CREATED BY
BIAMP™

CONTROL MANUAL

RS-232 and Telnet

November 3, 2002

Table of Contents

Overview Pgs 2 ~ 3

RS-232 Control Pg 4

Telnet Control Pg 5

ATP String Components:
- **Command** Pgs 6 & 7
- **Device Number** Pg 8
- **Attribute** Pgs 9 ~ 21
- **Instance ID Numbers** Pg 22
- **Index** Pg 23
- **Value** Pg 24

Responses Pg 25

Control Dialog
- **Overview** Pg 26
- **Levels** Pg 27
- **Presets** Pg 28
- **Meters** Pg 29

Overview

Audia can be controlled via the control dialogs in the Audia software, or via third-party controllers using RS-232 or Telnet.

For control of Audia, Biamp created ATP (Audia Text Protocol). This simply means that Audia will accept strings of ASCII characters to control and read settings of gain, mute, logic state, frequency, audio levels, and other parameters of DSP Blocks in Audia products.

ATP strings can be sent via third-party controllers using RS-232 (see page 4) or Telnet via TCP/IP (see page 5). A line feed needs to be sent after each command string sent.

The ATP string is structured in the following order: *Command DeviceNumber Attribute InstanceIDNumber Index1 Index2 Value* <LF>

ATP strings require a space between each parameter; the last character in the string needs to be a line feed <LF>.

For each control string a few components will need to be derived from the Audia software: *Device Number*, *InstanceIDNumber*, and *Index*. *Command* and *Attribute* are derived from this document. In a SET command, *Value* is used to specify what the DSP block attribute is to be set to. In an increment or decrement (INC or DEC) command *Value* is used to specify how much the DSP block attribute is to be changed by.

****Note**** Audia software will assign an Instance ID to each DSP block on the initial compile of the system. Subsequent compiles will not change the Instance IDs unless the "Reassign Instance IDs" check box is selected in the 'Compile' tab of the 'Options' screen (located on the Tools pull-down menu) in the Audia software.

Example: A string to control a fader might look like this:
SET 1 FDRLVL 2 1 9 <LF>. The individual components for this string are:

Command	Device Number	Attribute	Instance ID Number	Index1	Index2	Value	Line feed
SET	**1**	**FDRLVL**	**2**	**1**		**9**	**<LF>**

Notice that Index2 is not used since there is nothing entered in this parameter.

Example: A string to mute a standard mixer output might look like this:
SET 3 SMMUTEOUT 5 5 1 <LF> The individual components for this string are:

Command	Device Number	Attribute	Instance ID Number	Index1	Index2	Value	Line feed
SET	**3**	**SMMUTEOUT**	**5**		**5**	**1**	**<LF>**

Notice that Index1 is not used since there is nothing entered in this parameter.

Example: A string to turn up a cross point on a matrix mixer might look like this:
INC 2 MMLVLXP 4 3 2 1 <LF> The individual components for this string are:

Command	Device Number	Attribute	Instance ID Number	Index1	Index2	Value	Line feed
INC	**2**	**MMLVLXP**	**4**	**3**	**2**	**1**	**<LF>**

In this example Index1 and Index2 are both used, together they specify which cross point is to be changed.

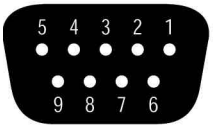
RS-232 Control

The RS-232 port on the back of an Audia unit is set to a default baud rate of 38400, 8 data bits, no parity, 1 stop bit, and no flow control. (38400:8:None:1). If multiple Audias are connected together in a system then only one RS-232 port needs to be connected to a third-party control system; communication data is shared via Ethernet through a switch.

When controlling multiple Audia units that are not part of the same DAP file, each Audia unit will need to be addressed via its own RS-232 port from a control system or PC. Audia units cannot be linked together via RS-232, like some other BIAMP Advantage products can.

(The RS-232 baud rate can be set to 9600, 19200, or 38400 – default is 38400)

A straight through PC Serial Cable is used to communicate from an RS-232 port on a third-party controller (or PC*) to the RS-232 port located on the back of an Audia unit.



pin #1 = not used
pin #2 = Transmit Data (TxD) output
pin #3 = Receive Data (RxD) input
pin #4 = not used
pin #5 = ground
pin #6 = not used
pin #7 = not used
pin #8 = not used
pin #9 = not used

(* A PC can be used with a terminal emulator program, such as HyperTerminal etc, to send/receive ATP Strings…Audia software must be connected via Ethernet in order to communicate.)

