TESIRA®

Printed Manual

Rev 2.3.0

December 2014

Table of Contents

Introduction

Features

TESIRA

- Is a Digital Audio Platform, which provides distributed digital audio, signal processing, and control.
- Is a networked, decentralized audio system, which is easy to configure & program.
- Allows the installer to quickly & accurately define the exact sound system required for each job.
- Is software programmable, easily expandable, and remotely controllable.
- Utilizes Audio Video Bridging (AVB), CobraNet® and Ethernet for enhanced system networking and control.
- Is a completely customizable, yet cost-effective, solution for sound system design.
- Is covered by a five-year warranty.

ALGORITHMS

- Mixers: standard, automatic, matrix, combiners
- Equalizers: graphic, parametric, feedback
- Filters: High Pass Filters (HPF) , Low Pass Filters (LPF), high shelving, low shelving, all-pass
- Crossovers: 2-way, 3-way, 4-way
- Dynamics: levelers, compressors, limiters, duckers, gates, ambient noise compensators
- Routers: 2x1 to 56x56
- Source Selectors: 2x1 to 16x1 with logic inputs and outputs
- Delays: 0 to 2,000ms
- Controls: levels, mutes, presets, logic, command strings
- Meters: signal present, peak reading, RMS reading
- Generators: single tone, sweep, pink-noise, white-noise
- Diagnostics: transfer function

SERVER HARDWARE

- Ethernet communications for software control and configuration
- Support of TCP/IP, UDP and ICMP (ping) networking standards
- Battery backed real-time clock and calendar
- Dual RS-232 serial ports - Port 1 for controlling other devices, Port 2 for connection to 3rd party control devices
- Compatible with all other CobraNet compliant devices.
- Compatible with all other AVB compliant devices.
- Works with standard Ethernet switches
- Multiple system-wide presets and current settings stored in flash memory
- Layout drawing information stored in flash memory

SOFTWARE

Minimum Software requirements -

- Windows® 7 Professional, SP1, 32-bit or 64-bit.
- Windows® 8 Professional 32-bit or 64-bit.

    - 1GB RAM
    - Pentium® 4, 2.4GHz or faster
    - 1280x1024 screen resolution (recommended)

*Virtual machines or tablets running Windows are not supported at this time.

- Fully dockable Menu and Toolbar support
- Bird's-eye viewer for easy panning and zooming of large layout files
- Fully customizable Processing Library bar for storing default and custom DSP objects
- Workspace state saved at program shutdown, including current documents, Toolbar,
- Catalog & Birds eye viewer states
- User configurable data file and Processing Library catalog
- Simple object attribute control: colors, line widths, hatching, text font, size, style & alignment, border widths, etc.
- Multiple simultaneous line (wire) drawing with no special editing modes required
- Special text block object to enter freeform text and label information
- Easy-to-use tools for object alignment, sizing, packing, spacing & centering
- Fully supported object drag-and-drop between Catalog and view, and between views
- Fully supported multiple layers within a drawing
- Support of Clipboard operation
- Programmable presets
- Export file types: DXF (Drawing interchange Format) & EMF (Enhanced Meta Files)

Architect and Engineer's Specifications

Tesira Audio Platform

The digital audio networking and processing platform shall include various hardware devices, software configuration and control and digital processing and networking of audio signals. The platform shall be configurable in signal flow, processing and routing. Processing algorithms shall include but are not limited to, level, mute, filtering, equalization, compression, limiting, automatic-gain control, mixing, automatic mixing, routing, room combining, ducking, acoustic echo cancellation (AEC), ambient noise compensation (ANC), delay, and metering. The platform shall include specialty algorithms for adaptive processing that can identify human speech from other ambient audio stimuli and that can identify ambient noise from primary program material. In addition to audio processing, the platform shall allow for software configuration and control that shall include but are not limited to logic signals, preset programming and command strings for providing control messages to systems that are not a part of the platform. The platform shall operate using Audio Video Bridging (AVB) as covered in IEEE 802.1 for the transport of all audio signals within the platform. The platform shall operate using Ethernet protocol for connection and communication between the platform and computers running Windows for diagnostic, configuration and control operations. The platform shall utilize various communication protocols for transmitting and receiving control signals with devices outside of the platform by way of TelNet, SSH, RS-232 and GPIO. The platform shall include hardware devices for processing, management and control of the platform and shall be known as "sever-class" devices. Server-class devices shall be available in multiple formats based on the maximum capacity for DSP resources and/or analog input and output resources. The platform shall include end point devices and shall be known as "expander" devices. Expander devices shall include audio endpoints, as well as logic and control endpoints, and shall include fixed configurations as well as modular configurations. The AEC algorithm shall be Sona by Biamp Systems. The specialty adaptive processing algorithms shall be SpeechSense and AmbientSense by Biamp Systems. The platform shall be Tesira by Biamp Systems.

Tesira SERVER

The digital audio network server shall be designed exclusively for use with Tesira systems. The server shall support AVB digital audio and control networking by means of a 420 x 420 modular card. The server shall also support an additional 420 x 420 channel AVB networking card or one 32 x 32 channel CobraNet or 64x64 Dante networking card. The server shall be configured with at least one DSP card and shall be capable of supporting a total of eight cards. The server will also accommodate one standard analog I/O card in lieu of one network card. The server shall provide dual Ethernet ports for configuration and control connection. The server shall provide front panel LED identification of server power, status, alarm, and activity as well as system-wide alarm. The server shall provide front panel LCD display for server and system information. The server shall be rack mountable (3RU) and feature software-configurable signal processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools. The server shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The server shall be a Tesira SERVER.

Tesira SERVER-IO

The digital audio network server shall be designed exclusively for use with Tesira systems. The server shall support AVB digital audio and control networking by means of a modular

420 x 420 channel card. The server shall also support use of one or two 32 x 32 channel CobraNet or one 64x64 Dante digital networking cards. The server shall be configured with at least one DSP card and shall be capable of supporting a total of three cards. The server shall provide dual Ethernet ports for configuration and control connection. The server shall be configurable for up to 48 channels of local audio input and output, including microphone and line level, VoIP, and telephone interface. The server shall also support modular I/O cards for acoustic echo cancellation and ambient noise compensation. The server shall provide front panel LED identification of server power, status, alarm, and activity as well as system-wide alarm. The server shall provide front panel LCD display for server and system information. The server shall be rack mountable (3RU) and feature software-configurable signal processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools. The server shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The server shall be a Tesira SERVER-IO.

Tesira SIC-4

The mic/line input shall be a 4-channel card designed exclusively for use with Tesira Server devices. The modular card shall provide 4 balanced inputs on plug-in barrier strip connections. Software configuration and control for each input shall include: gain with clip indication, phantom power on/off, mute, level, and signal invert. Analog-to-Digital conversion shall be 24-bit with a sampling rate of 48kHz. Performance specifications (20Hz-20kHz) shall be: Frequency Response +0/-0.25dB; THD+N <0.006% (line), <0.040% (mic); EIN <-125dBu; and Dynamic Range >108dB. The modular input card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira SIC-4.

Tesira SEC-4

The acoustic echo cancellation shall be a 4-channel card designed exclusively for use with Tesira Server devices. The modular card shall provide 4 balanced mic or line level inputs on plug-in barrier strip connections. Software configuration and control for each input shall include: gain with clip indicator, phantom power on/off, mute, level, and signal invert. The acoustic echo cancellation algorithm shall be configured and controlled separately in software and include processing for high-pass filtering, automatic gain control and noise reduction. Programmable parameters shall include: conferencing mode, noise reduction, threshold, mute and level. The modular input card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira SEC-4.

Tesira SAC-4

The ambient noise compensation shall be a 4-channel card designed exclusively for use with Tesira Server devices. The modular card shall provide 4 balanced mic or line level inputs on plug-in barrier strip connections. Software configuration and control for each input shall include: gain with clip indicator, phantom power on/off, mute, level, and signal invert. The ambient noise compensation algorithm shall be configured and controlled separately in software. Programmable parameters shall include: threshold, response time and compensation. The modular input card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira SAC-4.

Tesira STC-2

The telephone interface shall be a two-line, modular card for use with Tesira server devices, and shall allow direct connection to standard analog telephone lines. Each channel shall provide normal 2-wire to 4-wire "hybrid" functions, as well as line-echo cancellation, noise suppression, caller ID decoding, ring detection/validation, DTMF tone dialing, DTMF decoding, and call progress tone decoding. The telephone interface shall allow the Tesira system to respond to DTMF control commands such as preset recalls from any telephone system. The telephone interface may be used for audio input-only applications such as paging access; or for audio output-only applications such as broadcast feeds and remote system monitoring. The telephone interface card shall incorporate AES48-2005 Grounding & EMC practices, and shall be compliant with EU Directive 2002/95/EC, the RoHS directive. Warranty shall be 5 years. The telephone interface shall be a Tesira STC-2.

Tesira SOC-4

The line level output shall be a 4-channel card designed exclusively for use with Tesira Server devices. The modular card shall provide 4 balanced outputs on plug-in barrier strip connections. Software configuration and control for each output shall include: mute, level, signal invert and selectable output reference level (24dBu, 18dBu, 12dBu, 6dBu, 0dBu, -31dBu). Digital-to-Analog conversion shall be 24-bit with a sampling rate of 48kHz. Performance specifications (20Hz-20kHz) shall be: Frequency Response +0/-0.25dB; THD+N < 0.0035%; and Dynamic Range > 110dB. The modular output card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The output card shall be Tesira SOC-4.

TesiraFORTÉ AI and AI AVB

The digital audio network server shall be designed exclusively for use with Tesira systems. The AVB model server shall support Audio Video Bridging (AVB) digital audio networking that shall allow up to 128 x 128 channels. The AVB Networking connection shall be implemented on a RJ-45 connector on the AVB model. The server shall support Ethernet connection for programming and control on a RJ-45 connector. The server shall have internal DSP processing. The server shall include 4 channels of General Purpose Input and Output connection (GPIO) for sending or receiving logic signals. The programming of the GPIO ports shall be software configurable. The server shall include a RS-232 connection for control data transmission into or out of the server and such operation shall be software programmable. The server shall include a Universal Serial Bus (USB) connection on a standard USB-B type connector. The server shall be software configurable to stream up to 8 channels of digital USB Class 1 Audio transmission either into or out of the server or simultaneous input and output. The server shall provide 12 balanced input connections for receiving of microphone or line level analog audio signals on screw-down, removable connectors. The server shall provide 8 balanced output channels for the transmission of microphone or line level analog audio signals on screw-down, removable connectors. Each individual channel shall have its own dedicated connection. The server shall provide front panel LED identification of server power, status, alarm, and activity as well as system-wide alarm. The server shall be rack mountable (1RU) and feature software-configurable signal processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools. The server shall control and proxy all Tesira expander-class devices (AVB model only) and Tesira control devices. The server shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The server shall be TesiraFORTÉ AVB AI (for AVB model) or TesiraFORTÉ AI (for non-AVB model).

TesiraFORTÉ CI and CI AVB

The digital audio network server shall be designed exclusively for use with Tesira systems. The AVB model server shall support Audio Video Bridging (AVB) digital audio networking that shall allow up to 128 x 128 channels. The AVB Networking connection shall be implemented on a RJ-45 connector on the AVB model. The server shall support Ethernet connection for programming and control on a RJ-45 connector. The server shall have internal DSP processing. The server shall include 4 channels of General Purpose Input and Output connection (GPIO) for sending or receiving logic signals. The programming of the GPIO ports shall be software configurable. The server shall include a RS-232 connection for control data transmission and such operation shall be software programmable. The server shall include a Universal Serial Bus (USB) connection on a standard USB-B type connector. The server shall be software configurable to stream up to 8 channels of digital USB Audio Class 1 transmission either into or out of the server or simultaneous input and output. The server shall provide 12 balanced input connections for receiving of microphone or line level analog audio signals on screw-down, removable connectors. The input connections shall include Sona™ Acoustic Echo Cancellation (AEC) hardware and firmware, the parameters, routing and operation of which shall be software programmable. The server shall provide 8 balanced output channels for the transmission of microphone or line level analog audio signals on screw-down, removable connectors. Each individual channel shall have its own dedicated connection. The server shall provide front panel LED identification of server power, status, alarm, and activity as well as system-wide alarm. The server shall be rack mountable (1RU) and feature software-configurable signal processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools. The server shall control and proxy all Tesira expander-class devices (AVB model only) and Tesira control devices. The server shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The server shall be TesiraFORTÉ AVB CI (for AVB model) or TesiraFORTÉ CI (for non-AVB model).

TesiraFORTÉ TI and TI AVB

The digital audio network server shall be designed exclusively for use with Tesira systems. The AVB model server shall support Audio Video Bridging (AVB) digital audio networking that shall allow up to 128 x 128 channels. The AVB Networking connection shall be implemented on a RJ-45 connector on the AVB model. The server shall support Ethernet connection for programming and control on a RJ-45 connector. The server shall have internal DSP processing. The server shall include 4 channels of General Purpose Input and Output connection (GPIO) for sending or receiving logic signals. The programming of the GPIO ports shall be software configurable. The server shall include a RS-232 connection for control data transmission into or out of the server and such operation shall be software programmable. The server shall include a Universal Serial Bus (USB) connection on a standard USB-B type connector. The server shall be software configurable to stream up to 8 channels of digital USB Audio Class 1 transmission either into or out of the server or simultaneous input and output. The server shall provide 12 balanced input connections for receiving of microphone or line level analog audio signals on screw-down, removable connectors. The inputs shall include Sona™ Acoustic Echo Cancellation (AEC) hardware and firmware, the parameters, routing and operation of which shall be software programmable. The server shall provide 8 balanced output channels for the transmission of microphone or line level analog audio signals on screw-down, removable connectors. Each individual channel shall have its own dedicated connection. The server shall integrate to standard telephony communications on a RJ-11 connector for a single line of telephone communication. The server shall provide front panel LED identification of server power, status, alarm, and activity as well as system-wide alarm. The server shall be rack mountable (1RU) and feature software-configurable signal processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools. The server shall control and proxy all Tesira expander-class devices (AVB model only) and Tesira control devices. The server shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The server shall be TesiraFORTÉ AVB TI (for AVB model) or TesiraFORTÉ TI (for non-AVB model).

TesiraFORTÉ VI and VI AVB

The digital audio network server shall be designed exclusively for use with Tesira systems. The AVB model server shall support Audio Video Bridging (AVB) digital audio networking that shall allow up to 128 x 128 channels. The AVB Networking connection shall be implemented on a RJ-45 connector on the AVB model. The server shall support Ethernet connection for programming and control on a RJ-45 connector. The server shall have internal DSP processing. The server shall include 4 channels of General Purpose Input and Output connection (GPIO) for sending or receiving logic signals. The programming of the GPIO ports shall be software configurable. The server shall include a RS-232 connection for control data transmission into or out of the server and such operation shall be software programmable. The server shall include a Universal Serial Bus (USB) connection on a standard USB-B type connector. The server shall be software configurable to stream up to 8 channels of digital USB Audio Class 1 transmission either into or out of the server or simultaneous input and output. The server shall provide 12 balanced input connections for receiving of microphone or line level analog audio signals on screw-down, removable connectors. The input connections shall include Sona™ Acoustic Echo Cancellation (AEC) hardware and firmware, the parameters, routing and operation of which shall be software programmable. The server shall provide 8 balanced output channels for the transmission of microphone or line level analog audio signals on screw-down, removable connectors. Each individual channel shall have its own dedicated connection. The server shall integrate to Voice Over Internet Protocol (VoIP) systems on a RJ-45 connector for two lines of VoIP communication and shall support Session Initiation Protocol (SIP) v2.0 or later. The server shall provide front panel LED identification of server power, status, alarm, and activity as well as system-wide alarm. The server shall be rack mountable (1RU) and feature software-configurable signal processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools. The server shall control and proxy all Tesira expander-class devices (AVB model only) and Tesira control devices. The server shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The server shall be TesiraFORTÉ AVB VI (for AVB model) or TesiraFORTÉ VI (for non-AVB model).

Tesira EX-MOD

The modular expander shall be designed exclusively for use with Tesira SERVER and SERVER-IO devices. The expander shall be capable of handling up to 3 cards of 4 channels each for a total of 12 channels of analog audio inputs and outputs in various combinations. The expander shall utilize the AVB network for all audio networking as well as software configuration and control. The expander shall have a universal internal power supply: 100 – 240VAC, 50/60Hz. The expander shall provide front panel LED identification of device power, status, alarm, and activity. The expander shall be rack mountable (1RU) and shall be CE marked, UL listed and shall be compliant with the RoHS directive. Warranty shall be five years. The expander shall be a Tesira EX-MOD.

Tesira EEC-4

The acoustic echo cancellation shall be a 4-channel card designed exclusively for use with Tesira Server devices. The modular card shall provide 4 balanced mic or line level inputs on plug-in barrier strip connections. Software configuration and control for each input shall include: gain with clip indicator, phantom power on/off, mute, level, and signal invert. The acoustic echo cancellation algorithm shall be configured and controlled separately in software and include processing for high-pass filtering, automatic gain control and noise reduction. Programmable parameters shall include: conferencing mode, noise reduction, threshold, mute and level. The modular input card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira EEC-4.

Tesira EIC-4

The mic/line input shall be a 4-channel card designed exclusively for use with Tesira EX-MOD. The modular card shall provide 4 balanced inputs on plug-in barrier strip connections. Software configuration and control for each input shall include gain with clip indication, phantom power on/off, mute, level, and signal invert. Analog-to-Digital conversion shall be 24-bit with a sampling rate of 48kHz. Performance specifications (20Hz-20kHz) shall be: Frequency Response +0/-0.25dB; THD+N <0.006% (line), <0.040% (mic); EIN <-125dBu; and Dynamic Range >108dB. The modular input card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira EIC-4.

Tesira EIOC-4

The I/O card shall have 2 input and 2 output channels designed exclusively for use with Tesira EX-MOD. The modular card shall provide 2 balanced inputs and 2 balanced outputs on plug-in barrier strip connections. Software configuration and control for each input shall include gain with clip indication, phantom power on/off, mute, level, and signal invert; whereas for each output shall include mute, level, signal invert, and selectable output reference level (24dBu, 18dBu, 12dBu, 6dBu, 0Bbu, -31dBu). Analog-to Digital and Digital-to-Analog conversion shall be 24-bit with a sampling rate of 48kHz. Performance specifications (20Hz-20kHz) shall be: Frequency Response +0/-0.25dB. The modular I/O card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira EIOC-4.

Tesira EOC-4

The line level output shall be a 4-channel card designed exclusively for use with Tesira EX-MOD. The modular card shall provide 4 balanced outputs on plug-in barrier strip connections. Software configuration and control for each output shall include: mute, level, signal invert and selectable output reference level (24dBu, 18dBu, 12dBu, 6dBu, 0dBu, -31dBu). Digital-to-Analog conversion shall be 24-bit with a sampling rate of 48kHz. Performance specifications (20Hz-20kHz) shall be: Frequency Response +0/-0.25dB; THD+N < 0.0035%; and Dynamic Range > 110dB. The modular output card shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input card shall be Tesira EOC-4.

Tesira EX-IN, EX-OUT, EX-IO and EX-AEC

The 4-channel expanders shall be designed exclusively for use with Tesira Server devices. The expanders shall be built in a half-rack chassis and be powered from PoE+. The expander shall utilize the AVB network for all audio networking as well as software configuration and control. The input and input/output expander shall receive mic or line level analog input on plug-in barrier strip connection. The output and input/output expander shall deliver line level analog output on plug-in barrier strip connection. Analog-to-Digital and Digital-to-Analog conversion shall be 24-bit with a sampling rate of 48kHz. The expanders shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The input expander shall be Tesira EX-IN. The output expander shall be Tesira EX-OUT. The input/output expander shall be Tesira EX-IO. The AEC input expander shall be the Tesira EX-AEC.

Tesira EX-LOGIC

The logic expanders shall be designed exclusively for use with Tesira Server devices. The expanders shall be built in a half-rack chassis and be powered from PoE. Connection to the server for software configuration and control shall be via Ethernet. The expander shall have 16 connections that may be configured as input or output logic controls. As inputs the connections shall accept a contact closure or 5V TTL input signal. As outputs the connections shall be open-collector and deliver up to 40V. Additionally, 4 of the connections shall be configurable to operate as variable voltage control connections. These connections shall accept a variable input or provide a variable output from ground up to +5V. The expanders shall incorporate AES48-2005 Grounding and EMC practices and shall be compliant with EU Directive 2002/95/EC, the RoHS Directive. Warranty shall be 5 years. The logic expander shall be a Tesira EX-LOGIC

Warranty

BIAMP SYSTEMS IS PLEASED TO EXTEND THE FOLLOWING 5-YEAR LIMITED WARRANTY TO THE ORIGINAL PURCHASER OF THE PROFESSIONAL SOUND EQUIPMENT DESCRIBED IN THIS MANUAL

1. BIAMP Systems warrants to the original purchaser of new products that the product will be free from defects in material and workmanship for a period of 5 YEARS from the date of purchase from an authorized BIAMP Systems dealer, subject to the terms and conditions set forth below.

2. If you notify BIAMP during the warranty period that a BIAMP Systems product fails to comply with the warranty, BIAMP Systems will repair or replace, at BIAMP Systems' option, the nonconforming product. As a condition to receiving the benefits of this warranty, you must provide BIAMP Systems with documentation that establishes that you were the original purchaser of the products. Such evidence may consist of your sales receipt from an authorized BIAMP Systems dealer. Transportation and insurance charges to and from the BIAMP Systems factory for warranty service shall be your responsibility.

3. This warranty will be VOID if the serial number has been removed or defaced; or if the product has been altered, subjected to damage, abuse or rental usage, repaired by any person not authorized by BIAMP Systems to make repairs; or installed in any manner that does not comply with BIAMP Systems' recommendations.

4. Electromechanical fans, electrolytic capacitors, and normal wear and tear of items such as paint, knobs, handles, and covers are not covered under this warranty.

5. THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED. BIAMP SYSTEMS DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

6. The remedies set forth herein shall be the purchaser's sole and exclusive remedies with respect to any defective product.

7. No agent, employee, distributor or dealer of Biamp Systems is authorized to modify this warranty or to make additional warranties on behalf of Biamp Systems. Statements, representations or warranties made by any dealer do not constitute warranties by Biamp Systems. Biamp Systems shall not be responsible or liable for any statement, representation or warranty made by any dealer or other person.

8. No action for breach of this warranty may be commenced more than one year after the expiration of this warranty.

9. BIAMP SYSTEMS SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS OR LOSS OF USE ARISING OUT OF THE PURCHASE, SALE, OR USE OF THE PRODUCTS, EVEN IF BIAMP SYSTEMS WAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Biamp Systems, 9300 SW Gemini Drive, Beaverton, Oregon 97008
(503) 641-7287

Documentation

The information contained in this Help file can be printed in manual form (with Table of Contents and Index). Two PDF documents are provided with the software for this purpose. The file TESIRA-ltr.pdf is intended for printing on Letter (8.5" x 11") size paper. The file Tesira-A4.pdf is intended for printing on A4(210mm x 297mm) size paper. These are printable Help files. Similar PDF files are also available with the software for the 'RS-232 & Telnet Protocol' and the 'Quick Start Guide & Safety Information' documents.

Tesira Software Releases

Help File Build Date: 13:51:55 December 16,2014

2.3

Released December 2014
Full details not shown here. Please see
http://support.biamp.com/Tesira/Miscellaneous/Tesira_release_notes for full release notes.

New Features:

1. AVB 1722.1 Interoperability -Explicit AVB blocks for connecting to other 1722.1 compliant devices and networks.
2. Enhancements to system status reporting - Servers and expanders provide more details of particular faults. See the System Status and Fault Reporting section for more details
3. Dante Audio-Technica mic support.

Updates:

1. Updated and enhanced call progress tones

Resolved Issues:

Please see http://support.biamp.com/Tesira/Miscellaneous/Tesira_release_notes

2.2

New Features:

1. Added support for D series Lab.Gruppen Tesira amplifier.

Updates:

1. Properties for a link are now disabled when displaying a signal path or showing an implicit AVB network connection in the layout.
2. Document mode now shown in main title bar.

Resolved Issues:

1. Corrected an issue when activating knees on the Compressor block if the knee smoothing factor is set to a large value.
2. Corrected an issue where the compression ratio meter doesn't always return to 0 when a multi-channel compressor is bypassed.
3. Corrected an issue for updating fine-grain presets with level attributes that were created before the user-defined level min/max ranges were introduced.
4. Corrected an issue where mixers could be copied to a TesiraFORTÉ-only layout, even though the number of input and output channels exceeds the maximum available in TesiraFORTÉ.

