs as this is a direct mirror of the front panel LEDs.
audPeak	AUDIO_LEVEL_PEAK	3	
audRms	AUDIO_LEVEL_RMS	3	
rfAntStats	ANTENNA_STATUS	2	Quadversity: XX for Quad = OFF XXXX for Quad = ON  FD-C: Append F2's set of RF
rfBitmap	RSSI_LED_BITMAP	2	Use for simple UIs as this is a direct mirror of the front panel LEDs.

	Key	Corresponding command string for value format	Num Char	Notes
				See the command example for how this scales to antenna A-D.
	rfRssi	RSSI	3	See the command example for how this scales to antenna A-D.
<b>Notes</b>	None.			

## ANTENNA\_STATUS

<b>Description</b>	Discovery of antenna status.
<b>Commands</b>	<p>Normal channel, both LEDs lit:</p> <pre>&lt; GET x ANTENNA_STATUS &gt; &lt; REP x ANTENNA_STATUS BB &gt;</pre> <p>Quadversity example where antenna A-Blue, B-Red, C-Off, D-Blue:</p> <pre>&lt; GET x ANTENNA_STATUS &gt; &lt; REP x ANTENNA_STATUS BRXB &gt;</pre>
<b>Variables</b>	Where <b>x</b> 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. Each character represents an antenna ABCD, where the value for each can be:</p> <p>X - Off</p> <p>R - Red</p> <p>B - Blue</p> <p>Example of ANTENNA_STATUS with QUADVERSITY_MODE=OFF: BB</p> <p>Example of ANTENNA_STATUS with QUADVERSITY_MODE=ON: BBBB</p>

## AUDIO\_LED\_BITMAP

<b>Description</b>	Discovery of the Audio LED bitmap.
<b>Commands</b>	<pre>&lt; GET x AUDIO_LED_BITMAP &gt; &lt; REP x AUDIO_LED_BITMAP 031 &gt;</pre>
<b>Variables</b>	Where <b>x</b> 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</p> <p>Each bit corresponds to an LED from bottom to top.</p> <p>LEDs 1-4 - Green</p> <p>LEDs 5-7 - Amber</p> <p>LED 8 - Red</p> <p>Example values:</p> <p>0 = b00000000 - all off</p> <p>1 = b00000001 - first LED on</p> <p>...</p> <p>131 = b10000011 - after OL that is held, and the 2 bottom green LEDs are on</p> <p>...</p> <p>255 = b11111111 - all LEDs on</p>
--------------	---

## 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>	<p>This is a metered property using SAMPLE; it does not generate a report on a value change. Numeric, 3-character fixed output</p> <p>Units: dBFS</p> <p>The actual value = the reported value - 120</p> <p>Value range as reported over command strings: 000 - 120</p> <p>Value range after conversion to the actual value: -120 - 0 dBFS</p> <p>The AD4 values fall in the range of about -100 to 0 dBFS.</p>

## 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>	<p>This is a metered property using SAMPLE; it does not generate a report on a value change.</p> <p>Format: Numeric, 3-character fixed output</p> <p>Units: dBFS</p> <p>The actual value = the reported value - 120</p> <p>Value range as reported over command strings: 000 - 120</p> <p>Value range after conversion to the actual value: -120 - 0 dBFS</p> <p>The AD4 values fall in the range of about -100 to 0 dBFS.</p>
--------------	--

## CHAN\_QUALITY

<b>Description</b>	Discovery of the channel quality.
<b>Commands</b>	<pre>&lt; GET x CHAN_QUALITY &gt; &lt; REP x CHAN_QUALITY 005 &gt;</pre>
<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.</p> <p>Numeric, 3-character fixed output</p> <p>000 - 005</p> <p>255 = Unknown</p>

## RSSI

<b>Description</b>	Discovery of the RSSI.
<b>Commands</b>	<p>For a normal channel (Quadversity = OFF)</p> <pre>&lt; GET x RSSI 0 &gt; &lt; REP x RSSI 1 083 &gt; &lt; REP x RSSI 2 064 &gt;</pre> <p>For a Quadversity channel (Quadversity = ON)</p> <pre>&lt; GET x RSSI 0 &gt; &lt; REP x RSSI 1 083 &gt; &lt; REP x RSSI 2 064 &gt; &lt; REP x RSSI 3 082 &gt; &lt; REP x RSSI 4 071 &gt;</pre>
<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.</p> <p>Numeric, 3-character fixed output per antenna</p> <p>Units: dBm</p>



