biamp.

Tesira

Text Protocol

3.4

DECEMBER 2017

BIAMP SYSTEMS 9300 SW GEMINI DRIVE, BEAVERTON, OREGON 97008 U.S.A. 503.641.7287 BIAMP.COM

Tesira Text Protocol

TTP Overview

Tesira can be controlled via the control dialogs in the Tesira software, via third-party controllers or via a computer based terminal application. Supported connection methods include serial RS-232 or Ethernet. If using Ethernet a Telnet or Secure Shell Console (SSH) session can be initiated.

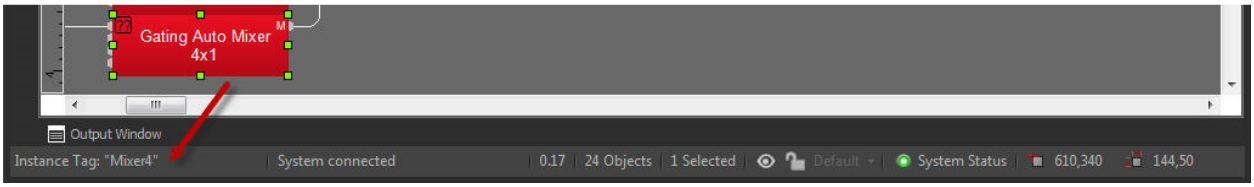
To facilitate external control of Tesira servers Biamp uses Tesira Text Protocol (TTP). This allows external control of a programmed Tesira system via ASCII characters.

TTP command strings allow the control of **Attributes** or **Services**. An **Attribute** defines the portion of the DSP Processing block to be controlled such as a fader level, crosspoint mute, and would depend on the specific DSP processing block Attribute Table. A **Service** defines an instruction and function specific to a DSP Processing block (such as the dialer block dial command), Tesira Hardware (Such as a Device Command referencing a Tesira Server) or to perform a system wide command such as recalling a Preset.

The command is case sensitive and uses upper and lower case characters. A line feed needs to be sent after each command.

TTP has built in error handling and the response will indicate the reason and location in the command where an error has been encountered. An error response will include **-ERR** at the beginning of the response. A successful response will include **+OK** at the beginning of the response. Review the Responses section for examples.

When Online with the Tesira Software any Attribute or Service changes made via TTP will update the values in real time.



When online - selecting a processing block will show the Instance ID in the Left hand Corner of the Status bar.

String Structure:

The commands outlined in this manual are formatted so that any command not in square brackets must be defined as part of the command. These include the **Instance Tag**, **Command** and **Attributes** of a command.

Any commands shown in square brackets (such as **[Index]** and **[Value]**)are dependant on the command being performed. They may not be required at all in which case no value is entered.

TTP in multiple device systems

Commands that act on the entire system (For Example- start audio) are forwarded to all the devices automatically, and commands that act on a block (such as set attribute) are automatically forwarded to the device hosting the block. In a redundant system, any server device in the system can handle TTP commands at any time. This is the same behavior as a system that does not have redundant pairs. If the block is in a redundant pair, the command is automatically forwarded to the active device in the pair.

TTP Syntax

The Services Code defines a instruction and function for a DSP block to perform. The Attribute Code defines the portion of the DSP block to be controlled such as a fader level. Each element of the command instruction is delimited by a single space. The commands are case sensitive and upper and lower case characters are used.

TTP string for Attribute Code:

To adjust an attribute of a DSP Processing Object is structured in the following order:

Instance_Tag Command Attribute [Index] [Value]

- **Instance Tag**: Is always required. Review the Instance Tag section for more details
- Command: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required should not be defined. Depending on the Attribute it can be made up of 1 or more indexes. Refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute	Index	Index	Value	Line Feed
MatMix_1	set	crosspointLevel	4	6	-4	<LF>

For Example: A get command will never use a [Value].
Mixer1 get crosspoint 1 1 +OK "value":false

For Example: A set command will always require a [Value]
Mixer1 set crosspoint 1 1 true +OK

TTP string for Service Codes:

The TTP string is structured in the following order:

Instance_Tag Service [Index] [Value]

- Instance Tag: Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Service being referenced.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be

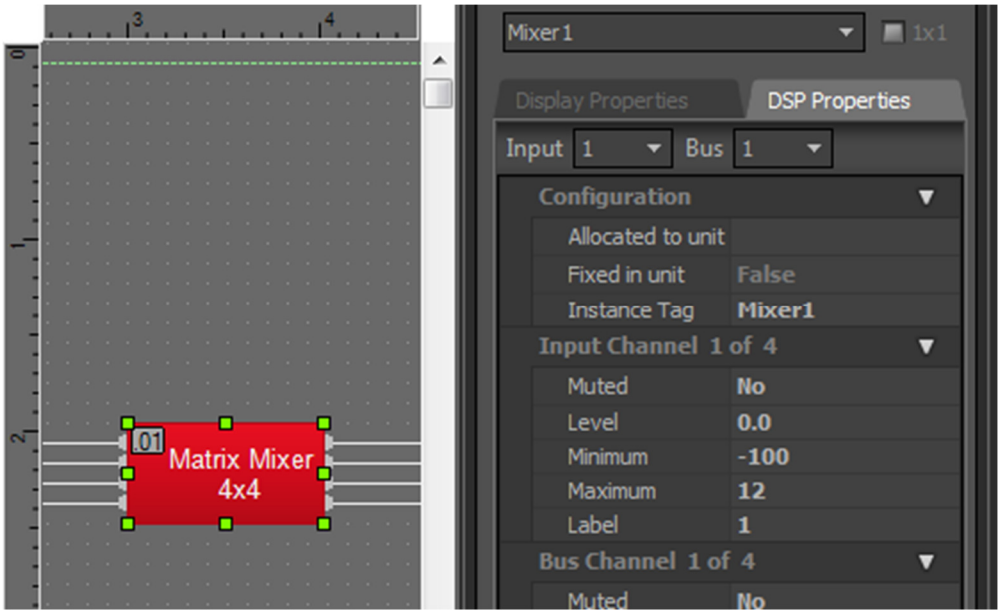
defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.

- **LF**: A Line feed or Carriage Return is used to define the end of the command.

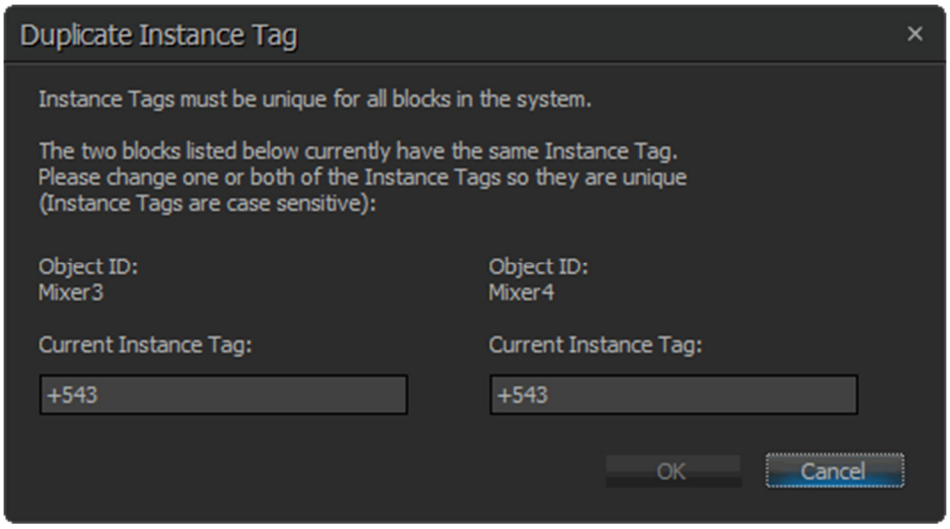
Instance Tag	Service Code	Value
DEVICE	recallPreset	1001

Instance Tag

The **Instance Tag** is case sensitive and is the unique name of a software object used in a Tesira project. The Instance Tag can be found when disconnected from the System in the **Processing object Properties>DSP Properties**. This defaults to the object code when compiled but can be customized by the user. The Tesira compiler will also check for duplicate Instance Tags. Instance tags can be defined within speech marks. If instance tags have no spaces they do not require speech marks. Instance tags can be numerical and contain spaces. Any Customized Instance tags that contain spaces must be defined within speech marks. The following Instance Tag characters are illegal / &



Duplicate instance tags are not allowed. If duplicates are created a dialog will appear allowing editing of the tags.



A SESSION command can be used to get a listing of available Instance Tags. Any devices that have an incomplete audio path will not be listed.

Example
SESSION get aliases +OK "list":["123" "AudioMeter1" "AudioMeter2" "AudioMeter3" "DEVICE" "Input1" "Mixer1" "Mute1" "Level1" "Output1"]

Example - When using a instance tag called **Level1**
Level1 get level 1 +OK "value":0.000000

Instance tags can contain spaces but must be enclosed in speech marks:

Example - When using a instance tag called **my level 2**
my level 2 get level 1 -ERR address not found: {"deviceId":0 "classCode":0 "instanceNum":0} "my level 2" get level 1 +OK "value":-10.000000

Instance tags can be numerical:

Example - When using a instance tag called **123**
123 get level 1 +OK "value":-10.000000

Commands

The Command field specifies what is to be done with the DSP processing block Attribute. Tesira Text Protocol supports different Attribute commands as listed below. These are case sensitive and the availability of the command would depend on the DSP object Attribute Code. The following table shows the Commands which only apply to Attribute Codes. An Attribute Code may not support all of them, but it will support at least one.

Command	Attribute Description
get	An attribute is to be read. The value will be returned in the response
set	An attribute is to be set to a specific value. String: **Instance_Tag Service [Index][Value]** Example: **Level1 set mute 1 true**
increment	An attribute is to be increased by the specified amount. Negative values will be decreased by the specified amount. String: **Instance_Tag Service [Index][Index]** Example: **Level1 increment level 1 3**
decrement	An attribute is to be decreased by the specified amount. Negative values will be increased by the specified amount. String: **Instance_Tag Service [Index][Index]**
toggle	An attribute is to be toggled. String: **Instance_Tag Service Attribute [Index]** Example: **Level1 toggle mute 1**
subscribe	An attribute is to be subscribed to.
unsubscribe	An attribute is to be unsubscribed from.

More details on subscriptions can be found in the Subscriptions section.

Attribute

The attribute Code defines the portion of the DSP Processing block to be controlled such as a fader level, crosspoint mute, etc. A full listing of the DSP block Attribute Codes are specified in the interface tables.

Service

The Services Code defines a instruction and function for a Hardware item to perform or a system wide command such as recalling a Preset. Currently the Device Instance Tag, TI Control Status, VoIP Control Status and Dialer Control Block support Service Code functions. Any Service Code commands do not use

Attribute Commands such as get, set, etc. Instead they use their own commands such as recallPreset or dial

Index

Attribute Codes use Index fields to refer to inputs, outputs, or cross attribute of a DSP Block. Due to the different types of DSP blocks, some attributes will not require and Index so no value should be used. Some DSP blocks require a single index such as a level control. Some DSP blocks require 2 indexes such as a matrix mixer. The first index would be the Input or Row and the second index would be the Output or Column. A full listing of the DSP block Attributes and Indexes are specified in the interface definition tables.

For a Crossover Index **band** is indexed by number from high to low, so in a four-way crossover high=1, mid high=2, low mid =3 and low=4. **filter** is indexed by number. 1 is the high cutoff frequency for each band while 2 is the low.

The Index values can be encased in double quotes. the following formats are both supported:

Example
Mixer1 set crosspoint 1 1 true +OK Mixer1 set crosspoint "1""1" true +OK

When a subscription command is configured a unique custom name can be used in the second Index of the command line. This is used as the identifier for the subscribed item.

Some Service Codes use index fields to define the hardware channel that is being controlled. For example a Dialer Block TI Control Status and, VoIP Control Status will require the **line** and **Call appearance** indexes to be specified.

Value

Value determines what a DSP block is being set to, incremented by, or decremented by. The interface definition tables define which type of value the string will need in order to execute the TTP string.

A TTP value will depend on the attribute being controlled. It can be:

- A number
- A string (in double quotes)
- A Boolean (true or false)
- null

Required action	Value example	Description
Turn On	true	Refers to the 'on' state of a processing object component with two states such as a crosspoint, mute or similar.
Turn Off	false	Refers to the 'off' state of a processing object component with two states such as a crosspoint, mute or similar.
Adjust level (set, increment, decrement)	1.0 -1.0 -15 etc.	A numerical decimal value used to represent the new state. Refer to the interface definition tables for the value range supported by the different component objects. For a 'set' command this will move the value to the specified level.

		For an increment it will adjust the value from the current value by the specified amount.
State	BUTTERWORTH	A text string can be used to represent a value such as a filter type
preset	1001	An Integer that is the required state.

Special Addresses

DEVICE - the local unit that you are currently connected to. See the Device Attribute table for a full listing of commands.

Instance Tag	Command	Attribute Code	Index	Line_Feed
DEVICE	get	ipStatus	interface	LF

SESSION - The current RS-232, Telnet or SSH text session. See the Session Attribute table for a full listing of commands

Instance Tag	Command	Attribute Code	Value	Line_Feed
SESSION	set	Verbose	false	LF

TTP Security

Security of Telnet, SSH and Serial Connections

Establishing an SSH connection to the TTP server ***requires*** login credentials by definition.

In a protected Tesira system, the same password access levels apply to all connections to the **Tesira Text Protocol (TTP) Server**. Please review the System Security settings that can be configured on the Tesira Servers.

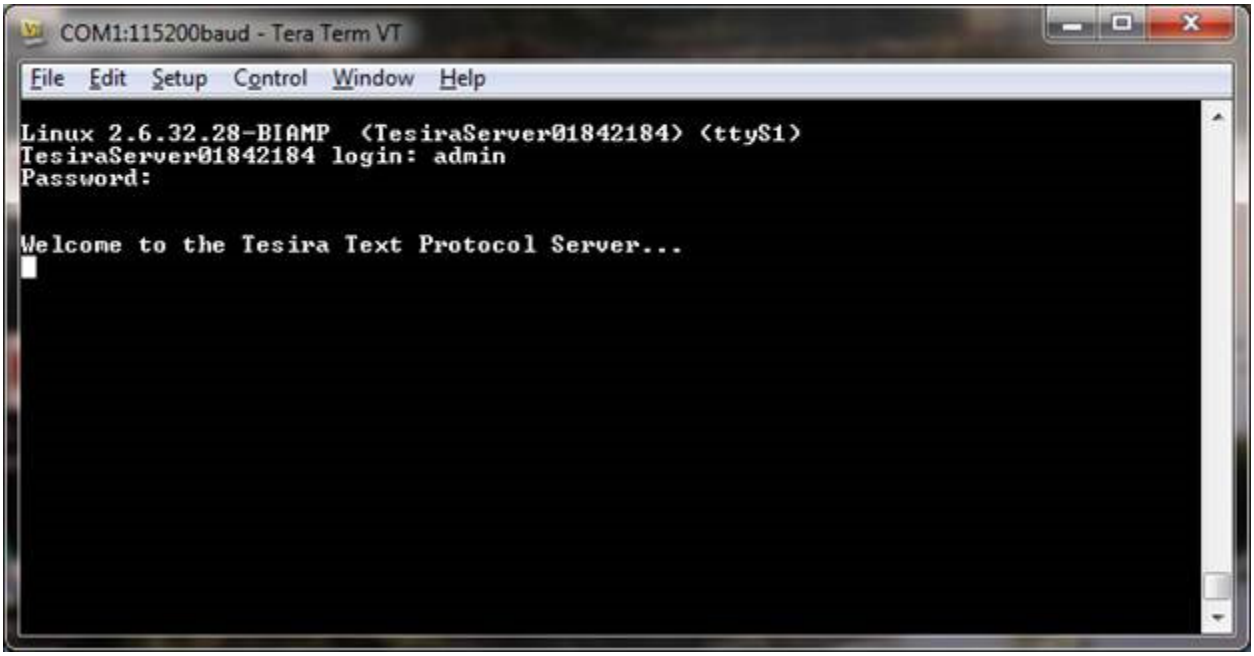
Opening a Telnet or SSH session to a Tesira Server results in a login prompt. Valid credentials must be provided to access the system in any way. One must be logged in as controller or higher level to make any changes to the system, while an observer can only query the system for levels and other current parameters.

In an unprotected system, the username and password are ‘default’ and ‘default’ respectively. In a protected system, the credentials configured in the system must be provided.

RS-232 Serial connections to the TTP servers also require authentication in protected systems. Making the serial connection and sending a line feed will reveal the login prompt.

If a system has security enabled the RS-232 will not require authentication until the connection is fully terminated using a 'exit' command. There will then be a requirement to authenticate at the next log on.

Once logged in to the TTP server via RS-232, this user has access until a 'exit' command is sent, even if the serial connection is removed and restored.


COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
Linux 2.6.32.28-BIAMP (TesiraServer01842184) (ttyS1)
TesiraServer01842184 login: admin
Password:
Welcome to the Tesira Text Protocol Server...

TTP Responses

Output Styles

A Verbose or non-verbose response can be configured as part of the Session Command type.

Verbose

+OK “time”:”12:00” “number”:”503-367-3568” “line”:”2”

Non-Verbose

+OK ”12:00” ”503-367-3568” ”2”

Example
SESSION set verbose true Mute1 get numChannels +OK "value":2 SESSION set verbose false +OK Mute1 get numChannels +OK 2

Tesira Text Protocol will provide user feedback if a command is incorrect. The response will vary depending on the command. The Tesira TTP error responses for the most common types of external programming errors include:

- can’t forward a request to a device that’s not on the network
- if an invalid address is used
- if an invalid attribute or service for a block type (it might be valid for a different object)
- right address, right attribute or service, but the request doesn’t make sense given the state of the target object
- case-and-spelling errors of various kinds

Please refer to the table below for some examples and details of some of the expected error responses.

TTP Command String	Message	Resolution
	+OK	The command was understood and completed successfully
Session get aliases	-ERR address not found: {"deviceId":0 "classCode":0 "instanceNum":0}	The requested address is not valid due to incorrect formatting. The Address field is case sensitive. **Session** commands must be in capitals. Reformat the command as **SESSION get aliases**.
SESSION Get aliases	-ERR Parse error at 8: verb was not one of the commands supported by Services	There is a problem 8 characters into the command. The **get** command is incorrectly formatted - it has a capital 'G'. Reformat the command as **SESSION get aliases**
SESSION get Aliases	-ERR 'Aliases' is not supported by TextSession::Attributes	**Aliases** is not correctly formatted. It has a capital 'A'. Reformat the command as **SESSION get aliases**
Mixer1 set inputMute 1	-ERR Parse error at 22: not enough parameters supplied	The command is missing the value. Reformat the command as **Mixer1 set inputMute 1 true** .
Mixer1 get inputLevel 1	+OK "value":0.000000	The command was delivered and the value of the Input level is 0.0dB
Input1 get gain channel1	-ERR Parse error at 16: could not parse value	Channel1 command is invalid. The Input block channel is numerical. Reformat the command as **Input1 get gain 1**
AudioMeter2 subscribe level 3 mymeter 1000	! "publishToken":"mymeter" "value":-100.000000 +OK	A subscribe of the meter refreshing every 1 second
MyLevel1 get level 10	-ERR INVALID_PARAMETER Index out of range:channelIndex min:1 max:8 received:10	Channel 10 not available. Index indicates channels 1 to 8 available.
	-ERR WRONG_STATE	VoIP card has received a command it cannot action (For example if the card is not connected to the Call Manager and is given a request to make a call)
	-CANNOT_DELIVER	Typically seen on a system with multiple Server devices when connected to one Server and addressing a DSP object in another

		server. Would indicate a communication issue between servers.
	-GENERAL_FAILURE	A 'catch all' error code. Can occur when referencing a Instance Tag that is not in the Tesira file.

TTP Subscriptions

Subscribing

Subscriptions enable the updating of metering and level values to be sent to a external control system without the control system requesting information.

Elements of a processing object can be subscribed to such as channel levels and meters. The Attribute tables will indicate which functions support subscription.

If subscriptions are used the Tesira server may be sending back replies that were not individually requested from the control system (they were subscribed to). All subscribed objects will be preceded by a **! "publishToken"** statement would indicate to the control system that the returned packet is from a subscription not a response to a command that was just sent.

Subscriptions are lost when the Tesira server is rebooted. Subscriptions can be revalidated by subscribing to the same block at regular intervals. If this is done ensure that the custom label used in **Index** is used in the re-subscription. If this label is not included it is possible to inadvertently open multiple subscriptions to the same call state.

Instance Tag Command Attribute [Index] [Index] [Value] LF

- Instance Tag: Is always required. Review the Instance Tag section for more details
- Command: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required should not be defined. Depending on the Attribute it can be made up of 1 or more indexes. Refer to the Index section for more details.
- **[Index]**: Is used to assign a custom label to the subscription. Is shown in [Brackets] as is not required but is recommended, especially if there is more than one subscription in the system. The label would indicate to the control system which object is providing the state change. Instance Tags are not included in subscription responses.
- **[Value]**: Is shown in [Brackets] as it is not required. Value can be used to throttle the rate of response to the control system. The value specified is in milliseconds. A subscription update is provided immediately after a state change, with updates spaced by the specified value. Updates are only sent when a change occurs. Consideration should be given to buffer sizes to make sure the subscribed responses can be handled correctly by any external control systems.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

To Subscribe to a level with a 500ms refresh

Instance Tag	Command	Attribute Code	Index	Index	Value
MyLevel1	subscribe	level	1	MyLevelName	500

Verbose Subscription Responses

When the subscription command is first sent the first reply will be:

! "publishToken":"[CustomName]" "value":[Value] +OK

Subsequent subscription replies will be formatted

! "publishToken":"[CustomName]" "value":[Value]

- The **[CustomName]** is used as an identifier. The identifier returned is specified in the Index field of the original subscribe command. This name can then be used in a parsing routine for the subscribed item. If no identifier is specified then empty double speech-marks ("") are shown in the response as a delimiter.
- The **[Value]** is the current state of the control being subscribed to. This will be formatted as an integer or boolean depending on the subscription attribute.

