

Command Strings for Axient Systems

The Axient system is connected via Ethernet to a control system, such as AMX or Crestron.

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

The Axient system has 4 types of strings, as follows:

1. GET – The GET command is used to find the status of a parameter. After the AMX/Crestron sends a GET command, the Axient system responds with a REPORT string.
2. SET – The SET command is used to change the status of a parameter. After the AMX/Crestron sends a SET command, the Axient system will respond with a REPORT string to indicate the new value of the parameter.
3. REP – When the Axient system receives a GET or SET command, it will reply with a REPORT command to indicate the status of the parameter. REPORT is also sent by the Axient system when a parameter is changed via the front panel or via Wireless Workbench.
4. SAMPLE – Used for metering RF levels and audio levels.

All messages sent and received are ASCII. Note that the level indicators and gain indicators are also in ASCII.

Most parameters will send a REPORT command then they change. Thus, it is not necessary to constantly query battery or interference parameters. The receiver will send a REPORT command when any of these parameters change.

Commands can be sent to the AXT400 receiver and the AXT900 battery charger as noted in the heading of the tables below.

The character "x" in all of the following strings represents the channel of that particular receiver and can be ASCII numbers 0, 1, or 2. With a single channel receiver, the channel is always 1. With a dual receiver, the left channel is 1 and the right channel is 2. Using the number 0 will report both channels on a dual receiver.

ATX400 Commands

View All	Command String:	< GET x ALL >	Where x is ASCII channel number: 0, 1, or 2. This command is intended to get all parameters on first power up.
	Axient Response:	< REP x CHAN_NAME {yyyyyyyy} > < REP x AUDIO_MUTE yy > < REP x AUDIO_GAIN yyy > < REP x GROUP_CHAN gg,cc > < REP x FREQUENCY yyyyyy > . . . etc.	The Axient device will respond with all parameters. See below for the definition of all REPORT commands. This is intended for use when first powering up a sound system.
View Device ID	Command String:	< GET DEVICE_ID >	The Device ID command does not contain the x channel character, as it is for the entire device.
	Axient Response:	< REP DEVICE_ID {yyyyyyyy} >	Where yyyyyyyy is 8 characters of the device ID. The Axient receiver always responds with an 8 character device ID.
Set Device ID	Command String:	< SET DEVICE_ID {yyyyyyyy} >	Where yyyyyyyy is 8 characters of the device ID. The device ID can be 1 to 8 characters long.
	Axient Response:	< REP DEVICE_ID {yyyyyyyy} >	Where yyyyyyyy is 8 characters of the device ID. The Axient receiver always responds with an 8 character device ID.
View Channel Name	Command String:	< GET x CHAN_NAME >	Where x is ASCII channel number: 0, 1, or 2.
	Axient Response:	< REP x CHAN_NAME {yyyyyyyy} >	Where yyyyyyyy is 8 characters of the channel name. The Axient receiver always responds with an 8 character name.
Set Channel Name	Command String:	< SET x CHAN_NAME {yyyyyyyy} >	Where yyyyyyyy is 8 characters of the channel name. The channel name can be 1 to 8 characters long.
	Axient Response:	< REP x CHAN_NAME {yyyyyyyy} >	
Get Mute Status	Command String:	< GET x AUDIO_MUTE >	
	Axient Response:	< REP x AUDIO_MUTE ON > < REP x AUDIO_MUTE OFF >	The Axient will respond with one of the two strings.
Mute Audio	Command String:	< SET x AUDIO_MUTE ON >	
	Axient Response:	< REP x AUDIO_MUTE ON >	
Un-mute Audio	Command String:	< SET x AUDIO_MUTE OFF >	
	Axient Response:	< REP x AUDIO_MUTE OFF >	
Toggle Mute	Command String:	< SET x AUDIO_MUTE TOGGLE >	
	Axient Response:	< REP x AUDIO_MUTE ON > < REP x AUDIO_MUTE OFF >	The Axient will respond with one of the two strings.
Get Audio Gain	Command String:	< GET x AUDIO_GAIN >	
	Axient Response:	< REP x AUDIO_GAIN yyy >	Where yyy takes on the ASCII values of 000 to 030. yyy minus 30 equals the value on the display of the Axient receiver.
Set Audio Gain	Command String:	< SET x AUDIO_GAIN yyy >	Where yyy takes on the ASCII values of 000 to 030.
	Axient Response:	< REP x AUDIO_GAIN yyy >	Where yyy takes on the ASCII values of 000 to 030.
Increase Audio Gain by n dB	Command String:	< SET x AUDIO_GAIN INC n >	Where n is the amount in dB to increase the gain. Valid n values are 1 through 30.
	Axient Response:	< REP x AUDIO_GAIN yyy >	Where yyy takes on the ASCII values of 000 to 030.
Decrease Audio Gain by n dB	Command String:	< SET x AUDIO_GAIN DEC n >	Where n is the amount in dB to decrease the gain. Valid n values are 1 through 30.
	Axient Response:	< REP x AUDIO_GAIN yyy >	Where yyy takes on the ASCII values of 000 to 030.