Telnet Control

Audia can be controlled using Telnet via TCP/IP. The same command strings that are used for RS-232 Control are used for Telnet.

When controlling multiple Audias that are not a part of the same DAP file, each Audia device will need to be addressed via its own Telnet session from a control system or PC.

SET – Tells Audia that a DSP attribute is to be set to a specific value – may contain negative numbers and/or decimal points
GET - Tells Audia that a DSP attribute is to be read – Response may contain a decimal point and/or a negative number.
INC - Tells Audia that a DSP attribute is to be incremented by a specific amount
DEC- Tells Audia that a DSP attribute is to be decremented by a specific amount
RECALL – Tells Audia that a preset is to be recalled.

SETL and GETL can be used if negative numbers and/or decimals are not supported by a control system.

SETL – Tells Audia that a DSP attribute is to be set to a specific value, no decimal places or negative numbers –To convert a dB number: add 100 to the desired level and then multiply by 10.

> Example: To set a level to –60.5dB, add 100 (-60.5 +100 = 39.5). Then multiply by 10 (39.5 X 10 = 395). Instead of Value being –60.5, Value after this SETL command will equal 395.

GETL - Tells Audia that a DSP attribute is to be read without negative numbers or decimals. To convert this number to dB: divide the number by 10, then subtract 100.

> Example: With a returned GETL response of 405, divide by 10 (405 / 10 = 40.5), and then subtract 100 (40.5 – 100 = -59.5dB)

For your reference, the SETL/GETL Table on the following page shows .5dB increments converted into the SETL/GETL format.

Some Attributes do not support all commands. The *Attribute* section defines which commands support SET/SETL, GET/GETL, INC, or DEC functions. RECALL is only used on preset commands.

*When GET or GETL is used, a *Value* will not need to be specified since GET/GETL is a request command. A Value must be specified in order for strings with SET/SETL, INC, DEC, and RECALL to work.

ATP String Component: Command

Level	Value	Level	Value	Level	Value	Level	Value	Level	Value
12	1120	-10.5	895	-33	670	-55.5	445	-78	220
11.5	1115	-11	890	-33.5	665	-56	440	-79.5	215
11	1110	-11.5	885	-34	660	-56.5	435	-79	210
10.5	1105	-12	880	-34.5	655	-57	430	-79.5	205
10	1100	-12.5	875	-35	650	-57.5	425	-80	200
9.5	1095	-13	870	-35.5	645	-58	420	-80.5	195
9	1090	-13.5	865	-36	640	-58.5	415	-81	190
8.5	1085	-14	860	-36.5	635	-59	410	-81.5	185
8	1080	-14.5	855	-37	630	-59.5	405	-82.5	175
7.5	1075	-15	850	-37.5	625	-60	400	-83	170
7	1070	-15.5	845	-38	620	-60.5	395	-83.5	165
6.5	1065	-16	840	-38.5	615	-61	390	-84	160
6	1060	-16.5	835	-39	610	-61.5	385	-84.5	155
5.5	1055	-17	830	-39.5	605	-62	380	-85	150
5	1050	-17.5	825	-40	600	-62.5	375	-85.5	145
4.5	1045	-18	820	-40.5	595	-63	370	-86	140
4	1040	-18.5	815	-41	590	-63.5	365	-86.5	135
3.5	1035	-19	810	-41.5	585	-64	360	-87	130
3	1030	-19.5	805	-42	580	-64.5	355	-87.5	125
2.5	1025	-20	800	-42.5	575	-65	350	-88	120
2	1020	-20.5	795	-43	570	-65.5	345	-88.5	115
1.5	1015	-21	790	-43.5	565	-66	340	-89	110
1	1010	-21.5	785	-44	560	-66.5	335	-89.5	105
.5	1005	-22	780	-44.5	555	-67	330	-90	100
0	1000	-22.5	775	-45	550	-67.5	325	-90.5	95
-.5	995	-23	770	-45.5	545	-68	320	-91	90
-1	990	-23.5	765	-46	540	-68.5	315	-91.5	85
-1.5	985	-24	760	-46.5	535	-69	310	-92	80
-2	980	-24.5	755	-47	530	-69.5	305	-92.5	75
-2.5	975	-25	750	-47.5	425	-70	300	-93	70
-3	970	-25.5	745	-48	520	-70.5	295	-93.5	65
-3.5	965	-26	740	-48.5	515	-71	290	-94	60
-4	960	-26.5	735	-49	510	-71.5	285	-94.5	55
-4.5	955	-27	730	-49.5	505	-72	280	-95	50
-5	950	-27.5	725	-50	500	-72.5	275	-95.5	45
-5.5	945	-28	720	-50.5	495	-73	270	-96	40
-6	940	-28.5	715	-51	490	-73.5	265	-96.5	35
-6.5	935	-29	710	-51.5	485	-74	260	-97	30
-7	930	-2.5	705	-52	480	-74.5	255	-97.5	25
-7.5	925	-30	700	-52.5	475	-75	250	-98	20
-8	920	-30.5	695	-53	470	-75.5	245	-98.5	15
-8.5	915	-31	690	-53.5	465	-76	240	-99	10
-9	910	-31.5	685	-54	460	-76.5	235	-99.5	5
-9.5	905	-32	680	-54.5	455	-77	230	-100	0
-10	900	-32.5	675	-55	450	-77.5	225		