Verbose Example
MyLevel1 subscribe level 1 MyLevelName 500 ! "publishToken":"MyLevelName" "value":-100.000000 +OK ! "publishToken":"MyLevelName" "value":-98.099998 ! "publishToken":"MyLevelName" "value":-77.800003 ! "publishToken":"MyLevelName" "value":-35.299999

Verbose Example
MyLevel1 subscribe level 1 ! "publishToken":"" "value":-100.000000 +OK ! "publishToken":"" "value":-98.099998 ! "publishToken":"" "value":-77.800003 ! "publishToken":"" "value":-35.299999

Non-Verbose Subscription Responses

If a non-verbose response is required this must be specified before as a SESSION command and must be configured before the subscription.

When the subscription command is first sent the first reply will be:

! "[CustomName]" [Value] +OK

Subsequent subscription replies will be formatted

! "[CustomName]" [Value]

- The **[CustomName]** is used as an identifier. The identifier returned is specified in the Index field of the original subscribe command. This name can then be used in a parsing routine for the subscribed item. If no identifier is specified then empty double speech-marks ("") are shown in the response as a delimiter.

- The **[Value]** is the current state of the control being subscribed to. This will be formatted as an integer or boolean depending on the subscription attribute.

Verbose Example
Welcome to the Tesira Text Protocol Server... SESSION set verbose false +OK MyLevel1 subscribe level 1 myLevelName 500 ! "myLevelName" -40.244328 +OK ! "myLevelName" -38.992748 ! "myLevelName" -41.044147 ! "myLevelName" -40.063908 ! "myLevelName" -38.674465

Unsubscribing

Once a value has been subscribed to, the unsubscribe command is used to cancel the request. If an Index and value have been specified in the original subscribe request they must be used in the unsubscribe request.

Instance_Tag Command Attribute [Index] [Value] LF

- Instance Tag: Is always required. Is the same Instance Tag used to originally subscribe.
- Command: Is always required. Is the same Command used to originally subscribe.
- **Attribute:** Is always required. Is the same Attribute used to originally subscribe.
- **[Index]**:Is required if specified as part of the Attribute. Is the same Attribute index or indexes used to originally subscribe.
- **[Index]**: Is required if specified as part of the original subscription. Must match the custom name given in the original subscription.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

To unsubscribe to a level.

Instance Tag	Command	Attribute Code	Index	Index
MyLevel1	unsubscribe	level	1	MyLevelName

Example
MyLevel1 subscribe level 1 MyLevelName 500 ! "publishToken":"MyLevelName" "value":-100.000000 +OK ! "publishToken":"MyLevelName" "value":-98.099998 ! "publishToken":"MyLevelName" "value":-77.800003

```
! "publishToken":"MyLevelName" "value":-35.299999

MyLevel1 unsubscribe level 1 MyLevelName
+OK
```

TTP Troubleshooting

Configuring a PC to connect to Tesira

Connecting a PC to a Tesira System to troubleshoot may be required. Using a PC allows testing of the strings and responses in real time to prove valid commands are being used. A terminal Emulator program is recommended to connect to the system. Suggested programs include TerraTerm or PuTTY.

Putty is used throughout this document in any examples given this allows connections using RS-232, Telnet or SSH.

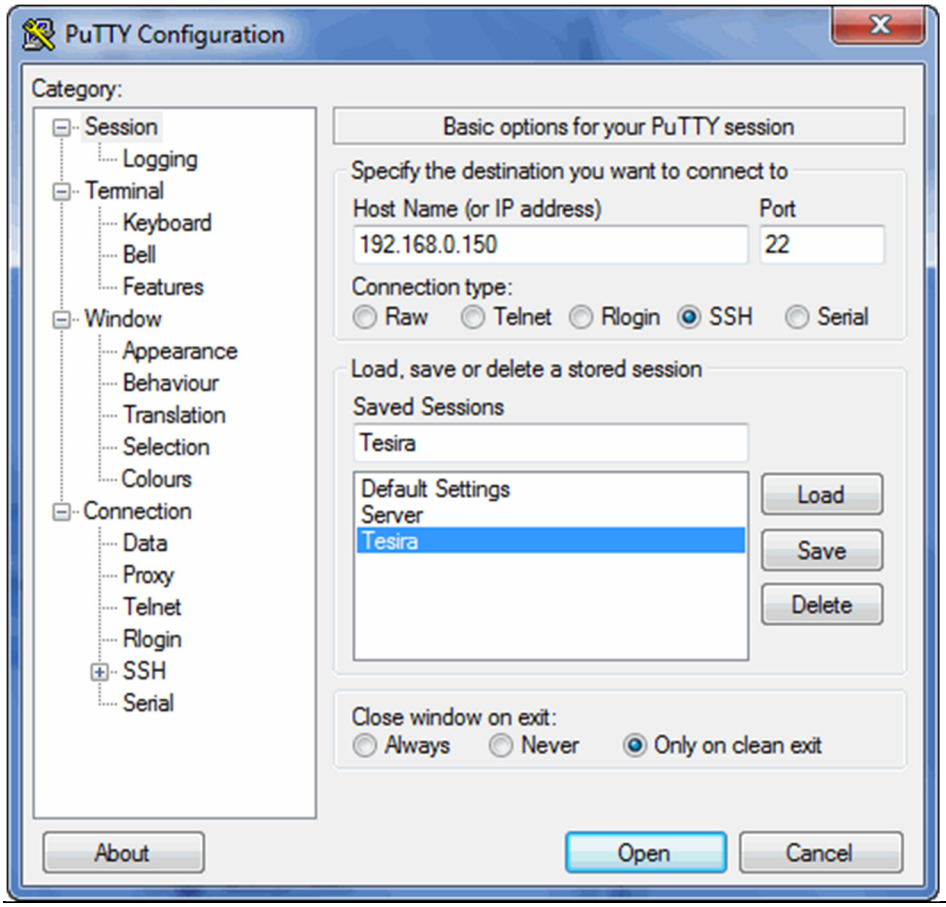
If using Telnet or SSH, ensure these services are enabled in Device Maintenance > **Network Settings** or via the Device TTP command.

Opening a Telnet or SSH session to a Tesira Server results in a login prompt. Valid credentials must be provided to access the system in any way. One must be logged in as controller or higher level to make any changes to the system, while an observer can only query the system for levels and other current parameters.

The SSH Login requires case sensitive User and Password authentication. In an unprotected system, the Username and Password are 'default' and 'default' respectively. In a protected system, the credentials configured in the system must be provided.

PuTTY is a free implementation of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator. This software can be downloaded from the following link:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Instructions on its use can be found here:http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html



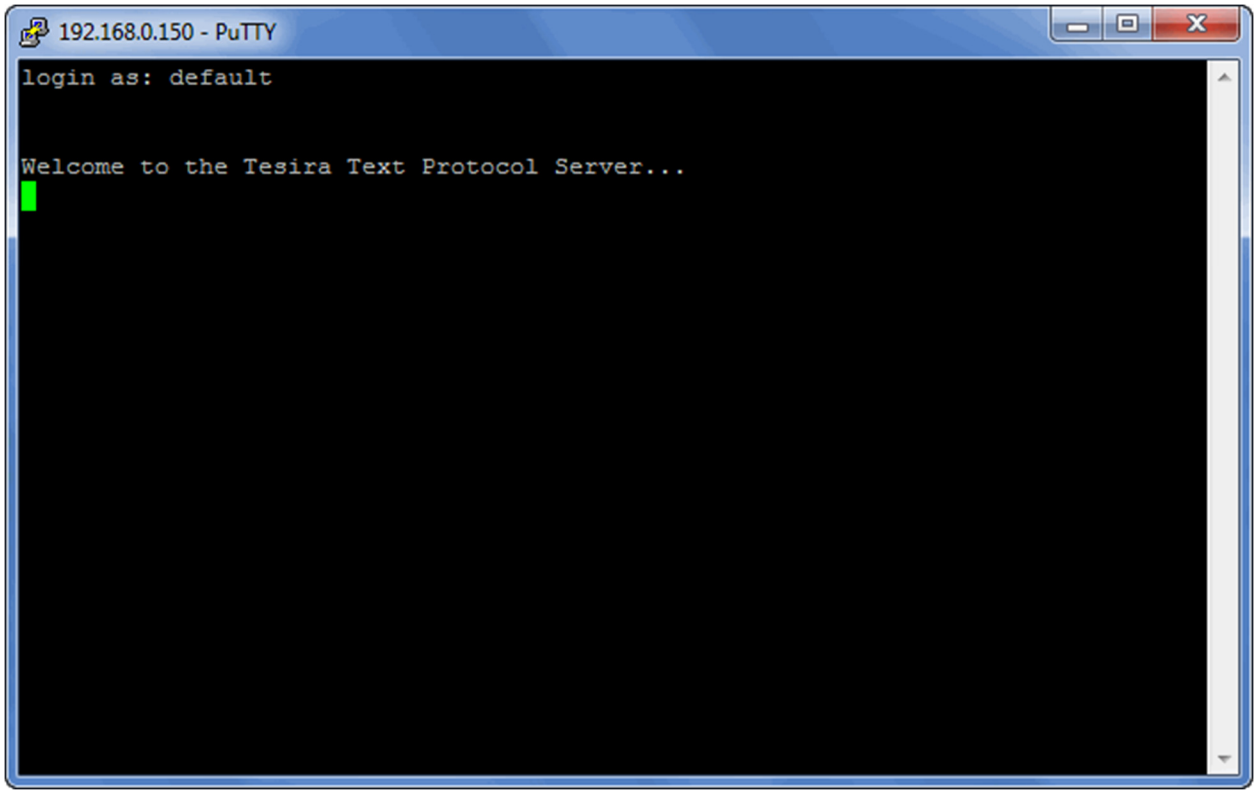
Configuring a PC to connect to Tesira using Telnet.

Telnet is enabled by default in Windows XP. If using windows Vista or Windows 7 it is not enabled by default in an attempt to make Windows more secure. If you require a secure method to connect to a Tesira Server, please refer to connecting via SSH.

The use of a terminal emulation program such as PuTTY is recommended in order to establish a command session to a Tesira Server.

If the convenience of using the Windows command prompt to initiate a Telnet session is required, you can use Windows Programs and Features to enable the Telnet Client.

- To enable Telnet navigate to: **Start > Control Panel > Programs & Features > Turn Windows Features on and off**
- Find the entry for **Telnet Client**
- Select the tick box.
- Select **OK.**
- To Initiate a TELNET session with a Tesira Server:
- Select **Start>programs>accessories> Command Prompt**
- At the command prompt type **telnet xxx.xxx.xxx.xxx** (xxx.xxx.xxx.xxx is the IP address of the Tesira Server.)



Tesira Text Protocol will provide user feedback if a command is incorrect. The response will vary depending on the command, please review the Responses section for more details.

RS-232

A Tesira SERVER or SERVER IO has two RS-232 ports. A TesiraFORTÉ, IDH-1 and OH-1 have one RS-232 port. Each Port can be configured to:

- send Command Strings for controlling other devices via the Command String Block
- accept full duplex TTP commands for Third Party control
- Both of the above
- None of the above

Please also review the Troubleshooting TTP which gives information on configuring a PC to connect to a Tesira system for testing purposes.

The baud rate can be adjusted in **Device Maintenance > Serial Port Settings** dialog. Baud rate of the RS-232 port can be set to 300,1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200

Device	Port Name	Default Setting
SERVER and SERVER IO	Serial 1	9600, Command String
SERVER and SERVER IO	Serial 2	115200, TTP
TesiraFORTÉ	RS-232	115200, Both
IDH-1	RS-232	115200, Both
OH-1	RS-232	115200, Both

If multiple servers are connected together in a system then only one RS-232 port needs to be connected to a third-party control system; TTP commands are proxied via the Ethernet port to other devices in the system. In an unsecured Tesira system RS-232 connections do not require authentication.

If a system has security enabled the RS-232 will not require authentication until the connection is fully terminated using a '**exit**' command. There will then be a requirement to authenticate at the next log on. Once logged in to a secured server via RS-232, this user has access until a 'exit' command is sent, even if the serial connection is removed and restored. Please review the TTP security setting for more details.

When controlling multiple Tesira units that are not part of the same TMF file, each Tesira server unit will need to be addressed via its own RS-232 port. Tesira units cannot be linked together via RS-232.

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 Tesira Server.

Serial Connection			
pin #1	not used	pin #6	not used
pin #2	Transmit data (TxD) Output	pin #7	not used
pin #3	Receive data (RxD) Output	pin #8	not used

pin #4	not used		pin #9	not used
pin #5	ground			

(* A PC can send/receive TTP Strings using a terminal emulator program such as HyperTerminal or PuTTY.)

Telnet

Please also review the Troubleshooting TTP which gives information on configuring a PC to connect to a Tesira system for testing purposes.

Telnet is configured by specifying the IP address of the Tesira Server and connecting via port 23. The ability for Tesira Server, Server IO, Tesira Amplifiers or TesiraFORTÉ devices to use Telnet can be enabled or disabled via a DEVICE TTP command or in the Device Maintenance Settings > **Network Settings** dialog.

When controlling multiple Tesira units that are not part of the same TMF file, each Tesira Server unit will need to be addressed via its own Telnet Session. Commands sent via Telnet are not encrypted.

VoIP Telnet

The option to disable Telnet connections (port 23) on the VoIP enabled FORTÉ and Server devices is available from within the VoIP Property Sheet. This is an engineering diagnostic interface only however for installations with security concerns about this port being open, it can be disabled. Also refer to the VoIP Property Sheet to disable HTTP access to the engineering diagnostic interface.

Negotiation required to establish a Telnet control session.

Session Options

Tesira implements a Telnet server on port 23. When the request from the control system to open a session is received, the Tesira Telnet server attempts to negotiate the session's options, following specifications described in the Telnet standard document RFC 854 as well as document RFC 855, Telnet Option Specifications.

A standard Telnet client would be able to negotiate the session options without problem, but several third party controllers do not implement a Telnet client by default. Instead, they implement control over TCP/IP using what's commonly known as a 'RAW' connection. If the Control System does not respond to the Telnet session options negotiations, the session will not go ahead. As such, the control system will have to be programmed to negotiate the Telnet options with Tesira's Telnet server. Many of the available options can be useful during a control session and indeed a programmer may choose to enable some of them, but if the desire is to continue using a 'RAW' connection, the simplest way to initiate a control session is for the control system to respond with a rejection to any option negotiation request from the server.

Negotiation

The best way to understand the Telnet options negotiation procedure is by looking at the data in Hex format. Notation will be "0xFF" for Hex character FF.

The Telnet commands we are concerned with are always three bytes long. The first is the **Interpret As Command** (IAC) character, and it is always 0xFF. The second character is the **Command** and the last character is the **Option** being negotiated.

Commands can be:

- WILL, or 0xFB
- DO, or 0xFD
- DON'T, or 0xFE
- WON'T, or 0xFC

Negotiated options can be (but not limited to*):

- Binary Transmission, 0x00
- Echo, 0x01
- Suppress Go Ahead, 0x03
- Status, 0x05
- Terminal Type, 0x18

*** There are many different Telnet options in existence; a master list is maintained by IANA http://www.iana.org/assignments/telnet-options**

The control system needs to react to any incoming string that begins with 0xFF, and decide whether the option is desired or not. If the intent is to control Tesira using a 'raw' connection, all that's required is to always reject the option negotiation. If Tesira sends a "WILL" Command, the control system shall respond with "DON'T", and if Tesira sends a "DO", the response should be "WON'T". The Option byte needs to be returned as received.

In essence, the mechanism is as follows:

When the server sends:	0xFF	WILL	<byte X>
The control system responds with:	0xFF	DON'T	<byte X>
When the server sends:	0xFF	DO	<byte X>
The control system responds with:	0xFF	WON'T	<byte X>

Examples

Source	IAC	Command	Option	Notes
Tesira Sever	0xFF	0xFD	0x01	Do Echo
control system / Client	0xFF	0xFC	0x01	Won't Echo

Source	IAC	Command	Option	Notes
Tesira Sever	0xFF	0xFB	0x03	Will Suppress Go Ahead
control system / Client	0xFF	0xFE	0x03	Don't Suppress Go Ahead

Once all options are negotiated, the Tesira server will send the message "Welcome to the Tesira Text Protocol Server", preceded and followed by 0x0D and 0x0A. The control system is now free to send TTP commands.

Other considerations

Please note that the Tesira server will usually end any string with either 0x0D (CR character) followed by 0x0A (LF character), but as per Telnet RCF it may also use 0x0D (CR character) followed by 0x00 (NUL character). As such, the third party control system must be able to read one more character after it sees a 0x0D, which will always be either 0x0A or 0x00, and handle them appropriately.

In addition, and while in practice most of the negotiations will always take place at the beginning of a session, Telnet allows for them to happen at any point during the session.

Example negotiation

Below is an example session options negotiation at the beginning of a Telnet session between Tesira and a TCP Client which was programmed to reject all options offered by the server. Please note this is for illustrations purposes only and the order and quantity of options negotiated may vary depending on firmware release. Strings have been organized below for clarity; however multiple Telnet strings may arrive from the Server in one Ethernet frame. Responses can be sent one at the time, or multiple responses in a single frame.

Source	IAC	Command	Option	Notes
Tesira Server	0xFF	0xFD	0x18	Do Terminal Type
Client	0xFF	0xFC	0x18	Won't Terminal Type
Tesira Server	0xFF	0xFD	0x20	Do Terminal Speed
Client	0xFF	0xFC	0x20	Won't Terminal Speed
Tesira Server	0xFF	0xFD	0x23	Display Location
Client	0xFF	0xFC	0x23	Won’t X Display Location
Tesira Server	0xFF	0xFD	0x27	Do New Environment Option
Client	0xFF	0xFC	0x27	Won't New Environment Option
Tesira Server	0xFF	0xFD	0x24	Do Environment Option
Client	0xFF	0xFC	0x24	Won't Environment Option
Tesira Server	0xFF	0xFB	0x03	Will Suppress Go Ahead
Client	0xFF	0xFE	0x03	Don’t Suppress Go Ahead
Tesira Server	0xFF	0xFD	0x01	Do Echo
Client	0xFF	0xFC	0x01	Won’t Echo
Tesira Server	0xFF	0xFD	0x22	Do Linemode
Client	0xFF	0xFC	0x22	Won’t Linemode
Tesira Server	0xFF	0xFD	0x1F	Do Negotiate About Window Size
Client	0xFF	0xFC	0x1F	Won't Negotiate About Window Size
Tesira Server	0xFF	0xFB	0x05	Will Status
Client	0xFF	0xFE	0x05	Don't Status
Tesira Server	0xFF	0xFD	0x21	Do Remote Flow Control
Client	0xFF	0xFC	0x21	Won't Remote Flow Control
Tesira Server	0xFF	0xFB	0x01	Will Echo
Client	0xFF	0xFE	0x01	Don’t Echo
Tesira Server	0xFF	0xFD	0x06	Do Timing Mark
Client	0xFF	0xFC	0x06	Won't Timing Mark
Tesira Server	0xFF	0xFD	0x00	Do Binary Transmission
Client	0xFF	0xFC	0x00	Won't Binary Transmission
Tesira Server	0xFF	0xFB	0x03	Will Suppress Go Ahead
Client	0xFF	0xFE	0x03	Don’t Suppress Go Ahead
Tesira Server	0xFF	0xFB	0x01	Will Echo
Client	0xFF	0xFE	0x01	Don’t Echo
Tesira Server	0xFF	0xFD	0x0A	
Tesira Server	0x0D 0x0A Welcome to the Tesira Text Protocol Server 0x0D 0x0A			

SSH

Please also review the Troubleshooting TTP which gives information on configuring a PC to connect to a Tesira system for testing purposes.

SSH is configured by specifying the IP address of the Tesira Server and connecting via port 22. The ability for Tesira Server, Server IO, Tesira Amplifiers or TesiraFORTÉ devices to use SSH can be enabled or disabled via a DEVICE TTP command or in the Device Maintenance Settings > **Network Settings** dialog.