Get current Group, Channel	Command String:	< GET x GROUP_CHAN >	
	Axient Response:	< REP x GROUP_CHAN gg, cc >	Where <i>gg</i> is Group Number and <i>cc</i> is Channel Number. If the receiver is on a frequency that does not line up with a group and channel, then <i>gg</i> and <i>cc</i> will report '--,--'.
Set Group and Channel	Command String:	< SET x GROUP_CHAN gg, cc >	Where <i>gg</i> and <i>cc</i> are the group and channel numbers.
	Axient Response:	< REP x FREQUENCY yyyyyy > < REP x GROUP_CHAN gg, cc >	Axient responds with both strings. Where <i>gg</i> is Group Number and <i>cc</i> is Channel Number. Where <i>yyyyyy</i> is the Frequency represented as <i>yyy.yyy</i> MHz. If the receiver is on a frequency that does not line up with a group and channel, then <i>gg</i> and <i>cc</i> will report '--,--'.
Get current Frequency	Command String:	< GET x FREQUENCY >	
	Axient Response:	< REP x FREQUENCY yyyyyy >	Where <i>yyyyyy</i> is the Frequency represented as <i>yyy.yyy</i> MHz.
Set Frequency	Command String:	< SET x FREQUENCY yyyyyy >	Where <i>yyyyyy</i> is the Frequency represented as <i>yyy.yyy</i> MHz.
	Axient Response:	< REP x FREQUENCY yyyyyy > < REP x GROUP_CHAN gg, cc >	Axient responds with both strings. Where <i>gg</i> is Group Number and <i>cc</i> is Channel Number. Where <i>yyyyyy</i> is the Frequency represented as <i>yyy.yyy</i> MHz. If the receiver is on a frequency that does not line up with a group and channel, then <i>gg</i> and <i>cc</i> will report '--,--'.
Get Battery Run Time	Command String:	< GET x TX_BATT_RUN_TIME >	
	Axient Response:	< REP x TX_BATT_RUN_TIME yyyyyy >	Where <i>yyyyyy</i> is the minutes until the transmitter turns itself off. <i>yyyyyy</i> =65535 when transmitter is off. <i>yyyyyy</i> =65534 when transmitter is calculating.
Get Battery Type	Command String:	< GET x TX_BATT_TYPE >	
	Axient Response:	< REP x TX_BATT_TYPE ALKA > < REP x TX_BATT_TYPE LION > < REP x TX_BATT_TYPE LITH > < REP x TX_BATT_TYPE NIMH > < REP x TX_BATT_TYPE UNKN >	The Axient will respond with one of the five strings.
Get Battery Bars	Command String:	< GET x TX_BATT_BARS >	
	Axient Response:	< REP x TX_BATT_BARS yyy >	Where <i>yyy</i> is the number of bars shown on the transmitter. Valid values are 000 through 005. (ex. 005 = 5 bars). <i>yyy</i> =255 when value is unknown.
Get Transmitter Type	Command String:	< GET x TX_TYPE >	
	Axient Response:	< REP x TX_TYPE AXT100 > < REP x TX_TYPE AXT200 > < REP x TX_TYPE AXT210 > < REP x TX_TYPE UR1 > < REP x TX_TYPE UR1M > < REP x TX_TYPE UR1H > < REP x TX_TYPE UR2 > < REP x TX_TYPE MW1 > < REP x TX_TYPE MW2 > < REP x TX_TYPE UNKN >	The Axient will respond with one of these strings.
Get Transmitter Offset	Command String:	< GET x TX_SENS >	
	Axient Response:	< REP x TX_SENS yyy >	Where <i>yyy</i> is the transmitter sensitivity. Values are 000, 010, and 025. <i>yyy</i> minus 10 equals value displayed on transmitter. <i>yyy</i> =255 when value is unknown.
Get Transmitter RF Power	Command String:	< GET x TX_RF_PWR >	