Table of SETL and GETL levels converted from integers.

ATP String Component: Device Number

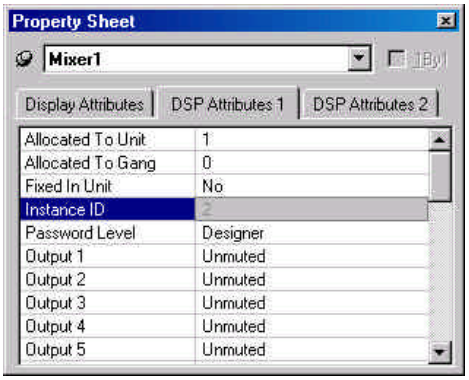
An Audia Device Number represents the physical Audia box's defined address. The Audia software automatically sets this number when a system is Compiled and loaded.

The Device number that a DSP block has been assigned to can be determined in 3 ways:

First Way:

1) Right click on the DSP block and select '*Properties*'.
2) Click on DSP 1 attributes tab and scroll down. The device that the block is assigned to will be displayed in the 'Allocated To Unit' field.

*Note: Each DSP block can be assigned to a device by changing 'Fixed In to Unit" to 'Yes' (this is defaulted to no).



Second Way:

1) In the *Display tab* of the *Options* screen select "Display Device Assignment in DSP Block info field".
2) This will display the device that each DSP block is assigned to on the main screen.

DSP block without Device number

DSP block with Device number

Third Way:

1) While connected an Audia's RS-232 port, type the string; GET 0 DEVID Audia will return the Device Number of the unit you are connected to.

ATP String Component: Attribute

The Attribute defines the portion of the DSP block to be controlled (fader level, crosspoint mute etc). The following tables show whether each ATP Attribute supports SET/SETL, GET/GETL, INC, and DEC *Commands*, as well as the *Value* range that the Attribute will accept. Index1/Index2 determines whether *Index1*, *Index2* or **BOTH** are needed for a ATP String to be complete.

ATP Strings can address:

Input/Output Blocks
Mixer Blocks
Equalizer Blocks
Filter Blocks
Crossover Blocks
Dynamic Blocks
Routers
Delays
Meters
Generators

Input/Output Blocks

Analog Inputs	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Input Gain	MICGAIN	Y	Y	Y	Index1	0 ~ 66 *
Input Level	INPLVL	Y	Y	Y	Index1	-100 ~ 12 *
Phantom Power	PHPWR	Y	Y	N	Index1	0 = off 1= on
Input Mute	INPMUTE	Y	Y	N	Index1	0=unmuted 1=muted
Invert Polarity	INPINVRT	Y	Y	N	Index1	0=normal 1=inverted

*** Can contain a decimal number.**

Example: In the command: SET 1 INPLVL 6 3 –10<LF>
We are telling Audia device 1 to set an input fader, Instance ID 6, input 3 (index1), to –10dB. (We set channel 3 input level of instance 6 to –10dB)