When controlling multiple Tesira units that are not part of the same TMF file, each Tesira server unit will need to be addressed via its own SSH Session

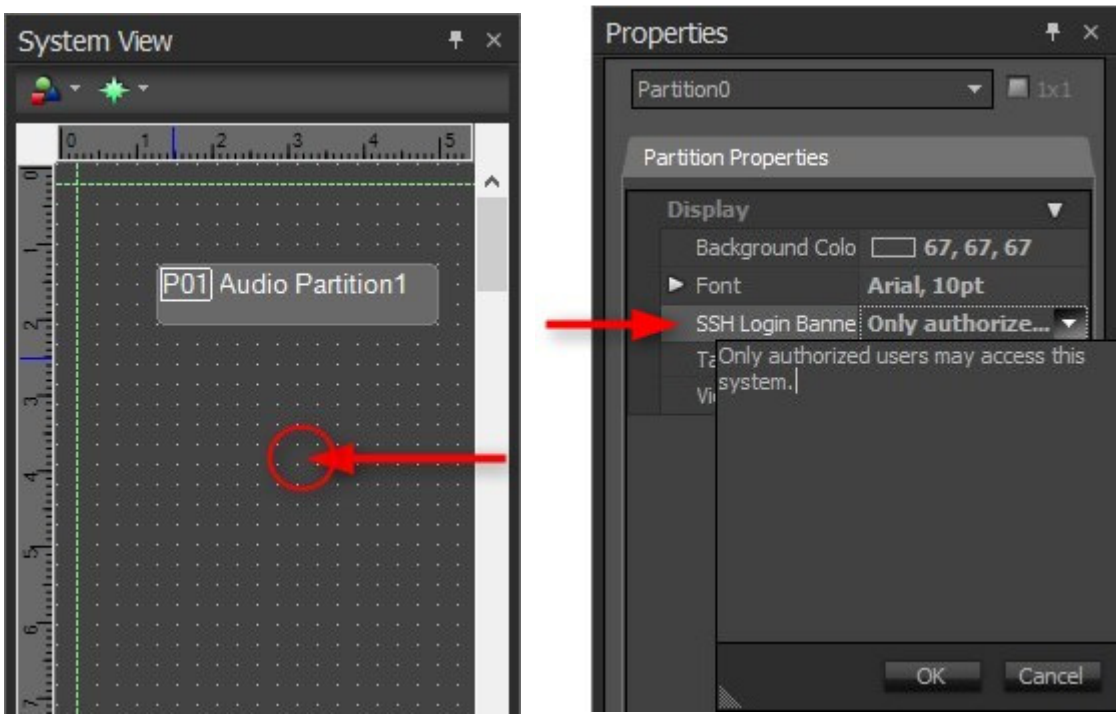
Commands sent via SSH are encrypted.

Opening a SSH session to a Tesira Server results in a login prompt. Valid credentials must be provided to access the system in any way. One must be logged in as controller or higher level to make any changes to the system, while an observer can only query the system for levels and other current parameters.

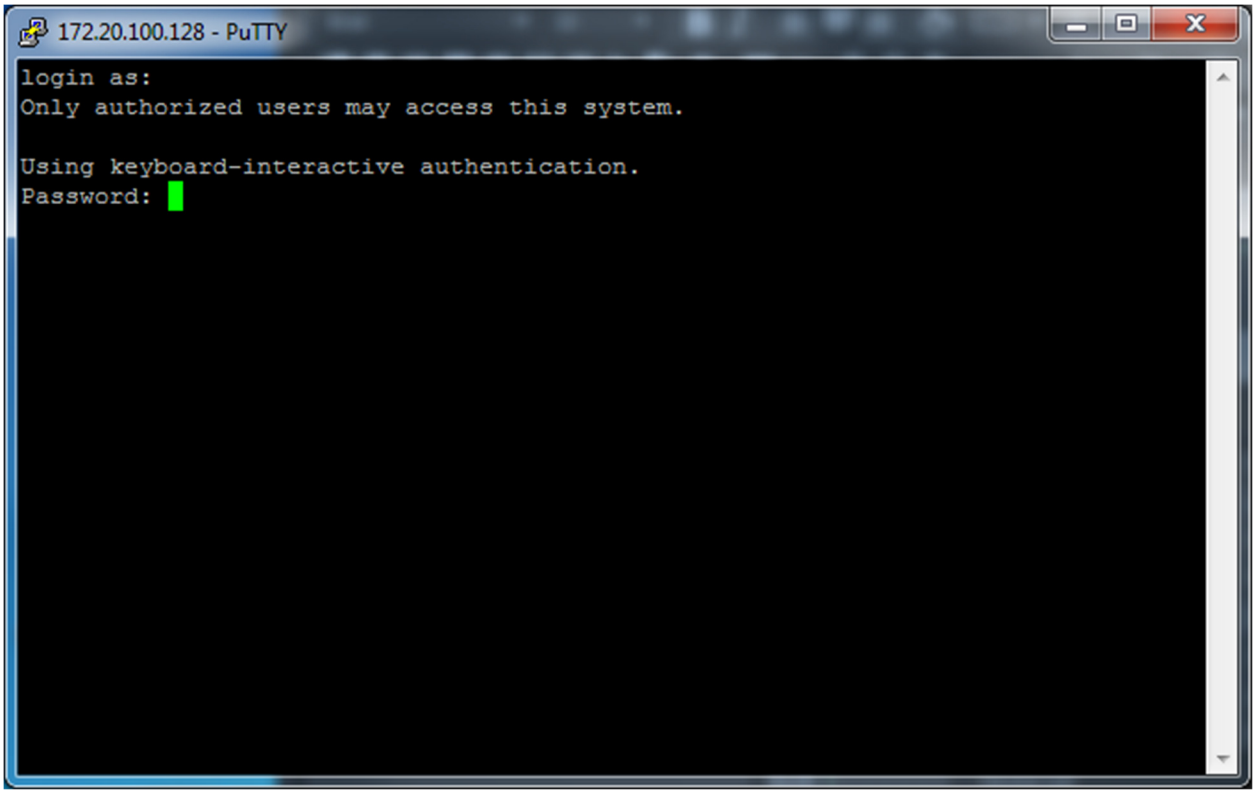
The SSH Login requires case sensitive User and Password authentication. In an unprotected system, the Username is **default** and Password is not required. In a protected system, the credentials configured in the system must be provided.

SSH Login Banner

A customized SSH logon banner can be defined by selecting the background of the System View. Then, in the Properties Sheet, 'Partition 0' will be shown with an option to define the SSH Login Banner line item. When selected, a text box appears allowing message customization.



Example:


172.20.100.128 - PuTTY
login as:
Only authorized users may access this system.
Using keyboard-interactive authentication.
Password:

Attribute tables

Interface tables

Service Addresses

Device Session

IO Blocks:

Audio Input Block Audio Output Block AVB.1 Input Block AVB.1 Output Block CobraNet Input Block CobraNet Output Block Dante Input Block Dante Output Block Audio-Technica Mic Block SHURE Mic Block Attero Tech Input Block Attero Tech Output Block USB Input Block USB Output Block AEC Input Block AEC Processing Block ANC Input Block ANC Processing Block TI Receive Block TI Transmit Block TI Control Status Block TC Call State Commands VoIP Receive Block VoIP Transmit Block VoIP Control Status Block VoIP Transfer Commands VoIP Call State Commands Dtmf Decode Block Labgruppen Amp Block Tesira Amplifier Block AV Input Block AV Output Block

Mixer Blocks

Gating Auto Mixer Block Gain Sharing Auto Mixer Block Standard Mixer Block Matrix Mixer Block Auto Mixer Combiner Block Room Combiner Block

Equalizer Blocks

Parametric Equalizer Block Graphic Equalizer Block Feedback Suppressor Block

Filter Blocks

Pass Filter Block Shelf Filter Block All Pass Filter Block Uber Filter Block

Crossover

Crossover Block

Dynamic Blocks

Leveler Block Compressor Block Peak Limiter Block Ducker Block Noise Gate Block AGC Block

Router Blocks

Router Block Source Selector Block AV Router Block

Delay Blocks

Audio Delay Block

Control Blocks

Level Control Block Invert Control Block Mute Control Block Preset Control Block Command String Block Dialer Block

Meter Blocks

Signal Present Meter Block Peak or RMS Meter Block

Generator Blocks

Tone Generator Block Noise Generator Block

Logic Blocks

Logic State Block Flip Flop Block Logic Delay Block Logic Meter Block Logic Input Block Logic Output Block Control Voltage Block

Service Addresses

Device

The DEVICE Instance Tag is case sensitive and must be in capital letters. It is used to send Device Services instructions or Device Attributes and Commands.

Device Services

The Following table summarizes DEVICE Service Codes. Due to the nature of the service being requested they do not require specific commands (get, set, etc)

Some service commands are specific to the connected device, such as 'reboot'. Other Service commands are design file specific, such as saving or recalling a Preset.

The TTP string is structured in the following order:

Instance_Tag Service [Value]

- Instance Tag : Is always required and will always be DEVICE.
- **Service** : Is always required please review the Device Services table below for the supported commands.
- **Value**: May be required depending on the **Service** Command being used.

Description	Service	Value
Manual Failover	manualFailover	unitNumber
Reboot Device you are connected to via SSH or Telnet	reboot	
Reset Device you are connected to via SSH or Telnet	deleteConfigData	
Reboot Expander class device	rebootERD	["*hostname*", "*2nd hostname*", "etc"]
Recall a Preset	recallPreset	Preset ID (Integer)
Recall a Preset and provide device for failures	recallPresetShowFailures	Preset ID (Integer)
Recall a preset by preset name	recallPresetByName	Preset name (a string)
Save a Preset	savePreset	Preset ID (Integer)
Save a preset by preset name	savePresetByName	Preset name (a string)
Start System Audio	startAudio	
Stop System Audio	stopAudio	
Start Partition Audio	startPartitionAudio	Partition ID (integer)
Stop Partition Audio	stopPartitionAudio	Partition ID (integer)

Examples:

To reboot the device you are connected to:

Instance Tag	Service

DEVICE	reboot

Result: DEVICE reboot

To start Audio on a device:

Instance Tag	Service
DEVICE	startAudio

Result: DEVICE startAudio

To reboot multiple expander devices.

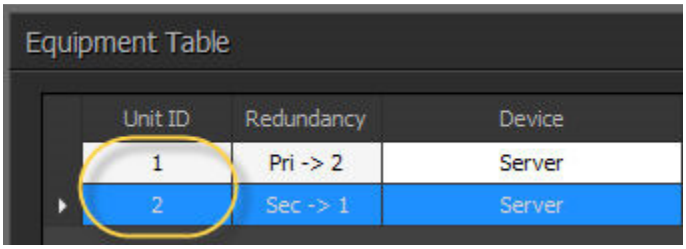
Instance Tag	Service
DEVICE	rebootERD ["EX-OUT-0000", "EX-IN-0001", EX-AEC-0001]

Result if all expanders are discoverable and accept the reboot command: +OK "failedDevices":[ ]

Result if all but the EX-IN expander are discoverable and accept the reboot command: +OK "failedDevices":["EX-IN-0001"]

Device manual Failover

A redundant server pair can be manually forced to failover. The unit number can be either unit ID (as specified in the equipment table) in the redundant pair that you want to force to fail over.



Instance Tag	Service	index
DEVICE	manualFailover	unitNumber

Result: DEVICE manualFailover 1

Device attributes and Commands

Additionally there are a number of DEVICE Instance Tag command Attributes. These would reference the device that has the current active Serial, SSH or TELNET session.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Active Faults	activeFaultList	get		
AVB Peer Delay Threshold	avbPDelayThreshold	get set increment decrement		0 - 2,147,483,647
Discovered Servers	discoveredServers	get		
DNS Config	dnsConfig	get/set		
DNS Status	dnsStatus	get		
Remote Device AVB Peer Delay Threshold	ERDavbPDelayThreshold	get set increment decrement	hostname	0 - 2,147,483,647
Host Name	hostname	get/set		*Host Name modification will not be available unless the device is in an un-configured state. Attempting to modify the Host Name via this command will result in an error message if the system is currently configured. A reset of the device is required to make changes to the Host Name first

				(**DEVICE deleteConfigData** command or a **Reset Device** via Device Maintenance). In the scenario where a system is configured and is Reset to change the Host Name, the Equipment Table will need to be re-opened and updated to reflect the new details, and the configuration re-sent to the system.
Resolver Hosts Table	hostTable	get/set		
Network Interface Config	ipConfig	get/set	interface name	control
Network Interface Status	ipStatus	get	interface name	control
Known Redundant Device States	knownRedundantDeviceStates	get subscribe unsubscribe		
mDNS Enabled	mDNSEnabled	get/set toggle		false true
Network Status	networkStatus	get		
Serial Number	serialNumber	get		
Telnet	telnetDisabled	get / set		false true
SSH	sshDisabled	get / set		false true
Firmware Version	version	get		

Instance Tag	Command	Attribute Code
DEVICE	get	serialNumber

Example

```
DEVICE get serialNumber

+OK "value":"01842224"
```

Example

```
DEVICE get networkStatus

+OK "value":{"schemaVersion":2 "hostname":"TesiraServer91" "defaultGatewayStatus":"0.0.0.0"
"networkInterfaceStatusWithName":[{"interfaceId":"control"
"networkInterfaceStatus":{"macAddress":"00:90:5e:13:3b:27" "linkStatus":LINK_1_GB
"addressSource":STATIC "ip":"10.30.150.62" "netmask":"255.255.0.0" "dhcpLeaseObtainedDate":""
"dhcpLeaseExpiresDate":"" "gateway":"0.0.0.0"}}] "dnsStatus":{"primaryDNSServer":"0.0.0.0"
"secondaryDNSServer":"0.0.0.0" "domainName":""} "mDNSEnabled":true" telnetDisabled":false}
```

ipConfig commands

The ipConfig command can set the DHCP state, IP address, Subnet mask and Gateway on a Tesira Server, Server IO and TesiraFORTÉ device. Only values that need to be changed are required to be specified.

To get the IP configuration of a device:

Instance Tag	Command	Attribute Code	Index
DEVICE	get	ipConfig	control

Example

```
DEVICE get ipConfig control
+OK "value":{"autoIPEnabled":true "ip":"" "netmask":"" "gateway":""}
```

To set a device to not use DHCP and with an IP address of 192.168.1.210, a subnet of 255.255.255.0 and no gateway:

Example

```
DEVICE set ipConfig control {"autoIPEnabled":false "ip":"192.168.1.210" "netmask":"255.255.255.0"
"gateway":"0.0.0.0"}
```

To set a device that is using a fixed IP address to use DHCP

Example

```
DEVICE set ipConfig control {"autoIPEnabled":true }
```

To change a device IP address to a new address in the same subnet (this example moves a device from 192.168.1.210 to 192.168.1.110) :

Example

```
DEVICE set ipConfig control { "ip":"192.168.1.110" }
```

Session

The SESSION Instance Tag is case sensitive and must be in capital letters. It is used to send session specific Attributes and Commands. This includes the response method and can be used to query the commands.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
SESSION	get	aliases

Example
SESSION get aliases +OK "list":["123" "AudioMeter1" "AudioMeter2" "AudioMeter3" "DEVICE" "Input1" "Mixer1" "Mute1" "Level1" "Output1"]

Attribute Description	Attribute Code	Command	Indexes	Value Range
Aliases	aliases	get		
Verbose Output Enabled	verbose	get / set toggle		false, true

Output Styles

A Verbose or concise response can be configured as part of the Session type.

- **Verbose:** +OK “time”:”12:00” “number”:”503-367-3568” “line”:”2”
- **Concise:** +OK ”12:00” ”503-367-3568” ”2”

Example
SESSION set verbose true Mute1 get numChannels +OK "value":2 SESSION set verbose false +OK Mute1 get numChannels +OK 2

IO Blocks

Audio Input Block

The following attribute tables that relate to any standard Mic/Line Input Blocks.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
Input1	get	gain	1

Example
Input1 get numChannels +OK "value":2 Input1 get gain 1 +OK "value":24.000000 Input1 set gain 1 12 +OK Input1 get gain 1 +OK "value":12.000000

Attribute Description	Attribute Code	Command	Indexes	Value Range
Gain	gain	get / set increment decrement	channel	0 - 66 dB in 6 dB increments

Invert	invert	get / set toggle	channel	false, true
Level	level	get / set increment decrement	channel	minLevel - maxLevel dB
Mute	mute	get / set toggle	channel	false, true
Channel Count	numChannels	get		1 - 24
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		
Phantom Power On	phantomPower	get / set toggle	channel	false, true

Audio Output Block

The following attribute tables relate to any standard Mic/Line Output Blocks.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
Output1	get	gain	1

Example
Output1 get numChannels +OK "value":2 Output1 set mute 1 true +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Full Scale	fullScale	get / set increment decrement	channel	-31 or 0 - 24 dB in 6 dB increments
Invert	invert	get / set toggle	channel	false, true
Level	level	get / set increment decrement	channel	minLevel - maxLevel dB
Mute	mute	get /set toggle	channel	false, true

Channel Count	numChannels	get		1 - 24

AVB.1 Input Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
AVB Data Format	format	get		LINEAR_PCM, FLOAT_32, GENERIC_32
Invert	invert	get / set toggle	channel	false, true
Level	level	get / set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
Max Level	maxLevel	get / set increment decrement	channel	minLevel - 12.0 dB
Min Level	minLevel	get / set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get / set toggle	channel	false, true
Channel Count	numChannels	get		1 - 60
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true

All Peaks	peaks	get subscribe unsubscribe		
Stream Connection Status	streamActive	get subscribe unsubscribe		false, true
AVB Stream Name	streamName	get		
Enable Redundant Stream	useCableRedundancy	get		false, true

AVB.1 Output Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
AVB Data Format	format	get		LINEAR_PCM, FLOAT_32, GENERIC_32
Invert	invert	get / set toggle	channel	false, true
Level	level	get / set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
Max Level	maxLevel	get / set increment decrement	channel	minLevel - 0.0 dB
Min Level	minLevel	get / set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get / set toggle	channel	false, true
Channel Count	numChannels	get		1 - 60
Stream Connection Status	streamActive	get subscribe unsubscribe		false, true
AVB Stream Name	streamName	get		

Enable Redundant Stream	useCableRedundancy	get		false, true

Dante Input Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channel Name (in Dante terms, 'RX Channel Label')	channelName	get	channel	Case-insensitive, up to 31 characters except '=' '.' '@' '\' '<' '>'
Invert	invert	get/set toggle	channel	false, true
Level	level	get/set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Max Level	maxLevel	get/set increment decrement	channel	minLevel - 12.0 dB
Min Level	minLevel	get/set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get/set toggle	channel	false, true

		subscribe unsubscribe		
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 16
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		

Dante Output Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channel Name (in Dante terms, 'TX Channel Label')	channelName	get	channel	Case-insensitive, up to 31 characters except '=' '.' '@' '\' '<' '>'
Invert	invert	get/set/toggle	channel	false, true
Level	level	get/set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Max Level	maxLevel	get/set increment decrement	channel	minLevel - 0.0 dB
Min Level	minLevel	get/set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get/set toggle subscribe unsubscribe	channel	false, true

All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 16

Audio-Technica Mic Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channel Name (Dante 'RX Channel Label')	channelName	get	channel	Case-insensitive, up to 31 characters except '=' '.' '@' '\' '<' '>'
Device Name (Dante 'Hostname of TX Device')	deviceName	get subscribe unsubscribe	channel	
All Device Names (Dante 'Hostnames of all TX Devices')	deviceNames	get subscribe unsubscribe		
Logic Output Enable	enableLogicOutputs	get		false, true
Gain	gain	get / set increment decrement	channel	30-50 dB in 10 dB increments
Invert	invert	get/set/toggle	channel	false, true
LED Logic	ledLogic	get		NONE, ONE_LOGIC_INPUT_ ALTERNATELY_DRIVES

				_TWO_LEDS, TWO_LOGIC_INPUTS _FOR_SEPARATE_ CONTROL_OF_TWO_LEDS
Level	level	get / set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Locate Mode Enable	locateMode	get / set toggle	channel	false, true
Low Cut	lowCut	get/set/toggle	channel	false, true
Max Level	maxLevel	get / set increment decrement	channel	minLevel - 12.0 dB
Microphone Mode	micMode	get		TOGGLE_MUTE, TOGGLE_TALK, PUSH_TO_TALK, PUSH_TO_MUTE, EXTERNAL
Microphone Model	micModel	get		ATND971, ATND8677, ATND8734, ANYTYPE
Microphone Mute Occurring	micMute	get subscribe unsubscribe	channel	false, true
All Microphone Mute Occurring States	micMutes	get subscribe unsubscribe		
Min Level	minLevel	get / set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 64

Logic Input Count	numLogicInputs	get		Zero to three per input channel
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		
Phantom Power	phantomPower	get set toggle	channel	false, true

SHURE Mic Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channel Name (Dante 'RX Channel Label')	channelName	get	channel	Case-insensitive, up to 31 characters except '=' '.' '@' '\' '<' '>'
Device Name (Dante 'Hostname of TX Device')	deviceName	get subscribe unsubscribe	channel	
All Device Names (Dante 'Hostnames of all TX Devices')	deviceNames	get subscribe unsubscribe		
Invert	invert	get/set/toggle	channel	false, true
Level	level	get / set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Max Level	maxLevel	get / set increment decrement	channel	minLevel - 12.0 dB

Microphone Model	micModel	get		MXA910 MXA310 ANYTYPE
Min Level	minLevel	get / set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 64
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		

CobraNet Input Block

The following attribute tables that relate to any CobraNet Input Blocks.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
CNInput1	set	enable	true

Example
CNInput1 get bundleNumber +OK "value":256 CNInput1 set enable true +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
CobraNet Bundle Number	bundleNumber	get /set increment decrement subscribe unsubscribe		1 - 255 if multicast, 256 - 65279 if not
Enabled	enable	get / set toggle		false, true
Invert	invert	get / set toggle	channel	false, true

Level	level	get /set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Multicast On	multicast	get / set toggle		false, true
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 8
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		