	<p>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> <p>Channel Number, Antenna Index</p> <p>1 : A  2 : B  3 : C  4 : D</p>
--	--

## RSSI\_LED\_BITMAP

<b>Description</b>	Discovery of the RSSI LED bitmap.
<b>Commands</b>	<p>Antenna A, 5 LEDs on</p> <pre>&lt; GET x RSSI_LED_BITMAP 1 &gt; &lt; REP x RSSI_LED_BITMAP 1 31 &gt;</pre> <p>Example of all for a Quad=OFF</p> <pre>&lt; GET x RSSI_LED_BITMAP 0 &gt; &lt; REP x RSSI_LED_BITMAP 1 63 &gt; &lt; REP x RSSI_LED_BITMAP 2 63 &gt;</pre> <p>Example of all for a Quad=ON</p> <pre>&lt; GET x RSSI_LED_BITMAP 0 &gt; &lt; REP x RSSI_LED_BITMAP 1 63 &gt; &lt; REP x RSSI_LED_BITMAP 2 63 &gt; &lt; REP x RSSI_LED_BITMAP 3 63 &gt; &lt; REP x RSSI_LED_BITMAP 4 63 &gt;</pre>
<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, 2-character fixed output per antenna:</p> <p>Represents a bitmap of which of the RSSI LEDs are ON from bottom to top.</p> <p>LEDs 1-5 : Amber</p> <p>LED 6 : Red (for Overload and RF Pad engaged indications)</p> <p>Example Variables</p> <pre>00 = b000000 = All LEDs off 01 = b000001 = LED1 ON (bottom) ...</pre>

```

31 = b0111111 = LEDs 1-5 ON

32 = b1000000 = LED6 ON (top, OL, all others OFF)

...

63 = b1111111 = All LEDs on

01 = b0000001 = LED1 ON (bottom)

Channel Number, Antenna Index

1 : A

2 : B

3 : C

4 : D

```

## Side Channel Command Strings

### TX\_BATT\_BARS

<b>Description</b>	Discovery of the transmitter Battery Bars.
<b>Commands</b>	<pre> &lt; GET x TX_BATT_BARS &gt; &lt; REP x TX_BATT_BARS 255 &gt;  Report when data becomes known:  &lt; REP x TX_BATT_BARS 004 &gt; </pre>
<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>

### TX\_BATT\_CHARGE\_PERCENT

<b>Description</b>	Discovery of the transmitter battery charge.
<b>Commands</b>	<pre> &lt; GET x TX_BATT_CHARGE_PERCENT &gt; &lt; REP x TX_BATT_CHARGE_PERCENT 255 &gt;  Report when data becomes known:  &lt; REP x TX_BATT_CHARGE_PERCENT 088 &gt; </pre>

<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	Numeric, 3-character fixed output 000 to 100 - Percent 255 - Unknown

## TX\_BATT\_CYCLE\_COUNT

<b>Description</b>	Discovery of the transmitter Battery Cycle Count.
<b>Commands</b>	< GET x TX_BATT_CYCLE_COUNT > < REP x TX_BATT_CYCLE_COUNT 65535 > Report when data becomes known: < REP x TX_BATT_CYCLE_COUNT 00019 >
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	Numeric, 5-character fixed output 00000 to 65534 - Number of cycles 65535 - Unknown

## TX\_BATT\_HEALTH\_PERCENT

<b>Description</b>	Discovery of the transmitter Battery Health.
<b>Commands</b>	< GET x TX_BATT_HEALTH_PERCENT > < REP x TX_BATT_HEALTH_PERCENT 255 > Report when data becomes known: < REP x TX_BATT_HEALTH_PERCENT 088 >
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	Numeric, 3-character fixed output 000 to 100 - Percent 255 - Unknown

## TX\_BATT\_MINS

<b>Description</b>	Discovery of the transmitter battery runtime minutes.
--------------------	---

<b>Commands</b>	<pre>&lt; GET x TX_BATT_MINS &gt; &lt; REP x TX_BATT_MINS 65535 &gt;</pre> <p>Report when data becomes known (example is 2 hours 5 minutes):</p> <pre>&lt; REP x TX_BATT_MINS 00125 &gt;</pre>
<b>Variables</b>	Where <b>x</b> 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> <p>65534 - Battery time calculating</p> <p>65535 - Unknown, or not applicable</p>