5. Corrected an issue preventing users from opening files that end in upper case TMF from the most-recently-used list.
6. Corrected an issue with control dialog scroll bars.
7. Corrected an issue with the Noise Gate control dialog not correctly returning from the minimized view.

2.1

Released June 2014
New Features:

1. Added support for the EX-AEC and EX-MOD EEC-4 card
2. Introduced multi-channel Peak Limiter block

Updates:

1. Enhanced the Compressor block to support multiple channels and simplified mode of operation
2. Added support for asynchronous synchronization with explicit feedback to TesiraFORTÉ USB audio interface. This is now the default behavior, but can be disabled through the USB Input block property sheet
3. Added ability to disable the Telnet interface, which may be desired in some installations due to security concerns with Telnet protocol
4. Active call appearance is now highlighted to make it easier to recognize
5. Renames "Audio Failover" to "Analog Failover"
6. The Property Sheet fields for the AGC block now include the units
7. Added flow text support for the AEC Reference, AGC, ANC, and Delay blocks
8. SIP user names are now being validated as they are entered in Liner Properties. Illegal usernames will not be allowed.

Resolved Issues:

1. Fixed a bug introduced in 2.0 that prevents some files from opening in some cases. Using 2.1 to open these files will fix the problem
2. Fixed a bug that could crash the software when more than two objects were selected with different pen widths and the user clicked the 'Pen Width' field in the property sheet
3. Fix a bug in FIR filters that could sometimes crash the software if a file with an extension other than .txt was opened
4. Fixed a bug that caused the 'Accept Compile Results' dialog to crash in some cases
5. Fixed a bug that added extra pixels to Source Selector
6. Fixed a bug that caused the Level control dialog to size incorrectly
7. Fixed a bug that caused the compile to fail if an ANC Input blocks was not fixed to a device
8. Fixed a bug in the TI transmit that did not correctly calculate DSP usage
9. Fixed a problem in global optimization which, when partition connectors were present, prevented a recompile from being done
10. Fixed a problem that prevented multi-partition layouts from compiling when a partition did not provide a full audio path
11. Addressed a problem where the addition of a Dialer in fixed mode could place the Dialer into a new TesiraFORTÉ incorrectly
12. Fixed a bug that caused the scroll bar to sometimes appear in the Meter's control dialog after disconnecting from a system
13. Fix a bug in Equalizers that changed the frequency settings for existing bands when adding a new band
14. Fixed a bug in VoIP dialer that prevented the incoming caller ID from being displayed
15. Fix a bug in the equipment table that disabled the Unit field in some cases.

16. Fixed the tab order in Dante initialization dialog
17. Corrected a problem where copy/paste made inter partition links invisible
18. Corrected an issue where the system status display sometimes did not update in a timely manner on a failover of a redundant server pair
19. The direct output channel levels and labels for the auto mixers now update the property sheet correctly
20. Fixed a bug that allowed duplicate Dante Input channel names in some cases
21. Fixed a bug that sometimes allowed duplicate Instance Tags
22. Fixed a bug in the compiler that would prevent the compiler from completing if an unconnected PCRx was in the layout
23. Fixed a bug that did not mark a layout as not compiled when partitions were linked together

2.0.0

Released April 2014
New Features:
1. Support for TesiraFORTÉ
2. Support for Tesira SERVER redundancy
3. Support for the DAN-1 card
4. Support for Logic IO and VCB functionality on Tesira SERVER and SERVER IO back panel GPIO pins

Updates:
1. Significantly improved handling of connector lines
2. Routers now support Input/output labels
3. Flow text from an AEC processing block is now allowed
4. SIP uri and SIPS uri are now configurable
5. The Compilation Output window will always be displayed
6. There is now a short delay between selecting a chart point and being able to drag it to prevent inadvertent changes to the chart settings
7. Control dialog for Partition Connectors will now update when changes are made to partition block text, port text, or to a partition name
8. Logic Input and Logic Output blocks can now be allocated to Servers and TesiraFORTÉ's
9. Added support for user defined ranges for Level values to additional blocks, including:
    1. Input
    2. Output
    3. CobraNet Input
    4. CobraNet Output
    5. AEC
    6. TI Receive
    7. TI Transmit
    8. VoIP Receive
    9. VoIP Transmit
    10. Dante Input
    11. Dante Output
    12. USB Input
    13. USB Output
    14. Crossover
    15. Ducker
    16. Tone Generator

17. Noise Generator

Resolved Issues:

1. Fixed a problem that prevented the ability to create multiple blocks using 'Shift-Click'
2. Improved the meter performance when there are multiple meters active in the layout
3. Corrected a problem with fine grain presets effecting All Pass Filters when the presets are created while online
4. Flowing text from logic ports is now handled correctly
5. The 'Current Gain Applied' meters will now continue to operate after the 'Channel Levels' dialog is closed. Previously this would interrupt the updates to the meters
6. All inputs are now turned on for N x 1 mixers
7. Fix a bug in Custom blocks so that DSP Resources are updated correctly when contents are modified
8. Partition Tab order is now persistent
9. Handling of out of range values is now more consistent for level settings
10. In control dialogs, the Min/Max for a Level will now respond correctly to Page Up and Page Down keys
11. Fixed a bug that allowed invalid connections to be made for signal paths that include AEC Reference blocks. Auto-mixers and other dynamic blocks are not allowed to be directly or indirectly connected to the output of the AEC Reference block
12. The TI Dialer 'Auto Answer' button is now active in minimized dialogs
13. Corrected condition on Fan-In OR Pulse blocks where an Undo may result in the number of visible ports not matching the number of actual channels
14. Fixed an issue that prevented fine grain presets for the 'Mute' attribute in a VoIP Receive block from working
15. Fixed a bug with the Ctrl-V (keyboard paste) that could cause some block/lines to lose alignment
16. The Ducker's 'Threshold' property can now be set in the Property Sheet, previously it was read only in the property sheet
17. Corrected a problem that allowed duplicate preset names which could cause the software to crash
18. AEC Input and AEC blocks will not be able to directly connect to another AEC Reference block in layouts created in 2.0. If the layout is older, removing direct links between these blocks will result in the inability to re-link them.

1.2.1

Released October 2013

New Features:

1. No new features added in this release

Updates:

1. There are no updates

Resolved Issues:

1. Solved potential problem with Biamp Canvas refusing to load if the Tesira configuration file has partition connectors with duplicate instance tags as a result of cut/copy handling
2. Fixed an issue in Biamp Canvas that could result in meters not updating after partitions were created and later deleted

1.2.0

Released August 2013

New Features:

1. Added support for Biamp Canvas

2. Launched new licensing algorithm and Web portal
3. New Global Optimization function
4. Server RS-232 Ports now supports multiple modes of operation

Updates:

1. Help file has been updated
2. License Information for GNU can now be retrieved from Device Information window
3. Improved the way Tesira software handles PC resources, making it more efficient to run
4. Improved compilation time of complex logic designs
5. CDI files are stored when the tmf document is stored, and not just when it is closed
6. Created System menu, hosting start/stop audio, Partitions, Network, and Security settings, as well as Equipment Table and Delay Equalization Groups
7. Software now provides a warning message if trying to send a configuration to a system running older firmware
8. Serial Command String block now allows selection of which serial port to use
9. Added support for advanced parameters for AEC, and Automixer's Edit Block Parameters preset window
10. Extended VoIP Local Dial Plan field to 256 characters
11. Bring to Front and Send to Back functions now have key shortcuts (Ctrl+F and Ctrl+B)
12. Adjusted some default AEC Noise Reduction settings so that the function performs better

Resolved issues:

1. Resolved problem that could result in Tesira software crashing when an Uber filter was deleted
2. Resolved problem that could result in Tesira software crashing when a via block was used but no valid audio path was connected to output blocks
3. Fixed problem that could result in Tesira software crashing when trying to copy/paste blocks into locked or invisible Layers
4. Prevented a situation that could result in Tesira software crashing if invalid data was entered by the user as destination printer and the Print function was used
5. Fixed propagation delay reporting for Custom Block, Split Pass Through and Partition Connector blocks.
6. Fixed problem that could result in "Sequence contains more than one matching element" being reported when compiling if some blocks had been deleted and replaced in a specific manner
7. Fixed problem that could result in "Sequence contains more than one matching element" being reported when compiling if some AEC or ANC blocks were left disconnected
8. Fixed problem that could result in "Sequence contains no matching element" being reported when compiling if an AEC input block is connected to an output but the AEC processing and reference blocks are not in a valid audio path
9. Ignored AEC blocks now reported correctly in Output Window
10. Fixed problem that could result in presets containing ANC blocks being corrupted if Select Block Attributes was used
11. Fixed problem that could occasionally result in firmware update not being performed on all selected devices
12. Resolved issue that could result in Filter Type and Locked column settings not being transferred between Uber Filters when using the Copy/Paste DSP Data function
13. Fixed a problem that could result in some Advanced parameters not updating correctly on gating Auto Mixers when recalled via preset
14. Tesira software now correctly allows blocks and their lines to be dragged and dropped between instances of the software

15. Fixed issue that could result in blocks being ignored if connected to Logic and Audio Pass through
16. Resolved problem that could result in being unable to delete lines on some layouts
17. Resolved issue that could result in an Observer user to have its mouse trapped if attempting to edit blocks with charts, like EQs or Filters
18. Fixed problem that could result in some Partition Connector mappings being displayed incorrectly if used to connect between Automixers and Combiners
19. Prevented software from displaying propagation delay data for some logic connectors on some audio blocks
20. Fixed the ERLE meter on an AEC block so it's now the same size as the other meters
21. Fixed problem that could result in the AGC Advanced control dialog to display incorrectly
22. AEC meters now reset correctly to default value after software disconnects from system
23. Resolved problem that could result in Device IO data displaying as ?? on AEC and ANC dialogs after a preset containing those blocks was recalled
24. Fixed problem that could result in being unable to select Telephone as AEC Conferring mode if the mode had been changed multiple times already
25. Room Group is now displayed as an option in Select Block Attributes window for Room Combiners
26. Prevented software from allowing invalid selection of serial port for Expanders on Command String block

1.1.2.14

Released March 2013

New Features:

1. No new features added in this release

Updates:

1. SVC-2 IP configuration now allows subnet masks with non-zero fourth octet
2. Removed (Microsoft) Lync from the VoIP Proxy Vendor List as it isn't supported
3. The CDI file is now saved every time the user saves the configuration (tmf) file
4. Further improvements to the way the Compiler adds hardware to a configuration

Resolved issues:

1. Including a password protected custom block in a preset no longer causes a crash during system configuration
2. Repaired problem that could cause Tesira software to crash when opening configuration files created with a newer version of the software
3. Clicking on the Set All button on the Pre/Post NOM channel setting window on a Gating Automixer with Direct Outs no longer causes a software crash
4. Changing the block color of a custom block no longer prevents the configuration file from opening
5. Fixed problem that could result in unexpected disconnection if connecting through a VPN
6. Fixed issue that would result in the DSP Usage not always being retained if set as conservative
7. Output block channel ordering now follows the same logic regardless of whether a serial number has been added to the Equipment able or not
8. Fixed problem that could result in "Object reference not set to an instance of an object" being reported when compiling if a large mixer was connected directly to ANC processing input port

9. Fixed problem that could result in "Object reference not set to an instance of an object" being reported when compiling if a Gating Auto Mixer with large number of direct outputs was included in the layout
10. Fixed problem that could result in "Could not break up a big (auto)mixer. Unable to complete DSP assignment for the unit allocation in position 1. No solution to the assignment problem found." when compiling if a mixer have between 198 and 211 channels

1.1.0.11

Released October 2012

New Features:

1. Added support for TEC-1
2. Added BUFFER logic gate.

Updates:

1. Help file has been updated
2. Instance Tags of blocks are now displayed in the Status bar whenever the block is selected. This feature works both off- and online.
3. Processing Library now allows adding blocks even if no Catalogs exist
4. Preset names are now allowed to contain characters that are not valid in XML, like '&' or characters with umlaut.
5. AVB links are now also displayed in dotted-blue (default) when going through Via blocks
6. Improved the way the compiler handles inputs and outputs added to a design that had been compiled already, resulting in better hardware allocation.
7. The Transfer Function block is now able to compute propagation delay even if the reference port is the output node of an audio Split Pass Through block
8. Improved response on Comp/Limiter graph when multiple knees are used
9. Object ID Inspector is now available when online

Resolved issues:

1. Software no longer crashes when compiling layout with many EX-Mods and a large mixer
2. Fixed problem in compiler that will result in "The given key was not present in the dictionary" error message when units are removed from Equipment Table.
3. Fixed problem in compiler that will result in "The given key was not present in the dictionary" error message if after a compilation a block controlled by a Voltage Control channel ends up being assigned to a new unit
4. Tesira software no longer crashes after a click on 'Line Properties' in the property sheet of a VOIP Control Status block
5. Deleting copied Split Pass Though block no longer causes a serious error being reported
6. Fixed a problem that could result in software crashing during compilation after Delay Equalization mode was changed in Document Settings, General Settings.
7. Prevented software from crashing if an audio Split Pass Through block is reduced in size such that connected port is removed
8. Tesira software no longer crashes if the print function is used on a PC with no printers installed
9. Prevented software crash when using the Active band drop-down menu on an All Pass Filter block after various drag points have been selected.
10. IP address now showing up correctly every time in remote device maintenance dialog
11. Prevented remote expanders from rebooting unnecessarily if their Network Settings window was accessed but nothing that required a reboot was changed

12. Adding a Text block to a compiled partition no longer requires the partition to re-compile before sending configuration.
13. Export to DXF function no longer ignores blocks inside Custom Blocks is they are in expanded view when the export is done.
14. Adding blocks to the Processing Library no longer resets their Delay Equalization group membership to default
15. Removed the Print Scale property from Partition Properties window as this is not a valid option.
16. Fixed the "locked layer" icon so it shows correct status of layer after a reboot.
17. Presets with only a few attributes of a Control Voltage block selected can now be sent to a device without errors.
18. Preset Buttons are now listed as unacceptable blocks in a Custom Block. Previously, the blocks weren't listed, but since they aren't permitted in Custom clocks, trying to add them would result in an unclear error message.
19. Removing an item from TEC-1 block and then undoing the delete won't break presets containing the TEC-1 anymore.
20. Text on blocks will not be reset to default if Edit Block Parameter changes are made on the block.
21. When duplicating a block, the Fixed in Unit setting is no longer resetting
22. Fixed Paste DSP Data function from a Feedback Suppressor to a Parametric EQ
23. Copy/Paste DSP menu option in the room combiner is now disabled, as the block doesn't support this function
24. Cut and Paste operations on a DTMF decode button now work correctly, no longer creating a new block.
25. Fixed logic simulation for multi-channel NOT blocks
26. Fixed problem preventing Signal Present LED from working correcting on some type of signals when the control dialog of the block was maximized
27. Fixed inconsistency between software and TTP VoIP Control/Status - VAD Threshold maximum setting
28. When duplicating custom blocks protected with password, the new instance of the custom block can now have their passwords changed
29. Added "No delay information available" on the status bar when signal coming from a Via block is selected.
30. The Uber Filter block now correctly flows port text from input to output.
31. Corrected problem that could result in the filter slope column to temporarily go blank on an Uber Filter block
32. Entries for Level Out settings in the property sheet of a Room Combiner block are no longer allowed to be outside the min-max range.
33. Room Combiner's settings now always work correctly even if the room controlled isn't assigned to a group
34. Min and Max Compensation limits on an ANC block now work as expected every time
35. Fixed a problem resulting on the ANC meters being stuck at minimum values under certain circumstances despite the block being used.
36. Crossover port labels are now always correct even if changes are applied to the block using Edit Block Parameters menu
37. The Active band drop-down menu on an All Pass Filter block now updates correctly is more bands are added through TTP
38. Prevented issue that resulted in Propagation Delay being incorrectly calculated on some Automixer outputs
39. The following problems have been resolved in the Comp/Limiter block:
    - Tab order is now correct
    - Graph changes no longer break the undo/redo queue

- Fixed problem that could result in two undo's required to see one event change in the control dialog
- Stopped knees from jumping when thresholds are set too close together
- Parameters of a Comp/Limiter block with only one knee are no longer limited too conservatively
- Changing a threshold value no longer automatically changes the compression ratio
- Prevented the compression curve breaking when changes are applied to the soft knee radius
- The down arrow on the Threshold field now works correctly

1.0.1

Released July 2012

New Features:

- No new features added in this release

Updates:

1. Help file has been updated
2. Improved graphical interface of Comp/Limiter block
3. Made improvements to the way Delay Equalization is reported and implemented

Resolved Issues:

1. Prevented software from occasional crashes when working with Edit Block Parameters
2. Prevented software from crashing occasionally when sending configuration after some partitions were deleted
3. Fixed software hanging when connecting to a device that was recently powered up
4. Prevented software from giving serious error and restarting if an Undo was done after an Edit Block Parameters operation was done to an Uber Filter
5. Stopped software from crashing when a Preset containing a Gain Sharing Automixer was recalled, if the Gain Sharing Automixer had its Direct Outputs enabled after the preset was created.
6. Fixed problem resulting in the inability to delete or move lines after copy-paste from an online layout
7. Prevent Feedback Suppressors from resetting bands when control dialog is closed
8. Fixed problem preventing fine-grained presets containing the Dialer block from working while online
9. Fixed problem resulting in the Country field on VOIP DSP Properties to always show USA even when other Country is specified.
10. Fixed problem preventing the Dialer dialog from opening if it was minimized when software disconnected from system
11. Made the Object ID Inspector to refresh contents properly
12. Prevented blocks from resetting their port properties on certain situations
13. Made spacing and text on Room Combiner block persistent
14. Prevented presets that had only a few attributes of a block selected from being removed if the related block parameters were changed after the preset was created.
15. Prevented the Select Block Attributes window of an AEC block from displaying values which are unavailable to the user
16. ANC meters now reset after disconnecting from the system
17. Corrected error that could lead to some Current Gain Applied meters on a Gain Sharing Automixer not updating properly.

1.0 Update 1

Released May 2012

New Features:
- No new features added in this release

Updates:
- There are no updates

Resolved Issues:
- Fixed the way licensing timeout was calculated

1.0

First public release, May 2012

Tesira Firmware Releases

Help File Build Date: 13:51:54 December 16,2014

2.3

Released Dec 2014
Not shown here -Please see
http://support.biamp.com/Tesira/Miscellaneous/Tesira_release_notes for full release notes.

2.2

Released September 2014, aligned with software release 2.2

New Features:

1. Added support for D series Lab.Gruppen Tesira amplifier.

Updates:

1. There are no updates in this release

Resolved Issues:

1. No issues were resolved in this release.

2.1.1

Released July 2014, aligned with software release 2.1

New Features:

1. No new features added in this release

Updates:

1. There are no updates in this release

Resolved Issues:

1. Fixed several problems related to the stability and robustness of Audio and Logic Expanders. Specifically, problems were corrected which could occasionally cause expanders to spontaneously reboot.
2. Fixed two issues with compressor blocks. Bypass now disables make-up gain. Also, enabling/disabling bypass no longer produces audible artifacts.
3. Several minor SVC-2 enhancements, including:
4. Calls not handled properly if terminated while far end is still ringing

2.1

Released June 2014, aligned with software release 2.1
New Features:
1. Added support for the EX-AEC and EX-MOD's EEC-4 card
2. Added support for multi-channel Peak Limiter block

Updates:
1. Enhanced the Compressor block to support multiple channels and simplified mode of operation
2. Added support for asynchronous synchronization with explicit feedback to TesiraFORTÉ USB audio interface. This is now the default behavior, but can be disabled through the USB Input block property sheet
3. Changed USB Input block Host Master Volume maximum value from +12 to 0 dB
4. Added ability to disable the Telnet interface
5. Updated AVB firmware

Resolved Issues:
1. Corrected audio distortions when using TesiraForte USB with Mac by virtue of change to asynchronous synchronization.
2. Stability and robustness enhancements to the USB implementation
3. Several minor SVC-2 enhancements, including:
    - Interoperability with Cisco Call Manager 10.0
    - IP Address conflict fault message now cleared
    - Failed DTMF negotiation
4. Corrected some security vulnerabilities in the operating environment, specifically as related to the Telnet server
5. Modified the Telnet server to detect connection of simple clients such as Crestron and suppress transmission of NULL pad bytes.
6. Stability and robustness enhancements to Audio and Logic Expander firmware including improved resilience to heavy network IP broadcast traffic.

2.0.1

Released April 2014, aligned with software release 2.0

New Features:
- No new features added in this release

Updates:
- There are no updates

Resolved Issues:
1. Corrected a bug in USB audio playback compensation which could result in audio pops and clicks being heard over USB inputs
2. Corrected a problem in which an AVB card which had been intentionally disabled using a TTP command would result in a reported fault
3. Fixed a problem in which a statically defined default gateway was not applied correctly

2.0

Released April 2014, aligned with software release 2.0

New Features:
1. Support for TesiraFORTÉ
2. Support for Tesira SERVER redundancy

3. Support for the DAN-1 card
4. Support for Logic IO and VCB functionality on Tesira SERVER and SERVER IO back panel GPIO pins

Updates:

1. Improved the response time of some dynamic audio processing blocks
2. Added fault detection and reporting for cooling fan malfunctions on SERVER and SERVER IO
3. Added non-volatile persistence to device maintenance features on EX-LOGIC. Serial port settings are now retained through power cycle
4. Calibrations on Logic and VC IO are now cleared when device is reset to factory settings
5. Tesira SERVER and SERVER IO devices which have no DSP cards installed now raise "No DSP cards found" fault.
6. Serial port TTP session will now prompt for a password immediately after a device is protected
7. Numerous SVC-2 enhancements, including
    - better operation in DHCP environments
    - improved handling of DNS requests
    - improved handling of session timer
    - more advanced detection and reporting of duplicate IP addresses.
    - achieved interoperability certification with Avaya Session Manager versions through 6.3
    - improved behavior of TTP subscriptions
8. NTP daemon only runs if NTP is enabled through Device Maintenance
9. GPIO ports on server network cards (SNC-1 and SNC-2) and TesiraFORTÉ are now operational
10. Various other performance improvements

Resolved Issues:

1. Calibrations on Logic and VC IO are now cleared when device is reset to factory settings
2. Ducker block logic outputs are no longer affected by Invert control when logic outputs are disabled
3. Gating Automixer block logic outputs now behave correctly with respect to "Logic Outputs Follow Mic Logic" feature
4. Serial port CommandString blocks no longer output when serial port usage is set to None or TTP
5. Changing time zone on a SERVER or SERVER IO now updates the device's time and date accordingly
6. Reversed the direction of the timeout settings on the SERVER and SERVER IO front panel so the up button increases the time and the down button decreases the time
7. "Unable to Communicate with DSP Card in slot" faults now report the slot number as it appears on the PCB silkscreen
8. Improved TTP subscription handling on TIControlStatus (lineIntrusion and CallState) as well as Level, RoomCombiner, and GainSharingAutoMixer (level)

1.2.1

Released October 2013, aligned with software release 1.2.1

New Features:

1. No new features added in this release

Updates:

1. Improved automated testing of some IO cards at the factory

Resolved Issues:

1. No other issues resolved in this release

1.2.0

Released October 2013, aligned with software release 1.2.0

New Features:

1. Support for Biamp Canvas
2. New serial port features: both Serial1 and Serial2 on Server(IO) are now configurable for TTP input, control string output, both, or neither
3. Added the ability to disable the AVB card in a Server(IO). This allows multiple Server(IO) devices to participate on the same CobraNet network without relying on AVB for media clock synchronization.

Updates:

1. Reset button now works regardless of the state of firmware, even if in error state
2. Added "gateway" to the TTP response for "DEVICE get networkStatus".
3. Improved responsiveness of network status icon on front panel display
4. Updated SVC-2 with general improvements and enhanced 3rd party interoperability
5. Significant performance improvements for logic signal handling and propagation
6. Improved Line Echo Cancelation on STC-2
7. STC-2 received audio is no longer ducked during delayed dialing
8. STC-2 no longer hangs up if Wait For Dial Tone is enabled and a valid dial tone isn't recognized. This change is made so that STC-2 behaves the same as the TI-2
9. Improved AEC Noise Reduction performance - particularly when using microphones with high background noise - for example ceiling microphones
10. Servers and Servers IO have improved recovery if a configuration upload fails

Resolved Issues:

1. Eliminated gating effect on the outputs which was noticeable in some circumstances
2. Prevented a problem that could result in Gateway information not correctly displayed in Device Network Settings
3. Fixed issue that could result in some control input not functioning on mute blocks if not all control inputs were connected
4. Prevented a situation that could lead to some control labels on a TEC-1 not being visible if left unconfigured
5. Multiple simultaneous preset recall requests are now handled by multi-device systems
6. Fixed numerous other internal issues and provided many stability enhancements.