Analog Outputs	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Output Level	OUTLVL	Y	Y	Y	Index2	-100 ~ 0 *
Output Mute	OUTMUTE	Y	Y	N	Index2	0=unmuted 1=muted
Invert Polarity	OUTINVRT	Y	Y	N	Index2	0=normal 1=inverted
Mic level pad	OUTPAD	Y	Y	N	Index2	0 = off 1= on

*** Can contain a decimal number.**

Example: In the command: SET 2 OUTMUTE 3 4 1<LF>
We are telling Audia device 2 to set the output mute Instance ID 3, output 4 (index1) to on. (We muted the 4th output of Instance ID 3 of an Audia)

CobraNet	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
CobraNet RX	CNINBNDL	Y	Y	N	None	0 ~ 61439
CobraNet TX	CNOUTBNDL	Y	Y	N	None	0 ~ 61439
CobraNet Input	CNNLVLIN	Y	Y	Y	Index1	-100 ~ 12 *
CobraNet Output	CNNLVLOUT	Y	Y	Y	Index1	-100 ~ 12 *

**** Can contain a decimal number.***

Mixer Blocks

Automixer	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Input Level	**AMLVLIN**	**Y**	**Y**	**Y**	**Index1**	**-100 ~ 12 ***
Output Level	**AMLVLOUT**	**Y**	**Y**	**Y**	**None**	**-100 ~ 12 ***
Input Mute	**AMMUTEIN**	**Y**	**Y**	**N**	**Index1**	**0=unmuted 1=muted**
Output Mute	**AMMUTEOUT**	**Y**	**Y**	**N**	**None**	**0=unmuted 1=muted**
Crosspoint Mute	**AMMUTEXP**	**Y**	**Y**	**N**	**Index1**	**1=unmuted 0=muted**
Logic Output	**AMLOGOUT**	**Y**	**Y**	**N**	**Index 2**	**0=off 1=on**

*** Can contain a decimal number.**

Example: In the command: SET 1 AMMUTEXP 3 1 0<LF>
We are telling Audia device 1 to set the automixer's crosspoint (row 1, column 1) on instance ID 3 to off.

Note: For attributes with 2 Index fields, *Index1* is the row, and *Index2* is the column.

Matrix Mixer	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Input Level	**MMLVLIN**	**Y**	**Y**	**Y**	**Index1**	**-100 ~ 0 ***
Output Level	**MMLVLOUT**	**Y**	**Y**	**Y**	**Index2**	**-100 ~ 0 ***
Input Mute	**MMMUTEIN**	**Y**	**Y**	**N**	**Index1**	**0=unmuted 1=muted**
Output Mute	**MMMUTEOUT**	**Y**	**Y**	**N**	**Index2**	**0=unmuted 1=muted**
Crosspoint Level	**MMLVLXP**	**Y**	**Y**	**Y**	**Index1 & Index2**	**-100 ~ 0 ***
Crosspoint Mute	**MMMUTEXP**	**Y**	**Y**	**N**	**Index1 & Index2**	**1=unmuted 0=muted**

*** Can contain a decimal number.**

Note: For attributes with 2 Index fields, *Index1* is the row, and *Index2* is the column.

Standard Mixer	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Input Level	SMLVLIN	Y	Y	Y	Index1	-100 ~ 12 *
Output Level	SMLVLOUT	Y	Y	Y	Index2	-100 ~ 12 *
Input Mute	SMMUTEIN	Y	Y	N	Index1	0=unmuted 1=muted
Output Mute	SMMUTEOUT	Y	Y	N	Index2	0=unmuted 1=muted
Crosspoint Mute	SMMUTEXP	Y	Y	N	Index1 & Index2	1=unmuted 0=muted

**** Can contain a decimal number.***

Note: *For attributes with 2 Index fields,* Index1 *is the row, and* Index2 *is the column.*

Room Combiner	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Wall State	RMCMBWALL	Y	Y	N	Index1	0 = down 1 = up
Input Level	RMCMBLVL	Y	Y	Y	Index1	-100 ~ 12 *
Input Mute	RMCMBMUTE	Y	Y	N	Index1	0=unmuted 1=muted

**** Can contain a decimal number.***

Note: *For wall state, Index1 represents the wall being opened or closed**.***

Mix-minus Combiner	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Group State	MCMBGROUP	Y	Y	N	Both	0 = not grouped 1 = grouped

.