## TX\_BATT\_TEMP\_C

<b>Description</b>	Discovery of the transmitter battery temperature in Celsius.
<b>Commands</b>	<pre>&lt; GET x TX_BATT_TEMP_C &gt; &lt; REP x TX_BATT_TEMP_C 255 &gt;</pre> <p>Report as data becomes known or changes:</p> <pre>&lt; REP x TX_BATT_TEMP_C 062 &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 to 254 - Temperature</p> <p>255 - Unknown, or not applicable</p> <p>The actual value = the reported value - 40.</p>

## TX\_BATT\_TEMP\_F

<b>Description</b>	Discovery of the transmitter battery temperature in Fahrenheit.
<b>Commands</b>	<pre>&lt; GET x TX_BATT_TEMP_F &gt; &lt; REP x TX_BATT_TEMP_F 255 &gt;</pre> <p>Report as data becomes known or changes:</p> <pre>&lt; REP x TX_BATT_TEMP_F 062 &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.

<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 to 254 - Temperature</p> <p>255 - Unknown, or not applicable</p> <p>The actual value = the reported value - 40.</p>
--------------	---

## TX\_BATT\_TYPE

<b>Description</b>	Discovery of the transmitter's battery type.
<b>Commands</b>	<p>&lt; GET x TX_BATT_TYPE &gt;</p> <p>&lt; REP x TX_BATT_TYPE UNKN &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_BATT_TYPE LION &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Report responses:</p> <p>LION</p> <p>ALKA</p> <p>NIMH</p> <p>LITH</p> <p>UNKN - Either no transmitter or not supported by transmitter</p>

## TX\_DEVICE\_ID

<b>Description</b>	Discovery of the transmitter's Device ID.
<b>Commands</b>	<p>&lt; GET x TX_DEVICE_ID &gt;</p> <p>&lt; REP x TX_DEVICE_ID {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} &gt;</p> <p>Report of a transmitter being received:</p> <p>&lt; REP x TX_DEVICE_ID {LeadVoxyyyyyyyyyyyyyyyyyyyyy} &gt;</p>
<b>Variables</b>	<p>Where x is the channel number.</p> <p>Where the repeating y represents or pads the 31-character string</p>
<b>Notes</b>	<p>31-character string - Device ID of the transmitter being received</p> <p>Unknown - blank, all spaces</p>

## TX\_INPUT\_PAD

<b>Description</b>	Discovery of the transmitter Input Pad.
<b>Commands</b>	<p>&lt; GET x TX_INPUT_PAD &gt;          &lt; REP x TX_INPUT_PAD 255 &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_INPUT_PAD 000 &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 - Input Pad is ON (-12 dB)</p> <p>012 - Input Pad is OFF (0 dB)</p> <p>255 - Unknown, or Input Pad is not applicable to the transmitter variant</p> <p>The actual value = the reported value - 12.</p> <p>Input Pad is applicable to AD1, ADX1. All others indicate unknown.</p>

## TX\_LOCK

<b>Description</b>	Discovery of the transmitter Lock.
<b>Commands</b>	<p>&lt; GET x TX_LOCK &gt;          &lt; REP x TX_LOCK UNKNOWN &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_LOCK MENU &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Report responses:</p> <p>NONE          POWER          MENU          ALL          UNKNOWN</p>

## 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:</p> <p>&lt; REP x TX_MODEL AD2 &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>Report responses:</p> <p>AD1          AD2          ADX1          ADX1M          ADX2          ADX2FD          UNKNOWN</p>

## TX\_MUTE\_MODE\_STATUS

<b>Description</b>	Discovery of the transmitter Mute Mode Status.
<b>Commands</b>	<p>&lt; GET x TX_MUTE_MODE_STATUS &gt;          &lt; REP x TX_MUTE_MODE_STATUS UNKNOWN &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_MUTE_MODE_STATUS MUTE &gt;</p>
<b>Variables</b>	Where x is the channel number.
<b>Notes</b>	<p>ON - Audio is being transmitted (power switch in "ON" position)</p> <p>MUTE - Audio is being locally muted, but still transmitting RF (power switch is in "OFF" position)</p> <p>UNKNOWN - Either no transmitter or not supported by transmitter</p>