1.1.3

Released May 2013, maintenance release

New Features:

1. No new features added in this release

Updates:

1. Made enhancements for improved internal communication on Server/Server IO, and between Server/Server IO and expanders

Resolved issues:

1. Resolved potential DHCP problems with SVC-2 on some network switches
2. Resolved problems that could cause AVB audio to sound distorted
3. Fixed situation that could result in the AVB-1 card becoming unresponsive
4. Prevented problem that could result in a unit failing to load its configuration data after a firmware update was done on a configured system

5. Fixed several other internal issues

1.1.2

Released March 2013, aligned with software release 1.1.2.14

New Features:
- No new features added in this release

Updates:
1. Updated SVC-2 with general improvements and enhanced third party interoperability
2. Improved robustness of control port on Server and Server IO to heavy UDP traffic
3. Uploading potentially damaging firmware version on a unit is prevented by imposing a minimum criteria before an update

Resolved issues:
1. Fixed numerous other internal issues and provided many stability enhancements

1.1.1

Released November 2012, maintenance release

New Features:
1. No new features added in this release

Updates:
1. No updates added in this release

Resolved issues:
1. Fixed a problem in which some settings specified in initial configuration, power-up configuration, or presets to non-default values are not actually applied to AEC or ANC Inputs, TI Receive and TI Transmit
2. Fixed other internal issues

1.1.0

Released October 2012, aligned with software release **1.1.0.11**

New Features:
1. Added support for TEC-1

Updates:
1. Made significant general performance improvements
2. Server and Server IO now always report a fault if another device in the same system goes offline
3. Improved responsiveness of Peak meters
4. Improved SVC-2 third party interoperability

Resolved issues:

1. Prevented a situation that could result in a failed firmware upgrade being reported as successful
2. Fixed potential loss of persistent data (serial port settings, time zone, passwords, etc.) when updating firmware using the Restore version
3. Prevented problem that could result in various error messages being reported if a preset was recalled externally while a preset's name was edited with Tesira software
4. Reset/Initialize function now also removes SESSION aliases from memory
5. Prevented problem that could result in TTP subscriptions to peak meters and levels returning duplicate data

6. Fixed problem that could result in a Preset Button block in a protected system unable to recall presets
7. Prevented audio expanders from occasionally falsely reporting a major fault after power on
8. Fixed numerous other internal issues and provided many stability enhancements

1.0.4

Released August 2012, maintenance release

New Features:
1. No new features added in this release

Updates:
1. No updates added in this release

Resolved issues:
1. Fixed occasional problems when updating only some (or one) partition on multi-partition systems that included audio expanders
2. Fixed various other internal issues

1.0.3

Released July 2012, aligned with software release 1.0.1

New Features:
1. No new features added in this release

Updates:
1. Added support for improved Comp Limiter control dialog behavior in Tesira Software
2. If configured for DHCP, Expanders now provide their hostname to the DHCP server
3. Improved AVB reliability and functionality, particularly with many streams
4. Improved communication between devices
5. Improved behavior of SVC-2

Resolved Issues:
1. Fixed problems in AEC and ANC cards which caused them to sometimes fail to process all channels
2. Fixed various audio problems when audio passed between partitions or fanned out within a Tesira device
3. Fixed problem that could result in a single AEC processing output channel appearing to be muted
4. Fixed a problem that could result in a VoIP conference run by the SVC-2 to be interrupted if the remote phone put the call on hold and then resumed
5. Fixed a problem that could result in a peak meter not displaying correctly in software of it had received a TTP Subscription which was subsequently canceled
6. Fixed a problem that could result in STC-2 hook flash duration not being properly accepted if set via TTP
7. Prevented problem that could result in corrupted audio when signal was send from one block's output node to multiple input nodes (fan out)
8. Fixed various other internal issues

1.0.1.0

Released June 2012, maintenance release

New Features:
1. No new features added in this release

Updates:

1. No updates added in this release

Resolved Issues:

1. Corrected a problem in AEC and ANC cards which caused them to sometimes fail to process all channels.

1.0.0.8

First public release May 2012, aligned with software release 1.0

Software Tools

Basic Screen Elements

The main screen of the **Tesira®** software has several sections.

The Layout occupies the largest portion of the main screen, at the lower-right. This is the area where system design actually occurs, with the placement & connection of Component Objects. Component Objects represent the individual audio devices (processing blocks) within the system. Other objects include Lines (for connecting components) and Text (for labeling the system).

There are two default tabs to a new Layout: The system Overview that shows the signal flow between all partitions, and the first partition layout.

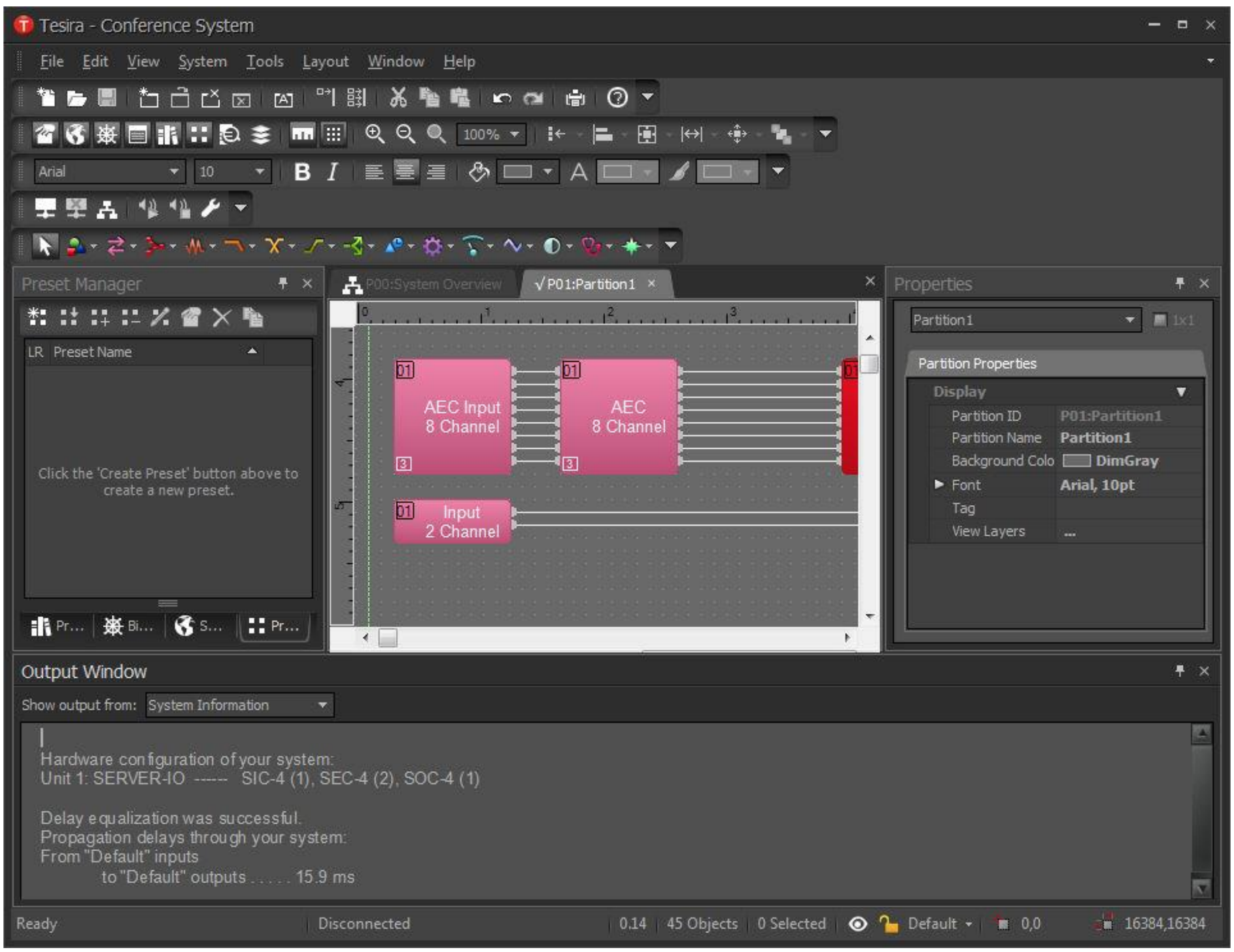
If a system becomes too big to fit the Layout, a Bird's Eye View is available to aid navigation. A Property Sheet, at the lower-left of the main screen, provides an editable table of attributes regarding the Layout and its associated objects.

A Processing Library of available Component Objects is displayed, at the lower-left of the main screen, for drag & drop placement into the Layout. However, the Processing Library can be closed, increasing the width of the Layout, and Component Object selection can then be accomplished using the Object Bar, located directly above the Layout.

The Object Toolbar also provides the option of either a select or a text cursor. Above the Object Bar is the Format Bar, which allows customizing of text & colors used in the Layout and associated objects. Above the Format Toolbar is the Layout Toolbar, which affects certain aspects of Layout, such as the grid, rulers, zooming, & alignment of Component Objects. The Layout Toolbar can also open editing/information sheets for properties, objects, & layers. Above the Layout Toolbar is the Network Bar, with functions related to communications, configuration, maintenance, and testing of the system network. Above the Network Toolbar is the Standard Bar, with file functions such as new, open, & save, plus additional functions such as cut, copy, paste, print, & help. The Standard Toolbar also includes the Compile function, which will generate a new configuration file, while checking system layout/connections & determining DSP resource allocation. Above the Standard Toolbar are the Main Menus, which provides all of the toolbar functions mentioned above, with several more in-depth editing functions. Along the bottom edge of the main screen is a Status Bar, which gives indication of object quantity, location, & size, as well as layer name, visibility, & locking. The location and shape of all toolbars, including the Processing Library, may be changed to fit the user's preference.

Layout

The Layout occupies the largest portion of the main screen, at the lower-right. This is the area where system design actually occurs, with the placement & connection of Component Objects. Component Objects represent the individual audio devices (processing blocks) within the system. Other objects include Lines (for connecting components) and Text (for labeling the system). Component objects can be placed into the Layout from the Processing Library or the Object Toolbar, or the Processing Library Menu. A Property Sheet provides an editable table of attributes regarding the Layout and its associated objects. The Layout has fixed dimensions of 16384x16384 pixels (approx. 163.75" square). Therefore, horizontal & vertical scroll bars are used to navigate within the Layout. Zoom In/Out and a Bird's Eye View are also available as navigational aids, and can be accessed from the Layout Toolbar or View Menu. The Ruler and Grid may be turned on/off from the Layout Toolbar, and the background color of the Layout may be changed from the Format Toolbar. Grid Settings, such as snap-to-grid, grid spacing, and guideline spacing, are available from the Layout Menu. Right-clicking over the Layout provides a pop-up menu of options.



Keyboard Shortcuts

Tesira software supports 'standard' windows keyboard shortcuts and methods.

'Alt' key

The Menu Bar can be navigated using the 'Alt' key to go into navigation mode and then referencing the corresponding underlined letter.

For example, to reference the 'Undo' function of the Edit menu the following keystrokes would be used:

- Alt - enters Menu Bar navigation mode
- E - Opens the Menu Bar 'Edit' menu
- U - Selects the 'Undo' function.

Each menu has a corresponding underlined letter. Please review the software menu's to get more details of the correct letter combinations.

'Alt' + Function Key

The following 'Alt + Function commands are supported

Service	Shortcut	Description
Exit	Alt + f4	Closes the **Tesira**® Software

'Ctrl' Key

The following commands can be accessed by holding the control ('Ctrl') key and corresponding letter key.

Service	Shortcut	Description
New	Ctrl + N	Begins a new **Tesira**® system design file (.TMF).
Open	Ctrl + O	Opens an existing **Tesira**® system design file (.TMF).
Save	Ctrl + S	Saves the working **Tesira**® system design file (.TMF) in its current location. The default file path is \my documents\Biamp\Tesira\DataFiles.
Copy	Ctrl + C	Copies the selected Item to the Clipboard
Paste	Ctrl + V	Pastes previously Cut or Copied item to the cursor location
Undo	Ctrl + Z	reverts to the last changed item
Redo	Ctrl + Y	Re-does the previous Undo
Copy DSP data	Ctrl + U	Copy DSP data allows the user to copy the settings of a specific block for pasting onto another block of the same kind.

Paste DSP data	Ctrl + T	Paste DSP data takes data copied from another DSP block of the same size and type and pastes it onto the specified block.
Select All	Ctrl + A	Selects all objects
Duplicate	Ctrl + D	Duplicated the currently selected items
Manage Partition Audio	Ctrl + P	Opens the Manage Partition Audio Dialog.

'Shift' Key

Service	Shortcut	Description
Connect Audio	Shift + Right	Connects Audio between the Selected DSP objects
Connect Logic	Shift + Up	Connects Logic between the Selected DSP objects

Context Menus

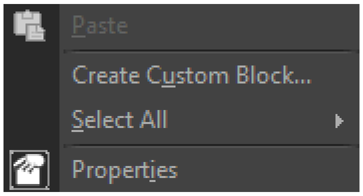
Context Menus

Tesira Software also makes uses of a 'right click' context menu. The options available will change depending what is selected.

- Surface Context Menu
- Line Context Menu
- Object Context Menu

Surface Context Menu

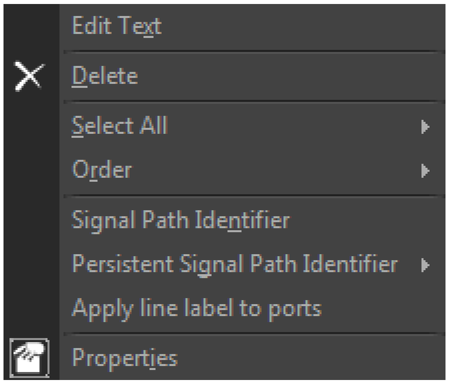
If you select an empty area of the surface the options relevant to the surface are available



Action	Description
Paste	Paste the contents of the clipboard to the selected areas. The clipboard must contain suitable Biamp Canvas controls
Create Custom Block	Adds a Custom Block to the layout
Select All	Provides an easy way to select all objects on the Layout. Also allows all objects of a specific type to be selected. Also available via the Edit Menu
Properties	Enables the Property Sheet

Line Context Menu

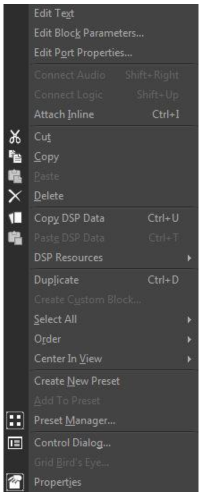
If you right click an Audio Connection the following Context Menu appears.



Name	Description
Edit Text	Adds labeling to the line
Delete	Deletes the selected Items
Select All	Provides an easy way to select all objects on the Layout. Also allows all objects of a specific type to be selected. Also available via the Edit Menu
Order	Moves the line forward or back in the layout. See Order Menu
Signal Path Identifier	Shows audio routing in a non-persistent mode. See Signal Path Identifier
Persistent Signal Path Identifier	Shows audio routing in a persistent mode. See Signal Path Identifier
Apply Line Labels to Ports	Propagates the text on the line to the transmitting and receiving Port Labels. Note: if the line's label is blank, the port labels will also become blank.
Properties	Opens or closes the Property Sheet

Object Context Menu

If you select a DSP or Logic object on the surface the relevant options are available.



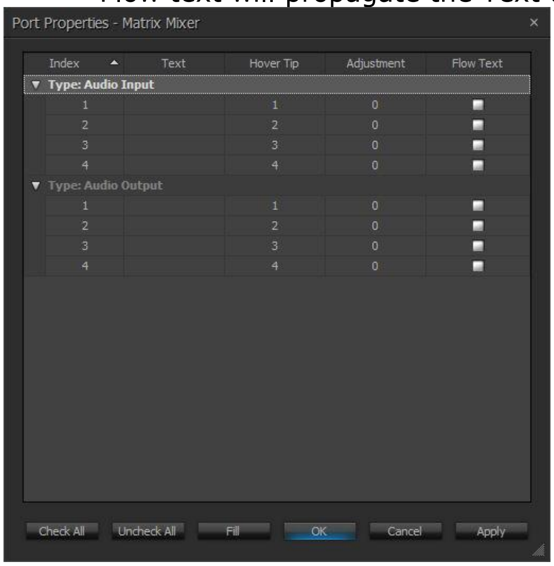
Name	Description
Edit Text	Places the block text into edit mode.
Edit Block Parameters	Opens the Initialization dialog for the selected block
Edit Port Properties	Opens the Port Properties Dialog
Connect Audio	Connects un-wired audio between twos DSP objects
Connect Logic	Connects un-wired logic between twos DSP objects
Attach Inline	
Cut	Removes the selected object(s) from the Layout, and places them in the Clipboard.
Copy	Places a copy of the selected object(s) into the Clipboard.
Paste	Places a copy of the object(s) from the Clipboard into the Layout.
Delete	Removes the selected object(s) from the Layout, without placing a copy into the Clipboard.
Copy DSP Data	Places a copy of the DSP data from the selected object into the Clipboard. DSP Data represents the current settings of that Component Object. See Component Object Properties . DSP Data can be copied from only one Component Object at a time. ***NOTE:*** *Software supports Copy/Paste DSP Data between objects which are of the same type, but which have different sized configurations. Examples: 4x4 & 8x8 Matrix Mixers; 3-band & 5-band Parametric EQ.*
Paste DSP Data	Places a copy of the DSP data from the Clipboard into the selected Component Object. DSP Data represents the current settings of that Component Object. See Component Object Properties . DSP Data can be pasted simultaneously into multiple Component Objects. ***NOTE:*** *Software supports Copy/Paste DSP Data between blocks which are of the same type, but which have different sized configurations. Examples: 4x4 & 8x8 Matrix Mixers; 3-band & 5-band Parametric EQ.*
DSP	Indicates the DSP resource required of the selected DSP objects.

Resources	Indicates DSP used in TesiraFORTÉ devices or DSP-2 Cards
Duplicate	Places a copy of the selected Component or Text Object directly into the Layout. Duplicate also places a copy of the object into the Clipboard. Duplicate works with only one object at a time.
Create Custom Block	Adds the selected DSP objects to a new Custom Block
Select All	Provides an easy way to select all objects on the Layout. Also allows all objects of a specific type to be selected. Also available via the Edit Menu
Order	Opens the Order Menu to move selected objects forward or back on the layout
Center in View	Centers the selected objects
Create New Preset	Creates and adds the selected objects to a New Preset
Add to Preset	Adds the selected objects to the current Preset
Preset Manager	Opens or closes the Preset Manager
Control Dialog	Opens the Control Dialog for the selected object.
Grid Birds Eye	Opens the Grid Birds Eye for processing object that support it. These include am mixer objects including Gating Auto Mixer, Gain Sharing Auto Mixer, Standard Mixer and Matrix Mixer
Properties	Opens or closes the Property Sheet

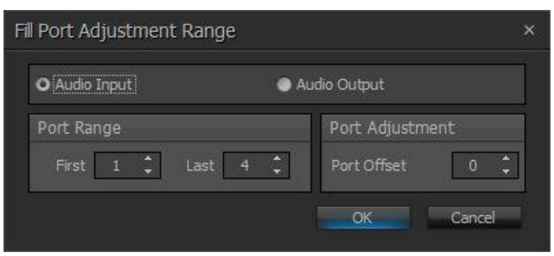
Port Properties

The Port Properties dialog can be used to review the channels of a component object.

- The Index is the channel number and is not adjustable.
- Text allows a custom name to be assigned to the channel
- Hover tip allows a custom name to be assigned when selecting the component object node
- Adjustment allows the node to be spaced for easier line drawing
- Flow text will propagate the Text to component objects upstream.



- **Check All** - Will check all the Flow text check boxes
- **Uncheck All** - will uncheck all the flow text boxes
- **Fill** will open the Fill Port Adjustment dialog to select multiple ports adjustments



Toolbars

Standard Toolbar

The Standard Toolbar provides file functions such as New, Open, & Save, plus additional functions such as New Partition, Open Partition, Close Partition, Delete Partition, Rename Partition, Cut, Copy, Paste, Undo, Redo, Print, & Help. The Standard Toolbar also includes the Compile and Compile All functions, which generate a new configuration file, while checking system layout/connections & determining DSP resource allocation. See File Menu and Edit Menu for more options.

Icon	Name	Description
	New (Ctrl + N)	Begins a new **Tesira**® system design file (.TMF).
	Open (Ctrl + O)	Opens an existing **Tesira**® system design file (.TMF).
	Save (Ctrl + S)	Saves the working **Tesira**® system design file (.TMF) in its current location. The default file path is \my documents\Biamp\Tesira\DataFiles.
	New Partition	Creates a new **Tesira**® DSP partition, which creates a new tab in the working system layout.
	Open Partition	Opens an existing **Tesira**® DSP partition.
	Close Partition	Closes an open **Tesira**® DSP partition.
	Delete Partition	Deletes an existing **Tesira**® DSP partition.
	Rename Partition	Renames a **Tesira**® DSP partition.
	Compile	Provides system design analysis and calculates DSP processing requirements for the selected partition in the current system layout. In this operation, determinations are made of the quantity/type of **Tesira**® devices needed, **AVB** channel assignments, and allocation of DSP resources. The compiler also provides indication of system design errors.
	Compile All	Compiles ALL partitions within the current system layout. In this operation, determinations are made of the quantity/type of **Tesira**® devices needed, **AVB** channel assignments, and allocation of DSP resources. The compiler also provides indication of system design errors.
	Cut	Removes the selected object(s) from the Layout, and places them in the clipboard.
	Copy	Places a copy of the selected object(s) into the clipboard.
	Paste	Places a copy of the object(s) from the clipboard into the Layout.

	Undo	Will undo last operation.
	Redo	Will redo most recent Undo operation
	Print	Opens a print dialog box, to adjust printer settings and print the Layout.
	Help	Opens the Tesira Help File - You're looking at it.

Network Toolbar

The Network Toolbar provides functions related to communications, configuration, maintenance, and testing of the system network (see System Network Considerations).

Network Toolbar function includes: Connect To System, Disconnect From System, Send Configuration, Start Audio, Stop Audio, and Device Maintenance.

Icon	Name	Description
	Connect To System	Establishes communication with, and retrieves data from, selected **Tesira**® systems on the network. Opens System Connect dialog box (see System Network Considerations). Password protection is then available from the Tools Menu. When connected to a system, Component Object Properties may be changed, but system design (objects & connections) cannot.
	Disconnect From System	Ends communications with selected **Tesira**® systems on the network. System design data is retained in software after disconnect. See System Network Considerations
	Send Configuration	Transmits system design data to selected **Tesira**® devices in the system. See System Network Considerations. Before data can be transmitted, a system design file (.TMF) must first be opened, then connected to a system, and have **Tesira** device IP addresses assigned (see Device Maintenance). Send Configuration will automatically Compile the system design, and reset the hardware devices, before sending the new configuration.
	Start Audio	Enables audio signal flow within the selected system. Start Audio is available only after Send Configuration has been successfully performed. See System Network Considerations.
	Stop Audio	Disables audio signal flow within the selected system. See System Network Considerations.
	Device Maintenance	Provides an editable table of network related settings for selected **Tesira**® devices. Opens Device Maintenance dialog box. Device Maintenance settings include Date/Time, IP Address, Description, Serial Number, Reset/Initialize, Update Firmware, and Disconnect From Network. See System Network Considerations

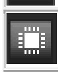
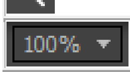
Format Toolbar

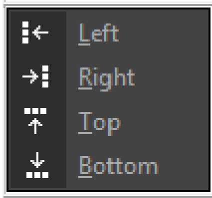
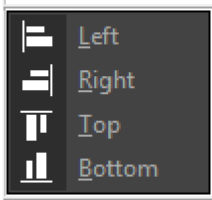
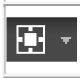
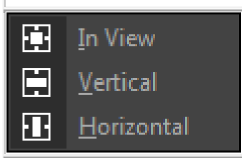
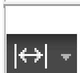
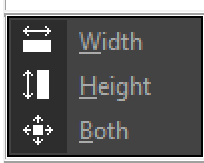
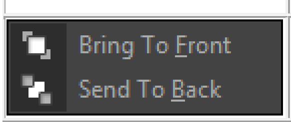
The Format Toolbar allows customizing of text and colors used in the Layout and associated Component, Line and Text Objects. The available tools are: Font, Size, Bold, Italic, Align Left, Center, Align Right, Background Color, Text Color, Border Color. The Format Toolbar may be opened and closed from the View Menu. The location and shape of all toolbars may be changed to fit the user's preference.