Note*: Index1 is the input number being addressed. Index2 is the group number; this is represented by a letter in the software, but addressed as a number in ATP commands. (Example: A=1, B=2, C=3….)*

Equalizer Blocks

Graphic EQ	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
EQ Band Level	**GEQLVLBND**	**Y**	**Y**	**Y**	**Index1**	**-27 ~ 15 ***
Bypass All Bands	**GEQBYPALL**	**Y**	**Y**	**N**	**None**	**0 = active 1 = bypassed**

- **Can contain a decimal number.**
-

The Index field represents the filter to be controlled. EQ filter farthest to the left is #1, next one is #2 and so forth. Upper left hand area of Graphic EQ dialog screen displays, which numbered filter, is being adjusted, use this number as the Index1 for Graphic EQ strings.

Example: In the string SET 2 GEQBYPALL 9 1 <LF>
We are telling Audia to bypass all bands on Graphic EQ Instance ID 9.

Parametric EQ	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
EQ Bandwidth	**PEQBWBND**	**Y**	**Y**	**Y**	**Index1**	**0.01 ~ 4.0 ***
Bypass EQ Band	**PEQBYPBND**	**Y**	**Y**	**N**	**Index1**	**0 = active 1 = bypassed**
Center Freq.	**PEQFCBND**	**Y**	**Y**	**Y**	**Index1**	**20 ~ 20000**
EQ Band Level	**PEQLVLBND**	**Y**	**Y**	**Y**	**Index1**	**-27 ~ 15 ***
Bypass All Bands	**PEQBYPALL**	**Y**	**Y**	**N**	**None**	**0 = active 1 = bypassed**

- **Can contain a decimal number.**

Upper left hand area of Parametric EQ dialog screen displays, which numbered filter, is being adjusted, use this number as the Index1 for Parametric EQ strings.

Example: In the string INC 1 PEQLVLBND 11 3 2<LF>
We are telling Audia to increment the second parametric EQ filter on Instance ID 11 by 2dB.

Filters Blocks

HPF	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cut off freq.	HPFLTFC	Y	Y	N	None	20 ~ 20000 *
Filter Bypass	HPFLTBYP	Y	Y	N	None	0 = active 1= bypassed

LPF	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cut off freq.	LPFLTFC	Y	Y	N	None	20 ~ 20000 *
Filter Bypass	LPFLTBYP	Y	Y	N	None	0 = active 1 = bypassed

High Shelf	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cut off freq.	HSFLTFC	Y	Y	N	None	20.0 ~ 20000 *
Gain	HSFTGAIN	Y	Y	Y	None	-27.0 ~ 9.0
Filter Bypass	HSFLTBYP	Y	Y	N	None	0 = active 1 = bypassed

Low Shelf	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cut off freq.	LSFLTFC	Y	Y	N	None	20.0 ~ 20000 *
Gain	LSFTGAIN	Y	Y	Y	None	-27 ~ 9
Filter Bypass	LSFLTBYP	Y	Y	N	None	0 = active 1 = bypassed

*** Can contain a decimal number.**

Example: SET 1 HSFTGAIN 100 –10<LF>
We are telling Audia to set the High Shelf filter gain on instance ID 100 to –10dB.

Crossover blocks

2-Way	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cutoff Freq.	**XOVR2FC**	**Y**	**Y**	**N**	**Index1**	**20 ~ 20000 ***

3-Way	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cutoff Freq.	**XOVR3FC**	**Y**	**Y**	**N**	**Index1**	**20 ~ 20000 ***

4-Way	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cutoff Freq.	**XOVR4FC**	**Y**	**Y**	**N**	**Index1**	**20 ~ 20000 ***

*** Can contain a decimal number.**

Example: GET 2 XOVR3FC 40 1 <LF>
We are asking Audia to tell us the 3-way crossover low-pass cutoff frequency on instance ID 40.