CobraNet Output Block

The following attribute tables that relate to any CobraNet Input Blocks.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
CNOutput1	set	enable	true

Example

CNOutput1 get bundleNumber
+OK "value":300

CNOutput1 set enable true
+OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
CobraNet Bundle Number	bundleNumber	get / set increment decrement subscribe unsubscribe		1 - 255 multicast, 256 - 65279 Unicast
Enabled	enable	get / set toggle		false, true
Invert	invert	get / set toggle	channel	false, true

Level	level	get / set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Multicast On	multicast	get / set toggle		false, true
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 8

USB Input Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Connection Status	connected	get subscribe unsubscribe		false, true
Host Master Mute Status	hostMasterMute	get subscribe unsubscribe		false, true
Host Master Volume Control Level	hostMasterVol	get subscribe unsubscribe		-100.0 - 12.0 dB
Level	level	get/set increment decrement	channel	minLevel - maxLevel dB
All Levels	levels	get		
Max Level	maxLevel	get/set increment decrement	channel	minLevel - 12.0 dB
Min Level	minLevel	get/set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get/set/toggle	channel	false, true
All Mute States	mutes	get		
Channel Count	numChannels	get		1 - 8

Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		
Streaming Status	streaming	get subscribe unsubscribe		false, true

USB Output Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Connection Status	connected	get subscribe unsubscribe		false, true
Host Master Mute Status	hostMasterMute	get subscribe unsubscribe		false, true
Host Master Volume Control Level	hostMasterVol	get subscribe unsubscribe		-100.0 - 0.0 dB
Level	level	get/set increment decrement	channel	minLevel - maxLevel dB
All Levels	levels	get		
Max Level	maxLevel	get/set increment decrement	channel	minLevel - 0.0 dB
Min Level	minLevel	get/set increment decrement	channel	-100.0 - maxLevel dB
Mute Status	mute	get/set toggle	channel	false, true
All Mute States	mutes	get		
Channel Count	numChannels	get		1 - 8

Streaming Status	streaming	get subscribe unsubscribe		false, true

AEC Input Block

The following attribute tables relate to any AEC Input processing Blocks.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
Aec1	get	aecEnable	1

Confirm number of channels and set Input gain on Channel 1

Example
AecInput1 get numChannels +OK "value":2 AecInput1 get gain 1 +OK "value":0.000000 AecInput1 set gain 1 48 +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Gain	gain	get / set increment decrement	channel	0 - 66 dB in 6 dB increments
Channel Count	numChannels	get		1 - 24
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true

All Peaks	peaks	get subscribe unsubscribe		
Phantom Power On	phantomPower	get / set toggle subscribe unsubscribe	channel	false, true
All Phantom Power States	phantomPowers	get subscribe unsubscribe		

AEC Processing Block

The following attribute tables that relate to any AEC processing Blocks.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
Aec1	get	aecEnable	1

Confirm processing on Aec1 Instance tag is enabled

Example
Aec1 get aecEnable 1 +OK "value":true

Attribute Description	Attribute Code	Command	Indexes	Value Range
AEC Enabled	aecEnable	get set toggle	channel	false, true
Bypass AGC	agcBypass	get set toggle	channel	false, true
Hold Time	holdTime	get set increment decrement	channel	0 - 350000 s
HPF Bypass	hpfBypass	get set toggle	channel	false, true

HPF Center Freq.	hpfCutoff	get set increment decrement	channel	20.0 - 500.0 Hz
Invert	invert	get set toggle	channel	false, true
Level	level	get set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		
Limiter Enabled	limiterEnable	get set toggle	channel	false, true
Max Attenuation	maxAttenuation	get set increment decrement	channel	0.0 - 12.0 dB
Max Gain	maxGain	get set increment decrement	channel	0.0 - 12.0 dB
Max Gain Adj. Rate	maxGainAdjRate	get set increment decrement	channel	0.0 - 5.0 dB/s
Max Level	maxLevel	get set increment decrement	channel	minLevel - 12.0 dB
All Meter States	meters	get subscribe unsubscribe	channel	
Min Level	minLevel	get set increment decrement	channel	-100.0 - maxLevel dB
Min SNR	minSnr	get set increment decrement	channel	10.0 - 50.0 dB
Min Threshold	minThreshold	get set	channel	-30.0 - 10.0 dBu

		increment decrement		
Mute	mute	get set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Nonlinear Processing Mode	nlpMode	get set	channel	NLPMODE_NONE, NLPMODE_LOW, NLPMODE_MEDIUM, NLPMODE_HIGH
Noise Reduction	nrdMode	get set	channel	OFF, LOW, MED, HIGH
Channel Count	numChannels	get		1 - 24
Speech Mode	speechMode	get set toggle	channel	false, true
Target Level	targetLevel	get set increment decrement	channel	-10.0 - 10.0 dB

ANC Input Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
AncInput1	get	numChannels

Example
AncInput1 get numChannels +OK "value":2

Attribute Description	Attribute Code	Command	Indexes	Value Range
Gain	gain	get / set increment decrement	channel	0 - 66 dB in 6 dB increments
Channel Count	numChannels	get		1 - 16
Peak Occurring	peak	get subscribe unsubscribe	channel	false, true
All Peaks	peaks	get subscribe unsubscribe		
Phantom Power On	phantomPower	get / set toggle subscribe unsubscribe	channel	false, true

All Phantom Power States	phantomPowers	get subscribe unsubscribe		

ANC Processing Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Anc1	get	numChannels

Example
Anc1 get numChannels +OK "value":2

Attribute Description	Attribute Code	Command	Indexes	Value Range
Ambient Threshold	ambThreshold	get / set increment decrement	channel	-100.0 - 0.0 dBu
Bypass	bypass	get / set toggle	channel	false, true
Compensation Max	maxGain	get / set increment decrement	channel	0.0 - 25.0 dB
All Meter States	meters	get subscribe unsubscribe	channel	
Channel Count	numChannels	get		1 - 16
Compensation Ratio	ratio	get / set increment decrement	channel	0.25 - 1.0

Response Time Down	responseTimeDown	get / set increment decrement	channel	500.0 - 300000.0 ms
Response Time Up	responseTimeUp	get / set increment decrement	channel	500.0 - 300000.0 ms
RT-60	rt60	get / set increment decrement	channel	300.0 - 8000.0 ms

TI Receive Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
TIReceive1	get	level

Example
TIReceive get level +OK "value":0.000000

Attribute Description	Attribute Code	Command	Value Range
Line Echo Cancel	lec	get /set toggle	false, true
Input Level	level	get / set increment decrement	minLevel - maxLevel dB
Max Input Level	maxLevel	get / set increment decrement	minLevel - 12.0 dB
Min Input Level	minLevel	get / set increment decrement	-100.0 - maxLevel dB
Mute	mute	get / set toggle subscribe unsubscribe	false, true
Channel Count	numChannels	get	Always 1

Ring Tone Level	ringLevel	get / set increment decrement	-100.0 - 0.0 dB

TI Transmit Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
TITransmit1	get	level

Example
TITansmit get level +OK "value":0.000000

Attribute	Attribute Code	Commands	Value Range
Input Level	level	get / set increment decrement	minLevel - maxLevel dB
Max Input Level	maxLevel	get / set increment decrement	minLevel - 12.0 dB
Min Input Level	minLevel	get / set increment decrement	-100.0 - maxLevel dB
Mute	mute	get / set toggle	false, true
Channel Count	numChannels	get	Always 1

TI Control/Status Block

The TI Control/Status blocks allows TTP control of a number of TI Service Codes that can be used for call based functions. It also enables a number of STC Call State commands that allows monitoring and feedback to a control system as well as TI Control Status Attributes for controlling general STC-2 functions.

When a STC-2 card is used and a Dialer is added and associated with the respective Control/Status block there are also a number of dialer specific attributes. Please refer to the Dialer section for more information.

TI Service Codes

The Following table summarizes TI Service Codes. Due to the nature of the service being requested they do not require specific Attribute commands (get, set, etc). Adding a Dialer Component object will allow many more calling functions. Please refer to the Dialer Block section for more information.

Instance_Tag Service [Index] [Value] LF

- Instance Tag: Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **Index**: Is shown in [Brackets] as may be required depending on the Service being referenced. The Index is two space delimited numbers. The first number is the Line which is 1 or 2 and the Call Appearance Index which is 1.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the [Value] section for more details.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Service Code	Value
TIControlStatus1	dial	+15036417287

Description	Service Code	Value
Redial	redial	
End	end	
Flash	flash	
Dial (Used when On Hook Only)	dial	Number to Dial (A String)
DTMF (Used when Off Hook only)	dtmf	One number between 0 - 9, * or #
Answer	answer	

TI Control Status Attributes

When a STC-2 card is used it also allows access to all the dialer functions. Please refer to the Dialer section for more information.

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
TIControlStatus1	set	autoAnswer	true

Attribute Description	Attribute Code	Command	Value Range
Auto Answer	autoAnswer	get / set toggle	false, true
Auto Answer Ring Count	autoAnswerRingCount	get / set	AA_ONE_RING, AA_TWO_RINGS, AA_THREE_RINGS, AA_FOUR_RINGS, AA_FIVE_RINGS
Auto Disconnect Type	autoDisconnect	get / set	AD_NONE, AD_LOOP_DROP, AD_CALL_PROGRESS, AD_LOOP_DROP_PLUS_CALL_PROGRESS
Busy Tone Detected	busyToneDetected	get subscribe unsubscribe	false, true
Caller ID Enabled	callerIdEnable	get / set toggle	false, true
Call State	callState	get subscribe unsubscribe	
Simple Caller ID	cid	get	
Full Caller ID	cidUser	get	

Dialing	dialing	get subscribe unsubscribe	false, true
Dial Tone Detected	dialToneDetected	get subscribe unsubscribe	false, true
Dial Tone Level	dialToneLevel	get / set increment decrement	-70 - 0 dB
Line Fault	faultCondition	get subscribe unsubscribe	LINE_NO_FAULT, LINE_OVERCURRENT_FAULT, LINE_UNDERVOLTAGE_FAULT, LINE_UNDERCURRENT_FAULT, LINE_OVERVOLTAGE_FAULT, LINE_POLARITY_REVERSAL_FAULT
Flash	hookFlash	set	Value ignored
Flash Duration	hookFlashDuration	get / set increment decrement	100 - 800 ms
Hook State	hookState	get / set subscribe unsubscribe	OFFHOOK, ONHOOK
Last Number Dialed	lastNum	get subscribe unsubscribe	
Line Fault	lineFault	get subscribe unsubscribe	false, true
Line Intrusion	lineIntrusion	get subscribe unsubscribe	false, true
Line In Use	lineInUse	get subscribe unsubscribe	false, true
Line Ready	lineReady	get subscribe unsubscribe	false, true
Line Voltage	lineVoltage	get subscribe unsubscribe	Actual line voltage
DTMF Local Level	localDtmfToneLevel	get / set increment decrement	-100 - 0 dB
Loop Current	loopCurrent	get subscribe unsubscribe	Actual loop current

Channel Count	numChannels	get	Always 1
Ring Back Tone Detected	ringBackToneDetected	get subscribe unsubscribe	false, true
Ringing	ringing	get subscribe unsubscribe	false, true
Use Redial	useRedial	get / set toggle	false, true
Wait For Dial Tone	waitForDialTone	get / set toggle	false, true

STC Call State Commands

Using the TTP Call State Command with the STC-2 Card

The analog Control Status Block supports the use of Call State monitoring in order to poll information about the current call state of the telephone card. The response will include multiple information fields for the line. Call State is also available as a subscribed service to allow unsolicited feedback to a connected control system via TTP. A full call state subscription update will be sent if any single part of the call state has changed.

Definitions

Line

A single extension on the STC-2 card. A line will have a dedicated phone number and the voice signals for this line are available as an independent input and output in the Tesira system. Each STC-2 card supports two lines and these lines may be used at the same time.

Call Appearance

A call appearance can be viewed as a voice connection point on a line. Each line supports a single call appearance. The call appearance will always indicate a 0 in the Tesira STC card. Note that this does not mean that the card does not support call waiting or line conferencing, it simply means that this would be a function of the phone system.

Call State Requests

Get the status of the Call State:

Instance Tag	Command	Attribute Code
TIControlStatus1	get	callState

- This command will give a onetime indication of the current state of the analog phone.
- Note that the Instance Tag field is variable and needs to match what is running in the current configuration.

Subscriptions

Subscribe to a Call State:

This command will set a subscription to a VoIP card's current state. Please review the subscriptions section for more details. If any portion of the card's call state changes, a subscription response will be provided indicating the current status of all call states.

The response of the subscription depends on the SESSION verbose State that was active at the time the subscription was setup. Examples will be given to show the response of a call state in both verbose and non-verbose formats.

Instance Tag	Command	Attribute Code	Index	Value
TIControlStatus1	subscribe	callState	[CustomLabel]	[Time(ms)]

- **Index** can be used to assign a custom label to the subscription. This label is not required but is recommended, especially if there is more than one STC-2 card in the system. The label would indicate to the control system which card is providing the state change. Instance Tags are not included in call state subscriptions responses.

- **Value** can be used to throttle the rate of response to the control system. Since a call state subscription update is only provided after a state change there should be no need to place a value in this field. Placing a value, especially if it is too high, could introduce a missed update effectively getting the STC card and the control system out of sync. By default the call state subscription has a 200ms delay, this ensures that the as many changed states as possible are included in a single call state response.

- Subscriptions are lost when the Tesira server is rebooted.
- Subscriptions can be revalidated by subscribing to the same block at regular intervals. If this is done ensure that the custom label used in **Index** is used in the re-subscription. If this label is not included it is possible to inadvertently open multiple subscriptions to the same call state.

Unsubscribing from a Call state.

This command will cancel a previously set subscription.

Instance Tag	Command	Attribute Code	Index
TIControlStatus1	unsubscribe	callState	[CustomLabel]

Call State Indication Fields

A Call State response will provide information for the requested STC-2 card line. Every Call State response will include the following information fields.

State

The State response gives the current operating conditions of the call on the analog line.

- The verbose indicator for the State field is: “state”
- Non-Verbose indicator responses will be numeric and are shown below.

Below is a list of the possible state responses from a STC-2 card:

Verbose	Non-Verbose	Description
TI_CALL_STATE_IDLE	1	The analog line is on hook and ready to make a call
TI_CALL_STATE_DIALING	2	A number has been entered in the STC card and it is currently dialing.
TI_CALL_STATE_RINGBACK	3	The far end is ringing
TI_CALL_STATE_BUSY_TONE	4	The far end has presented a busy indication
TI_CALL_STATE_ERROR_TONE	5	The STC card has received an error tone on the line
TI_CALL_STATE_CONNECTED	6	The call to the far end has been connected
TI_CALL_STATE_RINGING	7	A STC card has detected an incoming call
TI_CALL_STATE_DROPPED	8	The far end has hung up the call
TI_CALL_STATE_INIT	12	The card is booting

TI_CALL_STATE_FAULT	13	A fault has been detected on the phone line (reference the prompt field for more information)
TI_CALL_STATE_CONNECTED_MUTED	14	A call has been connected but the SVC receive block mute has been engaged

Line ID

Each STC-2 card supports two phone lines. A line is indicated as a unique extension on the analog system. The Line ID field indicates which line of the card the particular Call State response is located for.

- A Call State response is only valid for a single line; the line of the Control Status block that the request was sent to
- The first line is indicated as Line ID 0 and the second line is Line ID 1.
- The verbose indicator for Line ID is: “lineId”. Note the upper case “I” in this indicator.

Call ID

Unlike the Tesira SVC-2 (VoIP) card, each line of the STC-2 card only supports a single call appearance. A call appearance is defined as a separate phone connection point of a single phone extension. The Call ID field indicates which call appearance the particular Call State response is reporting. This will vary in the SVC-2 card but the STC-2 card will always report 0. Although the information contained in this Call State response field may not be pertinent to the STC operation, it has been left in so the same control system parser can be used for both types of telephony cards.

The verbose indicator for Call ID is: “callId”. Note the upper case “I” in this indicator.

Action

The Action field of the Call State response is a function of the Tesira SVC-2 (VoIP) interface. Although the information contained in this Call State response field may not be pertinent to the STC operation, it has been left in so the same control system parser can be used for both types of telephony cards.

The information provided in this field for the STC card will **always** be:

Verbose	Non-Verbose	Description
UI_DISPLAY_STATUS	1	Call State response

- The verbose indicator for Action is: “action”

Caller ID

If caller ID information is available it will be included in the Call State response

Format
"\"MMDDHHmm\"\"incoming_number\"\"caller_Name\""

- If no caller ID is available the Call State response for this field will be ""
- The first set of quotes contains the date and time in the format MMDDHHmm.
- The second set of quotes represents the incoming phone number in the format 5036417287.
- The third set of quotes contains the name of the caller. If there are quotes contained within the name, there will be a backslash preceding the quotes within the name, i.e. “John \"Johnny\" Doe"

- A Backslash (\ ) is used as a separator in the caller ID string

Example of a caller ID response with all information provided
"\"07131134\"\"15036260281\"\"Biamp Systems\""

Example of a caller ID response without all information provided
"\"07131134\"\"15036260281\"\"\""

- The verbose indicator for Caller ID is: “cid”

Prompt

The function of the prompt field in the STC Call State response is to provide further information on fault states detected on the analog line.

The verbose indicator for Prompt is: “prompt”

Below is a list of the possible prompt responses from a STC-2 card:

Verbose	Non-Verbose	Description
FAULT_NONE	1	No line fault has been detected
FAULT_OVERCURRENT	2	STC-2 card has detected excessive current on the phone line. * See **Note** below
FAULT_UNDERVOLTAGE	3	STC-2 card has detected a low voltage condition on the phone line. * See **Note** below
FAULT_UNDERCURRENT	4	STC-2 card has detected a low current condition on the phone line. * See **Note** below
FAULT_OVERVOLTAGE	5	STC-2 card has detected excessive voltage on the phone line. * See **Note** below
FAULT_POLARITY_REVERSAL	6	The + & - legs of the analog telephone line are reversed

*** Note:** the trigger point of a voltage fault is dependent on the Country of Origin settings defined in Tesira software

Syntax of the Call State Response

Call State response information order:

The Call State response will present the information listed above for each line and call appearance of the STC card. If a subscription to a Call State response is setup, the subscription will update if a change is detected in any of the information fields. Call State is available in both verbose and non-verbose responses. Below is an example of the order of information in a Call State response.

HEADER_TOKEN:[{STATE: LINE_ID: CALL_ID: ACTION: CALLER_ID: PROMPT}]

Call State full command examples:

In the following examples a Call State response will be given in both verbose and non-verbose formats. This information is intended to show a clear example of the expected response order.

All subscription responses will start with the “!” character for easy recognition. The response will also include token information in the form of the custom label associated with the subscription. Custom labels are defined in the **Index** of the Call State command when the subscription is setup.

In the following examples custom label was defines as “Room_1”. The call in each example shows the call state immediately after a call has been placed on line 0. Caller ID information is also included.

Verbose Format
! "publishToken":" Room 1" "value":{"callStateInfo":[{"state":TI_CALL_STATE_DIALING "lineId":0 "callId":0 "action":UI_DISPLAY_STATUS "cid":"\"07131038\"\"146\"\"\"" "prompt":FAULT_NONE}]}

Non-Verbose Format
! "Room_1" [[[2 0 0 2 "\"07131038\"\"146\"\"\"" 1]]]

VoIP Receive Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
VoIPReceive1	get	level	1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Level	level	get / set increment decrement	line	minLevel - maxLevel dB
Max Level	maxLevel	get / set increment decrement	line	minLevel - 12.0 dB
Min Level	minLevel	get / set increment decrement	line	-100.0 - maxLevel dB
Mute	mute	get / set toggle subscribe unsubscribe	line	false, true
Line Count	numChannels	get		Always 2

VoIP Transmit Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
VoIPTransmit1	get	level	1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Level	level	get / set increment decrement	line	minLevel - maxLevel dB
Max Level	maxLevel	get / set increment decrement	line	minLevel - 12.0 dB
Min Level	minLevel	get / set increment decrement	line	-100.0 - maxLevel dB
Mute	mute	get / set toggle subscribe unsubscribe	line	false, true
Line Count	numChannels	get		Always 2

VoIP Control/Status Block

The VoIP Control/Status blocks allows TTP control of a number of Dialer Service Codes that can be used for call based functions. It also enables a number of VoIP Call State Commands that allows monitoring and feedback to a control system as well as VoIP Control Status attributes for controlling general VoIP functions.

When a SVC-2 card is used and a Dialer is added and associated with the respective VoIP Control/Status block there are also a number of dialer specific attributes. Please refer to the Dialer section for more information.

Dialer Service Codes

The Tesira SERVER or SERVER IO SVC-2 VoIP card or TesiraFORTÉ VI can support two independent phone lines. Each independent line can support up to 6 call appearances. Each call appearance can be a call to a different far end. However, there are limitations on active call appearances that apply to each line independently within an SVC-2 card.

- Two active call appearances -The maximum number of active call appearances (i.e. call appearances that are not on hold) per line is two. When two call appearances are active, no other call appearances can be used for any purpose (an active call or a call on hold).
- **Less than two active call appearances** -If there is only one active call appearance (or none), then all of the remaining call appearances can have calls on hold. In this case, the SVC-2 card will allow a call appearance to be put on hold and a different call appearance made active.