Icon	Name	Description
	Font	Provides a drop-down menu of lettering styles for use in Component or Text Objects.
	Size	Provides a drop-down menu of lettering sizes for use in Component or Text Objects.
	Bold	Changes the selected text to a thicker version of the chosen font.
	Italics	Changes the selected text to an italicized version of the chosen font.
	Align Left	Aligns the selected text to the left margin of the Component or Text Object.
	Center	Centers the selected text between the left and right margins of the Component or Text Object.
	Align Right	Aligns the selected text to the right margin of the Component or Text Object.
	Background Color	Provides a drop-down menu for changing the background color of the Layout, or of selected Component or Text Objects. Left-clicking the icon applies the color which was selected last.
	Text Color	Provides a drop-down menu for changing the text color in selected Component or Text Objects. Left-clicking the icon applies the color which was selected last.
	Border Color	Provides a drop-down menu for changing the border color around selected Component or Text Objects. Left-clicking the icon applies the color which was selected last.

Layout Toolbar

The Layout Toolbar affects certain aspects of Layout such as the grid, rulers, zooming, and alignment of Component Objects. The Layout Toolbar can also open editing/information sheets for properties, objects, & layers. The available tools are: Property Sheet, System View, Bird's Eye View, Output Window, Processing Library, Preset Manager, Object ID Inspector, Layers Sheet, Toggle Rule, Toggle Grid, Zoom In, Zoom Out, Zoom 1:1, Zoom Level, Pack Objects, Align Edges, Center In View, Space Make Same Size, and Order. The Layout Toolbar may be opened/closed from the View Menu. The location and shape of all toolbars may be changed to fit the user's preference.

Icon	Name	Description
	Property Sheet	Provides an editable table of attributes regarding the Layout and its associated objects. Only Display Attributes are shown for the Layout (and Lines). Both Display and DSP Attributes are shown for Component Objects. See Layout Property Sheet, Line Property Sheet, Object Property Sheet
	System View	Opens the System View Sheet that gives an overview of the partitions in a system and how they connect
	Birds Eye View	The Bird's Eye View sheet allows one to see an overview of the layout of the selected tab in the main application window
	Output Window	The Output Window allows the user to see information about system hardware and software.
	Processing Library	The Processing Library allows for organization of DSP objects into catalogs for easy addition to a layout.
	Preset Manager	Opens the Preset Manager sheet.
	DSP resources	Opens the DSP Resources Docking Window
	Device Import/Export	Opens the Device Import/Export Docking Window
	Object ID Inspector	The Object ID Inspector reveals the details of each DSP block in the system including Partition blocks. Review the Object ID inspector details for more information.
	Layers	Opens the Layers Sheet and provides an editable table of Layer properties.
	Toggle Ruler	Turns Layout Ruler on/off.
	Toggle Grid	Turns Layout Grid on/off.
	Zoom In	Increases magnification of Layout in 25% increments.
	Zoom Out	Decreases magnification of Layout in 25% increments.
	Zoom 1:1	Returns magnification of Layout to 100%
	Zoom Level	Provides a drop-down menu of available Zoom

		magnifications (50% ~ 200% in 25% increments).
	Pack Objects	Provides a drop-down menu for packing selected objects next to each other. The target location for packing left/right is the top-most selected object. The target location for packing top/bottom is the left-most selected object.
	Pack Objects sub-menu	The icon changes to indicate the packing reference last selected, which becomes the default choice the next time it is used.
	Align Edges	Provides a drop-down menu for aligning the edges of selected objects. The primary selected object (green handles) provides the target edges for alignment.
	Align edges sub-menu	The icon changes to indicate the alignment reference last selected, which becomes the default choice the next time it is used.
	Center In View	Provides a drop-down menu for centering the Layout view on the selected objects.
	Center in View sub-menu	The icon changes to indicate the centering reference last selected, which becomes the default choice the next time it is used.
	Space	Provides a drop-down menu for evenly spacing selected objects. Spacing is determined between the two most distantly spaced selected objects.
	Spacing sub-menu	The icon changes to indicate the spacing reference last selected, which becomes the default choice the next time it is used.
	Make same Size	Provides a drop-down menu for matching the dimensions of selected objects. The primary selected object (green handles) provides the target dimensions. Objects cannot be smaller than original size.
	Make Same Size sub-menu	The icon changes to indicate the sizing reference last selected, which becomes the default choice the next time it is used.
	To Front or Back / Order	Provides a drop-down menu for changing the order in which overlapping objects appear on the Layout. The selected object(s) will either move in front of overlying objects, or will move behind underlying objects.
	Ordering sub-menu	The icon changes to indicate the reference last selected, which becomes the default choice the next time it is used.

Object Toolbar

The Object Toolbar, located directly above the Layout, allows Component Object selection for placement into the Layout. The Component Objects are organized in the following categories: Graphic Elements, Input Output, Mixers, Equalizers, Filters, Crossovers, Dynamics, Routers, Delays, Controls, Meters, Generators, Logic, Diagnostics, and Specialty.

Each category is represented by an icon, with a drop-down menu to the right. To place a Component Object, first choose the appropriate category and then select the desired component from the drop-down menu. Once the component has been selected, simply left-click at the desired location on the Layout. Left-clicking a category icon will select the component which occurs first in the menu list. When using the Object Toolbar to place components into the Layout, certain components will present the user with a pop-up window of configuration options.

In addition to DSP blocks, the Object Toolbar provides the selection pointer (cursor) and graphic elements. The selection pointer is the default selection for the application. Graphic Element can be placed in the layout just as the DSP blocks. The text can be edited by a right-click on the graphic element and then choosing Edit Text from the drop-down menu.

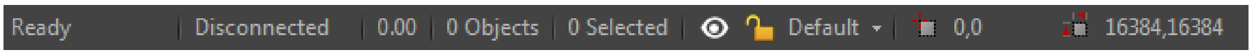
The Object Toolbar may be opened/closed from the View Menu. The location and shape of all toolbars may be changed to fit the user's preference.

Status Bar

The Status Bar, along the bottom of the main screen, provides system information. The left side of the Status Bar indicates system status, tool tips, instance ID's or flash update information. The center of the Status Bar indicates system/network connection status (with progress bar) and session privilege level. Prior to Tesira 2.0 the status bar of each partition there was a number that represented DSP Usage. This has been replaced by a DSP Resources Docking Window.

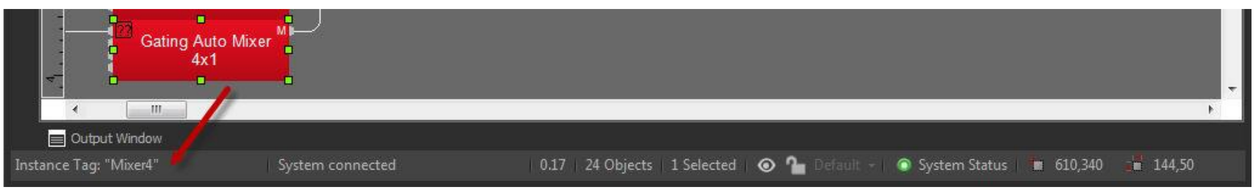
The right side of the Status Bar indicates total number of system objects, number of objects selected, Layer status, last selected object location (pixels), and last selected object size (pixels).

Status bar when disconnected:



Status Bar when connected:

When connected to a system a **System Status** dialog is shown. Selecting this will open to display any System Status and Fault messages.

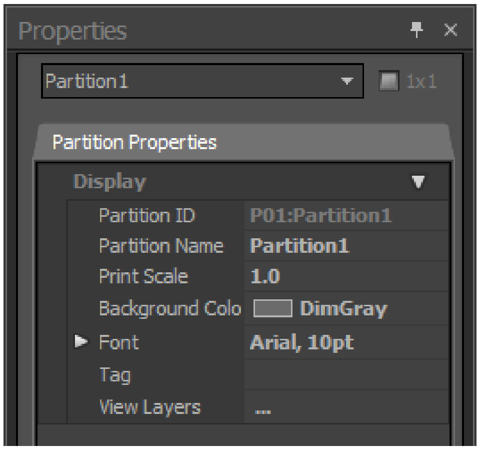


When online - selecting a processing block will show the Instance ID in the Left hand Corner of the Status bar. See the TTP Overview section.

Docking Windows

Property Sheets

Layout Property Sheet

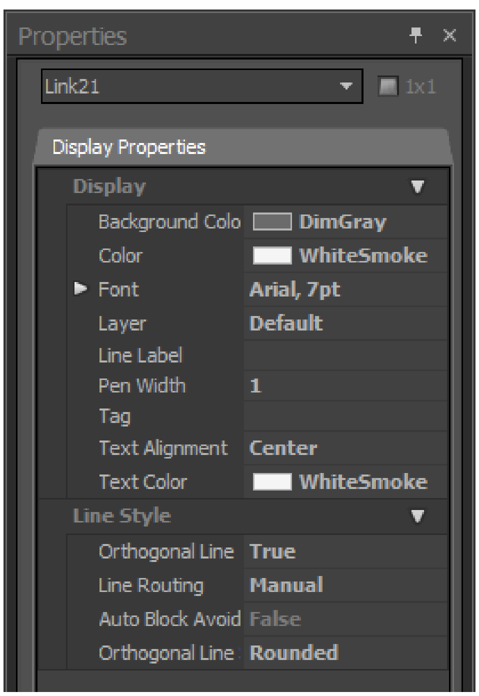


Provides an editable table of attributes regarding the Layout of the current partition. Only Display Attributes are shown for the Layout. Most Display Attributes duplicate functions found in the Format Toolbar However, some exceptions are as follows:

- **PrintScale** is entered as decimal information, where 0.5 = 50% or 2.0 = 200%.
- **ViewLayers** opens the Layers Sheet.
- **Tag** is for user comments or other identifying text.

A **Menu** icon (upper-right of title-bar) allows the Property Sheet to be docked or floating, and to be hidden (closed) or to utilize Auto Hide (if docked). The menu may also be accessed by right-clicking over the Property Sheet.
The **thumb-tack** allows the Property Sheet to remain open while selecting other objects (disables Auto Hide).
1By1 allows any group of selected components (multi-selection) to appear on the menu.

Line Property Sheet

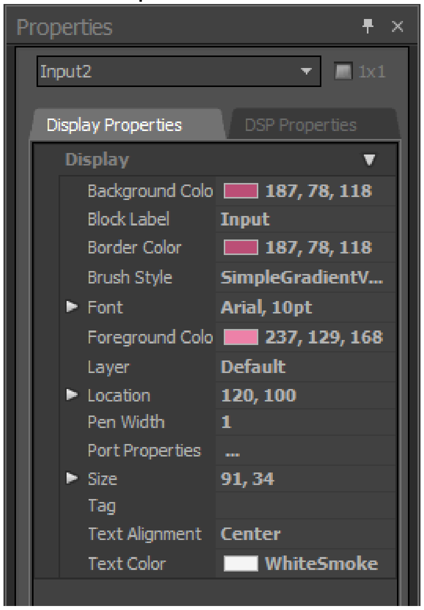


Only Display Attributes are shown for Lines (component connections). Most Display Attributes duplicate functions found in the Format Toolbar. However, some exceptions are as follows:

- **Layer** determines which layer the Line is assigned to.
- **Line Label** allows identifying text to be placed on the line itself.
- **Tag** is for user comments or other identifying text.
- **Font** and **Alignment** affect the line text.

A **Menu** icon (upper-right of title-bar) allows the Property Sheet to be docked or floating, and to be hidden (closed) or to utilize Auto Hide (if docked). The menu may also be accessed by right-clicking over the Property Sheet.
The **thumb-tack** allows the Property Sheet to remain open while selecting other objects (disables Auto Hide).
1By1 allows any group of selected components (multi-selection) to appear on the menu.

Object Property Sheet



Both Display and DSP Attributes are shown for component objects.

Most Display Attributes duplicate functions found in the Format Toolbar. However, some exceptions are as follows:

- **Brush Style** specifies the look of the object whether it be solid or one of a number of gradient patterns.
- **Layer** determines which layer the component is assigned to.
- **Location** specifies object location (top left in x,y pixels).
- **Pen Width** selects the weight of the border around the object.
- **Port Properties** brings up a dialog in which individual input and output ports can be labeled. The dialog provides a text label for each port on the object, a hover tip (seen when one mouse's over the connection node) and a spacing adjustment.
- **Flow Text** propagates the entered text labels throughout the signal chain.
- **Tag** is for user comments or other identifying text.

Most DSP Attributes duplicate functions found in the individual component Control Dialog boxes (see Component Object Properties). However, some exceptions are as follows:

- **Allocated To Unit** assigns the component (DSP block) to a particular unit in the system.
- **Fixed In Unit** prevents changes to Allocated Unit.
- **Instance Tag** allows a custom name to be used in lieu of the Instance ID number.
- **Instance ID** is the system-wide identifier number for the component (DSP block).

A **Menu** icon (upper-right of title-bar) allows the Property Sheet to be docked or floating, and to be hidden (closed) or to utilize Auto Hide (if docked). The menu may also be accessed by right-clicking over the Property Sheet.

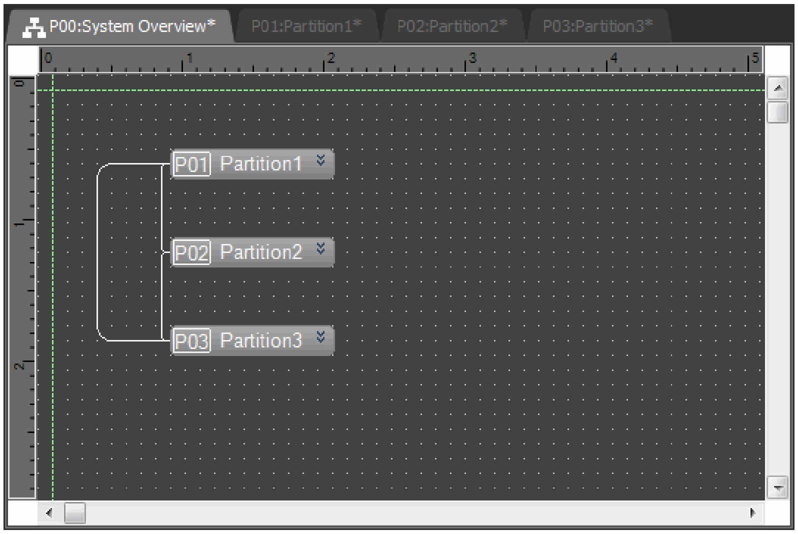
The **thumb-tack** allows the Property Sheet to remain open while selecting other objects (disables Auto Hide).

1By1 allows any group of selected components (multi-selection) to appear on the menu.

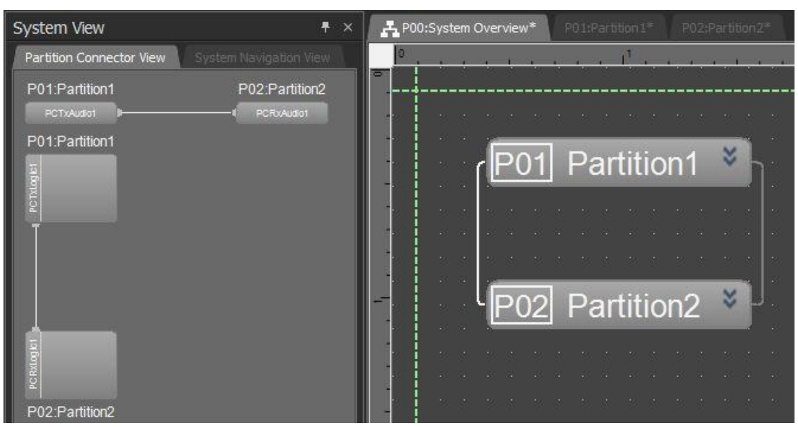
System View

System View Sheet

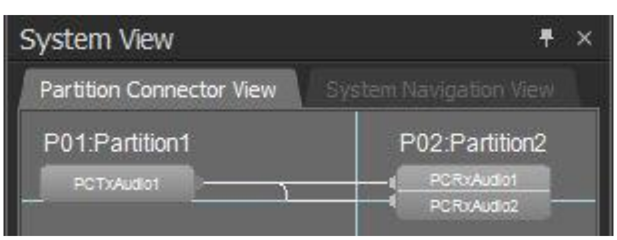
The **System View** sheet shows the partitions included in a system and how they connect. The Partition Connector View shows each partition as a block and the routing between partitions. The System Navigation View allows one to select the area seen in the System Overview Tab in the main application window. This is done by dragging the arrowed icon up, down, right or left.



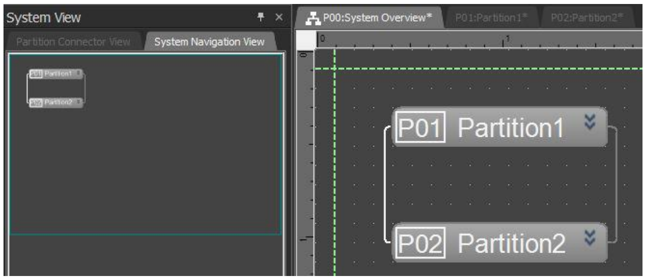
Partition Connector View lists the audio and logic connections between partitions and gives details of how any partitions are configured and any audio or logic interconnects that are in use.



The Partition Connector view allows nodes to be wired and associated from one partition to another in a similar way to wiring between DSP objects. The use of Partition Connector Audio Transmitter, Partition Connector Audio Receiver, Partition Connector Logic Transmitter, Partition Connector Logic Receiver is necessary to facilitate these interconnects.



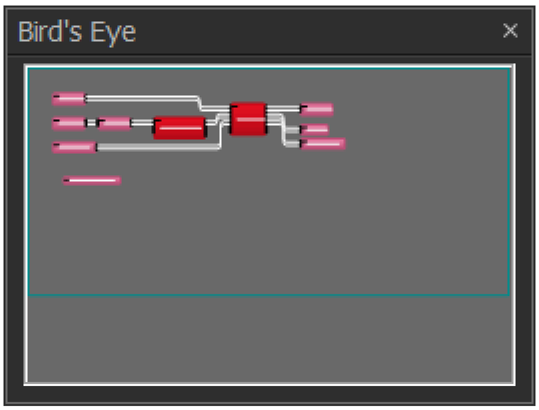
System Navigation View can be used to give a bird's eye indication of all partitions being used



Birds Eye View

Bird’s Eye View Sheet

The Bird's Eye View provides a thumb-nail sketch of the entire system design, to aid navigation within the Layout. Bird's Eye View initially covers an area of only 8" x 5", but will automatically increase size to cover a larger system design. A black rectangle frames the viewable area. The rectangle may be dragged to view any location within the system design. The rectangle also has handles to re-size the viewable area, from a minimum of 4" x 2.5" to a maximum of 16" x 10". The Bird’s Eye View can be made dockable. If Bird’s Eye View docking is enabled, a double-click anywhere in the title bar will restore it to the last docked position. To choose a docking position, drag the Bird’s Eye Viewer until positioning windows appear over the layout. Dragging the Bird’s Eye Viewer to the top, bottom, left or right positioning window chooses the docked orientation. To auto hide the Bird’s Eye Viewer, click on the thumb-tack to the top right of the title bar. A click on the X closes the Bird’s Eye View.

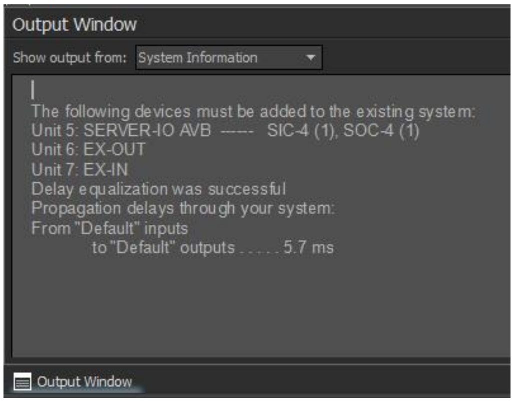


Output Window

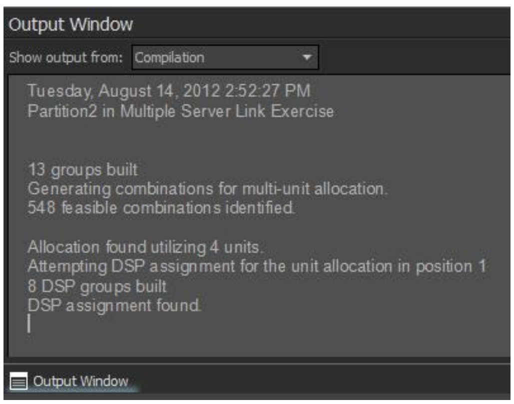
Output Window

The Output Window allows the user to see information about system hardware and software.

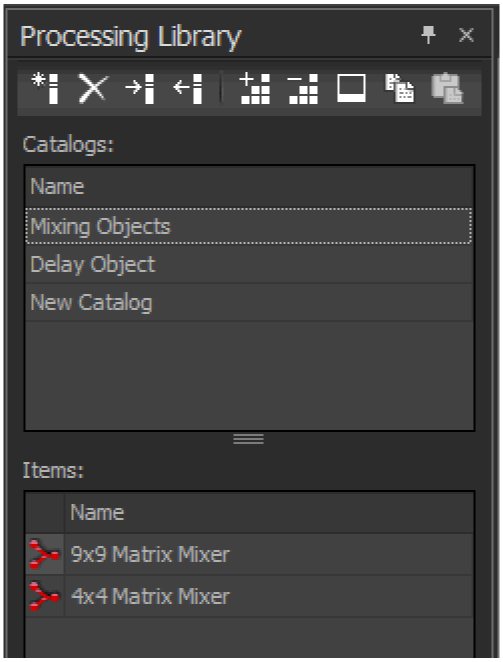
System Information shows information on the hardware layout and latency of the system.



Compilation shows any errors or comments generated from the last system compilation.

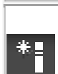
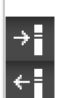
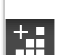


Processing Library View Sheet



The Processing Library allows for organization of DSP objects into catalogs for easy addition to a layout. Within the Processing Library View Sheet, one has the option to Create, Delete, Import and Export catalogs as well as add/remove object within a selected catalog, and copy/paste object to/from the clipboard. The upper field of the sheet displays the list of available catalogs. The lower field displays the object within a catalog selected in the upper field (mouse click to select).

The Processing Library can be made dockable. If Processing Library docking is enabled, a double-click anywhere in the title bar will restore it to the last docked position. To choose a docking position, drag the Processing Library until positioning windows appear over the layout. Dragging the Processing Library to the top, bottom, left or right positioning window chooses the docked orientation. To auto-hide the Processing Library, click on the thumb-tack to the top right of the title bar. A click on the X closes the Processing Library.

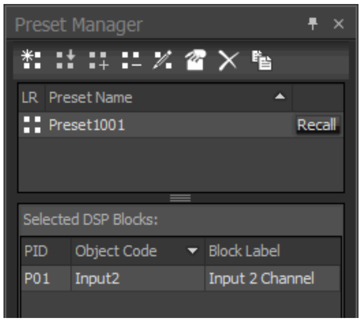
Icon	Name	Description
	New Catalog	This adds a catalog in the Catalogs field named ‘New Catalog’. By clicking on the new heading, the user can give the New Catalog a name.
	Delete Catalog	To delete a catalog, select one in the Catalogs field and hit Delete Catalog.
	Import/Export Catalog	Tesira allows processing catalogs to be imported and exported as files with the .tlf extension. To import a catalog, click the Import icon. Browse for the Catalog file (.tlf file) to import and select it. The Catalog will then appear in the list. To export, select a catalog and click on the Export button. Name the file and save it to storage.
	Item from Selection	Items can be added to a selected catalog by selecting an object in the main program window and clicking on the Item from Selection Icon. If multiple objects are selected, they will be added to the catalog as a block, meaning they can only be added to a layout in that combination. Selected items can also be dragged into a catalog by holding down the CTRL key while dragging them into the Items field.
	Delete Item	A selected item can be deleted from a catalog by clicking on the Delete Item Icon. This can also be done by right-clicking on the item in the list and choosing the Delete option from the pop-up menu.
	Change Image	To change the image associated with a specific catalog item, select the item (mouse click) and click the Change Image Icon. The standard object images (matching object toolbar images) will appear along with an Import Image selection. Note that imported images are limited to 16x16 pixels. All larger images will be

		clipped to 16x16 starting from top left. Supported file formats are .bmp, .jpeg and .png.
	Item to Clipboard	To place multiples of a catalog item into a layout, select the item and click on the Item to Clipboard Icon. The item can now be placed in the layout by using Paste from the edit menu (CTRL+V).
	Paste from Clipboard	To place and item from the clipboard into a catalog, select the item in the layout and copy it (CTRL+C), Copy from the Edit menu or copy from right-click on selected object. Paste the object into the selected catalog by clicking on the Paste from Clipboard Icon.

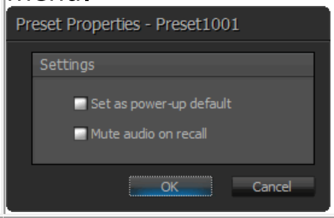
Preset Manager

Preset Manager View Sheet

The Preset Manager View Sheet is used to create, edit and recall system presets. The upper field in the Preset Manager View Sheet shows each preset by name. The field below, Selected DSP Blocks, shows the DSP objects on which a selected (mouse click to select) preset operates. A toolbar at the top of the sheet gives the user several options.