2-way crossover:

Index1 = 1 is the low-pass cutoff frequency
Index1 = 2 is the high-pass cutoff frequency

3-Way crossover:

Index1 = 1 is low-pass cutoff frequency
Index1 = 2 is the lower slope of the mid cutoff frequency
Index1 = 3 is the higher slope of the mid cutoff frequency
Index1 = 4 is the high-pass cutoff frequency

4-way crossover:

Index1 = 1 is the low-pass cutoff frequency
Index1 = 2 is the lower slope of the low-mid cutoff frequency
Index1 = 3 is the higher slope of the low-mid cutoff frequency
Index1 = 4 is the lower slope of the high-mid cutoff frequency
Index1 = 5 is the higher slope of the high-mid cutoff frequency
Index1 = 6 is the high-pass cutoff frequency

Dynamic Blocks

Leveler	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Bypass	LVLRBYP	Y	Y	N	None	0=active 1=bypassed

Comp / Limiter	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Bypass	CLIMBYP	Y	Y	N	None	0=active 1=bypassed

Ducker	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Input level	DKRLVLIN	Y	Y	N	None	-100 ~ 12 *
Level sense	DKRLVLSENSE	Y	Y	N	None	-100 ~ 12 *
Bypass Ducker	DKRBYP	Y	Y	N	None	0= active 1= bypassed
Mute sense	DKRMUTESENSE	Y	Y	N	None	0 = unmuted 1 = muted
Input mute	DKRMUTEIN	Y	Y	N	None	0 = unmuted 1 = muted
Logic in enable	DKRENLOGIN	Y	Y	N	None	0 = disabled 1= enabled
Logic out enable	DKRENLOGOUT	Y	Y	N	None	0 = disabled 1= enabled
Logic input invert	DKRINVLOGIN	Y	Y	N	None	0= normal 1= inverted
Logic output invert	DKRINVLOGOUT	Y	Y	N	None	0= normal 1= inverted

*** Can contain a decimal number.**

Noise Gate	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Bypass	NGBYP	Y	Y	N	None	0 = active 1 = bypassed

Example: SET 3 CLIMBYP 55 0<LF>
We are telling Audia to set the Comp/Limiter on instance ID 55 to active.

Router Blocks

Router	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Cross point	**RTRMUTEXP**	**Y**	**Y**	**N**	**Both**	**1 = off** **0 = on**

Example: SET 1 RTRXP 98 4 5 0<LF>

Note: For attributes with 2 Index fields, *Index1* is the row, and *Index2* is the column.

Delay Blocks
No Commands specified at this time

Control Blocks

Level Faders	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Fader Levels	**FDRLVL**	**Y**	**Y**	**Y**	**Index1**	**-100 ~ 12 ***
Mute Fader	**FDRMUTE**	**Y**	**Y**	**N**	**Index1**	**0 = unmuted 1 = muted**

*** Can contain a decimal number.**

Presets	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Preset	**PRESET**	**N**	**N**	**N**	**None**	**1001 to max preset number**

Note: Instead of SET/SETL, GET/GETL, INC, or DEC; the preset attribute uses RECALL as the Command. Preset numbers begin at 1001, regardless of preset name (the first preset defined is 1001, the next 1002, and so forth).

Example: in the string: RECALL 1 PRESET 1001<LF>
We are telling Audia to recall the first preset on an Audia system. Since presets apply to entire systems, or DAP files, the Device Number will always be 1 for Preset strings.

Meter Blocks

Meters	Attribute	SET / SETL	GET / GETL	INC / DEC	Index1/ Index2	Value Range
Signal	**SPMTRLVL**	**N**	**Y**	**N**	**Index1**	**-100 ~ 36 ***
Peak	**PKMTRLVL**	**N**	**Y**	**N**	**Index1**	**0 = off 1 = peak**
RMS	**RMSMTRLVL**	**N**	**Y**	**N**	**Index1**	**-100 ~ 36 ***