If you have 3 calls on hold you can choose any one of those to become an active call, but to conference in a second call appearance you need to disconnect the 3rd call before the conferencing can take place. Similarly, if 2 calls are in conference, any attempt to have a 3rd appearance dial in will result in a busy tone / redirect to voicemail / etc. since the system is already fully engaged. If a user tries to initiate a 3rd call appearance from the Tesira VoIP they will get an audible error tone / warble indicating they cannot complete the action.

Each element of the Service Code instruction is delimited by a single space. The commands are case sensitive and upper and lower case characters are used. The TTP string is structured in the following order:

Instance_Tag Service [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **Index:** Is shown in [Brackets] as may be required depending on the Service being referenced. The first number is the Line which is 1 or 2 and the Call Appearance Index which is 1,2,3,4,5 or 6.
- **[Value]:** Is shown in [Brackets] as may be required depending on the Command or Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Service	Index	Index	Value
VoIPControlStatus1	dial	1	1	15036417287

Description	Service	Index	Value
Redial	redial	Line,Call Appearance	

End	end	Line,Call Appearance	
Flash	flash	Line,Call Appearance	
Send	send	Line,Call Appearance	
Dial (Used when On Hook Only)	dial	Line, Call Appearance	Number to Dial (A String)
DTMF (Used when Off Hook only)	dtmf	Line	One number between 0 - 9, * or #
Answer	answer	Line,Call Appearance	
Conference	lconf	Line,Call Appearance	
Resume	resume	Line, Call Appearance	
Leave Conference	leaveConf	Line, Call Appearance	
Specify call appearance	callAppearance	Line, Call Appearance (0 -5)	
Resume	resume	Line,Call Appearance	
Hold	hold	Line,Call Appearance	
Go Off Hook	offHook	Line,Call Appearance	
Go On Hook	onHook	Line,Call Appearance	
Go On Hook	onHook	Line,Call Appearance	
Go On Hook	onHook	Line,Call Appearance	
Go On Hook	onHook	Line,Call Appearance	
Transfer	transfer	Line,Call Appearance	*Refer to the VoIP Transfer Commands section*

VoIP Call State Commands

The VoIP Control Status Block supports the use of Call State monitoring in order to poll information about the current call state of the telephone card. The response will include multiple information fields for all lines and call appearances of the card. Call State is also available as a subscribed service to allow unsolicited feedback to a connected control system via TTP. A full call state subscription update will be sent if any single part of the call state has changed.

Please refer to the VoIP Call State commands for more information.

VoIP Control Status Attributes

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.

- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Index
VoIPControlStatus1	get	lineInUse	1	1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Auto Answer	autoAnswer	get / set toggle	line	false, true
Auto Answer Ring Count	autoAnswerRingCount	get / set	line	AA_IMMEDIATELY, AA_ONE_RING, AA_TWO_RINGS, AA_THREE_RINGS
Call State	callState	get subscribe unsubscribe		*Refer to the VoIP Call State Commands section*
Statistics	cardStat	get subscribe unsubscribe		
Simple Caller ID	cid	get subscribe unsubscribe	line (1 or 2), call appearance index (1-6)	
Full Caller ID	cidUser	get subscribe unsubscribe	line (1 or 2), call appearance index (1-6)	
Codec Priorities	codecPriority	get / set	line	
Call Progress Tone Level	cptLevel	get / set increment decrement	line	-100.0 - 0.0 dB
Dialing Timeout	dialingTimeOut	get / set increment decrement	line	0 - 20 seconds
Direct URL Dialing Enabled	directUrlDialing	get/set/toggle	line	false, true

Do Not Disturb Enabled	dndEnable	get / set toggle	line	false, true
Do Not Disturb Response Code	dndMode	get/set	line	DND_480, DND_486, DND_603
DTMF Off Time	dtmfOffTime	get / set increment decrement	line	40 - 1000 ms
DTMF On Time	dtmfOnTime	get / set increment decrement	line	40 - 1000 ms
DTMF via SIP Info	dtmfSipInfo	get / set	line	DTMF_SIP_INFO_OFF, DTMF_SIP_INFO_NORMAL, DTMF_SIP_INFO_SIMPLE
Last Number Dialed	lastNum	get subscribe unsubscribe	line	
Line In Use	lineInUse	get subscribe unsubscribe	line (1 or 2), call appearance index (1-6)	false, true
Line Ready	lineReady	get subscribe unsubscribe	line	false, true
DTMF Local Mute	localDtmfMute	get / set toggle	line	false, true
DTMF Local Level	localDtmfToneLevel	get / set increment decrement	line	-100.0 - 0.0 dB
NAT Info	nat	get subscribe unsubscribe		
Network Info	network	get subscribe unsubscribe		
Out-Of-Band DTMF Enabled	oobDtmf	get / set toggle	line	false, true
Out-Of-Band DTMF Payload Type	oobDtmfPayload	get / set increment decrement	line	97 - 127
Protocol Info	protocols	get subscribe unsubscribe		

Redial Enabled	redialEnable	get / set toggle	line	false, true
RFC 2543-Style Hold Enabled	rfc2543StyleHold	get / set toggle	line	false, true
Ringing	ringing	get subscribe unsubscribe	line (1 or 2), call appearance index (1-6)	false, true
Ring Type	ringType	get set	line	RING_TYPE_CLASSIC, RING_TYPE_SILENT
Synchronized Time	syncTime	set		hh:mm:ss:MM:DD:YYYY
VAD Enabled	vad	get / set toggle	line	false, true
VAD Threshold	vadThreshold	get / set increment decrement	line	-64.0 - 24.0 dB

Synchronized Time format is

- hh = Hours
- mm = minutes
- ss = Seconds. Leap seconds (SS=60) specification are forbidden.
- MM =month of year 1-12
- DD =day of month 1-(28,29,30,31) according to the month and year
- YYYY = Year must be >= 2000
- Spaces are not permitted after the : and before YYYY so ": 2000" is not valid.

Set Synchronized Time

Instance Tag	Command	Attribute Code	Index
VoIPControlStatus1	set	syncTime	"00:00:00:02:29:2014"

Example
VoIPControlStatus1 set syncTime "00:00:00:02:29:2014"

Call State Command

Using the TTP Call State Command with the SVC-2 Card

The VoIP Control Status Block supports the use of Call State monitoring in order to poll information about the current call state of the telephone card. The response will include multiple information fields for all lines and call appearances of the card. Call State is also available as a subscribed service to allow unsolicited feedback to a connected control system via TTP. A full call state subscription update will be sent if any single part of the call state has changed.

Definitions

- **Line** - A single extension on the SVC-2 card. A line will have a dedicated phone number and the voice signals for this line are available as an independent input and output in the Tesira system. Each SVC-2 card supports two lines and these lines may be used at the same time.

- **Call Appearance -** Each line supports up to 6 call appearances. A call appearance can be viewed as a voice connection point on a line. A call appearance can be used to open another call from the same line by placing an active call on hold. Call appearances also allow the SVC-2 card to support call waiting.

- **Conference** - The SVC card can create a local conference by joining two call appearances into a single active call. There is no support for a conference larger than a 3-way conference (two call appearances). It is possible to have an active 3-way conference on both lines of the SVC card at the same time.

Call State Requests

This command will give a onetime indication of the current state of the VoIP phone. The Instance Tag is variable and needs to match what is running in the current configuration.

Get the status of the Call State:

Instance Tag	Command	Attribute Code
VoIPControlStatus1	get	callState

Subscriptions

Subscribe to a Call State

This command will set a subscription to a VoIP card's current state. Please review the subscriptions section for more details. If any portion of the card's call state changes, a subscription response will be provided indicating the current status of all call states.

The response of the subscription depends on the SESSION verbose State that was active at the time the subscription was setup. Examples will be given to show the response of a call state in both verbose and non-verbose formats.

Instance Tag	Command	Attribute Code	Index	Value
VoIPControlStatus1	subscribe	callState	[CustomLabel]	[Time(ms)]

- **Index** can be used to assign a custom label to the subscription. This label is not required but is recommended, especially if there is more than one SVC-2 card in the system. The label would

indicate to the control system which card is providing the state change. Instance Tags are not included in call state subscriptions responses.

- **Value** can be used to throttle the rate of response to the control system. Since a call state subscription update is only provided after a state change there should be no need to place a value in this field. Placing a value, especially if it is too high, could introduce a missed update effectively getting the SVC card and the control system out of sync. By default the call state subscription has a 200ms delay, this ensures that the as many changed states as possible are included in a single call state response.
- Subscriptions are lost when the Tesira server is rebooted.
- Subscriptions can be revalidated by subscribing to the same block at regular intervals. If this is done ensure that the custom label used in **Index** is used in the re-subscription. If this label is not included it is possible to inadvertently open multiple subscriptions to the same call state.

Unsubscribing from a Call state.

This command will cancel a previously set subscription.

Instance Tag	Command	Attribute Code	Index
VoIPControlStatus1	unsubscribe	callState	[CustomLabel]

Call State Indication Fields

A Call State response will provide information for the entire SVC-2 card. The response will include both VoIP lines with 6 call appearances per line.

Example - Note Line feeds are shown to aid readability
! "publishToken":" Room1" "value":{"callStateInfo":[{"state":VOIP_CALL_STATE_RINGBACK "lineId":0 "callId":0 "action":UI_DISPLAY_STATUS "cid":"\"07131038\"\"146\"\"\"" "prompt":VOIP_PROMPT_CONNECTING} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":1 "action": UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":2 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":3 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":4 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":5 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":0 "action":UI_DISPLAY_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":1 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":2 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":3 "action":UI_CLEAR_STATUS"cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":4 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":5 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED}]}

Call Appearance - Information included in Call State Response

Each call appearance provides the following information fields in the Call State response.

State

- The State response gives the current operating conditions of the call appearance on the VoIP line.
- The verbose indicator for the State field is: “state”
- Non-Verbose indicator responses will be numeric and are shown below.

Below is a list of the possible state responses from a SVC-2 card:

Verbose	Non-Verbose	Description
VOIP_CALL_STATE_INIT	1	The call appearance is initializing indicating general setup is in place; DHCP in progress, registration is taking place, etc. This can also indicate that the line has not been configured. The SVC-2 card will not be able to dial when this state is displayed.
VOIP_CALL_STATE_FAULT	2	General Fault condition; Network link is down, IP address conflict in place. The SVC-2 card will not be able to dial when this state is displayed.
VOIP_CALL_STATE_IDLE	3	Call Appearance is part of a registered connection to a Proxy Server and is ready to make or receive a call.
VOIP_CALL_STATE_DIALTONE	4	Call appearance is off hook and dial tone is present.
VOIP_CALL_STATE_SILENT	5	User has started dialing numbers but has yet to hit send
VOIP_CALL_STATE_DIALING	6	User has hit send on the call appearance and the card has sent an INVITE to the proxy or the called party. No response has been received at this point.
VOIP_CALL_STATE_RINGBACK	7	The far end is ringing
VOIP_CALL_STATE_RINGING	8	The call appearance has an incoming call
VOIP_CALL_STATE_ANSWER_CALL	9	The call has been answered but the call isn't active yet
VOIP_CALL_STATE_BUSY	10	The far end is busy
VOIP_CALL_STATE_REJECT	11	User has rejected the incoming call
VOIP_CALL_STATE_INVALID_NUMBER	12	The user has dialed an invalid number on this call appearance

VOIP_CALL_STATE_ACTIVE	13	A call has been connected to the call appearance
VOIP_CALL_STATE_ACTIVE_MUTED	14	A call is established but audio is muted in the VoIP Receive block
VOIP_CALL_STATE_ON_HOLD	15	The near end has placed the call appearance on hold
VOIP_CALL_STATE_WAITING_RING	16	The call appearance has received a call waiting indication
VOIP_CALL_STATE_CONF_ACTIVE	17	The call appearance has been placed in a local conference
VOIP_CALL_STATE_CONF_HOLD	18	The call appearance is part of a local conference that has been placed on hold
VOIP_CALL_STATE_XFER_INIT	19	The call appearance is initializing
VOIP_CALL_STATE_XFER_SILENT	20	The call appearance is silent
VOIP_CALL_STATE_XFER_REQ_DIALING	21	The call appearance is awaiting number to be dialed
VOIP_CALL_STATE_XFER_PROCESS	22	The call appearance is in a process of transferring
VOIP_CALL_STATE_XFER_REPLACES_PROCESS	23	The call appearance is updating the transfer process
VOIP_CALL_STATE_XFER_ACTIVE	24	The call appearance transfer is active
VOIP_CALL_STATE_XFER_RINGBACK	25	The call appearance is seeing DTMF tones from the proxy server
VOIP_CALL_STATE_XFER_ON_HOLD	26	The call appearance is on hold
VOIP_CALL_STATE_XFER_DECISION	27	The call appearance is awaiting confirmation to transfer
VOIP_CALL_STATE_XFER_INIT_ERROR	28	The call appearance has experienced an error initializing the transfer process
VOIP_CALL_STATE_XFER_WAIT	29	The call appearance is waiting

Line ID

- Each SVC-2 card supports two phone lines. A line is indicated as a unique extension on the VoIP system. The Line ID field indicates which line the particular Call State response is located on.
- The first line is indicated as Line ID 0 and the second line is Line ID 1.
- The verbose indicator for Line ID is: “lineId”. Note the upper case “I” in this indicator.

Call ID

- Each line of the SVC-2 card supports six call appearances. A call appearance is defined as a separate phone connection point of a single phone extension. The Call ID field indicates which call appearance the particular Call State response is reporting.
- The first call appearance of a line is indicated as Call ID 0 and the last call appearance of a line is indicated as Call ID 5.

- The verbose indicator for Call ID is: “callId”. Note the upper case “I” in this indicator.

Action

- The Tesira user interface supports the shifting of focus of a call appearance selection. For example if a call is in place on call appearance 1 and call appearance 2 rings, the user can shift focus in the UI to call appearance 2 to check Caller ID. This action would shift the focus from appearance 1 to 2.
- The Call State response will indicate which call appearance is the point of focus for each line in the Action field. A control system program could track this action if multiple devices are providing VoIP dialer control.
- There can only be a single focused call appearance per line.
- The verbose indicator for Action is: “action”

Possible action responses from a SVC-2 card:

Verbose	Non-Verbose	Description
UI_CLEAR_STATUS	1	This call appearance is not the current point of focus in the user interface.
UI_DISPLAY_STATUS	2	This call appearance is the current point of focus in the user interface.

Caller ID

If caller ID information is available it will be included in the Call State response

Format
"\"MMDDHHmm\"\"incoming_number\"\"caller_Name\""

- If no caller ID is available the Call State response for this field will be ""
- The first set of quotes contains the date and time in the format MMDDHHmm.
- The second set of quotes represents the incoming phone number in the format 5036417287.
- The third set of quotes contains the name of the caller. If there are quotes contained within the name, there will be a backslash preceding the quotes within the name, i.e. “John \"Johnny\" Doe"
- A Backslash (\ ) is used as a separator in the caller ID string

Example of a caller ID response with all information provided
"\"07131134\"\"15036260281\"\"Biamp Systems\""

Example of a caller ID response without all information provided
"\"07131134\"\"15036260281\"\"\""

The verbose indicator for Caller ID is: “cid”

Prompt

The Tesira user interface provides prompting indications of the state of the call appearance that is currently in focus. This prompting information is also included in the Call State response. A control

system can use the prompt indications to provide users information about the individual call appearance states. Note that a prompt is provided for each call appearance in the Call State response.

The verbose indicator for Prompt is: "prompt"

Below is a list of the possible prompt responses from a SVC-2 card:

Verbose	Non-Verbose	Description
VOIP_PROMPT_NONE	1	Nothing to display in prompt field
VOIP_PROMPT_STARTING	2	SVC-2 card is booting. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_REGISTERING	3	SVC-2 is registering to a Proxy Server. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_SIP_USER_NOT_CONFIGURED	6	SIP User field has not been configured on the line properties page. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_ENTER_NUMBER	7	SVC-2 card is off hook and waiting for a number entry
VOIP_PROMPT_CONNECTING	8	Connecting to the number dialed
VOIP_PROMPT_INCOMING_CALL_FROM	9	Incoming call from a far end
VOIP_PROMPT_PEER_BUSY	10	The far end device is busy
VOIP_PROMPT_CALL_CANNOT_BE_COMPLETED	11	The number called from the SVC-2 card cannot be completed
VOIP_PROMPT_ON_HOLD	12	The SVC-2 card has placed the call on hold
VOIP_PROMPT_CALL_ON_HELD	13	The far end device has placed the call on hold
VOIP_PROMPT_CONFERENCE	14	The SVC-2 card has placed this call appearances into a conference
VOIP_PROMPT_CONFERENCE_ON_HOLD	15	The SVC-2 card has placed a conference on hold
VOIP_PROMPT_CONNECTED	16	The call appearance is connected to a far end device
VOIP_PROMPT_CONNECTED_MUTED	17	The call appearance is connected to a far end device but the VoIP Receive block has been muted
VOIP_PROMPT_AUTH_FAILURE	18	Authentication to Proxy Server has failed

VOIP_PROMPT_PROXY_NOT_CONFIGURED	19	A Proxy Address has not been entered in the SVC line properties page
VOIP_PROMPT_NETWORK_INIT	20	The SVC-2 card is setting up network communications. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_DHCP_IN_PROGRESS	21	The SVC-2 card is requesting an IP address via DHCP. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_NETWORK_LINK_DOWN	22	The SVC-2 network link sees no connection. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_NETWORK_LINK_UP	23	The SVC-2 network port sees a connection point but cannot make use of it due to its current IP settings. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_IPADDR_CONFLICT	24	An IP Address is conflict has been detected. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_NETWORK_CONFIGURED	25	The SVC network interface has been configured. The SVC-2 card will not be able to dial when this prompt is displayed.
VOIP_PROMPT_CODEC_NEGOTIATION_FAILURE	26	Codec negotiation between the endpoints has failed
VOIP_PROMPT_UNEXPECTED_ERROR	27	The SVC card has encountered an unexpected error
VOIP_PROMPT_AUTH_USER_NOT_CONFIGURED	28	Authentication Username has not been configured in the SVC line properties page
VOIP_PROMPT_AUTH_PASSWORD_NOT_CONFIGURED	29	Authentication Password has not been configured in the SVC line properties page
VOIP_PROMPT_DND	30	Do Not Disturb
VOIP_PROMPT_INVALID_NUMBER	31	Invalid Dialed Number (Number not routable by proxy server)
VOIP_PROMPT_TEMP_UNAVAILABLE	32	Temporary Not Available (The callee is temporarily

		unavailable such as DND is on)
VOIP_PROMPT_DECLINED	33	Call is Declined (the call is declined by the far end or the server)
VOIP_PROMPT_SERVICE_UNAVAILABLE	34	Service Unavailable (such as a PSTN call but PSTN gateway isn't configured or no rule to get there)
VOIP_PROMPT_FORBIDDEN	35	Call Forbidden (The call is prohibited because of policy)
VOIP_PROMPT_BEING_XFER_TO	36	Call is Being Transfer to
VOIP_PROMPT_XFER_IN_PROCESS	37	Transfer in Process
VOIP_PROMPT_XFER_TIME_OUT	38	Transfer Timeout (Transfer not finished in a designated time)
VOIP_PROMPT_PROXY_UNAVAILABLE	39	Proxy Unavailable (such as the configurable proxy is down)

Syntax of the Call State Response

Call State response information order:

The Call State response will present the information listed above for each line and call appearance of the SVC card. If a subscription to a Call State response is setup, the subscription will update if a change is detected in any of the information fields. Call State is available in both verbose and non-verbose responses. Below is an example of the order of information in a Call State response. Note that the "{….}" field indicates the additional lines and call appearances on the SVC card.

HEADER_TOKEN:[{STATE: LINE_ID: CALL_ID: ACTION: CALLER_ID: PROMPT} {….} {….}]

Call State subscription header examples

All subscription responses will start with the "!" character for easy recognition. The response will also include token information in the form of the custom label associated with the subscription. Custom labels are defined in the Index command when the subscription is setup. Below is an example or the subscription header of a Call State response in both verbose and non-verbose formats. In each case the custom label was defines as "Room_1" and the "{…}" symbol indicates the additional responses from the specific call appearances.

Verbose Format
! "publishToken":" Room_1" "value":{"callStateInfo":[{….} {….}]}

Non-Verbose Format
! "Room_1" [[….] [….]]

Single Call Appearance response examples

Below is an example of a response from a single call appearance in both verbose and non-verbose formats. This information is intended to show a clear example of the response order of a single appearance.

The call in each example shows the call state after a call was placed on line 0, call appearance 3, with the far end currently ringing. Caller ID information is also included.

Verbose Format
{"state":VOIP_CALL_STATE_RINGBACK "lineId":0 "callId":3 "action":UI_DISPLAY_STATUS "cid":"\"07131124\"\"146\"\"John Smith\"" "prompt":VOIP_PROMPT_CONNECTING}

Non-Verbose Format
[7 0 3 2 "\"07131124\"\"146\"\"John Smith\"" 8]

Call State full command examples

An actual Call State response will include two separate lines, each with 6 call appearances. An example of a full response is provided below in both verbose and non-verbose formats. The following responses show a ring-back on line 0, call appearance 0. All other call appearances on line 0 are idle. Line 1 has not been configured.