## TX\_OFFSET

<b>Description</b>	Discovery of the transmitter offset.
<b>Commands</b>	<p>&lt; GET x TX_OFFSET &gt;          &lt; REP x TX_OFFSET 255 &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_OFFSET 012 &gt;</p>

<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>Numeric, 3-character fixed</p> <p>Range - 000 to 033</p> <p>255 - Unknown</p> <p>The actual value = the reported value - 12. This means that the actual range is -12 to +21 dB in 1 dB increments.</p>

## TX\_POLARITY

<b>Description</b>	Discovery of the transmitter Polarity.
<b>Commands</b>	<p>&lt; GET x TX_POLARITY &gt;</p> <p>&lt; REP x TX_POLARITY UNKNOWN &gt;</p> <p>Report when data becomes known:</p> <p>&lt; REP x TX_POLARITY POSITIVE &gt;</p>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>Report responses:</p> <p>POSITIVE</p> <p>NEGATIVE</p> <p>UNKNOWN - Either no transmitter or not supported by transmitter</p>

## TX\_POWER\_LEVEL

<b>Description</b>	Discovery of the transmitter power level.
<b>Commands</b>	<p>&lt; GET x TX_POWER_LEVEL &gt;</p> <p>&lt; REP x TX_POWER_LEVEL 255 &gt;</p> <p>Report when normal power level is decoded:</p> <p>&lt; REP x TX_POWER_LEVEL 010 &gt;</p>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>Numeric, 3-character fixed length</p> <p>Transmit power level in mW (for example, 002, 010, or 020)</p> <p>255 - Unknown</p>



## TX\_TALK\_SWITCH

<b>Description</b>	Discovery of the transmitter Talk Switch activity.
<b>Commands</b>	<p>Before a Talk Switch has been pressed, or no transmitter has been decoded:</p> <pre>&lt; GET x TX_TALK_SWITCH &gt; &lt; REP x TX_TALK_SWITCH UNKNOWN &gt;</pre> <p>User presses Talk Switch, there is a report:</p> <pre>&lt; REP x TX_TALK_SWITCH ON &gt;</pre> <p>User releases the Talk Switch, there is a report:</p> <pre>&lt; REP x TX_TALK_SWITCH OFF &gt;</pre> <p>Now that the AD4 has knowledge that a Talk Switch is being used on this session:</p> <pre>&lt; GET x TX_TALK_SWITCH &gt; &lt; REP x TX_TALK_SWITCH OFF &gt;</pre> <p>When the Transmitter goes out of range, that session is lost and the state of the Talk Switch becomes unknown. There is a report (along with the reports for battery level unknown, model unknown, and so on):</p> <pre>&lt; REP x TX_TALK_SWITCH UNKNOWN &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number.
<b>Notes</b>	<p>ON - Talk Switch button press has been detected</p> <p>OFF - Talk Switch button has been released</p> <p>UNKNOWN - No transmitter detected and/or no Talk Switch button press has been detected up to this point in the transmitter's current connection.</p> <p><i>Note: OFF and UNKNOWN are treated the same by the AD4 algorithms with respect to Talk Switch routing.</i></p>

## Transmitter Slots Command Strings

### Transmitter slot general description

- Each channel of AD4 has 8 slots where a transmitter can be registered
  - Specify Channel number 0 to interact with ALL 8 slots on a channel
  - Specify Channel number 1 through 8 to interact with that specific slot on a channel
- The SLOT\_STATUS lets you know the status and what to expect from further operations
  - EMPTY - No transmitter exists in the SLOT.  
GET - You can GET the other properties but those report as "Unknown" values  
SET - Not supported and results in **REP ERR**
  - STANDARD - An AD (standard) transmitter is registered in the slot
  - LINKED.INACTIVE - An ADX (enhanced) transmitter is registered in the slot, but is not currently online

- STANDARD & LINKED.INACTIVE:  
GET - You can GET more useful data for the SLOT\_DEVICE\_ID and SLOT\_TX\_MODEL but the other properties report as "Unknown" values  
SET - Not supported and results in **REP ERR**
- LINKED.ACTIVE - An ADX (enhanced) transmitter is registered in the slot and is online.  
GET - You can GET more useful data for all SLOT properties  
SET - You can SET properties where it is applicable to do so. See the specific command details

**Note:** As transmitters get linked, unlinked, and moved around there will be corresponding REP events for all of the slot attributes that change. This is a great deal of data, but a control system can readily parse and use the data that it requires.