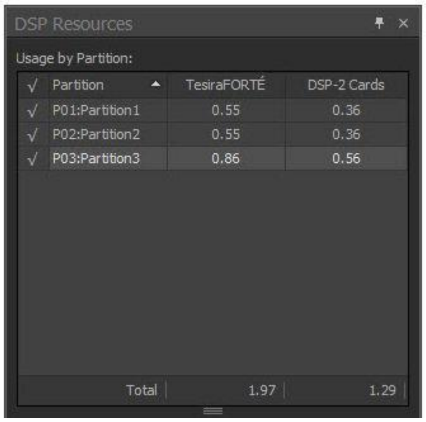
Icon	Name	Description
	Create New Preset	This adds a preset in the Preset field with a numerical name starting with 'Preset1001'. By clicking on the name of the new entry, the user can give the new preset a name.
	Save Last Recalled Preset	To save the state of a recalled preset after editing, click the Save Last Recalled Preset Icon
	Add Selected DSP Blocks	To add DSP objects to a preset, first select the preset in the upper field of the Preset Manager View Sheet (mouse click). Select one or more DSP objects in the DSP layout (CTRL+mouse click for multiples) so that each desired object is highlighted. Click the Add Selected DSP Blocks Icon to add the selection to the list of objects being controlled. The object(s) can also be selected and added to the list by holding CTRL while dragging the selection to the list from the layout.
	Remove DSP Blocks	To remove a DSP object from a selected preset, select the object in the Selected DSP Blocks field of the Preset Manager View Sheet. Click the Remove DSP Blocks Icon. Alternately, an object or all objects can be removed by right-clicking on an object in the Selected DSP Objects field and choosing Remove from Preset or Remove All from the resulting pop-up menu.
	Select Block Attributes	To keep certain controls of a DSP object from being altered by a preset, they must be omitted from the block attributes to be controlled. To edit this, select (mouse click) a block from the Selected DSP Blocks field for a given preset and hit the Select Block Attributes Icon. A dialog will appear of the available control attributes of that block. By default all attributes are controlled by the preset – to exclude a control, uncheck the attribute in the dialog. Select All and Clear All buttons exist on the bottom left of the dialog to aid in selection / de-selection of multiple items. A right-click on an Object in the Selected DSP Blocks field will result in a pop-up menu where Select Block Attributes is also available.
	Preset Properties	With a given selected preset, clicking on the Preset Properties Icon will result in a Preset Properties dialog. In this dialog one can choose the preset as the power-up default and whether to mute audio when the preset is recalled. This can also be done by right-clicking on a preset

		in the preset field and choosing properties from the resulting pop-up menu. 
	**Delete Preset**	To delete a preset, select the preset and click the Delete Preset Icon. This and all presets can also be deleted by right-clicking on a preset and choosing Delete Preset or Delete All Presets from the pop-up menu.
	**Duplicate Last Recalled Preset**	To create a new preset based on an existing one, recall a preset and click on the Duplicate Last Recalled Preset. Any preset is recalled by clicking the Recall in the same line as the preset name. Duplicated presets contain the same DSP blocks, attributes and control parameters of the original object. The preset properties (power-up/mute) are not duplicated.

If any selections are made in the properties dialogue for a preset, the properties are then shown in parentheses by the preset name (power-up/mute)

DSP Resources

Once a Compile is successful, the DSP allocation can be reviewed via the DSP Resources Docking Window. The Usage by Partition section shows a table which summarizes the current DSP required for each partition. The DSP Resources window has two parts, a Usage by Partition table on the top and a Usage by Device chart below it.
Usage by Partition
The **DSP resources Usage by Partition** table sums the required resources for each block of a partition if the block is able to be included in a compile. Blocks that are not included in a compile (as they are unconnected, for example) will not have their resources included. The table will display how many TesiraFORTÉ or DSP-2 cards would be required for that partition. This calculation is made live as lines are connected / disconnected. The check mark indicates the compiled state of a partition.



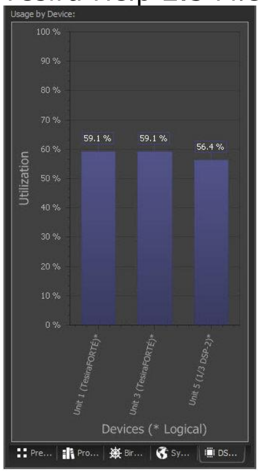
Name	Description	Scale
	Partition is compiled	Blank or ticked
Partition	Partition number and name	Pxx:<Partition_name>
TesiraFORTÉ	Percentage of DSP required if using a Forte Device	0.00 to 1.00 per device
DSP-2 Cards	Percentage of DSP required if using one or more DSP-2 cards	0.00 to 1.00 per card

Usage by Device
The DSP resources chart describes usage per device. This chart will only display information for compiled partitions. If the layout only has some partitions compiled the Usage by Device chart will only show information for the partitions that have been compiled. This chart reflects the current equipment in a system (according to the equipment table). It will also display DSP utilization of that equipment as partitions are compiled. When a compiled partition is altered (line added/removed), the DSP resource allocation for that partition is removed until the next compilation.

Based on the devices populated in the Equipment table:

- A TesiraFORTÉ device uses a fixed DSP resource, so each device used is shown as 1 logical device.
- A Server IO uses an expandable DSP resource. It has up to 3 DSP-2 Cards indicated as 1-3 of 3 used (shown as 1/3 DSP-2 to a maximum of 3/3 used)
- A Server uses an expandable DSP resource. It has up to 8 DSP-2 Cards indicated as 1-8 of 8 used (shown as 1/8 DSP-2 to a maximum of 8/8 used)

The Scale percentage indicates a 0 to 100% utilization value of the DSP used based on the available DSP Resource in the device.

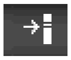


Name	Description	Scale
Utilization	A scale of how much resource is available on the unit	0 - 100% of available DSP
Devices	A description of the device used and DSP resource required	Forte, 1/3 DSP-2, 1/8 DSP-2

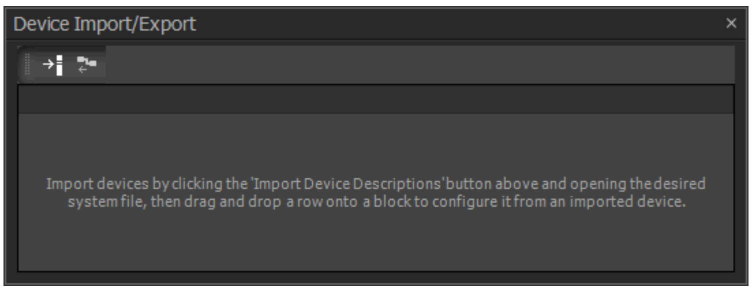
Device Import/Export

Device Import/Export

The **Device Import/Export** sheet allows importing and exporting of a LabGruppen System File.

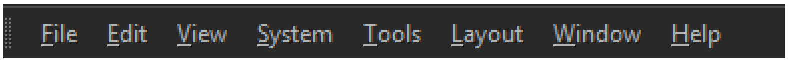
 The Import Icon opens a Import Device List dialog which allows users to browse to the file location where the saved LabGruppen System File is stored. The LabGruppen System File is created using the external Labgruppen Cafe Software suite.

The Export Icon is available when connected to the system. It allows the ability to export information about the labgruppen blocks into a file which can then be imported to the external Labgruppen Cafe Software suite.



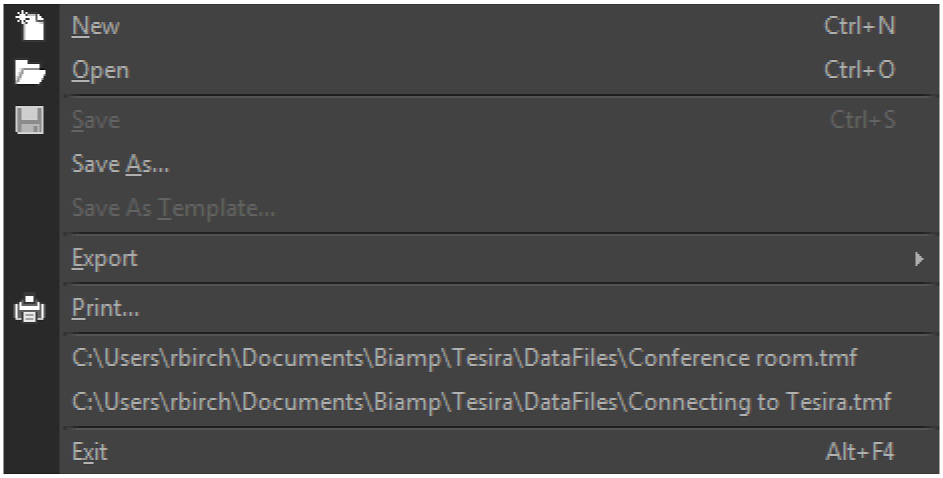
Main Menus

Main Menus



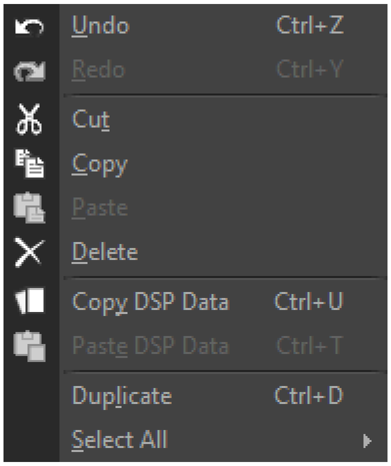
The Main Menus provide most of the toolbar functions mentioned previously, in Basic Screen Elements, as well as several more in-depth functions. Main Menus includes the following individual menus: File; Edit; View; System ; Tools; Layout; Window; and Help. Keyboard shortcuts are shown on the menus, where applicable.

File Menu



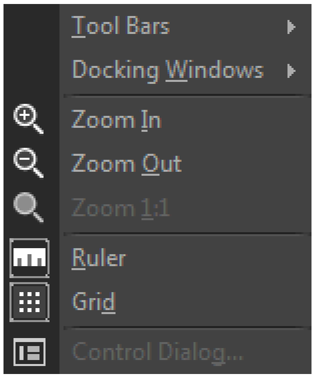
Icon	Name	Description
	New	Begins a new **Tesira**® system design file (.TMF). The user may be asked to specify the Software Mode required.
	Open	Opens an existing **Tesira**® system design file (.TMF).
	Save	Saves the current **Tesira**® system design file (.TMF). The default file path is \My Documents\Biamp\Tesira\Datafiles\
	Save As	Saves the current **Tesira**® system design file (.TMF), with choice of directory location and file name.
	Save As Template	Saves the current **Tesira**® system design file (.TMF) without equipment table and system information so it can be recalled later for use with another set of hardware. More details are available in the Using Templates section.
	Export	Export Menu items save the DSP layout of the active partition to .DXF (drawing interchange format) or .EMF (enhanced metafile file) format.
	Print	Opens a print dialog box, to adjust printer settings and print the Layout. The system layout and any/all partitions are available to print.
	Recent File	Provides a list of recently saved files for convenient access.
	Exit	Closes the **Tesira**® software program. Also provides prompt to save the current **Tesira** system design file if necessary. If a file is open during Exit, that file will automatically open at next session.

Edit Menu



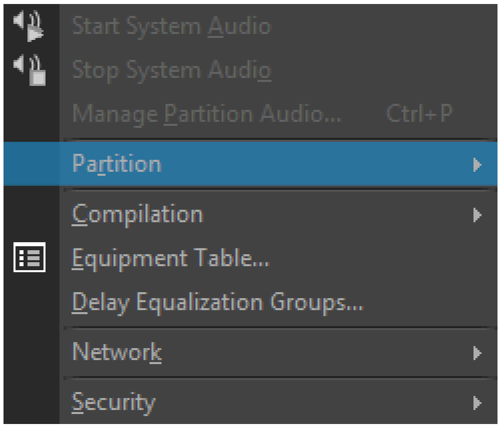
Icon	Name	Description
	Undo	Will undo last operation.
	Redo	Will redo most recent Undo operation.
	Cut	Removes the selected object(s) from the Layout, and places them in the Clipboard.
	Copy	Places a copy of the selected object(s) into the Clipboard.
	Paste	Places a copy of the object(s) from the Clipboard into the Layout.
	Delete	Removes the selected object(s) from the Layout, without placing a copy into the Clipboard.
	Copy DSP Data	Places a copy of the DSP data from the selected object into the Clipboard. DSP Data represents the current settings of that Component Object. See Component Object Properties. DSP Data can be copied from only one Component Object at a time. ***NOTE:*** *Software supports Copy/Paste DSP Data between objects which are of the same type, but which have different sized configurations. Examples: 4x4 & 8x8 Matrix Mixers; 3-band & 5-band Parametric EQ.*
	Paste DSP Data	Places a copy of the DSP data from the Clipboard into the selected Component Object. DSP Data represents the current settings of that Component Object. See Component Object Properties. DSP Data can be pasted simultaneously into multiple Component Objects. ***NOTE:*** *Software supports Copy/Paste DSP Data between blocks which are of the same type, but which have different sized configurations. Examples: 4x4 & 8x8 Matrix Mixers; 3-band & 5-band Parametric EQ.*
	Duplicate	Places a copy of the selected Component or Text Object directly into the Layout. Duplicate also places a copy of the object into the Clipboard. Duplicate works with only one object at a time.
	Select All	Provides an easy way to select all objects on the Layout. Also allows all objects of a specific type to be selected.

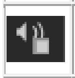
View Menu



Icon	Name	Description
	Toolbars	Provides a drop-down menu to turn on/off any of the toolbars (see Basic Screen Elements). Also, the appearance and functionality of new and existing toolbars and menus can be 'Customized'.
	Docking Windows	Provides a drop-down menu to view/hide the docking menu items such as the Layout Property Sheet, Birds Eye View, Output view, Processing Library, Preset Manager, DSP Resources and Device Import/Export
	Zoom In	Magnifies the active layout view.
	Zoom Out	Reduces the active layout view.
	Zoom 1:1	Restores the active layout view to its native size.
	Ruler	Toggles the layout ruler view.
	Grid	Toggles the layout grid view.
	Control Dialog	Reveals the control dialog options of a selected DSP object.

System Menu



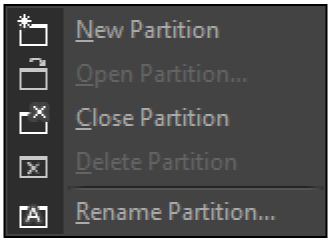
Icon	Name	Description
	**Start System Audio**	Enables audio signal flow within the selected system. Start Audio is available only after Send Configuration has been successfully performed. See System Network Considerations.
	**Stop System Audio**	Disables audio signal flow within the selected system. See System Network Considerations.
	Manage Partition Audio	Allows selective start/stop audio for each partition in the layout.
	Partition	Opens the Partition menu options
	Compilation	Allows different compilation tasks to be selected see the Compilation section for more details
	**Equipment Table**	Provides an editable table of **Tesira**® devices in the system design. **Add Unit** and **Remove Unit** can be used to manually change the hardware design. Please review the Equipment table section for more details.
	Delay Equalization Groups	Provides a way to group signal paths that are sensitive to small offsets in delay. Please review the Delay Equalization Groups section for more details.
	Network	Provides an editable table of network related settings for selected **Tesira**® devices. Provides the same networking functions as found on the Network Bar. See the Network section for more details
	Security	Shows the **System Security** and **Change Own Password** Option. System Security allows a system to be password protected. **Change Own Password c**hanges the password of the user currently logged on to the system. See System Security Section form more details.

System Menu Items

Partition Menu

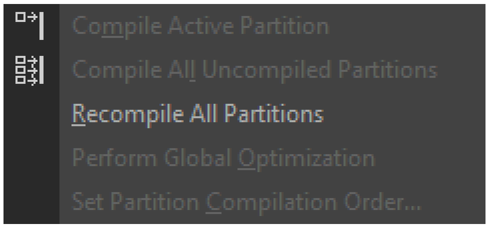
Partitions allow a configuration file to be segmented into different sections. Please review the Partitions section for more information.

Partition Menu



Icon	Name	Description
	New Partition	Creates a new tab in the system layout for an audio partition.
	Open Partition	Opens an existing partition in the system that has been closed.
	Close Partition	Closes and open audio partition within the system layout.
	Delete Partition	Deletes an audio partition from the system layout.
	Rename Partition	Renames an audio partition in the system layout.

Compilation



Compile Active Partitions

Compiles the DSP layout for the active partition in the overall system without regard to other partitions. In the compilation process the blocks are analyzed, checked against the system hardware (systems specified in the equipment table) and formatted to be sent to that hardware. The analysis and any logical errors in the system blocks are displayed in the output window. For more information on the process, see the Complier section.

Compile All Uncompiled Partitions

Compiles the DSP layout for all partitions in the entire system layout. In the compilation process the blocks are analyzed, checked against the system hardware (systems specified in the equipment table) and formatted to be sent to that hardware. The analysis and any logical errors in the system blocks are displayed in the output window. For more information on the process, see the Complier section.

Recompile All Partitions

Recompiles the DSP layout for all partitions in the entire system layout even if there have been no changes.

Perform Global Optimization

Compiles all uncompiled partitions and analyzes the compilation results to find an equivalent, lower cost equipment allocation, if one exists. By default, the compiler finds an optimum equipment list for each partition in the system individually, and the accumulated result of this may not be cost optimal across the entire configuration, particularly in systems having a large number of partitions. Perform Global Optimization may add or remove hardware from the Equipment Table if the list is functionally equivalent and lower cost. Devices in the Equipment Table with serial numbers assigned (physical devices) are not subject to removal by the optimization. Likewise, devices having DSP objects with fixed allocation will not be removed. If Global Optimization finds an improvement, the equipment table will be updated, and the entire layout will be recompiled. If no improvement is found, the previous compilation results are kept.

Set Partition Compilation Order

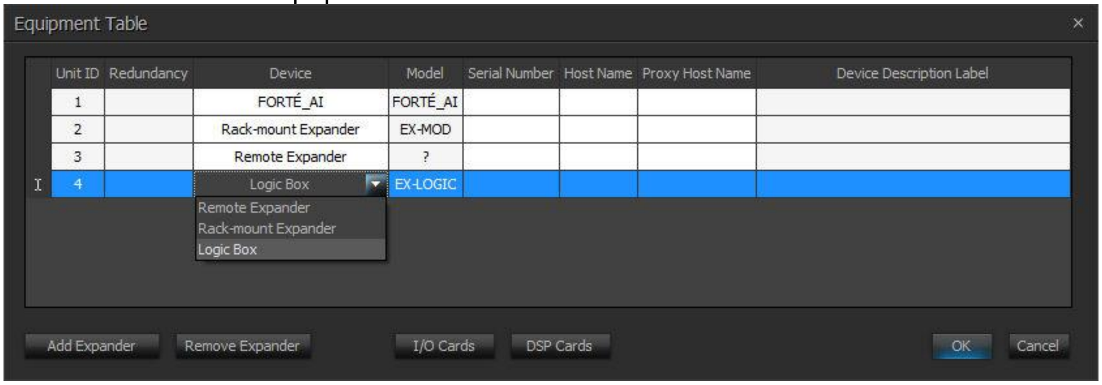
Manually sets the order in which partitions will be compiled when the **Recompile All Partitions** function is chosen. By default the partitions are compiled in the order in which they were created.

Equipment table

Correlates specific **Tesira**® hardware components to the system design. Compiling a system layout will populate the equipment table with the most efficient hardware solution for that system. The specific pieces of hardware (by serial number or hostname) can then be chosen for each component part. This must be done before a configuration can be sent to any Tesira devices.
Equipment table and modes.
The Equipment table will display different options depending on the software mode being used.

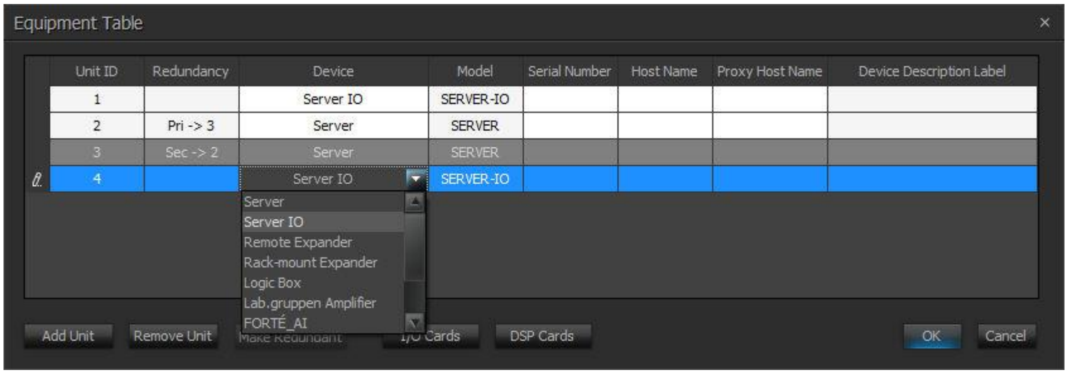
- **TesiraFORTÉ mode** - TesiraFORTÉ devices are added to the equipment table automatically when TesiraFORTÉ IO channels are specified in the layout. This is done by adding TesiraFORTÉ IO channels via the TesiraFORTÉ IO Component objects. Expander devices can be specified and the following devices are available - EXMOD, EX-In, EX-Out, EX-IO, Lab.gruppen Amplifier and EX-LOGIC devices.
- **Tesira SERVER mode** and **Both TesiraFORTÉ and SERVER Mode** - Allows addition of SERVER and SERVER IO devices. Allows for SERVER devices to be specified as redundant pairs. I/O Channels placed in the layout have an equipment filter to instruct the compiler to use certain hardware. TesiraFORTÉ devices are added to the equipment table automatically when TesiraFORTÉ IO channels are specified in the layout. This is done by adding TesiraFORTÉ IO channels via the TesiraFORTÉ IO Component objects. Expander devices can be specified and the following devices are available - EXMOD, EX-In, EX-Out, EX-IO, Lab.gruppen Amplifier and EX-LOGIC devices.

TesiraFORTÉ mode Equipment table



- Add Expander - Allows a expander device to be specified if IO is not already included in the layout.
- Remove Expander - will remove the selected expanders
- I/O cards - will give details of available IO channels
- DSP cards - will give information on the available DSP processing (TesiraFORTÉ, SERVER and SERVER IO only)

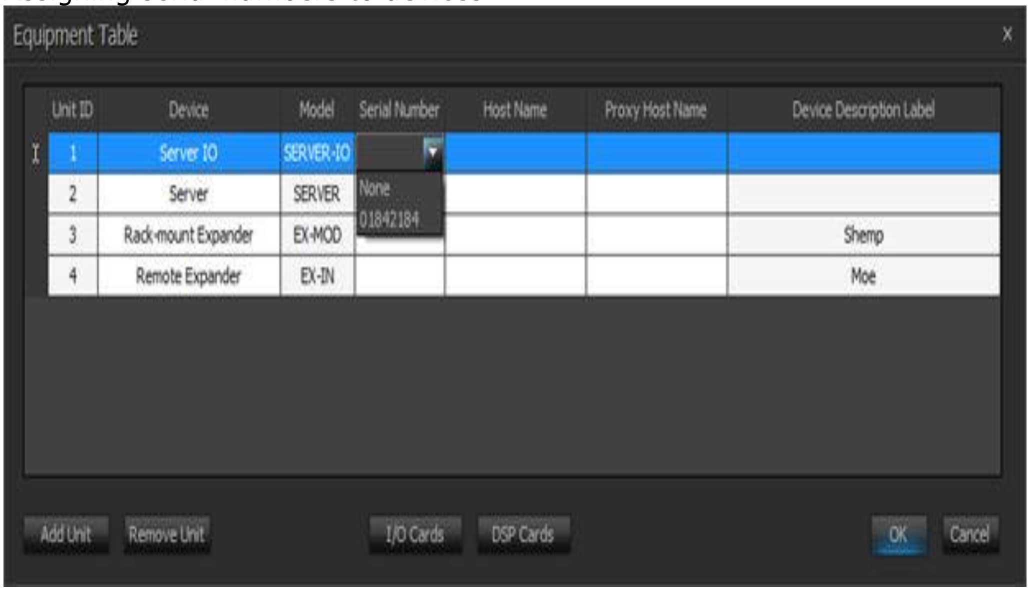
Tesira SERVER mode and Both TesiraFORTÉ and SERVER Mode



- Add Unit - Allows a Server, Server IO, Remote expander, Rack Mount Expander, Lab.gruppen Amplifier, Logic Box and any TesiraFORTÉ device to be specified.
- Remove Unit - will remove the selected expanders
- Make Redundant - If a SERVER device is in the equipment table, selecting it will make the **Make Redundant** option available. See Redundancy.