*** Can contain a decimal number.**

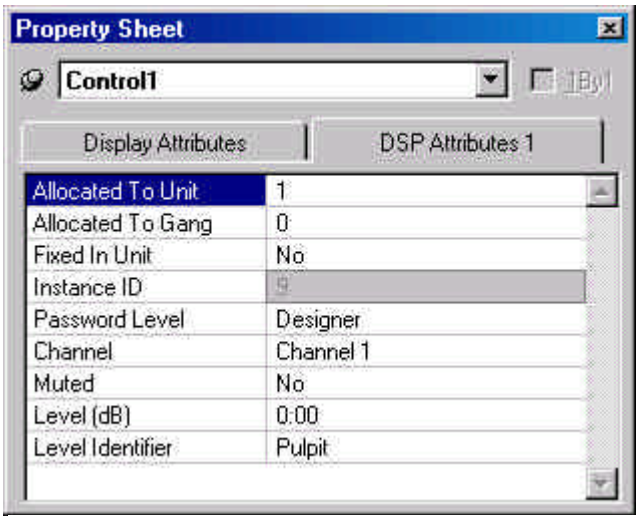
Example: GET 1 RMSMTRLVL 48 1 <LF>
We are asking Audia the RMS Meter 1 level on instance ID 48.

Generator Blocks

No Attributes are specified at this time

ATP String Component: Instance ID Number

Audia uses an instance ID number to specify the exact DSP block to be controlled. Right click on the DSP block and select “Properties”, the instance number can be found the DSP Attributes 1 tab.

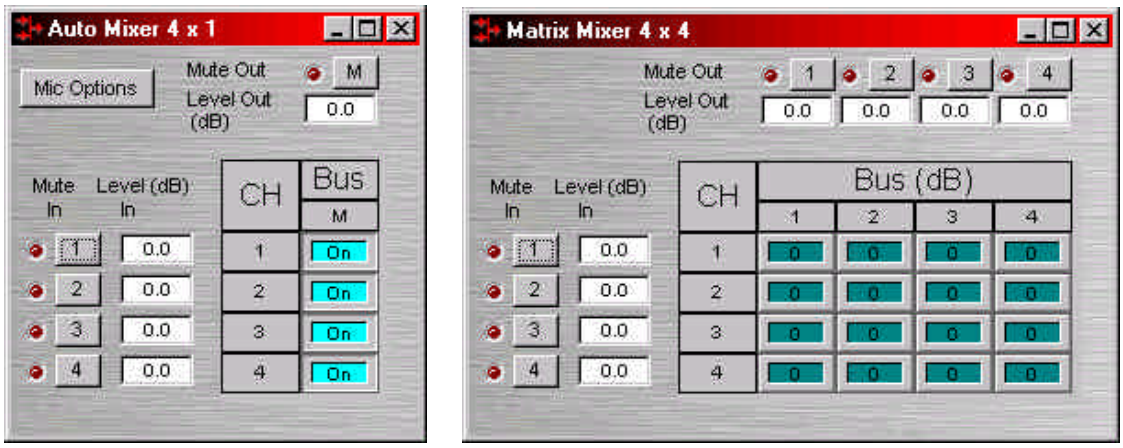


The property sheet can only be accessed in an offline screen.

****Note**** Audia software will assign an Instance ID to each DSP block on the initial compile of the system. Subsequent compiles will not change the Instance IDs unless the “Reassign Instance IDs” check box is selected in the ‘Compile’ tab of the ‘Options’ screen (located on the Tools pull-down menu) in the Audia software.

If one or more DSP blocks are disconnected from the audio path (when connections are not made to at least one input or output) the Instance IDs will be unassigned. New Instance ID(s) will be reassigned once the DSP block(s) are properly connected again.

Index refers to inputs, outputs, or crosspoints of an *Attribute*. Some Attributes will need an *Index1* (Input or Row) and/or an *Index2* (output or Column). The tables in the *Attribute* section will define which, if any, indexes are required for the string.



Example: INC 1 AMLVLIN 4 **1** 1<LF>
In an Automatic mixer on device 1 we are increasing the level of input 1 on instance ID 4 by 1dB. Index1 is used to represent the input number.

Example: DEC 2 AMLVLOUT 6 **1** 2<LF>
In an Automatic mixer on device 2 we are decreasing the level of output 1 on instance ID 6 by 2dB. Index2 is used to represent the output number.

Example: SET 1 MMLXP 5 **1 2** –5<LF>
In a matrix mixer on device 1 we are setting cross point 1:2 (row 1, column 2) level to –5dB on instance ID 5.

Note for commands with 2 Index fields, Index1 is the row, and Index2 is the column.

ATP String Component: Value

Value determines what a DSP block is being set to, incremented by, or decremented by. The *Attribute* section will define which type of value the string will need in order to execute the ATP string.