Verbose Format
! "publishToken":" Room 1" "value":{"callStateInfo":[{"state":VOIP_CALL_STATE_RINGBACK "lineId":0 "callId":0 "action":UI_DISPLAY_STATUS "cid":"\"07131038\"\"146\"\"\"" "prompt":VOIP_PROMPT_CONNECTING} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":1 "action": UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":2 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":3 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":4 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_IDLE "lineId":0 "callId":5 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_NONE} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":0 "action":UI_DISPLAY_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":1 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":2 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":3 "action":UI_CLEAR_STATUS"cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":4 "action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED} {"state":VOIP_CALL_STATE_INIT "lineId":1 "callId":5"action":UI_CLEAR_STATUS "cid":"" "prompt":VOIP_PROMPT_SIP_USER_NOT_CONFIGURED}]}

Non-Verbose Format
! "Room_1" [[[7 0 0 2 "\"07131038\"\"146\"\"\"" 8] [3 0 1 1 “” 1] [3 0 2 1 “” 1] [3 0 3 1 "" 1] [3 0 4 1 "" 1] [3 0 5 1 "" 1] [1 1 0 2 "" 6] [1 1 1 1 “” 6] [1 1 2 1 “” 6] [1 1 3 1 "" 6] [1 1 4 1 "" 6] [1 1 5 1 "" 6]]]

VoIP Transfer

Call transfer process

There are a number of supported workflows as part of transferring a call using third party control. The method used is dependent on external factors such as the phone system used. Below, the three different processes are defined. The method chosen would depend on the unique project requirements and functions of the phone system being used and its supported functions.

The 'transfer' function can be used for the default type of transfer when it has been defined in the Tesira software. The other two methods, the 'featureKey Blind' or 'featureKey Consultative' methods can be used as an 'on-the-fly' method of initiating a transfer.

Transfer

This function is a dedicated function for initiating the default type of call transfer. The transfer behavior is defined in the VoIP Line Properties block properties configured as part of the Tesira design file configuration. For some proxies, the default behavior is a 'blind transfer' for others it is a 'consultative transfer'. The control developers will need to know what kind of call transfer method is used then this should be configured in the Line properties > General Tab > Consultative Transfer setting.

Example - To transfer to extension 9175

VoIPControlStatus1 transfer 1 1
+OK

VoIPControlStatus1 dial 1 2 9175
+OK

Blind transfer

The function: **featureKey 1 1 FEATURE_KEY_BLIND_TRANSFER**, opens a new call appearance and allows the number to be transfered to to be entered. The process operates the same as off-hook dialing.

Once the transfer target answers the call, a manual confirmation of the transfer can be specified, alternatively, the transfer will complete itself after a timeout period (default 10 seconds)

To confirm the transfer, the feature key function with the new line instance is specified: **featureKey 1 2 FEATURE_KEY_BLIND_TRANSFER**.

If the transfer does not complete, the original call instance will remain on hold. This will need to be restored manually.

Example - To transfer to extension 9175

VoIPControlStatus1 featureKey 1 1 FEATURE_KEY_BLIND_TRANSFER

+OK

VoIPControlStatus1 dial 1 2 9175
+OK

VoIPControlStatus1 featureKey 1 2 FEATURE_KEY_BLIND_TRANSFER
+OK

Consultative transfer

The function: **featureKey 1 1 FEATURE_KEY_CONSULTATIVE_TRANSFER** opens a new call appearance and allows the number to be transfered to to be entered. The process operates the same as off-hook dialing.

Once the transfer target answers the call, a manual confirmation of the transfer must be specified. To confirm the transfer, the feature key function with the new line instance is specified by **featureKey 1 2 FEATURE_KEY_CONSULTATIVE_TRANSFER**.

Example - To transfer to extension 9175

VoIPControlStatus1 featureKey 1 1 FEATURE_KEY_CONSULTATIVE_TRANSFER
+OK

VoIPControlStatus1 dial 1 2 9175
+OK

VoIPControlStatus1 featureKey 1 2 FEATURE_KEY_CONSULTATIVE_TRANSFER
+OK

For a consultative transfer, the function featureKey(1, 1, FEATURE_KEY_CONSULTATIVE_TRANSFER) can be used to initiate a new call appearance. The number will be required to be inputted. This process looks like off-hook dialing. The original call is on hold and a new call is placed. Once the transfer target answers the call and would like to accept the transferred call, this function must be called again to complete the call transfer. In case of the failure, the original call is on hold and the call will be required to be restored manually.

DTMF Decode Block

DTMF Service Commands

Each element of the Service Code instruction is delimited by a single space. The commands are case sensitive and upper and lower case characters are used. The TTP string is structured in the following order:

Instance_Tag Service [Index] [Value] LF

- Instance Tag: Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **Index**: Is always required. The Index is two space delimited numbers. The first number is the Line which is 1 or 2 and the Call Appearance Index which is 1,2,3,4,5 or 6.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Service
DTMFDecode1	clear

Description	Service	Index	Value
Clear DTMF	clear		

DTMF Attribute Commands

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
DTMFDecode1	subscribe	dtmfs	MyCustomName	500

Command: **DTMFDecode1 subscribe dtmfs MyCustomName 500**

Result: changes to the DTMF Decode block number 1 will be sent every 500ms

Attribute Description	Attribute Code	Command	Value Range
Decoded Data	dtmfs	get subscribe unsubscribe	
Logic Enabled	enableLogic	get / set toggle	false, true

Labgruppen Amp

Service Codes

Each element of the Service Code instruction is delimited by a single space. The commands are case sensitive and upper and lower case characters are used. The TTP string is structured in the following order:

Instance_Tag Service [Index] [Value] LF

- Instance Tag: Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **Index:** Is shown in [Brackets] as may be required depending on the Service being referenced.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Service
LGAmp1	select

Description	Service	Index	Value
Identify Amplifier	select		

Status Attributes

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Amplifier Name	ampName	get		
Amplifier Power	ampPower	get / set toggle		false, true
Amplified Output Amp Status	ampStatus	get subscribe unsubscribe	channel	STATUS_OK, STATUS_WARNING, STATUS_ERROR, STATUS_UNKNOWN
Amplified Output Amp Status Reason	ampStatusReason	get	channel	Reason code for any indicator
Amplified Output Auto Power Down Threshold	apdThreshold	get / set increment decrement	channel	-100.0 - 0.0 dB
Auto Power Down Timeout	apdTimeoutMins	get / set increment decrement		0 - 60 min
Amplified Output Channel Name	channelName	get	channel	
Failover Input Gain	failoverGain	get / set increment decrement	channel	0 - 66 dB in 6 dB increments
All Failover Input Indicators	failoverIndicators	get subscribe unsubscribe		
Amplified Output Failover Input Channel	failoverInputChannel	get	channel	Failover input channel or 0 for none
Failover Input Invert	failoverInvert	get / set toggle	channel	false, true
Failover Input Level	failoverLevel	get / set increment decrement	channel	failoverMinLevel - failoverMaxLevel dB
Failover Input Level Max	failoverMaxLevel	get / set increment decrement	channel	failoverMinLevel - 12.0 dB
Failover Input Level Min	failoverMinLevel	get / set increment decrement	channel	-100.0 - failoverMaxLevel dB

Failover Input Mute	failoverMute	get / set toggle	channel	false, true
Failover Input Peak Indicator	failoverPeak	get subscribe unsubscribe	channel	false, true
Failover Input Phantom Power	failoverPhantomPower	get / set toggle	channel	false, true
Failover Input Signal Present Indicator	failoverSignalPresent	get subscribe unsubscribe	channel	false, true
Failover Input Signal Present Threshold	failoverSignalPresentThreshold	get / set increment decrement	channel	-64.0 - 30.0 dB
Amplified Output Failover Test	failoverTest	get / set toggle	channel	false, true
Amplified Output Failover Test Active Indicator	failoverTestActive	get subscribe unsubscribe	channel	false, true
Frame Status	frameStatus	get subscribe unsubscribe		STATUS_OK, STATUS_WARNING, STATUS_ERROR, STATUS_UNKNOWN
Frame Status Reason	frameStatusReason	get		Reason code for any indicator
All Frame Indicators	indicators	get subscribe unsubscribe		
Amplified Output Invert	invert	get / set toggle	channel	false, true
Amplified Output Level	level	get / set increment decrement	channel	minLevel - maxLevel dB
Amplified Output Load Status	loadStatus	get subscribe unsubscribe	channel	STATUS_OK, STATUS_WARNING, STATUS_ERROR, STATUS_UNKNOWN
Amplified Output Load Status Reason	loadStatusReason	get	channel	Reason code for any indicator

Amplified Output Level Max	maxLevel	get / set increment decrement	channel	minLevel - 0.0 dB
Amplified Output Level Min	minLevel	get / set increment decrement	channel	-100.0 - maxLevel dB
Amplified Output Mute	mute	get /set toggle	channel	false, true
Selected Time	selectedTime	get subscribe unsubscribe		0 - 2147483647 s
Amplified Output Signal Status	signalStatus	get subscribe unsubscribe	channel	STATUS_OK, STATUS_WARNING, STATUS_ERROR, STATUS_UNKNOWN
Amplified Output Signal Status Reason	signalStatusReason	get	channel	Reason code for any indicator

Tesira Amplifier Block

Service Codes

Each element of the Service Code instruction is delimited by a single space. The commands are case sensitive and upper and lower case characters are used. The TTP string is structured in the following order:

Instance_Tag Service [Index] [Value] LF

- Instance Tag: Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **Index:** Is shown in [Brackets] as may be required depending on the Service being referenced.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Service
TAmp1	select

Description	Service	Index	Value
Identify amplifier	select		

Status Attributes

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Amplifier Fault Indicator	ampFault	get subscribe unsubscribe		false, true
Amplified Output Mute All Channels	ampMuteAll	get set toggle subscribe unsubscribe		false, true
Amplifier Power	ampPower	get set toggle subscribe unsubscribe		false, true
Amplifier Standby Timeout	ampStandbyTimeout	get set		STANDBY_TIMEOUT_DISABLED, STANDBY_TIMEOUT_15, STANDBY_TIMEOUT_30, STANDBY_TIMEOUT_45, STANDBY_TIMEOUT_60
Amplifier Thermal Fault Indicator	ampThermalFault	get subscribe unsubscribe		THERMAL_NONE, THERMAL_WARNING, THERMAL_FAULT
Amplifier Warning Indicator	ampWarning	get subscribe unsubscribe		false, true
Amplified Output AVB Stream Present Indicator	AVBstreamPresent	get subscribe unsubscribe	channel	false, true
Amplified Output Expected Load Impedance	expectedImpedance	get set	channel	EXPECT_8_OHMS, EXPECT_4_OHMS
Amplified Output Failover Active Indicator	failoverActive	get subscribe unsubscribe	channel	false, true

Failover Input Gain	failoverGain	get set increment decrement	channel	0 - 66 dB in 6 dB increments
Amplified Output Failover Input Channel	failoverInputChannel	get set increment decrement	channel	Failover input channel or 0 for none
Failover Input Invert	failoverInvert	get set toggle	channel	false, true
Failover Input Level	failoverLevel	get set increment decrement subscribe unsubscribe	channel	failoverMinLevel - failoverMaxLevel dB
Failover Input Level Max	failoverMaxLevel	get set increment decrement	channel	failoverMinLevel - 12.0 dB
Failover Input Level Min	failoverMinLevel	get set increment decrement	channel	-100.0 - failoverMaxLevel dB
Failover Input Mute	failoverMute	get set toggle subscribe unsubscribe	channel	false, true
Failover Input Peak Indicator	failoverPeak	get subscribe unsubscribe	channel	false, true
Failover Input Phantom Power	failoverPhantomPower	get set toggle	channel	false, true
Failover Input Signal Present Indicator	failoverSignalPresent	get subscribe unsubscribe	channel	false, true
Failover Input Signal	failoverSignalPresentThreshold	get set increment decrement	channel	-64.0 - 30.0 dB

Present Threshold				
Amplified Output Failover Test	failoverTest	get set toggle subscribe unsubscribe	channel	false, true
Front Panel Lock	frontPanelLock	get set toggle subscribe unsubscribe		false, true
Amplified Output Sensitivity	gain	get set increment decrement	channel	0 - 24 dB in 6 dB increments
Amplified Output High Impedance Indicator	highImpedance	get subscribe unsubscribe	channel	HIGH_IMPEDANCE_NONE, HIGH_IMPEDANCE_OPEN
Amplified Output Input Meter	inputLevel	get subscribe unsubscribe	channel	-100.0 - 36.0 dB
Amplified Output Invert	invert	get set toggle	channel	false, true
Amplified Output Level	level	get set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
Amplified Output Limiter Attenuation	limiterAttenuation	get subscribe unsubscribe	channel	LIMITER_ATTENUATION_NONE, LIMITER_ATTENUATION_LIMITING, LIMITER_ATTENUATION_CLIPPING
Amplified Output Limiter Attenuation Level	limiterAttenuationLevel	get subscribe unsubscribe	channel	0.0 - 24.0 dB
Amplified Output	limiterEnable	get set toggle	channel	false, true

Limiter Enable				
Amplified Output Low Impedance Indicator	lowImpedance	get subscribe unsubscribe	channel	LOW_IMPEDANCE_NONE, LOW_IMPEDANCE_LOWZ, LOW_IMPEDANCE_SHORT
Amplified Output Level Max	maxLevel	get set increment decrement	channel	minLevel - 0.0 dB
Amplified Output Level Min	minLevel	get set increment decrement	channel	-100.0 - maxLevel dB
Amplified Output Mute	mute	get set toggle subscribe unsubscribe	channel	false, true
Amplified Output Current	outputCurrentLevel	get subscribe unsubscribe	channel	any value for Arms
Amplified Output Voltage	outputVoltageLevel	get subscribe unsubscribe	channel	any value for Vrms
Selected Time	selectedTime	get subscribe unsubscribe		0 - 2147483647 s
Amplified Output Standby Threshold	standbyThreshold	get set increment decrement	channel	-100.0 - 0.0 dB
Amplified Output Thermal Fault Indicator	thermalFault	get subscribe unsubscribe	channel	false, true
Amplified Output Thermal Warning Indicator	thermalWarning	get subscribe unsubscribe	channel	false, true

AV Input Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
AVInput1	get	embeddedAudioPresents	1

Example
AVInput1 get embeddedAudioPresents 1 +OK "value":[false false false false false false false false]

Attribute Description	Attribute Code	Command	Indexes	Value Range
Active Deinterlace Mode	activeDeinterlace	get subscribe unsubscribe	AV channel	false, true
Active Video Source	activeVideoSource	get subscribe unsubscribe	AV channel	VIDEO_SOURCE_HDMI, VIDEO_SOURCE_DISPLAYPORT
Total bandwidth allocated for all audio &	allocatedBandwidth	get subscribe unsubscribe	AV channel, AV auxilliary port	0.0 - 10.0Gbps

video talker streams				
Auxilliary Audio Delay	auxDelay	get set increment decrement	AV channel	0 - 64 ms
Auxilliary Audio Gain	auxGain	get set increment decrement	AV channel, AV auxilliary port	0 - 66dB in 6dB increments
Auxilliary Audio Invert	auxInvert	get set toggle	AV channel, AV auxilliary port	false, true
Auxilliary Audio Level	auxLevel	get set increment decrement	AV channel, AV auxilliary port	auxMinLevel - auxMaxLevel dB
Auxilliary Audio Max Level	auxMaxLevel	get set increment decrement	AV channel, AV auxilliary port	auxMinLevel - 12.0dB
Auxilliary Audio Min Level	auxMinLevel	get set increment decrement	AV channel, AV auxilliary port	-100.0 - auxMaxLevel dB
Auxilliary Audio Mute	auxMute	get set toggle	AV channel, AV auxilliary port	false, true
Auxilliary Audio Peak Occurring	auxPeak	get subscribe unsubscribe	AV channel, AV auxilliary port	false, true
All Auxilliary Audio Peaks	auxPeaks	get subscribe unsubscribe		
Auxilliary Audio Phantom Power On	auxPhantomPower	get set toggle	AV channel, AV auxilliary port	false, true

Auxilliary Audio Port Type	auxPortType	get	AV channel	MONO_PORT, STEREO_PORT
Bandwidth Limit	bandwidthLimit	get	AV channel	0.0 - 10.0 Gbps
Current bandwidth used	currentBandwidth	get subscribe unsubscribe	AV channel index	0.0 - 10.0 Gbps
Deinterlace Input Mode	deInterlace	get set	AV channel	Auto, Off
Embedded Audio Mute	embeddedAudioMute	get set toggle subscribe unsubscribe	AV channel	false, true
Embedded Audio Present Meters	embeddedAudioPresents	get subscribe unsubscribe	AV channel	
Signal Presence Threshold for Embedded Audio Signal Present Meters	embeddedAudioThreshold	get set increment decrement	AV channel	-64 - 24.0dBu
Input Device Connection State	inputDeviceConnected	get subscribe unsubscribe	AV channel	DEVICE_CONNECTED_NONE, DEVICE_CONNECTED_HDMI, DEVICE_CONNECTED_DISPLAYPORT, DEVICE_CONNECTED_BOTH
Currently reserved required network bandwidth	maxRequiredBandwidth	get subscribe unsubscribe	AV channel	0-10.0Gbps
Negotiated Input Frame Rate	negotiatedInputFrameRate	get subscribe unsubscribe	AV channel	0-60Hz
Negotiated Input Resolution	negotiatedInputResolution	get subscribe unsubscribe	AV channel	[0,0]-[4096,2160]
Current network	networkInterfaceType	get subscribe	AV channel	0-10.0Gbps

interface speed		unsubscribe		
Auxilliary Audio Port Count	numAuxPorts	get	AV channel	2 - 2
AV Channel Count	numAVChannels	get		1 - 1
Outgoing Frame Rate	outgoingFrameRate	get subscribe unsubscribe	AV channel index	0-60Hz
Outgoing Resolution	outgoingResolution	get subscribe unsubscribe	AV channel	[0,0]-[4096,2160]
Test Pattern Selection	testPattern	get set	AV channel	Off, ColorBar, Grid, HDMI420, JpegSafe
Video Bandwidth Parameters	videoBandwidthConfig	get set	AV channel	[resMax, frameRate, compressionFactor] -or- {"resMax":resMax "frameRate":frameRate "compressionFactor":compressionFactor} resMax:[r4096x2160, r3840x2160, r2560x1600, r1920x1200, r1920x1080, r1280x720, r800x600, r1280x800] frameRate:[fr60Hz, fr30Hz, fr15Hz] compressionFactor:[0-19] *0 == no compression* *1 == 2:1* *...* *19 == 20:1* *20 => out of range* e.g. [r1280x720, fr60Hz, 15]
Video Freeze	videoFreeze	get set toggle	AV channel	false, true
Video Mute	videoMute	get set toggle subscribe unsubscribe	AV channel	false, true

Video Source Format Selection	videoSource	get set	AV channel	VIDEO_SOURCE_HDMI, VIDEO_SOURCE_DISPLAYPORT

AV Output Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
AVOutput1	get	embeddedAudioPresents	1

Example
AVOutput1 get embeddedAudioPresents 1 +OK "value":[false false false false false false false false]

Attribute Description	Attribute Code	Command	Indexes	Value Range
Total bandwidth allocated for all audio & video talker streams	allocatedBandwidth	get subscribe unsubscribe	AV channel	0.0 - 10.0Gbps
Auxilliary Audio Delay	auxDelay	get set increment decrement	AV channel	0 - 64 ms

Auxilliary Audio Full Scale	auxFullScale	get set increment decrement	AV channel, AV auxilliary port	-31, 0, 6, 12, 18 or 24
Auxilliary Audio Invert	auxInvert	get set toggle	AV channel, AV auxilliary port	false, true
Auxilliary Audio Level	auxLevel	get set increment decrement	AV channel, AV auxilliary port	auxMinLevel - auxMaxLevel dB
Auxilliary Audio Max Level	auxMaxLevel	get set increment decrement	AV channel, AV auxilliary port	auxMinLevel - 0dB
Auxilliary Audio Min Level	auxMinLevel	get set increment decrement	AV channel, AV auxilliary port	-100.0 - auxMaxLevel dB
Auxilliary Audio Mute	auxMute	get set toggle	AV channel, AV auxilliary port	false, true
Auxilliary Audio Port Type	auxPortType	get	AV channel	MONO_PORT, STEREO_PORT
Current Bandwidth usage	currentBandwidth	get/subscribe/unsubscribe	AV channel	1-10Gbps
Embedded Audio Mute	embeddedAudioMute	get set toggle subscribe unsubscribe	AV channel	false, true
Embedded Audio Present Meters	embeddedAudioPresents	get subscribe unsubscribe	AV channel	
Signal Presence Threshold for Embedde	embeddedAudioThreshold	get set increment decrement	AV channel	-64 - 24.0dBu

d Audio Signal Present Meters				
Video Fill Color	fillColor	get set increment decrement	AV channel	0 - 4294967295
Incoming Frame Rate	incomingFrameRate	get subscribe unsubscribe	AV channel	0-60Hz
Incoming Resolution	incomingResolution	get subscribe unsubscribe	AV channel	[0,0]-[4096,2160]
Negotiated Output Frame Rate	negotiatedOutputFrameRate	get subscribe unsubscribe	AV channel	0-60Hz
Negotiated Output Resolution	negotiatedOutputResolution	get subscribe unsubscribe	AV channel	[0,0]-[4096,2160]
Network Interface Bandwidth	networkInterfaceType	get subscribe unsubscribe	AV channel	0-10.0Gbps
Auxilliary Audio Port Count	numAuxPorts	get	AV channel	2 - 2
AV Channel Count	numAVChannels	get		1 - 1
On Screen Display Message Duration	osdDuration	get set subscribe unsubscribe	AV channel	OSDOff, OSD5seconds, OSD15seconds, OSDOn
Output Device Connection State	outputDeviceConnected	get subscribe unsubscribe	AV channel	DEVICE_CONNECTED_NONE, DEVICE_CONNECTED_HDMI, DEVICE_CONNECTED_DISPLAYPORT, DEVICE_CONNECTED_BOTH
Test Pattern Selection	testPattern	get set subscribe unsubscribe	AV channel	Off, ColorBar, Grid, HDMI420, JpegSafe