- I/O cards - will give details of available IO channels
- DSP cards - will give information on the available DSP processing (TesiraFORTÉ, SERVER and SERVER IO only)

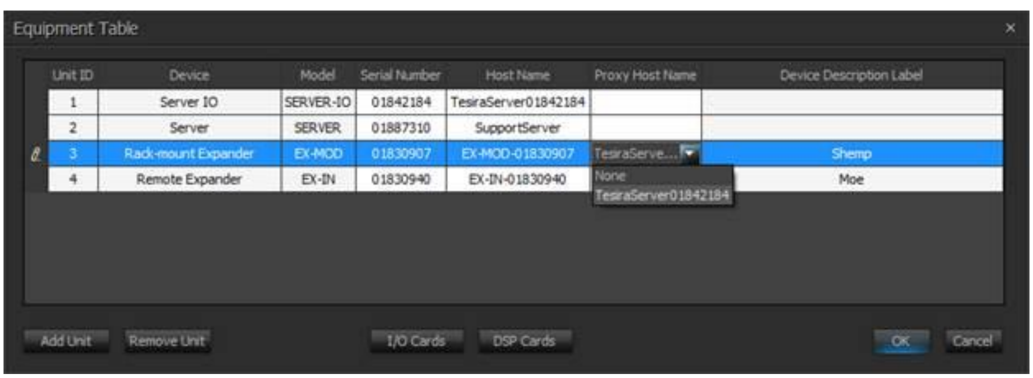
Assigning serial numbers to devices



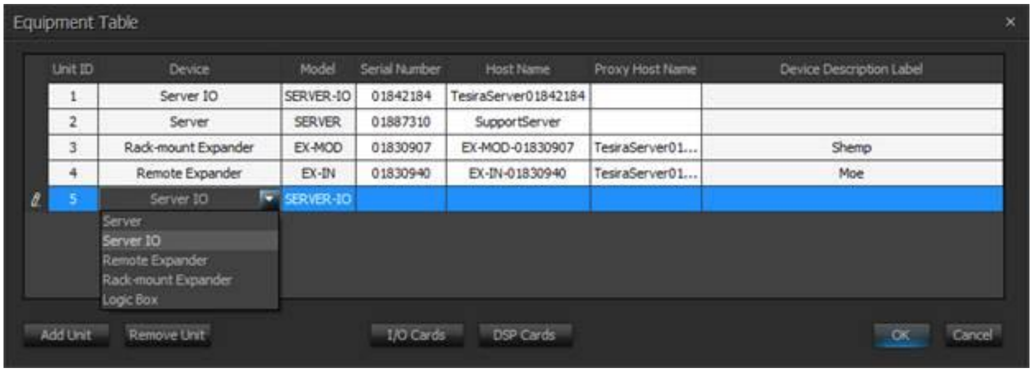
Click on either the **Serial Number** or **Host Name** field in the line for a given device. Select the appropriate serial number or hostname from the available units online of that type.

NOTE: In order for a unit to be available for assignment, it must not be part of another system, i.e., Device ID and System ID must be 0 as seen in the Device Maintenance dialog.

The **Proxy Host Name field** selects the TesiraFORTÉ, Server or Server IO unit that will become the 'proxy' for a given remote expander or rackmount expander unit. The chosen server is then responsible for firmware and configuration updates as well of real time control of the expander. To choose the proxy host name, click in the field on the entry for any expander unit.



To manually add a unit, click the **Add Unit** button. Click on the **Device** field for the new entry to select the hardware type. After this the Serial Number/ Host Name and Proxy Host Name (if expander unit) can be chosen.

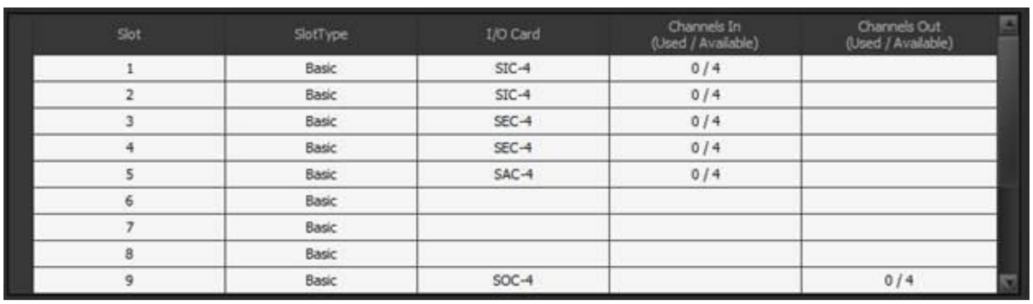

Equipment Table

Unit ID	Device	Model	Serial Number	Host Name	Proxy Host Name	Device Description Label
1	Server IO	SERVER-IO	01842184	TesiraServer01842184		
2	Server	SERVER	01887310	SupportServer		
3	Rack-mount Expander	EX-MOD	01830907	EX-MOD-01830907	TesiraServer01...	Shemp
4	Remote Expander	EX-IN	01830940	EX-IN-01830940	TesiraServer01...	Moe
5	Server IO	SERVER-IO				

To manually remove a unit, highlight that unit in the list and click on the **Remove Unit** button.

Viewing Unit Configuration

An **I/O Cards** button at the bottom of the Equipment Table dialog box provides a list of input/output cards of a selected Tesira SERVER, SERVER IO Lab.gruppen amplifier or EX-MOD rackmount expander. Click to expand the IO configuration of a given unit.

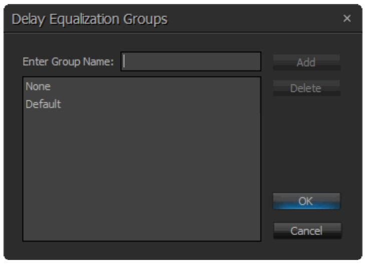


A **DSP Cards** button at the bottom of the Equipment Table dialog box shows the internal DSP hardware (and thus DSP resources available) contained in a selected **Tesira** SERVER, SERVER IO or TesiraFORTÉ unit.

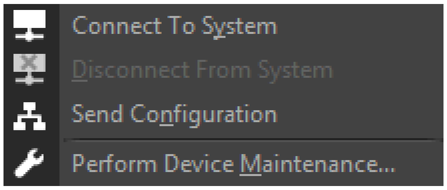
When Connected to the system, the Equipment table can be viewed but not edited. The IO and DSP configurations of any SERVER, SERVER IO, TesiraFORTÉ can be viewed while connected. Lab.gruppen amplifier or EX-MOD will display IO only.

Delay Equalization Groups

Delay Equalization Groups
Provides a way to group signal paths that are sensitive to small offsets in delay. System latencies through the system will be equalized for all paths within a defined group.



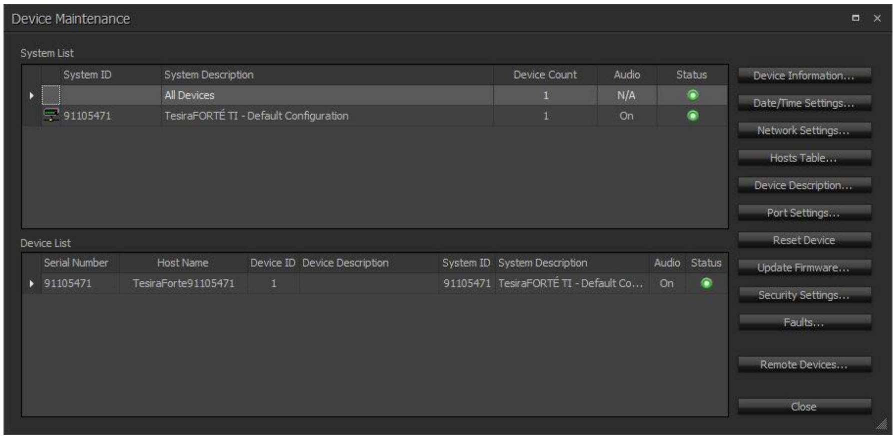
Network



Icon	Name	Description
	Connect To System	Establishes communication with, and retrieves data from, selected **Tesira**® systems on the network. Opens System Connect dialog box (see System Network Considerations). Password protection is then available from the Tools Menu. When connected to a system, Component Object Properties may be changed, but system design (objects & connections) cannot.
	Disconnect From System	Ends communications with selected **Tesira**® systems on the network. System design data is retained in software after disconnect. See System Network Considerations
	Send Configuration	Transmits system design data to selected **Tesira**® devices in the system. See System Network Considerations. Before data can be transmitted, a system design file (.TMF) must first be opened, then connected to a system, and have **Tesira** device IP addresses assigned (see Device Maintenance). Send Configuration will automatically Compile the system design, and reset the hardware devices, before sending the new configuration.
	Device Maintenance	Provides an editable table of network related settings for selected **Tesira**® devices. Opens Device Maintenance dialog box. Device Maintenance settings include Date/Time, IP Address, Description, Serial Number, Reset/Initialize, Update Firmware, and Disconnect From Network. See System Network Considerations

Device Maintenance

Device Maintenance



System List provides information on the discovered configured systems.

Device List provides information on the discovered Tesira SERVER, SERVER IO or TesiraFORTÉ devices.

Device Information provides information (such as firmware version, input/output configuration, etc.) regarding the selected device. Please review the Device Maintenance Settings section for more details

Date/Time Settings is used to configure time zone, daylight savings time adjustments and NTP (Network Time Protocol) servers. The date and time can be manually set or the unit can be synchronized to the clock of the host PC. Please review the Device Maintenance Settings section for more details.

Network Settings allows for manipulation of a device's network configuration. The hostname, DNS and IP configuration of the unit can be set as needed. Please review the Device Maintenance Settings section for more details.

The **Hosts Table** allows for hostname network mapping in the absence of a DNS server. The table simply maps an IP address to the hostname of another unit. Please review the Device Maintenance Settings section for more details.

Device Description allows the selected device to be given a descriptive name.

Port Setting selects the baud rate for the serial ports on the SERVER, SERVER IO or TesiraFORTÉ unit selected in the Device list. There is also an option to calibrate the Logic IO on a Server which allows attached Potentiometers to be configured. See the Port Settings sections for more details.

Reset Device clears all current system design data from the selected device. This happens automatically whenever a new system design file (.TMF) is uploaded (see Send Configuration).

Update Firmware allows a Firmware update file to be sent to discovered Tesira SERVER, SERVER IO or TesiraFORTÉ Devices. Please review the Device Maintenance Settings section for more details

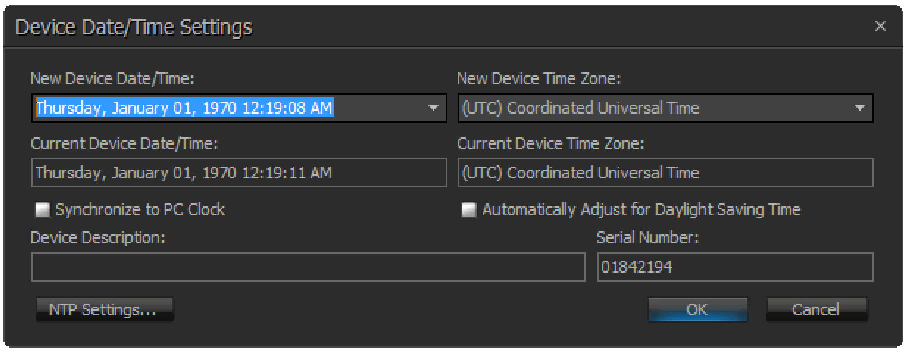
Security Settings allows security access and permissions to be implemented on a Tesira system. See the System Security section for more details.

Faults displays a list of active faults. If the Status is green then no faults are present. Review the System Status section for more details

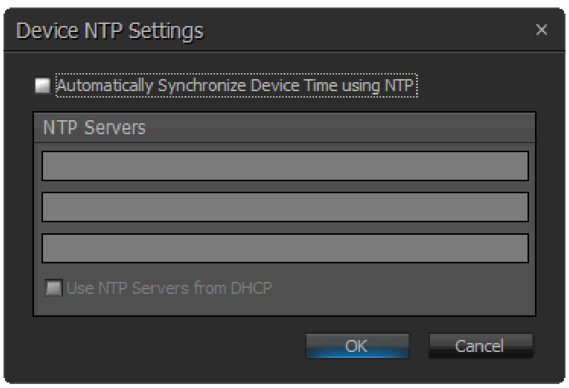
Remote Devices opens a list of any EX-MOD, EX-In, EX-Out, EX-IO, EX-Logic, Lab.gruppen Amplifiers and TEC-1 Ethernet devices connected to the control network of the selected Tesira device. Review the Expander Device Maintenance section for more details.

Device Maintenance Settings

Device Information
Device Information provides information (such as firmware version, input/output configuration, etc.) regarding the selected device.
Date Time Settings
Date/Time Settings is used to configure time zone, daylight savings time adjustments and NTP (Network Time Protocol) servers. The date and time can be manually set or the unit can be synchronized to the clock of the host PC.



NTP Settings provides a list of three NTP servers to which the device can automatically synchronize. If the network provides a DHCP server, it can also be chosen as the NTP server.



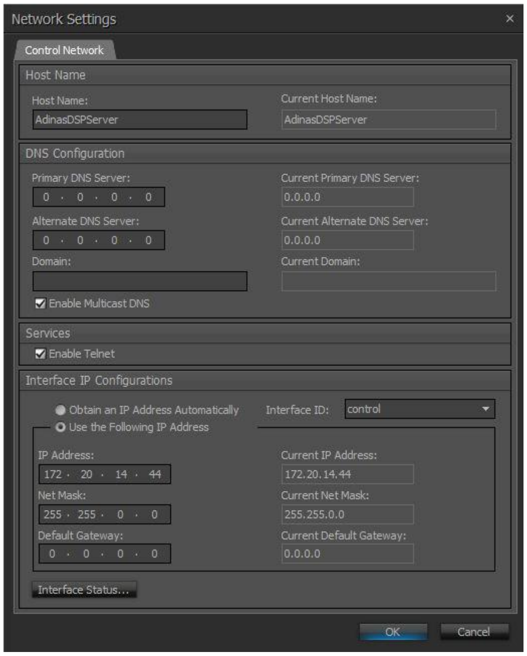
Network Settings
Network Settings allows for manipulation of a device's network configuration. There is dedicated Tabs for the Control Network and if a SERVER or SERVER IO is being used with a DAN-1 each DANTE card will appear on a separate tab.
Control Network Settings
The Network Settings dialog allows the following to be configured:
Host Name

- Host name - Alphanumeric and unique with no spaces
- DNS Configuration - a Primary and alternate DNS Server can be specified, as well as a domain and multicast DNS.

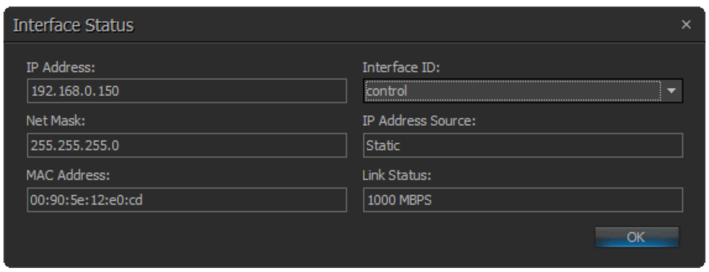
Services

- Ability to enable or disable the Telnet port (23) which is one option for Third Party Control

Interface IP configurations
Settings related to automatic or manual IP addressing can be defined here.

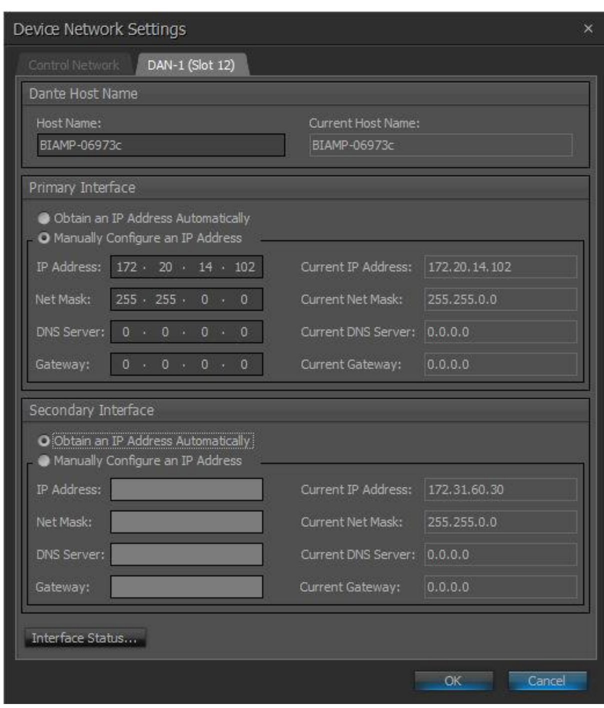


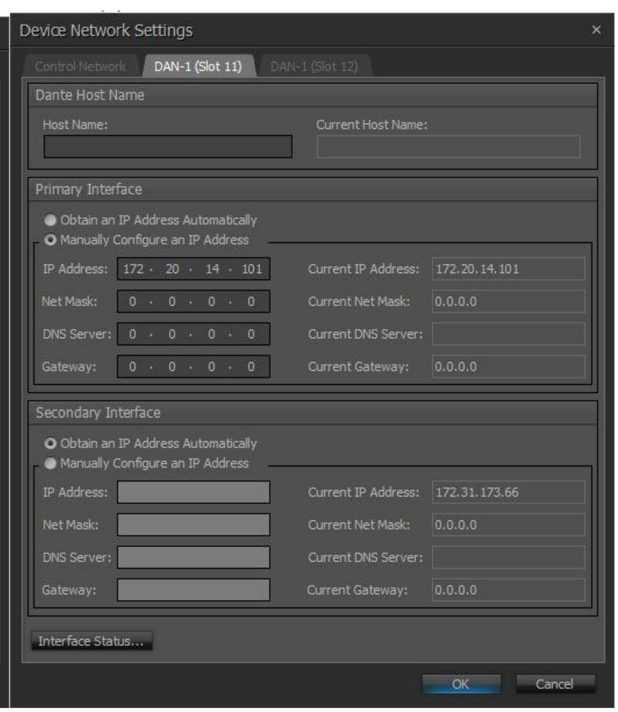
Interface Status shows the current connection details for the chosen interface.



DAN-1 Network Settings

The hostname, DNS and IP configuration of the DAN-1 can be set as needed.

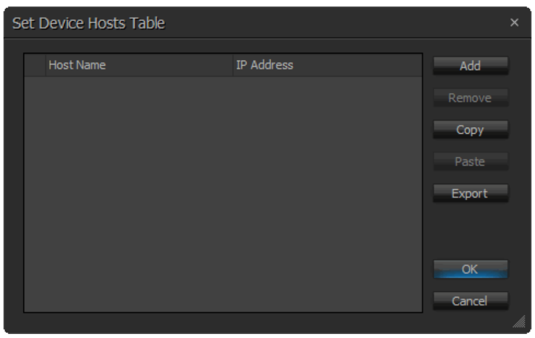




Interface Status shows the current connection details for the chosen interface.

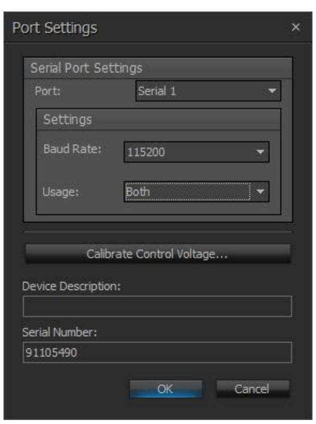
Hosts Table

The **Hosts Table** allows for hostname network mapping in the absence of a DNS server. The table simply maps an IP address to the hostname of another unit.

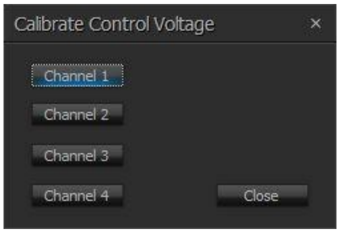


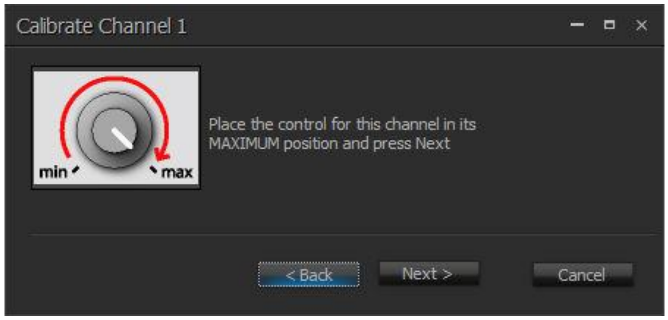
Device Description allows the selected device to be given a descriptive name.

Port Setting selects the baud rate for the RS-232 ports on the selected Tesira Server. Serial ports can be configured to use with Command string blocks, TTP inputs, both or none. See the RS-232 section for more details.



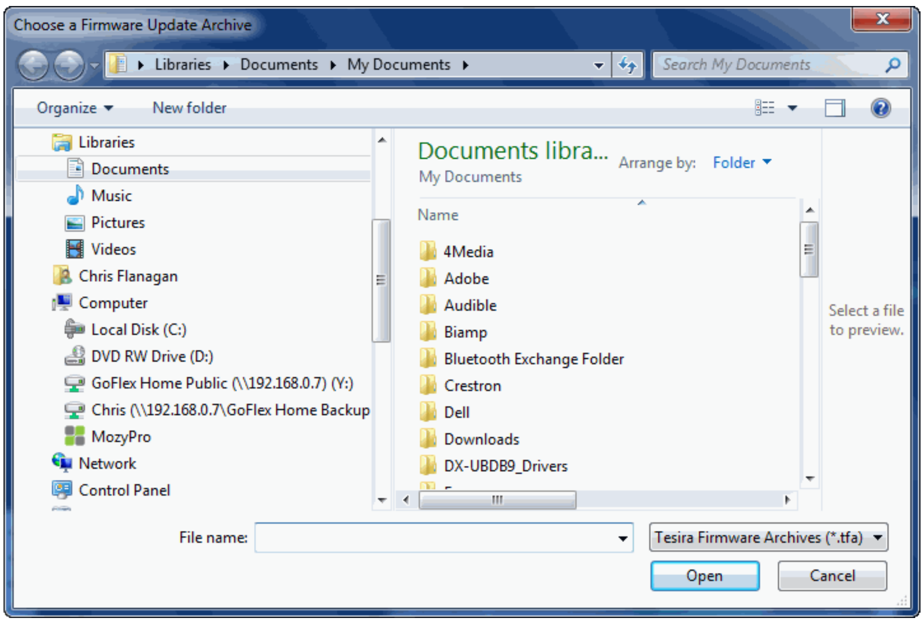
Calibrate Control Voltage is used to calibrate the GPIO connections on a Server SNC card or TesiraFORTÉ.



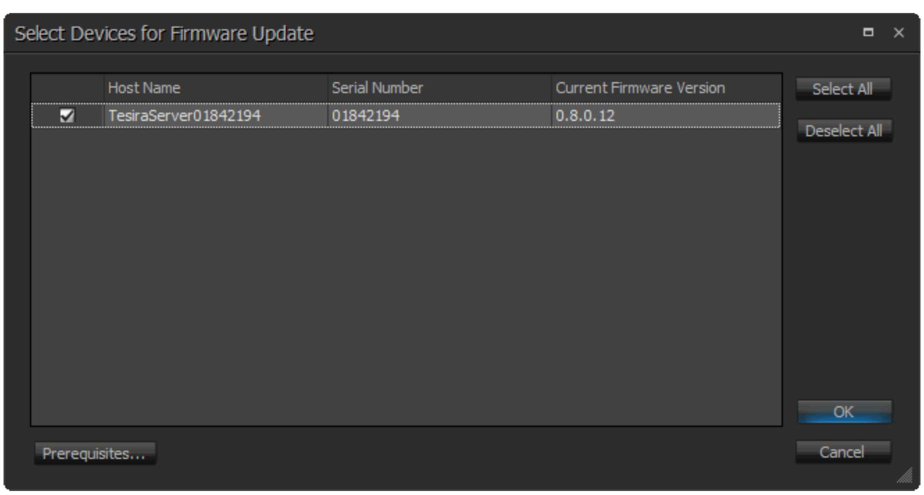


Update Firmware

Update Firmware produces a file browser window.