## SLOT\_BATT\_BARS

<b>Description</b>	Discovery of the transmitter's battery bars by slot.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <pre>&lt; GET x SLOT_BATT_BARS z &gt; &lt; REP x SLOT_BATT_BARS z 004 &gt;</pre> <p>Reports occur as the battery depletes:</p> <pre>&lt; REP x SLOT_BATT_BARS z 003 &gt; ... &lt; REP x SLOT_BATT_BARS z 002 &gt;</pre> <p>Transmitter is out of range:</p> <pre>&lt; GET x SLOT_BATT_BARS z &gt; &lt; REP x SLOT_BATT_BARS z 255 &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 to 005 - Number of bars</p> <p>255 - Unknown</p>

## SLOT\_BATT\_CHARGE\_PERCENT

<b>Description</b>	Discovery of the battery charge percent by slot.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <pre>&lt; GET x SLOT_BATT_CHARGE_PERCENT z &gt; &lt; REP x SLOT_BATT_CHARGE_PERCENT z 087 &gt;</pre> <p>Reports occur as the battery is first calculating, then depletes:</p> <pre>&lt; REP x SLOT_BATT_CHARGE_PERCENT z 087 &gt; ... &lt; REP x SLOT_BATT_CHARGE_PERCENT z 086 &gt;</pre>

	<p>...</p> <p>&lt; REP x SLOT_BATT_CHARGE_PERCENT z 085 &gt;</p> <p>Transmitter is out of range, or not applicable</p> <p>&lt; REP x SLOT_BATT_CHARGE_PERCENT z 255 &gt;</p>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 to 100 - Percent</p> <p>255 - Unknown</p>

## SLOT\_BATT\_CYCLE\_COUNT

<b>Description</b>	Discovery of the battery cycle count by slot.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <p>&lt; GET x SLOT_BATT_CYCLE_COUNT z &gt;</p> <p>&lt; REP x SLOT_BATT_CYCLE_COUNT z 00013 &gt;</p> <p>Transmitter is out of range, or not applicable:</p> <p>&lt; REP x SLOT_BATT_CYCLE_COUNT z 65535 &gt;</p>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Numeric, 5-character fixed output</p> <p>00000 to 65534 - Number of cycles</p> <p>65535 - Unknown</p>

## SLOT\_BATT\_HEALTH\_PERCENT

<b>Description</b>	Discovery of the battery health by slot.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <p>&lt; GET x SLOT_BATT_HEALTH_PERCENT z &gt;</p> <p>&lt; REP x SLOT_BATT_HEALTH_PERCENT z 097 &gt;</p> <p>Reports as the Transmitter is ON/OFF:</p> <p>&lt; REP x SLOT_BATT_HEALTH_PERCENT z 097 &gt;</p> <p>...</p> <p>&lt; REP x SLOT_BATT_HEALTH_PERCENT z 255 &gt;</p> <p>...</p> <p>&lt; REP x SLOT_BATT_HEALTH_PERCENT z 096 &gt;</p>

<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	Numeric, 3-character fixed output 000 to 100 - Percent 255 - Unknown

## SLOT\_BATT\_MINS

<b>Description</b>	Discovery of the battery runtime by slot.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <pre>&lt; GET x SLOT_BATT_MINS z &gt; &lt; REP x SLOT_BATT_MINS z 00360 &gt;</pre> <p>Reports occur as the battery is first calculating, then depletes:</p> <pre>&lt; REP x SLOT_BATT_MINS z 65534 &gt; &lt; REP x SLOT_BATT_MINS z 00300 &gt; ... &lt; REP x SLOT_BATT_MINS z 00299 &gt;</pre> <p>Transmitter is out of range:</p> <pre>&lt; GET x SLOT_BATT_MINS z &gt; &lt; REP x SLOT_BATT_MINS z 65535 &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot 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. (Check contacts)</p> <p>65534 - Battery time calculating</p> <p>65535 - Unknown, or not applicable</p>

## SLOT\_BATT\_TYPE

<b>Description</b>	Controls the transmitter slot's battery type.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <pre>&lt; GET x SLOT_BATT_TYPE z &gt; &lt; REP x SLOT_BATT_TYPE z LION &gt;</pre> <p>Transmitter is out of range, or a not-supported by Transmitter:</p>