On Screen Display Message Transition Mode	transition	get set	AV channel	FreezeAndFade, OSD, Instant
Video Freeze	videoFreeze	get set toggle	AV channel	false, true
Video Mute	videoMute	get set toggle subscribe unsubscribe	AV channel	false, true
Video Output Format	videoOutputFormat	get set	AV channel	vfEDIDPreferred, vf4096x2160p60, vf4096x2160p30, vf3840x2160p60, vf3840x2160p30, vf2560x1600p60, vf1920x1200p60, vf1920x1080p60, vf1920x1080p30, vf1280x720p60, vf800x600p60, vf4096x2160p50, vf4096x2160p25, vf3840x2160p50, vf3840x2160p25, vf1920x1080p50, vf1920x1080p25, vf1280x720p50, vf1280x800p60

Gating Auto Mixer Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
Mixer1	get	crosspoint	1

Example
Mixer 1 get crosspoint 1 Mixer2 set crosspoint 1 true

Attribute Description	Attribute Code	Command	Indexes	Value Range
Crosspoint On	crosspoint	get / set toggle	channel	false, true
Direct Output	directOutputLogic	get / set	channel	POST_GATE_PRE_NOM, POST_GATE_POST_NOM
Gate Hold Time	gateHoldTimeMs	get / set increment decrement	channel	0.0 - 6000.0 ms
Logic Output	gateLogic	get / set	channel	FOLLOWGATE, ON, OFF
Input Label	inputLabel	get / set	channel	
Input Level	inputLevel	get / set increment decrement	channel	inputMinLevel - inputMaxLevel dB

Max Input Level	inputMaxLevel	get / set increment decrement	channel	inputMinLevel - 12.0 dB
Min Input Level	inputMinLevel	get / set increment decrement	channel	-100.0 - inputMaxLevel dB
Input Mute	inputMute	get / set toggle	channel	false, true
Logic Output Invert	invert	get / set toggle	channel	false, true
Logic Outputs Follow Mic Logic	logicOutputsFollowMicLogic	get / set toggle		false, true
Channel Manual	manual	get / set toggle	channel	false, true
Mic Logic Type	micLogic	get / set		NONE, LASTHOLD, CHAN1, CHAN2, ...
Mix Output Label	mixOutputLabel	get / set		
NOM Gain Enabled	nomGainEnable	get / set toggle	channel	false, true
Open Mic Limit	nomLimit	get / set increment decrement		1 - lesser of numInputs-1 or 7
Open Mic Limit Enabled	nomLimitEnable	get / set toggle		false, true
Input Count	numInputs	get		2 - 256
Off Attenuation	offGain	get / set increment decrement	channel	-80.0 - -10.0 dB
Output Level	outputLevel	get / set increment decrement		outputMinLevel - outputMaxLevel dB
Max Output Level	outputMaxLevel	get / set increment decrement		outputMinLevel - 12.0 dB
Min Output Level	outputMinLevel	get / set increment decrement		-100.0 - outputMaxLevel dB
Output Mute	outputMute	get / set toggle		false, true

Gain Sharing Auto Mixer Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
Mixer1	get	crosspoint	1

Example
Mixer1 get crosspoint 1 Mixer2 set crosspoint 1 true

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channel Level	channelLevel	get / set increment decrement subscribe unsubscribe	channel	channelMinLevel - channelMaxLevel dB
All Channel Levels	channelLevels	get subscribe unsubscribe		
Max Channel Level	channelMaxLevel	get / set increment decrement	channel	channelMinLevel - 12.0 dB
Min Channel Level	channelMinLevel	get / set increment decrement	channel	-100.0 - channelMaxLevel dB

Channel Mute	channelMute	get / set toggle subscribe unsubscribe	channel	false, true
All Channel Mutes	channelMutes	get subscribe unsubscribe		
Crosspoint On	crosspoint	get / set toggle subscribe unsubscribe	channel	false, true
All Crosspoint States	crosspoints	get subscribe unsubscribe		
Gain Reduction	gainReduction	get subscribe unsubscribe	channel	-100.0 - 0.0 dB
All Gain Reductions	gainReductions	get subscribe unsubscribe		
Gain Response Time	gainResponseTimeMs	get / set increment decrement		1 - 100 ms
Input Label	inputLabel	get / set	channel	
Input Mute	inputMute	get / set toggle subscribe unsubscribe	channel	false, true
All Input Mutes	inputMutes	get subscribe unsubscribe		
Mic Isolation Factor	micIsolationFactor	get / set increment decrement		0.0 - 2.0
Mix Output Label	mixOutputLabel	get / set		
Input Count	numInputs	get		2 - 256
Output Level	outputLevel	get / set increment decrement subscribe unsubscribe		outputMinLevel - outputMaxLevel dB
Max Output Level	outputMaxLevel	get / set increment decrement		outputMinLevel - 12.0 dB
Min Output Level	outputMinLevel	get / set increment decrement		-100.0 - outputMaxLevel dB

Output Mute	outputMute	get / set toggle subscribe unsubscribe		false, true

Standard Mixer Bock

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Index	Value
Mixer1	set	crosspoint	1	1	true

Result: Sets Mixer1 Crosspoint of Input 1 and Output 1 to on.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Crosspoint On	crosspoint	get / set toggle		false, true
All Crosspoints	crosspointAll	set toggle		false, true
Crosspoint Column	crosspointColumn	set toggle	output	false, true
Crosspoint Diagonal	crosspointDiagonal	set toggle	input, output	false, true
Crosspoint Row	crosspointRow	set toggle	input	false, true
Input Label	inputLabel	get set	input	name
Input Level	inputLevel	get / set increment decrement	input	inputMinLevel - inputMaxLevel dB

Max Input Level	inputMaxLevel	get / set increment decrement	input	inputMinLevel - 12.0 dB
Min Input Level	inputMinLevel	get / set increment decrement	input	-100.0 - inputMaxLevel dB
Input Mute	inputMute	get / set toggle	input	false, true
Input Count	numInputs	get		2 - 256
Output Count	numOutputs	get		1 - 256
Output Label	outputLabel	get / set	output	name
Output Level	outputLevel	get / set increment decrement	output	outputMinLevel - outputMaxLevel dB
Max Output Level	outputMaxLevel	get / set increment decrement	output	outputMinLevel - 12.0 dB
Min Output Level	outputMinLevel	get / set increment decrement	output	-100.0 - outputMaxLevel dB
Output Mute	outputMute	get / set toggle	output	false, true

Matrix Mixer Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Index	Value
Mixer1	set	crosspointLevelState	1	1	true

Example
Mixer1 set crosspointLevelState 1 1 true +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Crosspoint Delay	crosspointDelay	get / set increment decrement	input, output	0.0 - 250.0 ms
Crosspoint Delay On	crosspointDelayState	get / set toggle	input, output	false, true
All Delay Crosspoints	crosspointDelayStateAll	set toggle		false, true
Delay Crosspoint Column	crosspointDelayStateColumn	set toggle	output	false, true
Delay Crosspoint Diagonal	crosspointDelayStateDiagonal	set toggle	input, output	false, true
Delay Crosspoint Row	crosspointDelayStateRow	set toggle	input	false, true

Crosspoint Level	crosspointLevel	get / set increment decrement	input, output	-100.0 - 0.0 dB
Crosspoint On	crosspointLevelState	get / set toggle	input, output	false, true
All Crosspoints	crosspointLevelStateAll	set toggle		false, true
Crosspoint Column	crosspointLevelStateColumn	set toggle	output	false, true
Crosspoint Diagonal	crosspointLevelStateDiagonal	set toggle	input, output	false, true
Crosspoint Row	crosspointLevelStateRow	set toggle	input	false, true
Delay Enabled	delayEnabled	get		false, true
Input Label	inputLabel	get / set	input	
Input Level	inputLevel	get / set increment decrement	input	inputMinLevel - inputMaxLevel dB
Max Input Level	inputMaxLevel	get / set increment decrement	input	inputMinLevel - 12.0 dB
Min Input Level	inputMinLevel	get / set increment decrement	input	-100.0 - inputMaxLevel dB
Input Mute	inputMute	get / set toggle	input	false, true
Input Count	numInputs	get		2 - 256
Output Count	numOutputs	get		1 - 256
Output Label	outputLabel	get set	output	
Output Level	outputLevel	get / set increment decrement	output	outputMinLevel - outputMaxLevel dB
Max Output Level	outputMaxLevel	get / set increment decrement	output	outputMinLevel - 12.0 dB
Min Output Level	outputMinLevel	get / set increment decrement	output	-100.0 - outputMaxLevel dB
Output Mute	outputMute	get / set toggle	output	false, true

Auto Mixer Combiner Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
AutoMixerCombiner1	get	nomLimit	inGroup:	1

Example
AutoMixerCombiner1 get nomLimit inGroup:1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Input Group	inputGroup	get / set increment decrement	channel	0 - channel count
Last Mic Hold Enabled	lastMicHoldEnable	get / set toggle	inGroup:	false, true
Open Mic Limit	nomLimit	get / set increment decrement	inGroup:	1 - 7
Open Mic Limit Enabled	nomLimitEnable	get / set toggle	inGroup:	false, true

Room Combiner Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
RoomCombiner1	get	wallState	1

Example
RoomCombiner1 get wallState 1 RoomCombiner1 set wallState 1 true

Attribute Description	Attribute Code	Command	Indexes	Value Range
Room Group	group	get / set increment decrement	room	0 - room count
Last Mic Hold Enabled	lastMicHoldEnable	get / set toggle		false, true
Input Level	levelIn	get / set increment decrement	room	levelInMin - levelInMax dB
Max Input Level	levelInMax	get / set increment decrement	room	levelInMin - 12.0 dB
Min Input Level	levelInMin	get / set increment decrement	room	-100.0 - levelInMax dB

Output Level	levelOut	get / set increment decrement subscribe unsubscribe	room	levelOutMin - levelOutMax dB
Max Output Level	levelOutMax	get / set increment decrement	room	levelOutMin - 12.0 dB
Min Output Level	levelOutMin	get / set increment decrement	room	-100.0 - levelOutMax dB
Source Level	levelSource	get / set increment decrement	room	levelSourceMin - levelSourceMax dB
Max Source Level	levelSourceMax	get / set increment decrement	room	levelSourceMin - 12.0 dB
Min Source Level	levelSourceMin	get / set increment decrement	room	-100.0 - levelSourceMax dB
Input Mute	muteIn	get / set toggle	room	false, true
Output Mute	muteOut	get / set toggle	room	false, true
Source Mute	muteSource	get / set toggle	room	false, true
Open Mic Limit	nomLimit	get / set increment decrement		1 - 7
Open Mic Limit Enabled	nomLimitEnable	get / set toggle		false, true
Wall Room Precedence	preferredRoom	get / set increment decrement	Wall Number	A room index
Room Label	roomLabel	get / set	room	
Source Label	sourceLabel	get / set	source	
Source Selection	sourceSelection	get / set increment decrement	room	0 - 4
Wall Closed	wallState	get / set toggle	wall number	false, true

Equalizer Blocks

Parametric Equalizer Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
ParametricEQ1	get	numbands

Example
ParametricEQ1 get numbands ParametricEQ1 set gain 1 5.0 ParametricEQ1 set bandwidth 1 0.5

Attribute Description	Attribute Code	Command	Indexes	Value Range
Bandwidth	bandwidth	get / set increment decrement	band	0.01 - 4.0 oct
Bypass	bypass	get / set toggle	band	false, true
Bypass All	bypassAll	get / set toggle		false, true

Center Frequency	frequency	get / set increment decrement	band	20.0 - 20000.0 Hz
Frequency & Gain	frequencyGain	get / set	band	[Frequency, gain] Frequency in Hz
Band Gain	gain	get / set increment decrement	band	minGain - maxGain dB
Band Max Gain	maxGain	get / set increment decrement	band	0.0 - 15.0 dB
Band Min Gain	minGain	get / set increment decrement	band	-30.0 - 0.0 dB
Band Count	numBands	get		1 - 16

Graphic Equalizer Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
GraphicEQ1	get	gain	25

Attribute Description	Attribute Code	Command	Indexes	Value Range
Bypass Band	bypass	get / set toggle	band	false, true
Bypass All	bypassAll	get / set toggle		false, true
Band Gain	gain	get / set increment decrement	band	minGain - maxGain dB
Band Max Gain	maxGain	get / set increment decrement	band	0.0 - 15.0 dB
Band Min Gain	minGain	get / set increment decrement	band	-30.0 - 0.0 dB
Band Count	numBands	get		10, 15, or 31

Band Number	Frequency 1/3 Octave (HZ)	Frequency 2/3 Octave (HZ)	Frequency 1 Octave (HZ)
1	20	25	31.5
2	25	40	63

3	31.5	63	125
4	40	100	250
5	50	160	500
6	63	250	1000
7	80	400	2000
8	100	630	4000
9	125	1000	8000
10	160	1600	16000
11	200	2500	
12	250	4000	
13	315	6300	
14	400	10000	
15	500	16000	
16	630		
17	800		
18	1000		
19	1250		
20	1600		
21	2000		
22	2500		
23	3150		
24	4000		
25	5000		
26	6300		
27	8000		
28	10000		
29	12500		
30	16000		
31	20000		

Feedback Suppressor Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
FeedbackSuppressor1	set	fixedAll	true

Attribute Description	Attribute Code	Command	Indexes	Value Range
Bandwidth	bandwidth	get / set increment decrement	band	0.01 - 4.0 oct
Bypass	bypass	get / set toggle	band	false, true
Bypass All	bypassAll	get / set toggle		false, true
All Bands Fixed	fixedAll	get / set toggle		false, true
Floating Band Max Depth	floatingBandMaxDepth	get / set increment decrement		-20.0 - 0.0
Floating Band Width	floatingBandWidth	get / set		NARROWBAND, WIDEBAND
Center Frequency	frequency	get / set increment decrement	band	20.0 - 20000.0 Hz
Frequency & Gain	frequencyGain	get / set	band	[Frequency, gain] Frequency in Hz

Band Gain	gain	get / set increment decrement	band	-30.0 - 0.0 dB
Band Fixed	isFixed	get / set toggle	band	false, true
Band Count	numBands	get		1 - 16
Reset Floating Bands	resetFloatingBands	set		Value ignored

Filter Blocks

Pass Filter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
PassFilter1	set	frequency	100

Example
PassFilter1 set frequency 100 +OK

Filter Type and Slope Values must be specified within square brackets -filter type must be specified before slope and both parameters MUST be included.

Example
PassFilter1 set filterTypeSlope [LINKWITZ_RILEY 24] +OK

The following format is also acceptable. Since "type" and "slope" are clearly denoted within the {braces}, the [value] variables can be provided in either order.

Example
PassFilter1 set filterTypeSlope {"type":LINKWITZ_RILEY "slope":24} +OK

Attribute Description	Attribute Code	Command	Value Range
Bypass	bypass	get / set toggle	false, true
Filter Type	filterType	get	BUTTERWORTH, LINKWITZ_RILEY, BESSEL
Filter Type & Slope	filterTypeSlope	get / set	[Type, slope] or {"type":Type "slope":slope} Type:BUTTERWORTH, Slope:6,12,18,24,30,36,42,48 Type: LINKWITZ_RILEY, Slope: 12, 24,36,48 Type: BESSEL Slope:6,12,18,24,30,36,42,48
Cutoff Frequency	frequency	get / set increment decrement	20.0 - 20000.0 Hz
Max Slope	maxSlope	get	Always 48 dB/oct
Filter Slope	slope	get	Linkwitz/Riley: 12, 24,36,48 Butterworth: 6,12,18,24,30,36,42,48 Bessel: 6,12,18,24,30,36,42,48

Shelf Filter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
ShelfFilter1	get	frequency

Example
ShelfFilter1 get frequency +OK "value":6350.116211

Attribute Description	Attribute Code	Command	Value Range
Bypass	bypass	get / set toggle	false, true
Cutoff Frequency	frequency	get / set increment decrement	20.0 - 20000.0 Hz
Gain	gain	get / set increment decrement	-27.0 - 9.0 dB

All Pass Filter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
AllPassFilter1	get	frequency	1

Example
AllPassFilter1 get frequency

Attribute Description	Attribute Code	Command	Indexes	Value Range
Bandwidth	bandwidth	get / set increment decrement	band	0.01 - 4.0 oct
Bypass	bypass	get / set toggle	band	false, true
Bypass All	bypassAll	get / set toggle		false, true
Center Frequency	frequency	get / set increment decrement	band	20.0 - 20000.0 Hz
Band Enabled	isUsed	get / set toggle	band	false, true
Band Count	numBands	get		1 - 16

Uber Filter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
UberFilter1	get	frequency	1

Example
UberFilter1 get frequency 1 UberFilter1 set frequency 1 4000

Attribute Description	Attribute Code	Command	Indexes	Value Range
Band Type	bandType	get	band	NONE, PARAMETRIC_EQ, PASS, SHELF
Bandwidth	bandwidth	get / set increment decrement	band	0.01 - 4.0 oct
Band Bypass	bypass	get / set toggle	band	false, true
Bypass All	bypassAll	get / set toggle		false, true
Band Frequency	frequency	get / set increment decrement	band	20.0 - 20000.0 Hz

Frequency & Gain	frequencyGain	get / set	band	[Frequency, gain] Must be a parametric or shelf Frequency = value in Hz
Band Gain	gain	get / set increment decrement	band	-27.0 - 9.0dB for shelf bands -30.0 - 15.0 dB for parametric bands
Locked Band Type	locked	get	band	false, true
Max Slope	maxSlope	get		Always 48 dB/oct
Band Count	numBands	get		1 - 16
Pass Filter Type	passFilterType	get	band	BUTTERWORTH, LINKWITZ_RILEY, BESSEL
Pass Filter Type & Slope	passFilterTypeSlope	get / set	band	[Type, slope] or {"type":Type "slope":slope} Type:BUTTERWORTH, Slope:6,12,18,24,30,36,42,48 Type: LINKWITZ_RILEY, Slope: 12, 24,36,48 Type: BESSEL Slope:6,12,18,24,30,36,42,48
Filter Slope	slope	get	band	Linkwitz/Riley: 12, 24,36,48 Butterworth: 6,12,18,24,30,36,42,48 Bessel: 6,12,18,24,30,36,42,48

Crossover Blocks

Crossover Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Crossover1	toggle	synchronize

- **band** is indexed by number from high to low, so in a four-way crossover high=1, mid high=2, low mid =3 and low=4,
- **filter** is indexed by number. 1 is the high cutoff frequency for each band while 2 is the low.

Filter Type and Slope Values must be specified within square brackets -filter type must be specified before slope and both parameters MUST be included.

Example
Crossover1 set filterTypeSlope 1 1 [LINKWITZ_RILEY 24] +OK

The following format is also acceptable. Since "type" and "slope" are clearly denoted within the {braces}, the [value] variables can be provided in either order.