	Axient Response:	< REP x TX_RF_PWR LOW > < REP x TX_RF_PWR NORMAL > < REP x TX_RF_PWR HIGH > < REP x TX_RF_PWR UNKNOWN >	The Axient will respond with one of the four strings.
Get Transmitter Lock	Command String:	< GET x TX_LOCK >	
	Axient Response:	< REP x TX_LOCK UNLOCKED > < REP x TX_LOCK LOCKED > < REP x TX_LOCK PWR > < REP x TX_LOCK FREQ+PWR > < REP x TX_LOCK FREQ > < REP x TX_LOCK UNKNOWN >	The Axient will respond with one of these strings.
Get Transmitter Audio Gain	Command String:	< GET x TX_AUDIO_GAIN >	
	Axient Response:	< REP x TX_BATT_BARS yyy >	Where yyy is the number of bars shown on the transmitter. Valid values are 000 through 050. yyy minus 10 equals value on transmitter display. yyy=255 when value is unknown.
Get Transmitter Device ID	Command String:	< GET x TX_DEVICE_ID >	
	Axient Response:	< REP DEVICE_ID {yyyyyyyy} >	Where yyyyyyyy is 8 characters of the device ID. The Axient receiver always responds with an 8 character device ID.
Get Interference Detection	Command String:	< GET x INTERFERENCE >	
	Axient Response:	< REP x INTERFERENCE NONE > < REP x INTERFERENCE ALERT > < REP x INTERFERENCE DETECTED > < REP x INTERFERENCE DISABLED >	The Axient will respond with one of the four strings. The Axient will send a REP command whenever this status changes. Thus, it is not necessary to constantly query this.
Turn Metering On	Command String:	< SET x METER_RATE sssss >	Where sssss is the metering speed in milliseconds. Setting sssss=0 turns metering off. Minimum setting is 100 milliseconds. Metering is off by default.
	Axient Response:	< REP x METER_RATE sssss > < SAMPLE x ALL nn aaa bbb eee >	See below.
Stop Metering	Command String:	< SET x METER_RATE 0 >	A value of 00000 is also acceptable.
	Axient Response:	< REP x METER_RATE 00000 >	

Notes on metering

- Where sssss is the metering speed in milliseconds. Setting sssss=0 turns metering off. Minimum setting is 100 milliseconds. Maximum setting is 99999 milliseconds. Metering is off by default.
- Where nn indicates the blue RF LED's from the receiver. These show the squelch status of the receiver and take on the following ASCII values.
 - AX – Antenna A on, Antenna B off
 - XB – Antenna A off, Antenna B on
 - AB – Antenna A on, Antenna B on
 - XX – Antenna A off, Antenna B off
- Where aaa and bbb is the value of the RF level received for antennas A and B respectively, and is 000-255.
- Where eee is the audio level and is 000-255.
 - 128= all LEDs on
 - 64 = G1,G2, G3, G4, Y5, Y6, Y7 are on
 - 32 = G1,G2, G3, G4, Y5 & Y6 are on
 - 16 = G1,G2, G3, G4, & Y5 are on
 - 8 = G1,G2, G3, G4 are on
 - 4 = G1,G2, & G3 are on
 - 2 = G1 & G2 are on
 - 1 = G1 LED is on
 - 0 = All LEDs off

Shure Incorporated
5800 Touhy Ave
Niles IL 60714

Phone: 847-600-8440
Fax: 847-600-8444
support@shure.com

Each receiver channel can have two transmitters linked to it for ShowLink control. These are referred to as "Slot 1" and "Slot 2". The following table uses these slot numbers. These strings are to be sent to the AXT400 receiver.