	<pre>&lt; GET x SLOT_POLARITY z &gt; &lt; REP x SLOT_POLARITY z UNKN &gt;  &lt; SET x SLOT_POLARITY z ALKA &gt; &lt; REP ERR &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Report responses:</p> <p>LION</p> <p>ALKA</p> <p>NIMH</p> <p>LITH</p> <p>UNKN - Either no transmitter or not supported by transmitter</p>

## SLOT\_INPUT\_PAD

<b>Description</b>	Controls the transmitter's slot input pad.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE, input pad not engaged:</p> <pre>&lt; GET x SLOT_INPUT_PAD z &gt; &lt; REP x SLOT_INPUT_PAD z 012 &gt;</pre> <p>To engage the input pad:</p> <pre>&lt; SET x SLOT_INPUT_PAD z 0 &gt; &lt; REP x SLOT_INPUT_PAD z 000 &gt;</pre> <p>Transmitter is out of range, or property not supported by the transmitter variant:</p> <pre>&lt; GET x SLOT_INPUT_PAD z &gt; &lt; REP x SLOT_RF_OUTPUT z 255 &gt;</pre> <pre>&lt; SET x SLOT_RF_OUTPUT z 0 &gt; &lt; REP ERR &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>000 - Input Pad is ON (-12 dB)</p> <p>012 - Input Pad is OFF (0 dB)</p> <p>255 - Unknown, or Input Pad is not applicable to the transmitter variant</p> <p>The actual value = the reported value - 12.</p> <p><i>Note: A remotely controllable Input Pad is applicable to ADX1. All others indicate unknown.</i></p>

## SLOT\_OFFSET

<b>Description</b>	Controls the transmitter's slot offset.
<b>Commands</b>	<p>Get for Linked.Active transmitter: (default)</p> <pre>&lt; GET x SLOT_OFFSET z &gt; &lt; REP x SLOT_OFFSET z 012 &gt;</pre> <p>SET for Linked.Active transmitter: to min value:</p> <pre>&lt; SET x SLOT_OFFSET z 0 &gt; &lt; REP x SLOT_OFFSET z 000 &gt;</pre> <p>SET for Linked.Active transmitter: (demonstrating INC and DEC)</p> <pre>&lt; SET x SLOT_OFFSET z INC 5 &gt; &lt; REP x SLOT_OFFSET z 005 &gt;</pre> <pre>&lt; SET x SLOT_OFFSET z INC 5 &gt; &lt; REP x SLOT_OFFSET z 010 &gt;</pre> <pre>&lt; SET x SLOT_OFFSET z DEC 2 &gt; &lt; REP x SLOT_OFFSET z 008 &gt;</pre> <p>Attempt to GET/SET for an unsupported or out of range transmitter:</p> <pre>&lt; GET x SLOT_OFFSET z &gt; &lt; REP x SLOT_OFFSET z 255 &gt;</pre> <pre>&lt; SET x SLOT_OFFSET z 012 &gt; &lt; REP ERR &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Numeric, 3-character fixed</p> <p>Range - 000 to 033</p> <p>255 - Unknown</p> <p>The actual value = the reported value - 12.</p> <p>The Actual Range is -12 to +21 dB in 1 dB increments.</p>

## SLOT\_POLARITY

<b>Description</b>	Controls the transmitter's slot polarity.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <pre>&lt; GET x SLOT_POLARITY z &gt; &lt; REP x SLOT_POLARITY z POSITIVE &gt;</pre>

	<pre>&lt; SET x SLOT_POLARITY z NEGATIVE &gt; &lt; REP x SLOT_POLARITY z NEGATIVE &gt;  Transmitter is out of range, or a not-supported transmitter such as ADX2:  &lt; GET x SLOT_POLARITY z &gt; &lt; REP x SLOT_POLARITY z UNKNOWN &gt;  &lt; SET x SLOT_POLARITY z POSITIVE &gt; &lt; REP ERR &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Report responses:</p> <p>POSITIVE</p> <p>NEGATIVE</p> <p>UNKNOWN - Either no transmitter or not supported by transmitter</p> <p><i>Note: Polarity is only supported by ADX1 and ADX1M.</i></p>