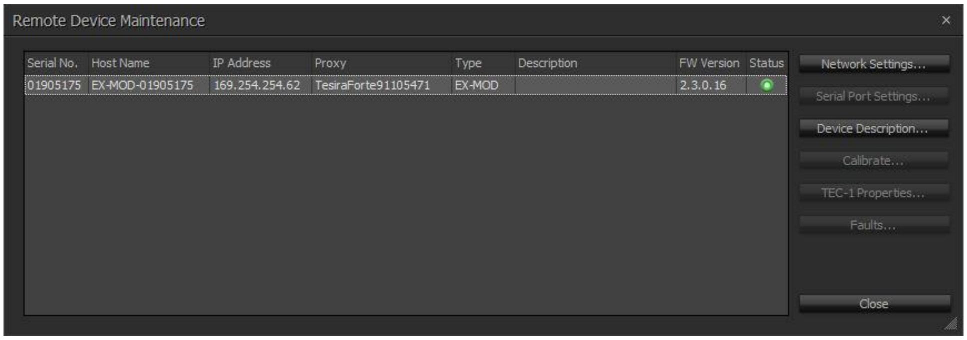


When a valid firmware file is selected, a dialog window is produced that shows all discovered devices in a table with columns indicating System ID, Device IP Address, and other details for each unit. To specify a unit for updating, place a check in that unit's Update box. Buttons are provided to Select All entries and Clear All entries. Press the Update button to perform the firmware update on the selected units and Cancel to exit this window.

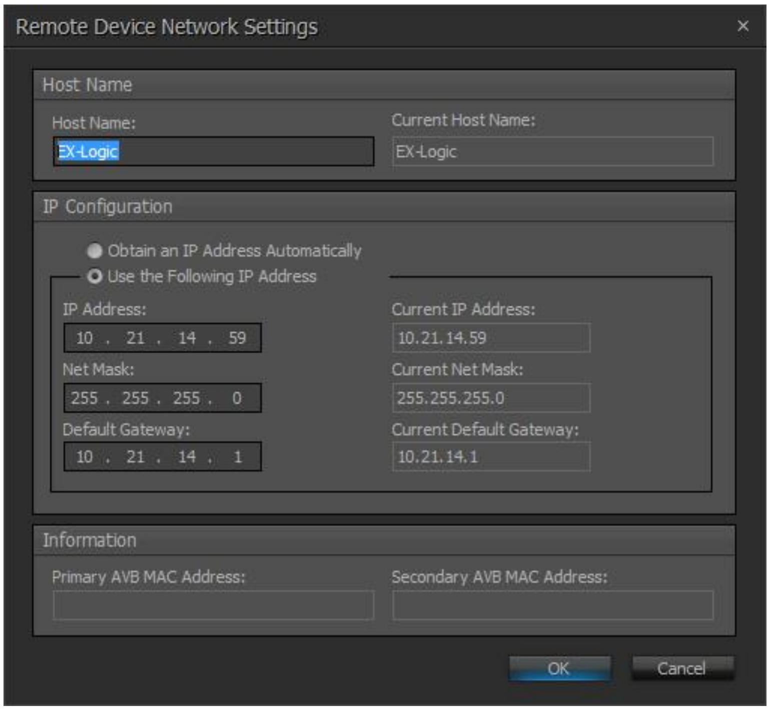


Expander Device Maintenance

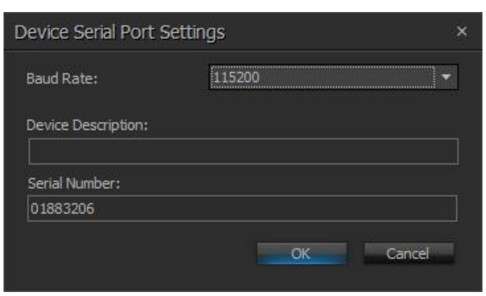
This area is used to define the IP and other Device maintenance settings of a Expander class device. This includes any EX-MOD, EX-AEC, EX-In, EX-Out, EX-IO, EX-Logic, Lab.gruppen Amplifiers and TEC-1 Ethernet devices connected to the control network of the selected Tesira device.



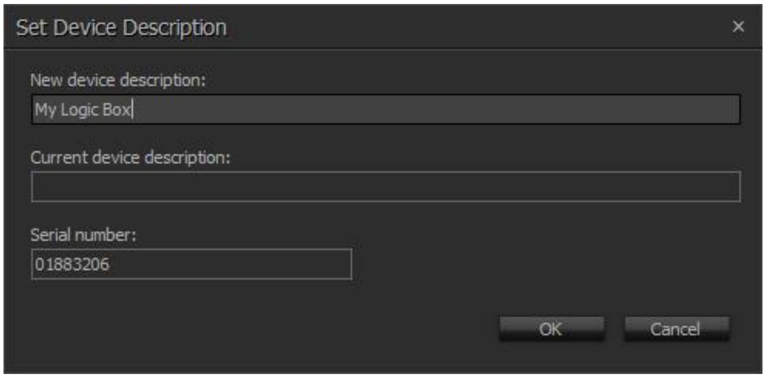
Using **Network Settings**, the IP properties and Host name of any device can be set.



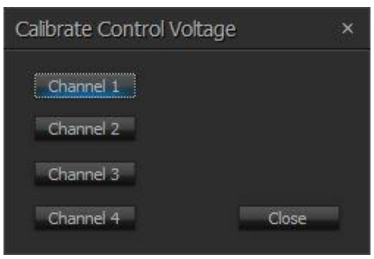
Serial port settings allows the Logic box serial port baud rate to be configured (default 115200 Baud)

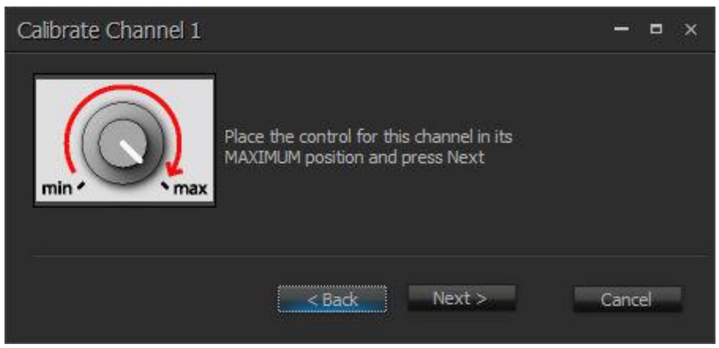


Device Description sets a logical description of the device,



Calibrate allows attached Potentiometers to be configured.



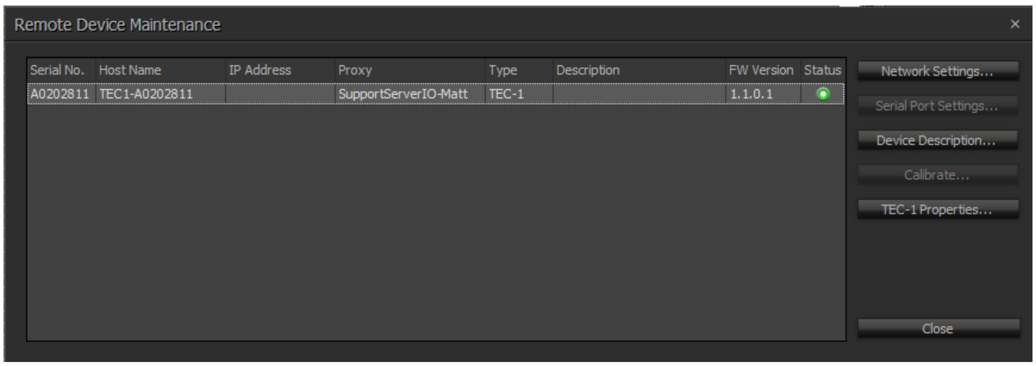


TEC-1 Properties is only used when a TEC-1 is selected. See TEC-1 Device Maintenance for more details.

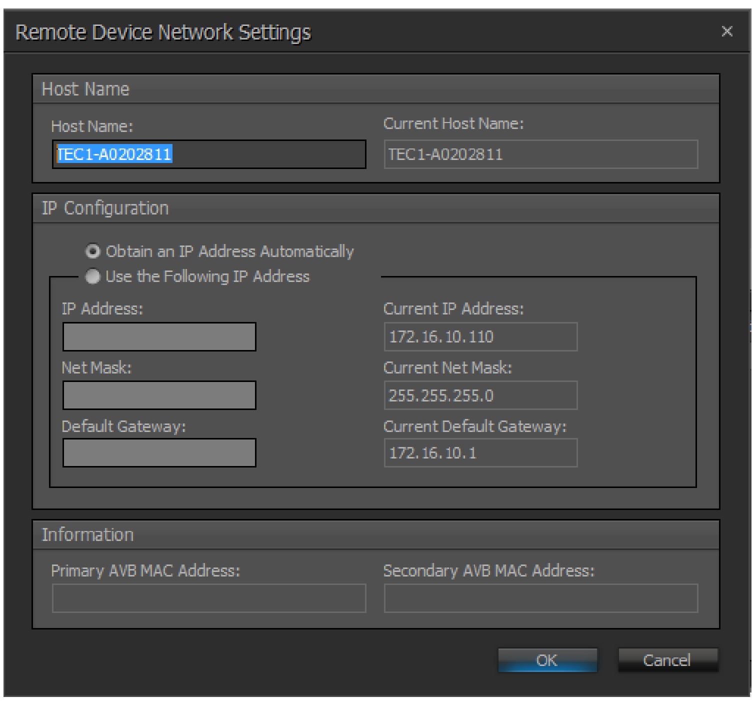
Faults will be available if a expander is reporting an issue. Please see the System Status and Fault Reporting sections for more details.

TEC-1 Device Maintenance

The resulting dialog allows for setting the network parameters, device description and other TEC-1 properties.
Note: the default network configuration of TEC-1 units is DHCP, so if there is no DHCP server on the Tesira network the units will revert to link local addressing (169.254.xxx.xxx, netmask 255.255.0.0) schemes. If the Tesira servers have been set to static addresses of a different subnet BEFORE this process is completed, a TEC-1 unit can be statically addressed by shorting the 'Locksmith' header pins on the back of the unit and using the dialogs on the unit to set it to a compatible subnet. Once the TEC-1s and the Tesira servers are on the same subnet, the units should be discoverable in Device Maintenance.

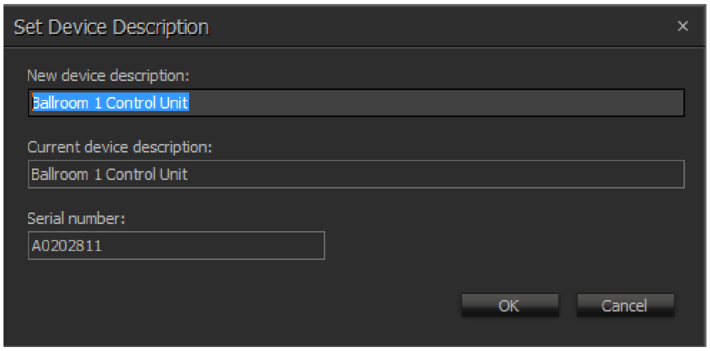


This view shows the DHCP host name of the unit, the IP Address (only if statically configured), the units proxy server, its description and firmware version.

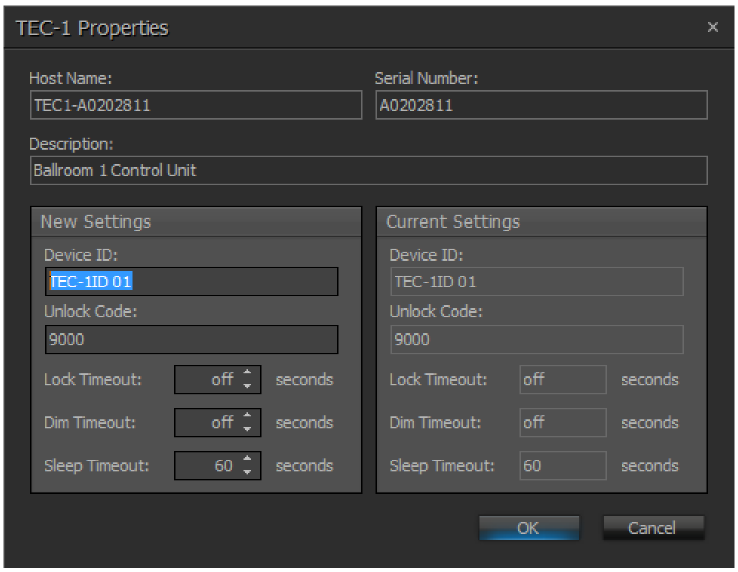


The Network Settings dialog allows for setting of the unit's Host Name and DHCP/Static IP addressing options.

The Device Description dialog allows one to set a logical description to a TEC-1 unit (separate from the Device ID used in the TEC-1 block).



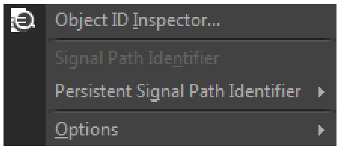
The **TEC-1 Properties** dialog allows for setting of the Device ID and other operational settings of the TEC-1.



- **Device ID**: This must match exactly with the device ID in the TEC-1 block for the unit to receive the proper configuration menus and control the correct parameters.
- **Unlock Code:** If the Lock Timeout is set, this Unlock code is required to unlock the unit. This is a four-digit number.
- **Lock Timeout:** The inactivity time after which the unit will lock (in seconds). The Unlock Code will then be required to operate the unit.
- **Dim Timeout:** The inactivity time after which the unit will dim the display (in seconds) to save power. A sensory input will bring the display back to full brightness.
- **Sleep Timeout:** The inactivity time after which the unit will go into sleep mode (in seconds) to save power. A sensory input will bring the unit out of sleep mode.

Tools

Tools Menu



Object ID Inspector

Provides a list of all objects within the Layout, along with their Object Codes, Block Types, Text Labels, Partition Names, Partition IDs, Unit numbers, and Instance Tags. Please review the Object Inspector section for more details.

Signal Path Identifier

Provides a temporary color-coded identification of all audio signal paths (Lines) or Logic nodes which are associated with a selected Object. See Signal Path Identifier for more details.

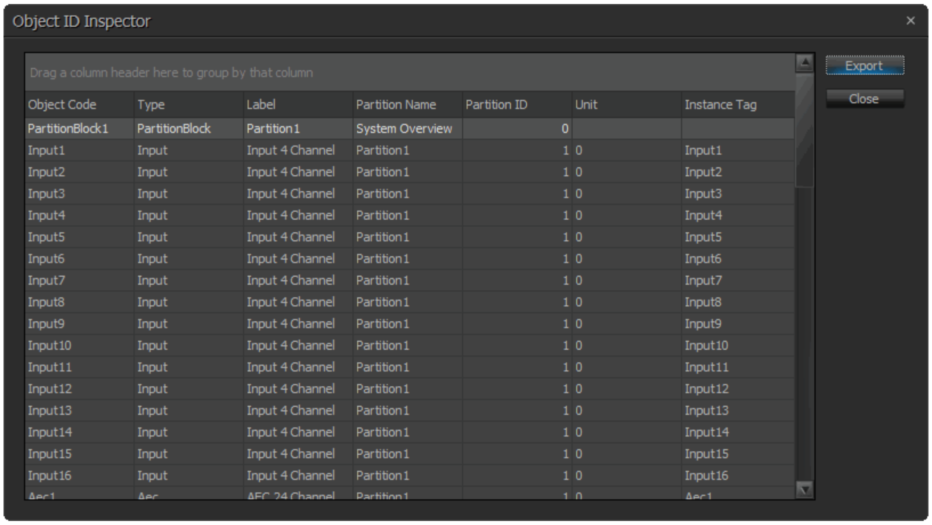
Options

Application Settings - Determines many base display and configuration options of the Tesira Application.
Document Settings - Allows adjustment of General and Network Settings.

Tools menu items

Object ID inspector

Object ID Inspector



Object Code	Type	Label	Partition Name	Partition ID	Unit	Instance Tag
PartitionBlock1	PartitionBlock	Partition1	System Overview	0		
Input1	Input	Input 4 Channel	Partition1	1	0	Input1
Input2	Input	Input 4 Channel	Partition1	1	0	Input2
Input3	Input	Input 4 Channel	Partition1	1	0	Input3
Input4	Input	Input 4 Channel	Partition1	1	0	Input4
Input5	Input	Input 4 Channel	Partition1	1	0	Input5
Input6	Input	Input 4 Channel	Partition1	1	0	Input6
Input7	Input	Input 4 Channel	Partition1	1	0	Input7
Input8	Input	Input 4 Channel	Partition1	1	0	Input8
Input9	Input	Input 4 Channel	Partition1	1	0	Input9
Input10	Input	Input 4 Channel	Partition1	1	0	Input10
Input11	Input	Input 4 Channel	Partition1	1	0	Input11
Input12	Input	Input 4 Channel	Partition1	1	0	Input12
Input13	Input	Input 4 Channel	Partition1	1	0	Input13
Input14	Input	Input 4 Channel	Partition1	1	0	Input14
Input15	Input	Input 4 Channel	Partition1	1	0	Input15
Input16	Input	Input 4 Channel	Partition1	1	0	Input16

The Object ID Inspector reveals the details of each DSP block in the system including Partition blocks. The Object Inspector view shows the Object Code, Type, the user-assigned Label, the Partition name and ID in which that object resides. It also shows the Tesira unit and Instance Tag of each object. Clicking on any object entry in the table will scroll the layout view to that object and show it as selected in the window. The object information can be exported into comma separated value format for outside reference if needed.

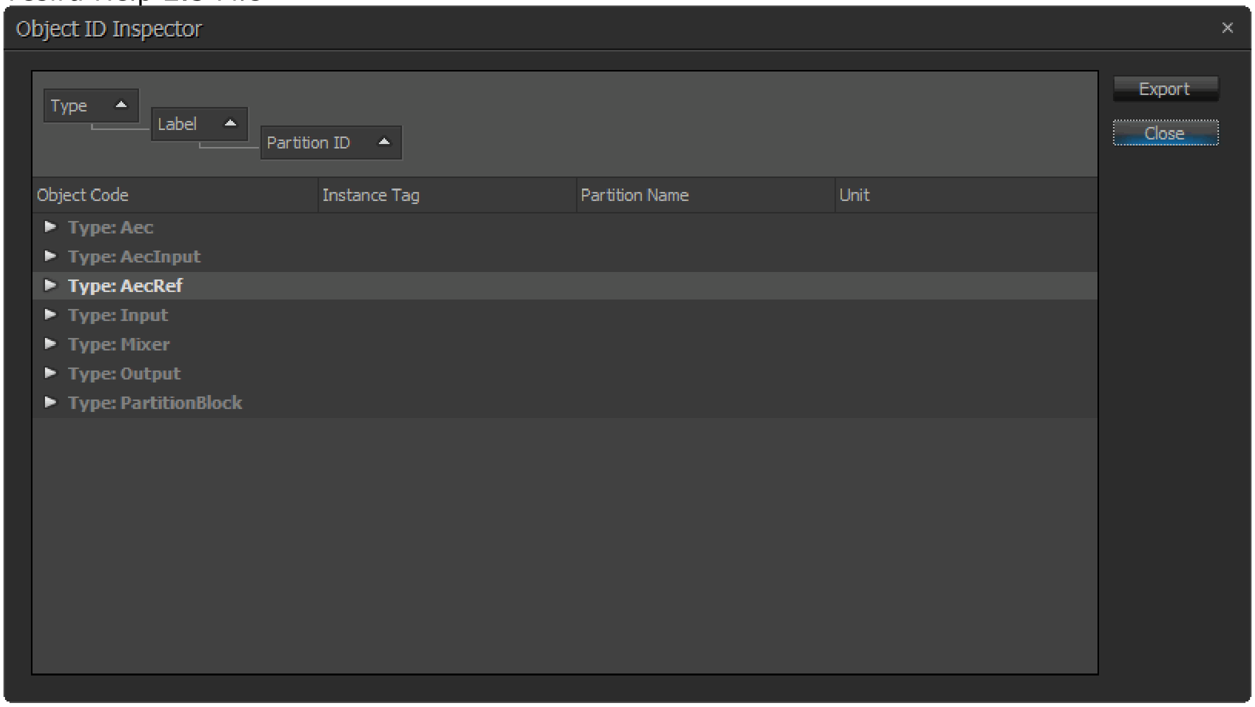
Filtering the Object ID Inspector View

Filtering Options

A click on any column header in the Object ID Inspector sorts in numerical and alphabetical order within that column. This can be reversed by clicking the column again. A mouse over of a column header reveals a second icon which allows for custom filtering within that column. A single object can be selected, all objects can be selected or objects with blank/non-blank fields. A custom selection allows the filter to be configured very specifically based on alphanumeric logic. When a custom filter is chosen, the filter can be disabled or edited by clicking on the icons that are created on the bottom of the Object ID Inspector dialog.

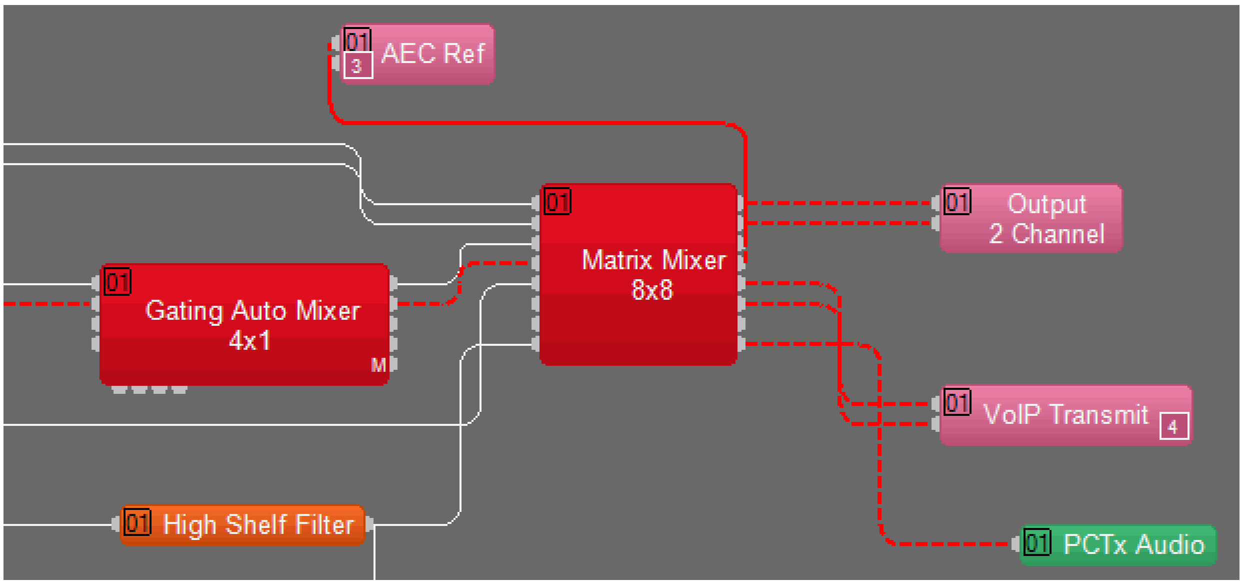
Grouping Options

Dragging a column header to the blank field at the top of the Object ID Inspector dialog group's objects according to that parameter. Arrows on the updated listing show groups that can be expanded with a click on the arrow and contracted with another click. Subgroups within those groups can be created by dragging another column into the group set. Filters and sorting options can be applied to the grouping and sorting methods.

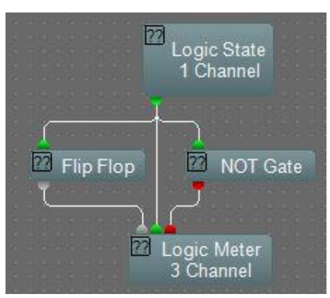

Object ID Inspector
Type
Label
Partition ID
Export
Close
Object Code
Instance Tag
Partition Name
Unit
Type: Aec
Type: AecInput
Type: AecRef
Type: Input
Type: Mixer
Type: Output
Type: PartitionBlock

Signal Path Identifier

Signal Path Identifier
Can be used on audio paths or to review Logic states.
Audio Signal Path Identifier
Provides a temporary color-coded identification of all audio signal paths (Lines) which are associated with a selected Line Object. See example below, where the selected input is shown as being routed to multiple outputs. Signal Path Identifier is represented by a thin dashed red line. However, the Signal Path Identifier color may be changed in Application Settings. When using Signal Path Identifier on a compiled design, propagation delay will appear in the Status Bar. Persistent Signal Path Identifier is available in Normal Mode (follows subsequent line selections), Locked Mode (remains on original line selection), or Off (temporary selection). If the selected Line Object includes identifying text (see Line Property Sheet), that text will be temporarily imposed on all lines being indicated by Signal Path Identifier**.**



Logic Signal Path Identifier
Logic paths and states can be reviewed by right clicking a Logic Node. Green is Logic High (1), Red is Logic Low(0).
Logic does not propagate through Flip Flop gates.