Get ShowLink Linked Device Status	Command String:	< GET x LINKED_DEVICE_STS z >	Where z is the slot number and can be 0, 1 or 2. 0 = slot 1 and slot 2 1 = slot 1 2 = slot 2
	Axient Response:	< REP x LINKED_DEVICE_STS z LINKED > < REP x LINKED_DEVICE_STS z UNLINKED >	The Axient will respond with one of these strings.
Get Showlink Status	Command String:	< GET x SHOWLINK_STS z >	Where z is the slot number and can be 0, 1 or 2.
	Axient Response:	< REP x SHOWLINK_STS z ONLINE > < REP x SHOWLINK_STS z OFFLINE > < REP x SHOWLINK_STS z UNKNOWN >	The Axient will respond with one of these strings.

AXT900 Commands

The character "v" in all of the following strings represents the battery bays in the AXT900 charger and can be ASCII numbers 0 through 8. The bays are numbered left to right, 1 through 8. Using the number 0 will report all bays on the charger.

Get Battery Charge State	Command String:	< GET v BATT_STATE >	Where v is ASCII channel number: 0 through 8.
	Axient Response:	< REP v BATT_STATE FULL > < REP v BATT_STATE CALCULATING > < REP v BATT_STATE NORMAL > < REP v BATT_STATE ERROR > < REP v BATT_STATE NO_BATT > < REP v BATT_STATE DISCHARGING > < REP v BATT_STATE DISCHARGE_CALC > < REP v BATT_STATE DISCHARGE_WARM > < REP v BATT_STATE WARM > < REP v BATT_STATE WARM_FULL > < REP v BATT_STATE HOT > < REP v BATT_STATE COLD > < REP v BATT_STATE PRECHARGE > < REP v BATT_STATE PRECHARGE_FAIL > < REP v BATT_STATE CHECK_CHARGER > < REP v BATT_STATE ERROR_FULL > < REP v BATT_STATE ERROR_WARM_FULL > < REP v BATT_STATE READY_TO_STORE >	The Axient will respond with one of these strings.
Get battery charge percentage	Command String:	< GET v BATT_CHARGE >	
	Axient Response:	< REP v BATT_CHARGE yyy >	Where yyy takes on the ASCII values of 000 to 100 representing percentage charged. yyy=255 when there is no bay, no battery, or battery read error.
Get battery cycles	Command String:	< GET v BATT_CYCLE >	
	Axient Response:	< REP v BATT_CYCLE yyyyy >	Where yyyyy is the number of battery cycles. yyyyy=65535 when there is no bay, no battery, or battery read error.
Get battery health	Command String:	< GET v BATT_HEALTH >	
	Axient Response:	< REP v BATT_HEALTH yyy >	Where yyy takes on the ASCII values of 000 to 100 representing percentage health. yyy=255 when there is no bay, no battery, or battery read error.
Determine if battery is in charger	Command String:	< GET v BATT_DETECTED >	
	Axient Response:	< REP v BATT_DETECTED YES > < REP v BATT_DETECTED NO >	The Axient will respond with one of these strings.
Get battery time to full	Command String:	< GET v BATT_TIME_TO_FULL >	
	Axient Response:	< REP v BATT_TIME_TO_FULL FULL > < REP v BATT_TIME_TO_FULL CALC > < REP v BATT_TIME_TO_FULL yyyyy >	The Axient will respond with one of these strings. Where yyyyy is the number of minutes until the battery is fully charged.