## SLOT\_RF\_OUTPUT

<b>Description</b>	Controls the transmitter's RF output.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <pre>&lt; GET x SLOT_RF_OUTPUT z &gt; &lt; REP x SLOT_RF_OUTPUT z RF_ON &gt;  &lt; SET x SLOT_RF_OUTPUT z RF_MUTE &gt; &lt; REP x SLOT_RF_OUTPUT z RF_MUTE &gt;  Transmitter is out of range: <pre>&lt; GET x SLOT_RF_OUTPUT z &gt; &lt; REP x SLOT_RF_OUTPUT z UNKNOWN &gt;  &lt; SET x SLOT_RF_OUTPUT z RF_MUTE &gt; &lt; REP ERR &gt;</pre> </pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Report responses:</p> <p>UNKNOWN</p> <p>RF_ON</p> <p>RF_MUTE</p>

## SLOT\_RF\_POWER

<b>Description</b>	<p>REports the actual power level of the unit in mW.</p> <p><i>Note: Use SLOT_RF_POWER_MODE to set the power level</i></p>
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <p>&lt; GET x SLOT_RF_POWER z &gt; &lt; REP x SLOT_RF_POWER z 002 &gt;</p> <p>Transmitter is LINKED.INACTIVE or out of range:</p> <p>&lt; GET x SLOT_RF_POWER z &gt; &lt; REP 255 &gt;</p> <p>Reports as the level changes:</p> <p>&lt; REP x SLOT_RF_POWER z 040 &gt;</p> <p>&lt; REP x SLOT_RF_POWER z 010 &gt;</p> <p>&lt; REP x SLOT_RF_POWER z 255 &gt;</p> <p>Attempt to SET SLOT_RF_POWER reports an ERR:</p> <p>&lt; SET x SLOT_RF_POWER z 10 &gt; &lt; REP ERR &gt;</p>
<b>Variables</b>	Where x is the channel number and z is the slot number.
<b>Notes</b>	<p><b>Use SLOT_RF_POWER_MODE to SET the power level</b></p> <p>Numeric, 3-character fixed output</p> <p>002, 010, 020, 035, 040, 050 - Typical values</p> <p>255 - Unknown</p>

## SLOT\_RF\_POWER\_MODE

<b>Description</b>	Controls the logical power level the unit is operating on.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <p>&lt; GET x SLOT_RF_POWER_MODE z &gt; &lt; REP x SLOT_RF_POWER_MODE z NORMAL &gt;</p> <p>Using SET (note both responses):</p> <p>&lt; SET x SLOT_RF_POWER_MODE z LOW &gt; &lt; REP x SLOT_RF_POWER_MODE z LOW &gt; &lt; REP x SLOT_RF_POWER z 002 &gt;</p> <p>Transmitter is out of range:</p>



	<p>&lt; SET x SLOT_RF_POWER z LOW &gt;          &lt; REP ERR &gt;</p> <p>Reports as the level changes:</p> <p>&lt; REP x SLOT_RF_POWER_MODE z HIGH &gt;          &lt; REP x SLOT_RF_POWER z 040 &gt;</p> <p>Unit changes frequency to a more restrictive HIGH Power allowed value:</p> <p>&lt; REP x SLOT_RF_POWER z 020 &gt;</p>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Report responses:</p> <p>UNKNOWN</p> <p>LOW</p> <p>NORMAL</p> <p>HIGH</p> <p><i>Note: Some RF Bands or modes do not support HIGH and any attempt to use HIGH returns a REP ERR.</i></p>

## SLOT\_SHOWLINK\_STATUS

<b>Description</b>	Discovery of the slot showlink status.
<b>Commands</b>	<p>Transmitter is LINKED.ACTIVE:</p> <p>&lt; GET x SLOT_SHOWLINK_STATUS z &gt;          &lt; REP x SLOT_SHOWLINK_STATUS z 005 &gt;</p> <p>Transmitter is out of range:</p> <p>&lt; GET x SLOT_SHOWLINK_STATUS z &gt;          &lt; REP x SLOT_SHOWLINK_STATUS z 255 &gt;</p> <p>Reports as the level changes:</p> <p>&lt; REP x SLOT_SHOWLINK_STATUS z 003 &gt;          &lt; REP x SLOT_SHOWLINK_STATUS z 004 &gt;          &lt; REP x SLOT_SHOWLINK_STATUS z 005 &gt;</p>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Numeric, 3-character fixed output</p> <p>001 to 005</p> <p>255 - Unknown</p>