Example
Crossover1 set filterTypeSlope 1 1 {"type":LINKWITZ_RILEY "slope":24} +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Filter Type	filterType	get	band, filter	BUTTERWORTH, LINKWITZ_RILEY, BESSEL
Filter Type & Slope	filterTypeSlope	get / set	band, filter	[Type, slope] or {"type":Type "slope":slope} Type:BUTTERWORTH, Slope:6,12,18,24,30,36,42,48 Type: LINKWITZ_RILEY, Slope: 12, 24,36,48 Type: BESSEL Slope:6,12,18,24,30,36,42,48
Cutoff Frequency	frequency	get / set increment decrement	band, filter	20.0 - 20000.0 Hz
Input Level	inputLevel	get / set increment decrement		inputMinLevel - inputMaxLevel dB
Input Mute	inputMute	get / set toggle		false, true
Band Count	numBands	get		2 - 4
Band Filter Count	numFilters	get	band	1 - 2
Output Invert	outputInvert	get / set toggle	band	false, true
Output Level	outputLevel	get / set increment decrement	band	outputMinLevel - outputMaxLevel dB
Output Mute	outputMute	get / set toggle	band	false, true
Filter Slope	slope	get	band, filter	Linkwitz/Riley: 12, 24,36,48 Butterworth: 6,12,18,24,30,36,42,48 Bessel: 6,12,18,24,30,36,42,48
Synchronize Bands	synchronize	get / set toggle		false, true

Dynamic Blocks

Leveler Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Leveler1	get	threshold

Example
Leveler1 get threshold Leveler1 set threshold -40

Attribute Description	Attribute Code	Command	Value Range
Bypass	bypass	get / set toggle	false, true
Gain Reduction	gainReductionLevel	get subscribe unsubscribe	-152.0 - 0.0 dB
Label	label	get / set	
Response Time	responseTime	get / set increment decrement	0.1 - 40000.0 ms
Threshold	threshold	get / set increment decrement	-60.0 up to +24.0 dBu

Compressor Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
GR Levels	allGainReduction	get subscribe unsubscribe		
Attack Time	attackTime	get / set increment decrement		1.0 - 2000.0 ms
Bypass	bypass	get / set toggle		false, true
Gain Reduction	gainReduction	get subscribe unsubscribe	channel	1 - 32
Makeup Gain	makeupGain	get / set increment decrement		0.0 - 12.0 dB
Channel Count	numChannels	get		1 - 32
Release Time	releaseTime	get / set increment decrement		5.0 - 10000.0 ms

Peak Limiter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Active LED	activeLED	get subscribe unsubscribe	channel	false, true
All Active LEDs	allActiveLEDs	get subscribe unsubscribe		
Bypass	bypass	get /set toggle		false, true
Channel Count	numChannels	get		1 - 32
Release Time	releaseTime	get /set increment decrement		1.0 - 10000.0 ms
Peak Threshold	threshold	get / set increment decrement		-20.0 - 28.0 dB

Ducker Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Ducker1	get	attackTime

Example
Ducker1 get attackTime

Attribute	Attribute Code	Commands	Value Range
Attack Time	attackTime	get / set increment decrement	0.1 - 2000.0 ms
Bypass	bypass	get / set toggle	false, true
Ducking Level	duckingLevel	get / set increment decrement	-100.0 - 0.0 dB
Input Level	inputLevel	get / set increment decrement	-100.0 - 12.0 dB
Input Mute	inputMute	get / set toggle	false, true
Logic In Enabled	logicInEnable	get / set toggle	false, true
Logic In Inverted	logicInInvert	get / set toggle	false, true

Logic Out Enabled	logicOutEnable	get / set toggle	false, true
Logic Out Inverted	logicOutInvert	get / set toggle	false, true
Max Input Level	maxInputLevel	get / set increment decrement	minInputLevel - 12.0 dB
Min Input Level	minInputLevel	get / set increment decrement	-100.0 - maxInputLevel dB
Mix Sense Enabled	mixSense	get/set toggle	false, true
Release Time	releaseTime	get / set increment decrement	0.1 - 40000.0 ms
Sense Level	senseLevel	get / set increment decrement	-100.0 - 12.0 dB
Sense Mute	senseMute	get / set toggle	false, true
Threshold	threshold	get / set increment decrement	-60.0 - 24.0 dBu

Noise Gate Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
NoiseGate1	get	threshold

Example
NoiseGate1 get threshold NoiseGate1 set threshold -40

Attribute Description	Attribute Code	Command	Value Range
Attack Time	attackTime	get / set increment decrement	0.1 - 2000.0 ms
Bypass	bypass	get /set toggle	false, true
Gain Reduction	gainReductionLevel	get subscribe unsubscribe	-152.0 - 0.0 dB
Label	label	get / set	
Release Time	releaseTime	get /set increment decrement	0.1 - 40000.0 ms
Threshold	threshold	get / set increment decrement	-60.0 - 24.0 dBu

AGC Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
AGC1	get	speech

Example
AGC1 get speech AGC set speech true

Attribute Description	Attribute Code	Command	Value Range
AGC Active	agcActive	get	false, true
Bypass	bypass	get / set toggle	false, true
Gain Level	gainLevel	get	-30.0 - 30.0 dB
Hold Time	holdTime	get / set increment decrement	0 - 350000 s
Input Level	inputLevel	get	-100.0 - 36.0 dBu
Limiter On	limiter	get / set toggle	false, true
Limiter Active	limiterActive	get	false, true

Max Attenuation	maxAtten	get / set increment decrement	0.0 - 30.0 dB
Max Gain	maxGain	get / set increment decrement	0.0 - 30.0 dB
Max Gain Adj. Rate	maxGainRate	get / set increment decrement	0.0 - 15.0 dB/s
All Meter States	meters	get subscribe unsubscribe	
Min SNR	minSnr	get / set increment decrement	10.0 - 50.0 dB
Min Threshold	minThreshold	get / set increment decrement	-30.0 - 20.0 dBu (Max Value equal to Target Level)
Noise Floor Level	noiseFloorLevel	get	-100.0 - 36.0 dBu
Side Chain Level	sideChainLevel	get	-100.0 - 36.0 dBu
SNR Level	snrLevel	get	0.0 - 136.0 dB
Speech On	speech	get / set toggle	false, true
Target Level	targetLevel	get / set increment decrement	-20.0 - 20.0 dB

Router Blocks

Router Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
Router1	set	input	1	1

Example
Router1 get input 1 +OK "value":0 Router1 set input 1 1 +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Selected Input	input	get / set increment decrement	output	Input index or 0 for no selected input
Input Label	inputLabel	get/set	input	Any alphanumeric string
Input Count	numInputs	get		1 - 256
Output Count	numOutputs	get		1 - 256
Output Label	outputLabel	get/set	output	Any alphanumeric string

Source Selector Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
SourceSelector1	set	sourceSelection	1

Example
SourceSelector1 set sourceSelection 1 +OK

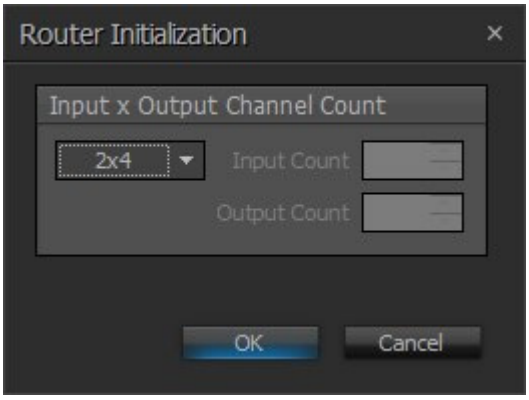
Attribute Description	Attribute Code	Command	Indexes	Value Range
Label	label	get / set	source	
Input Count	numInputs	get		2 - 64
Output Count	numOutputs	get		1 - 2
Source Count	numSources	get		2 - 32
Output Level	outputLevel	get / set increment decrement subscribe unsubscribe		outputMinLevel - outputMaxLevel dB
Max Output Level	outputMaxLevel	get / set increment decrement		outputMinLevel - 12.0 dB
Min Output Level	outputMinLevel	get / set increment decrement		-100.0 - outputMaxLevel dB

Output Mute	outputMute	get / set toggle subscribe unsubscribe		false, true
Source Level	sourceLevel	get / set increment decrement subscribe unsubscribe	source	sourceMinLevel - sourceMaxLevel dB
Max Source Level	sourceMaxLevel	get / set increment decrement	source	sourceMinLevel - 12.0 dB
Min Source Level	sourceMinLevel	get / set increment decrement	source	-100.0 - sourceMaxLevel dB
Source Selection	sourceSelection	get / set increment decrement subscribe unsubscribe		Source index or 0 for none
Stereo Enabled	stereoEnable	get		false, true

AV Router

Routers allow each input to be assigned to multiple outputs via **In** / **Out**. However, each output allows only one input assigned at a time. Therefore, Routers behave like a series of individual distribution amplifiers.

Initialization Dialog



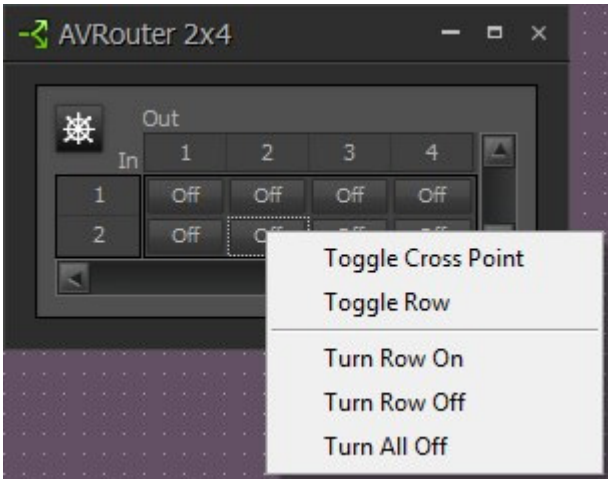
DSP Block Representation

AVRouter

Control Dialog

 **Birds Eye View** : This is used when large Routers are required as a means of navigating around the available crosspoints

Right-clicking over any cross point will provide a menu of additional options



Delay Blocks

Audio Delay Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Delay1	get	unitsDelay

Example
Delay1 get unitsDelay +OK "value":{"units":MILLISECOND "delay":47.3}

Attribute Description	Attribute Code	Command	Value Range
Bypass	bypass	get / set toggle	false, true
Delay Value	delay	get	0 - maxDelay ms converted to selected units
Max Delay	maxDelay	get	5, 10, 50, 100, 500, 1000, or 2000 ms
Delay Units	units	get	MILLISECOND, CENTIMETER, METER, INCH, FOOT
Delay Setting	unitsDelay	get / set	[unit delay] or {"units":units "delay":delay}

Level Control Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Level1	get	levels

Example
Level1 get numChannels +OK "value":4 Level1 get levels +OK "value":[0.000000 0.000000 0.000000 0.000000]

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channels Ganged	ganged	get		false, true
Label	label	get / set	channel	
Level	level	get / set increment decrement subscribe unsubscribe	channel	minLevel - maxLevel dB
All Levels	levels	get subscribe unsubscribe		

Max Level	maxLevel	get / set increment decrement	channel	minLevel - 12.0 dB
Min Level	minLevel	get / set increment decrement	channel	-100.0 - maxLevel dB
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 32
Ramp Interval	rampInterval	get / set increment decrement	channel	250.0 - 30000.0 ms in 250.0 ms increments
Ramp Step	rampStep	get / set increment decrement	channel	1.0 - 15.0 dB
Use Ramping	useRamping	get		false, true

Invert Control Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Invert1	get	inverts

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channels Ganged	ganged	get		false, true
Invert	invert	get / set toggle subscribe unsubscribe	channel	false, true
All Invert States	inverts	get subscribe unsubscribe		
Label	label	get / set	channel	
Channel Count	numChannels	get		1 - 16

Mute Control Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
Mute1	get	mutes

Example
Mute1 get numChannels +OK "value":3 Mute1 get mutes +OK "value":[false false false]

Attribute Description	Attribute Code	Command	Indexes	Value Range
Channels Ganged	ganged	get		false, true
Label	label	get / set	channel	
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 16

Preset Button Block

The Preset Button can be used to control a preset that is part of a Preset Button. Presets can also be directly accessed via TTP using the Device Service Commands

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
PresetButton1	set	preset	1	1001

Example
PresetButton1 get preset 1 +OK "value":1001 PresetButton1 set preset 1 1001 +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Preset ID	preset	get / set increment decrement	channel	ID of any preset

Command String Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
CommandString1	get	command	1

Example
CommandString1 get command 1 +OK "value":"my test string"

Example - Set Command ID and String
CommandString1 set labelCommand 1 {"label":"Hello" "command":"World"} CommandString1 set labelCommand1 ["Hello" "World"]

Attribute Description	Attribute Code	Command	Indexes	Value Range
Command String	command	get / set	channel	
Command ID	label	get / set	channel	
Command ID & String	labelCommand	get / set	channel	Set Supports the following format: {"label":"Hello" "command":"World"} ["Hello" "World"]

Network Config	networkConfig	get		
Serial Config	serialConfig	get		
Command Status	status	get subscribe unsubscribe		

HD-1 Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Speed Dial Entries	speedDialEntries	get/set/subscribe/unsubscribe		Parallel sequences of names and numbers

Dialer Block

The dialer block supports Service codes as well as Attribute codes. The Services Code defines a instruction and function for the dialer block to perform. The attribute Code defines the portion of the DSP block to be controlled such as a fader level.

Dialer Service Codes

The Following table summarizes Dialer Service Codes. Due to the nature of the service being requested they do not require specific Attribute commands (get, set, etc)

- Dialer blocks associated with STC-2 cards will always use a Call appearance of 1.
- Dialer blocks associated with SVC-2 cards currently support up to six call appearances per line, three call appearances are able to be used in a conference call. (The main call is Call appearance 1)

Inserting pauses in a **Dial** Service Code is supported by using commas between numbers. Each Comma insets a one second pause between numbers. Whenever pauses are used the number must be enclosed in "Double Quotes". See example below.

Each element of the Service Code instruction is delimited by a single space. The commands are case sensitive and upper and lower case characters are used.

The TTP string is structured in the following order:

Instance Tag Service [Index] [Value] LF

- Instance Tag: Is always required. Review the Instance Tag section for more details
- **Service:** Is always required. Review the Service section for more details.
- **Index:** Is shown in [Brackets] as may be required depending on the Service being referenced. For Dialers associated with the SVC-2 The first number is the Line which is 1 or 2 and the Call Appearance Index which is 1,2,3,4,5 or 6. For Dialers associated with the STC-2 The first number is the Line which is 1 or 2 and the Call Appearance Index which is 1.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Service being referenced. If not be required it should not be defined. Would not normally have spaces, if it does it can be defined in "Double Quotes". Can also be a numerical value. Refer to the Value section for more details.
- **LF**: A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Service Code	Index	Index	Value
Dialer1	dial	1	1	15036417287

Example - No Pauses
Dialer1 dial 1 1 15036417287

Example - With Pauses
Dialer1 dial 1 1 "1,5036417287"

Description	Service Code	Index 1	Value
Speed Dial	speedDial	Line, Call Appearance	Speed Dialer Entry

Redial	redial	Line, Call Appearance	
End	end	Line, Call Appearance	
Flash	flash	Line, Call Appearance	
Send	send	Line, Call Appearance	
Dial (Used when On Hook Only)	dial	Line, Call Appearance	Number to Dial (A String)
DTMF (Used when Off Hook only)	dtmf	Line	One number between 0 - 9, * or #
Answer	answer	Line, Call Appearance	
Conference (SVC Only)	lconf	Line, Call Appearance	
Resume (SVC Only)	resume	Line, Call Appearance	
Leave Conference (SVC Only)	leaveConf	Line, Call Appearance	
Specify call appearance (SVC Only)	callAppearance	Line, Call Appearance	
Hold (SVC Only)	hold	Line, Call Appearance	
Go Off Hook	offHook	Line, Call Appearance	
Go On Hook	onHook	Line, Call Appearance	

Dialer Attributes

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index
Dialer1	get	lastNum	1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Auto Answer	autoAnswer	get / set toggle subscribe unsubscribe	line	false, true
Call State	callState	get subscribe unsubscribe		
Display Name Label	displayNameLabel	get set		
Do Not Disturb Enabled	dndEnable	get / set toggle subscribe unsubscribe	line	false, true
Last Number Dialed	lastNum	get subscribe unsubscribe	line	
Line Label	lineLabel	get subscribe unsubscribe	line	
Line Count	numChannels	get		1 - 2
Speed Dial Label	speedDialLabel	get / set	line, speed dial entry	
Speed Dial Number	speedDialNum	get / set	line, speed dial entry	

Meter Blocks

Signal Present Meter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Index	Value
SignalPrstMeter1	subscribe	level	1	MyMeterName	500

Example
SignalPrstMeter1 subscribe level 1 MyMeterName 500 ! "publishToken":"MyMeterName" "value":-100.000000 +OK ! "publishToken":"MyMeterName" "value":-98.099998 ! "publishToken":"MyMeterName" "value":-77.800003

Attribute Description	Attribute Code	Command	Indexes	Value Range
Invert	invert	get / set toggle	channel	false, true
Label	label	get / set	channel	
Signal Level	level	get subscribe unsubscribe	channel	-100.0 - 36.0 dB
All Levels	levels	get subscribe unsubscribe		
Logic State	logicState	get	channel	false, true

Channel Count	numChannels	get		1 - 16
Off Delay	offDelay	get / set increment decrement	channel	0 - 60000 ms
On Delay	onDelay	get / set increment decrement	channel	0 - 60000 ms
Signal Present	present	get subscribe unsubscribe	channel	false, true
All Signal Indicators	presents	get subscribe unsubscribe		
Threshold	threshold	get / set increment decrement	channel	-64.0 - 30.0 dBu

Peak or RMS Meter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Index	Value
AudioMeter2	subscribe	level	3	myspecialmeter	5000

Example - To subscribe and unsubscribe to a meter.
AudioMeter2 subscribe level 3 myspecialmeter 5000 ! "publishToken":"myspecialmeter" "value":-100.000000 +OK ! "publishToken":"myspecialmeter" "value":-70.000000 ! "publishToken":"myspecialmeter" "value":-40.000000 AudioMeter2 unsubscribe level 3 myspecialmeter +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Hold Enabled	holdEnabled	get / set toggle	channel	false, true
Hold Time	holdTime	get / set increment decrement	channel	0.0 - 1000.0 ms
Hold Indefinitely	indefiniteHold	get / set toggle	channel	false, true
Label	label	get / set	channel	

Level	level	get subscribe unsubscribe	channel	-100.0 - 36.0 dB
All Levels	levels	get subscribe unsubscribe		
Channel Count	numChannels	get		1 - 32

Tone Generator Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
ToneGenerator1	set	sweepEnable	true

Attribute Description	Attribute Code	Command	Value Range
Frequency	frequency	get / set increment decrement	20.0 - 20000.0 Hz
Frequency Increment	frequencyInterval	get / set	OCTAVE_1, OCTAVE_2_3, OCTAVE_1_3, OCTAVE_1_6, OCTAVE_1_12, OCTAVE_1_24, OCTAVE_1_48, OCTAVE_1_96
Level	level	get / set increment decrement	minLevel - maxLevel dBu
Max Level	maxLevel	get / set increment decrement	minLevel - 36.0 dBu
Min Level	minLevel	get / set increment decrement	-100.0 - maxLevel dBu
Mute	mute	get / set toggle	false, true

Sweep Enabled	sweepEnable	get / set toggle	false, true
Sweep Start Frequency	sweepFrequencyStart	get / set increment decrement	20.0 - 20000.0 Hz
Sweep Stop Frequency	sweepFrequencyStop	get / set increment decrement	20.0 - 20000.0 Hz
Sweep Increment Time	timeInterval	get / set increment decrement	10 - 60000 ms

Noise Generator Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Value
NoiseGenerator1	set	mute	true

Example
NoiseGenerator1 set mute false +OK NoiseGenerator1 set level -100

Attribute Description	Attribute Code	Command	Value Range
Level	level	get / set increment decrement	minLevel - maxLevel dBu
Max Level	maxLevel	get / set increment decrement	minLevel - 36.0 dBu
Min Level	minLevel	get / set increment decrement	-100.0 - maxLevel dBu
Mute	mute	get / set toggle	false, true
Noise Type	type	get / set	WHITE, PINK

Logic Blocks

Logic State Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
LogicState1	set	state	1	true

Example
LogicState1 set state 1 true +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Label	label	get / set	channel	name
Set	state	get / set toggle	channel	false, true

Flip Flop Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
FlipFlop1	set	state	1	true

Example
FipFlop1 set state 1 true +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Label	label	get / set	channel	
Set	state	get / set toggle	channel	false, true

Logic Delay Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code	Index	Value
LogicDelay1	set	offDelayMs	1	1000

Example
LogicDelay1 set offDelayMs 1 1000 +OK

Attribute Description	Attribute Code	Command	Indexes	Value Range
Bypass	bypass	get / set toggle	channel	false, true
Off Delay	offDelayMs	get / set increment decrement	channel	0 - 60000 ms
On Delay	onDelayMs	get / set increment decrement	channel	0 - 60000 ms

Logic Meter Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
LogicMeter1	get	states

Example
LogicMeter1 get states LogicMeter1 subscribe state 1 mylogicstate 500

Attribute Description	Attribute Code	Command	Indexes	Value Range
Label	label	get / set	channel	
State	state	get subscribe unsubscribe	channel	false, true
All States	states	get subscribe unsubscribe		

Logic Input Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
LogicInput1	get	numInputs

Example
LogicInput1 get numInputs

Attribute Description	Attribute Code	Command	Indexes	Value Range
Invert	invert	get / set toggle	channel	false, true
Label	label	get / set	channel	
Input Count	numInputs	get		1 - 16

Logic Output Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
LogicOutput1	get	numOutputs

Example
LogicOutput1 get numOutputs

Attribute Description	Attribute Code	Command	Indexes	Value Range
Invert	invert	get / set toggle	channel	false, true
Label	label	get / set	channel	
Output Count	numOutputs	get		1 - 16
Powered Outputs Enabled	power	get		false, true

Control Voltage Block

Please refer to the TTP Overview section for more details on the controlling Tesira devices using the TTP protocol.

Each element of the command instruction is delimited by one or more spaces. The commands are case sensitive and upper and lower case characters are used.

The TTP string to adjust a DSP object attribute is structured in the following order:

Instance_Tag Command Attribute [Index] [Value] LF

- **Instance Tag:** Is always required. Review the Instance Tag section for more details.
- **Command**: Is always required. Review the Command section for more details.
- **Attribute:** Is always required. Review the Attribute section for more details.
- **[Index]**: Is shown in [Brackets] as may be required depending on the Attribute being referenced. If not required it should not be defined. Depending on the Attribute, it can be made up of one or more indexes. Please refer to the Index section for more details.
- **[Value]**: Is shown in [Brackets] as may be required depending on the Command or Attribute being referenced. If not be required it should not be defined. The Value would not normally have spaces, if it does it can be defined in "double quotes". It can also be a numerical value. Please refer to the **Value** section for more details.
- **LF:** A Line feed or Carriage Return is used to define the end of the command.

Instance Tag	Command	Attribute Code
ControlVoltage1	get	numchannels

Example
ControlVoltage1 get numChannels +OK "value":1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Label	label	get / set	channel	
Channel Count	numChannels	get		1 - 4