Example: INC 1 AMLVLIN 4 1 **1**<LF>
In an Automatic mixer input level string, the *Value* equal to **1** states that the fader is to be incremented by 1 dB.

Example: SET 2 MMLVLIN 5 2 **–100**<LF>
In a matrix mixer input level string, the value is set to –100dB.

Example: RECALL 1 PRESET **1004**<LF>
For a preset recall we are telling device 2 to recall the fourth preset.

*When GET is used, a *Value* will not need to be specified since GET is a request command. A *Value* must be set in order for SET, INC, DEC, and RECALL command strings to work.

Example: GET 4 MMLVLOUT 5 <LF>
We are asking the level of a matrix mixer's output. There is no *Value* required for this string.

Responses

When a successful SET, SETL, INC, DEC, or RECALL command is sent to an Audia device via RS-232 or Telnet, Audia will respond with: '+OK' followed by a carriage return and line feed.
Example: Sending the string: SET 2 AMLVLIN 4 2 1< LF> will result in an: +OK<CR><LF> response.

A successful GET or GETL command will result in a numerical response to the command string followed by <CR><LF>.

> The response from an ATP string with a GET command may contain a decimal point (with 4 numbers after the decimal) and/or negative numbers, depending on type of A*ttribute* addressed. If the control system does not support negative numbers or decimal places a GETL command may be used instead (see page 6).

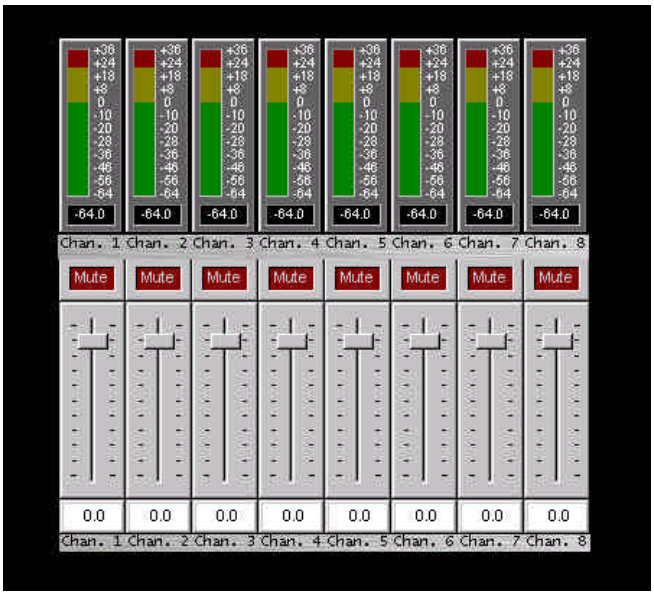
Example: After sending the above example to an Audia device the string: GET 2 AMLVLIN 4 2<LF> would result in the response: 1.0000<CR><LF> this tells us that the level is currently set to 1dB.

If an incorrect command string is sent, an Audia will respond with: -ERR <CR><LF>

Control Dialog - Overview

Audia software can be used to control Audia units real time via a network. After connecting to an Audia device, users, technicians, and designers (as determined by passwords) can access their appropriate level of control of the system. Depending on the authorized access level of control Levels, Meters, Presets, and layers can be accessed and changed.

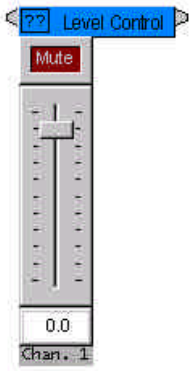
Level Controls, Meters, and, Preset control dialog screens can be minimized and arranged to provide a simple, and intuitive, user interface for the system. Connecting Audia's Ethernet port, through a switch, to network jacks in various locations can provide control to the system from different locations via a Laptop or Desktop computer.



*Note – Audia's control network can be easily shared on an existing network without compromising network bandwidth, but CobraNet data should be routed on its own network to ensure CobraNet audio and existing network reliability.

Control Dialog - Levels

Once level controls are placed in a signal path audio levels can be increased, decreased or muted as needed.



Control Dialog - Presets

Once *Presets* are created, Preset buttons can be placed on screen and used to easily recall different scenarios.

Control Dialog - Meters

If meters are placed on screen, users can view real time Signal Present, RMS, and Peak indications.