## SLOT\_STATUS

<b>Description</b>	Discovery of the slot status.
<b>Commands</b>	<p>An empty slot:</p> <pre>&lt; GET x SLOT_STATUS z &gt; &lt; REP x SLOT_STATUS z EMPTY &gt;</pre> <p>Slot 1 with an AD transmitter:</p> <pre>&lt; GET x SLOT_STATUS 1 &gt; &lt; REP x SLOT_STATUS 1 STANDARD &gt;</pre> <p>Slot 2 with an ADX transmitter that is online:</p> <pre>&lt; GET x SLOT_STATUS 2 &gt; &lt; REP x SLOT_STATUS 2 LINKED.ACTIVE &gt;</pre> <p>The transmitter goes offline: (among others)</p> <pre>&lt; REP x SLOT_STATUS 2 LINKED.INACTIVE &gt;</pre> <p>The user registers an AD1 to Slot 2:</p> <pre>&lt; REP x SLOT_STATUS 2 STANDARD &gt;</pre> <p>The user unlinks or cleans up Slot 2 or moves the transmitter:</p> <pre>&lt; REP x SLOT_STATUS 2 EMPTY &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Report responses:</p> <p>EMPTY - No transmitter is registered in the slot</p> <p>STANDARD - An AD (standard) transmitter is registered in the slot</p> <p>LINKED.INACTIVE - An ADX (enhanced) transmitter is registered, linked, but the receiver is not connected to the live transmitter at this time</p> <p>LINKED.ACTIVE - An ADX (enhanced) transmitter is registered, linked, and the receiver is connected. You can remotely adjust the transmitter now.</p>

## SLOT\_TX\_DEVICE\_ID

<b>Description</b>	Controls the device ID of the transmitter by slot.
<b>Commands</b>	<p>An empty, unregistered slot:</p> <pre>&lt; GET x SLOT_TX_DEVICE_ID z &gt; &lt; REP x SLOT_TX_DEVICE_ID z {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} &gt;</pre> <p>A slot with an AD (standard) transmitter:</p>

	<pre>&lt; GET x SLOT_TX_DEVICE_ID z &gt; &lt; REP x SLOT_TX_DEVICE_ID z {AD1yyyyyyyyyyyyyyyyyyyyyyyy} &gt;</pre> <p>An attempt to set the Device ID of an AD transmitter that is not supported:</p> <pre>&lt; SET x SLOT_TX_DEVICE_ID z {Name_1} &gt; &lt; REP ERR &gt;</pre> <p>An attempt to set the Device ID of a Linked.Active ADX transmitter:</p> <pre>&lt; SET x SLOT_TX_DEVICE_ID z {Name_1} &gt; &lt; REP x SLOT_TX_DEVICE_ID z {Name_1yyyyyyyyyyyyyyyyyyyy} &gt;</pre>
<b>Variables</b>	<p>Where <b>x</b> is the channel number and <b>z</b> is the slot number.</p> <p>Where the repeating <b>y</b> represents the spaces returned by the device to pad the response to 31 characters.</p>
<b>Notes</b>	<p>SET only accepts 1 to 8 characters from the set: A-Z,a-z,0-9,!"#\$%&amp;'()*+,-./:;&lt;=&gt;?@[]^_`~ and space</p> <p>The device always responds with a 31-character string</p> <p>Empty slot - blank, all spaces</p>

## SLOT\_TX\_MODEL

<b>Description</b>	Discovery of the transmitter slot's model.
<b>Commands</b>	<p>An empty slot:</p> <pre>&lt; GET x SLOT_TX_MODEL z &gt; &lt; REP x SLOT_TX_MODEL z UNKNOWN &gt;</pre> <p>An occupied slot:</p> <pre>&lt; GET x SLOT_TX_MODEL z &gt; &lt; REP x SLOT_TX_MODEL z ADX1 &gt;</pre>
<b>Variables</b>	Where <b>x</b> is the channel number and <b>z</b> is the slot number.
<b>Notes</b>	<p>Report responses:</p> <p>AD1</p> <p>AD2</p> <p>ADX1</p> <p>ADX1M</p> <p>ADX2</p>

	ADX2FD
--	--------

	UNKNOWN
--	---------