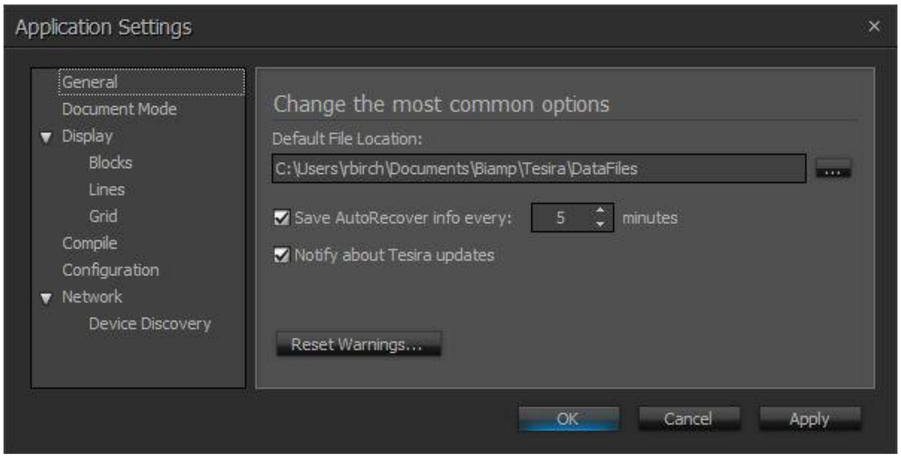
Options

Options

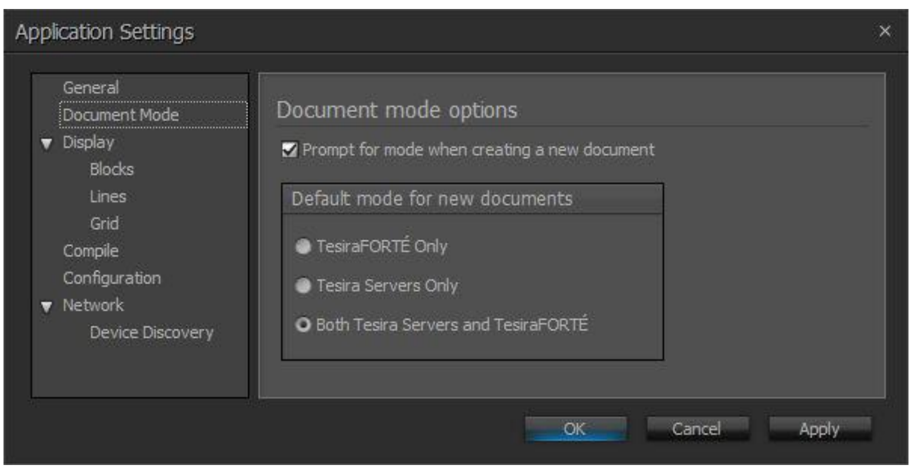
Options
Global settings are subdivided into two categories: Application Settings and Document Settings.

Application Settings
Determines many base display and configuration options of the Tesira Application. There are several options within the Applications Settings dialog General, Display, Compile, Configuration and Network.

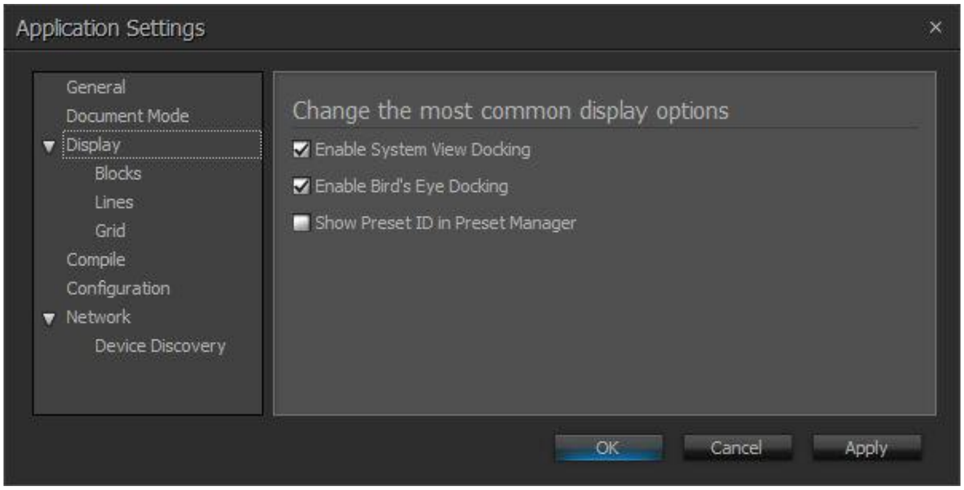
General Options
Allows Data Files (system designs) and Processing Libraries (Component Objects) to be saved to specified directory locations. Allows the Save AutoRecover time to be adjusted or disabled. Tesira update notification can be enabled or disabled. **Reset Warnings** will let any suppressed warnings to be reset.



Document Mode Options
Is used to specify the default Document Mode when creating new files, and whether the user is prompted for this selection every time a new file is created.



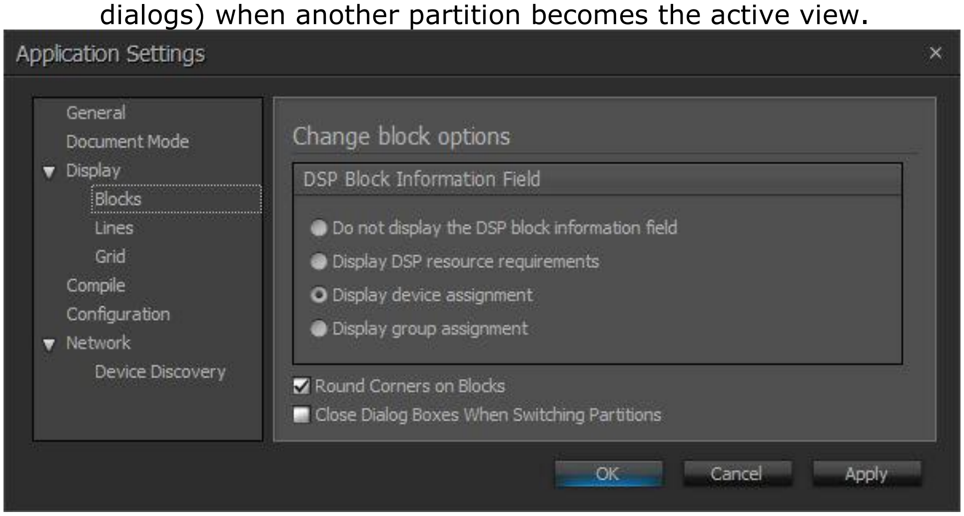
Display Options
The general display options allows docking to be enabled for the System View and Bird's Eye View sheets. Show Preset ID in Preset Manager inserts a column in the Preset Manager view sheet where the preset ID number is always displayed.



Block Options

This selects the details to be displayed on the DSP blocks in the layout. The options are as follows:

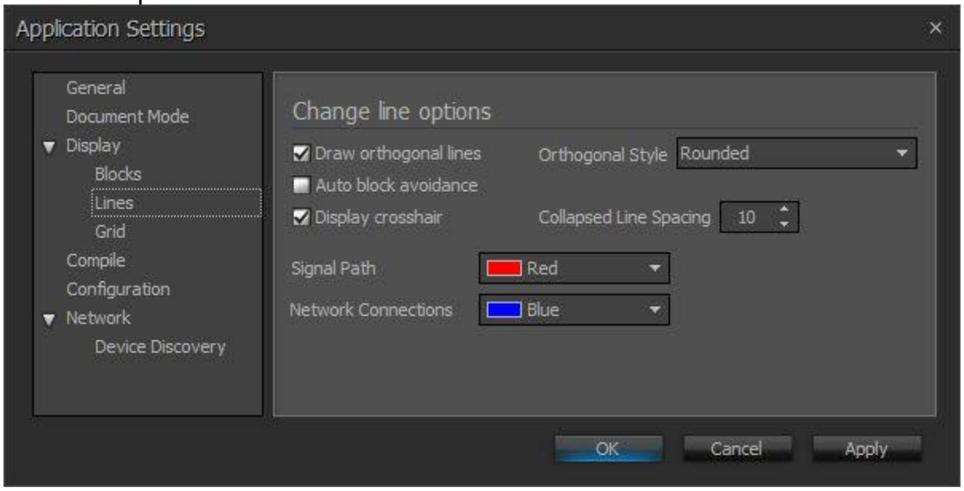
- Do not display the DSP block information field – display nothing on the DSP blocks.
- Display DSP resource requirements – display DSP resources for a block in percent.
- Display device assignment – display the hardware unit in which this DSP object resides.
- Display group assignment – display the DSP group in which this DSP object resides.
- Round corners on Blocks is a cosmetic feature.
- Close Dialog Boxes When Switching Partitions will close all open dialogs (i.e. DSP block control dialogs) when another partition becomes the active view.



Line Options

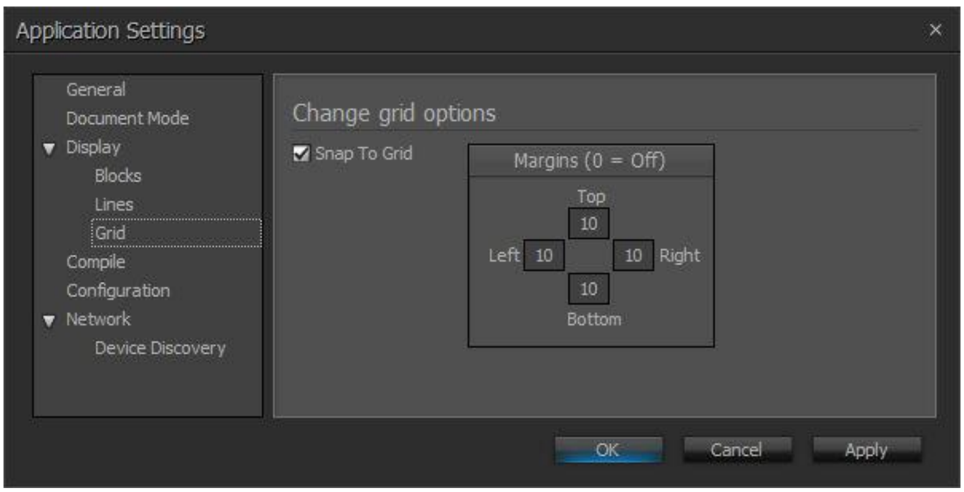
Allows the line drawing options to be configured.

- Draw orthogonal lines automatically bends lines at right angles to meet DSP block nodes. The style can be normal (right angles at turns), rounded (no right angles at turns) and rounded with jump-overs (lines jump over existing lines).
- Enabling of **Auto block avoidance** automatically turns lines around DSP blocks in the layout.
- Enabling of **Display crosshair** will use the crosshair rather than the hand pointer to draw the lines.
- **Collapsed Line Spacing** determines the horizontal separation of vertical lines in Orthogonal line drawing mode when the left arrow key is held.
- Any color can be chosen for the **signal path** (using the signal path identifier) and Audio **Network Connections**.



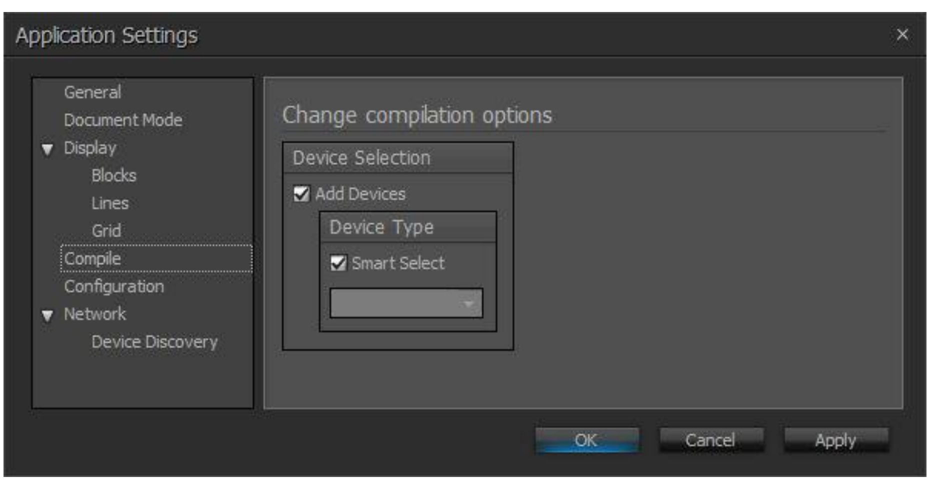
Grid Options

Enables the snap to grid functionality on placement of DSP blocks and lines. Margins can be chosen for blocks in the layout window (pixels).

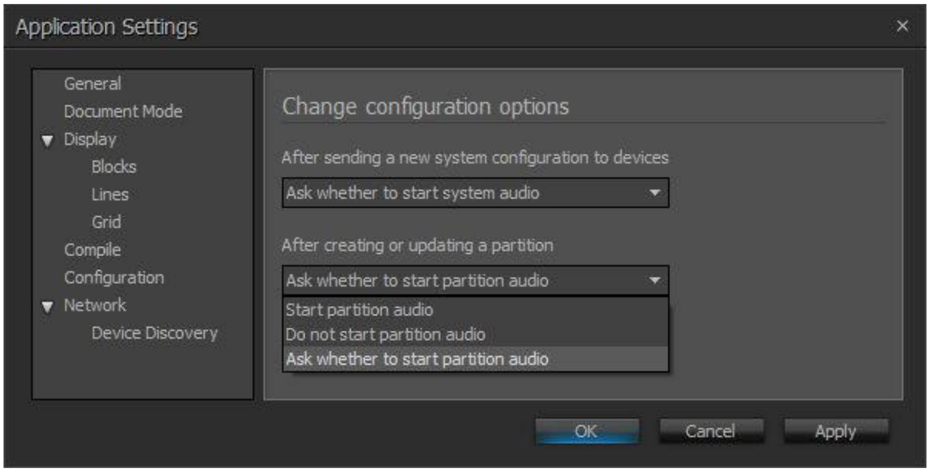


Compile Options

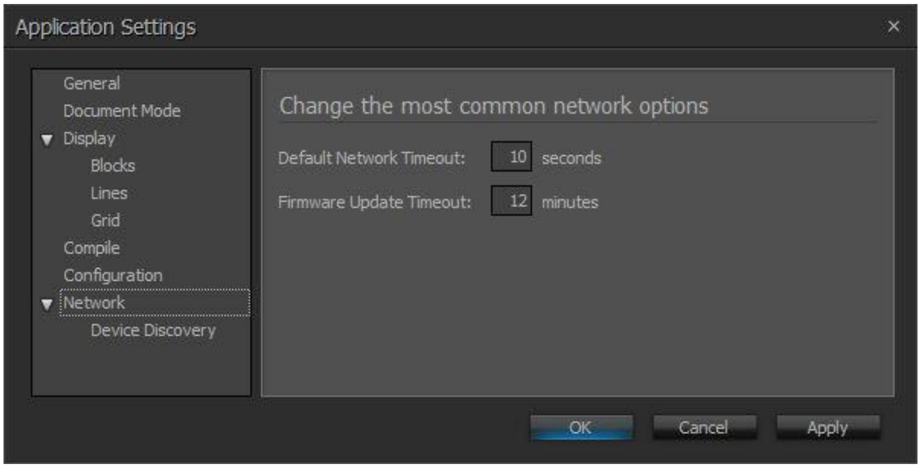
Determine how the Application handles hardware allocations during the compilation process. Add Devices allows the compiler to add **Tesira** hardware to the system equipment table and allocate DSP blocks to the new unit(s). When Smart Select is enabled, the compiler will use the most efficient hardware configuration possible. This is especially useful during the initial design of a system. Else the application can be configured to add only **Tesira** servers, expanders or mini-expanders.



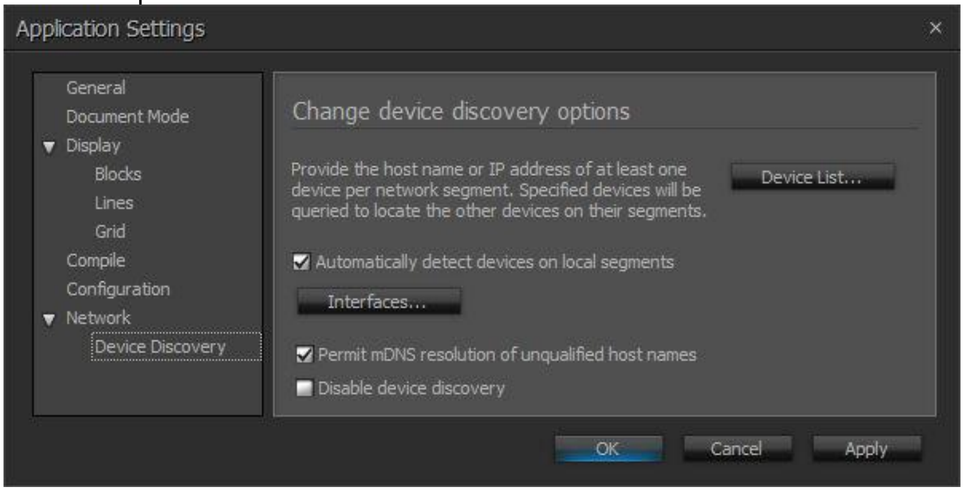
Configuration Options
Determine how the system prompts the user when one or more audio partitions are updated. The options can be set to prompt the user to start audio, automatically start audio or do not start audio.



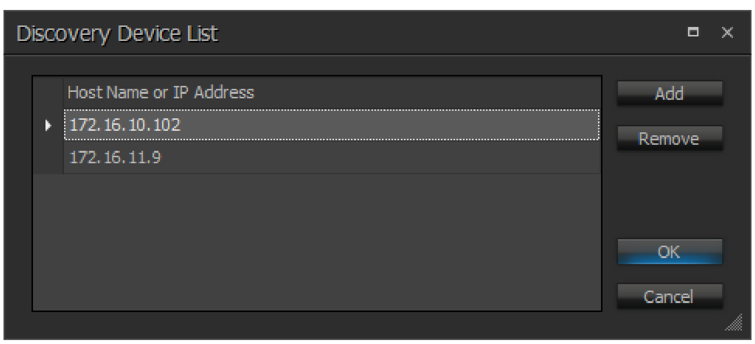
Network Options
Sets the timeouts for network control communications and firmware updates.



Device Discovery Options
Configures the way the **Tesira** application finds hardware on the network. Automatic discovery allows the application to automatically detect **Tesira** hardware on the local area network. Multiple Network Interface Cards can be chosen to detect **Tesira** hardware on multiple LAN's.



In the case that one Network Interface Card (NIC) is discovering **Tesira** units on a routed network, hostnames or IP address can be manually entered in the Device List for discovery.

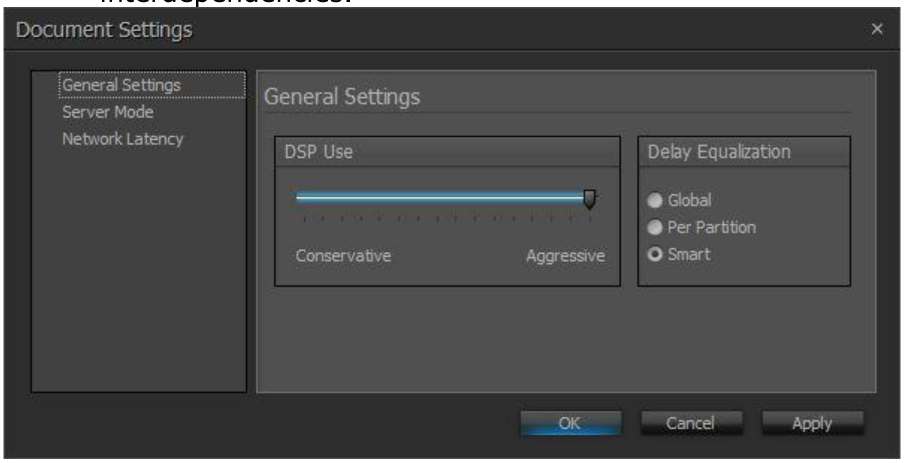


Permission of mDNS allows software resolve hostnames using the multicast DNS protocol, which requires no DNS server. Disabling device discovery defeats all previous options in this section.

Document Settings

General Settings
DSP Use
Configures the amount of DSP resources reserved for delay equalization of audio paths. This is a user selectable slider. When set as Aggressive the DSP will reserve the less resources to delay compensation. When set to Conservative the DSP will reserve more resources to delay compensation.
Delay Equalization
The Delay Equalization paths are chosen with radio buttons:

- Global - All signal paths in the system will have same latency from input to output.
- Per Partition – Audio paths within each audio partition will have the same latency from input to output. Latency may vary from partition to partition.
- Smart – the software analyzes DSP groups and applies Delay Equalization to those found to have interdependencies.

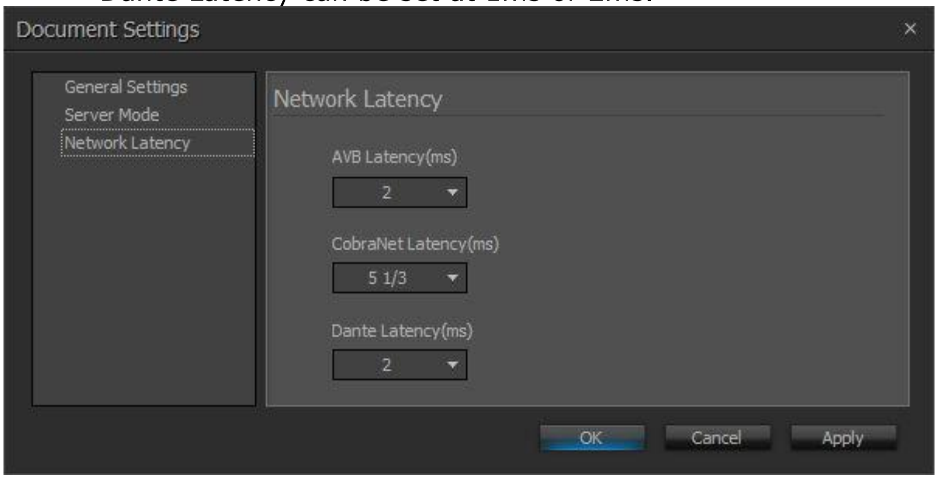


Server Mode
Defines the compiler rules for the Hardware that should be used to calculate the layout resources. Please note that if audio expanders are required then this will be included by default as long as a suitable TesiraFORTÉ, SERVER or SERVER IO device is available as a proxy

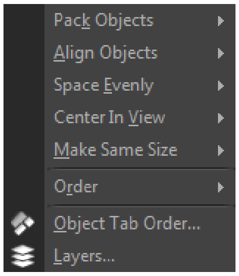
- **TesiraFORTÉ only** - The compiler will only consider TesiraFORTÉ devices in its DSP calculations and Equipment table allocation.
- **Tesira Servers Only** - The compiler will only consider Tesira Server and Server IO devices in its DSP calculations and Equipment table allocation.
- **Both Tesira Server sand TesiraFORTÉ** - The compiler will consider TesiraFORTÉ, Serve and server IO devices in its DSP calculations and Equipment table allocation.

Network Latency
Sets the latency of the AVB and CobraNet audio transports.

- AVB Latency can be set at 1ms or 2 ms.
- CobraNet Latency can be 5 1/3 ms (default) , 2 1/3 ms or 1 1/3 ms.
- Dante Latency can be set at 1ms or 2ms.



Layout



Name	Description
Pack Objects	opens the Pack Objects Menu
Align Objects	opens the Align Objects Menu
Space Evenly	opens the Space Evenly Menu
Center in View	opens the Center In View Menu
Make Same Size	opens the Make Same Size Menu
Order	opens the Order Menu
Objects Tab Order	allows the objects to be moved forward or back in the layout. Opens the Objects Tab Order Dialog
Layers	opens the Layers Shhet

Layout Menu Items

Pack Objects

Action	Description
Left	packs selected objects next to each other, aligned on the left.
Right	packs selected objects next to each other, aligned on the right. The target location for packing Left/Right is the top-most selected object.
Top	packs selected objects next to each other, aligned on the top.
Bottom	packs selected objects next to each other, aligned on the bottom. The target location for packing Top/Bottom is the left-most selected object.

Align Objects Menu

The Align function can be used when multiple control objects are selected.

Action	Description
Left	aligns selected objects on the left.
Right	aligns selected objects on the right.
Top	aligns selected objects on the top.
Bottom	aligns selected objects on the bottom.

Space Evenly

Action	Description
Across	spaces selected objects horizontally. The two most-distant objects become the reference for spacing evenly.
Down	spaces selected objects vertically. The two most-distant objects become the reference for spacing evenly.

Center in View

The Center in View controls assist in aligning control elements on a surface.

Action	Description
Both	centers selected objects within the visible Surface, both vertically and horizontally.
Vertical	centers selected objects vertically within the visible Surface.
Horizontal	centers selected objects horizontally within the visible Surface.

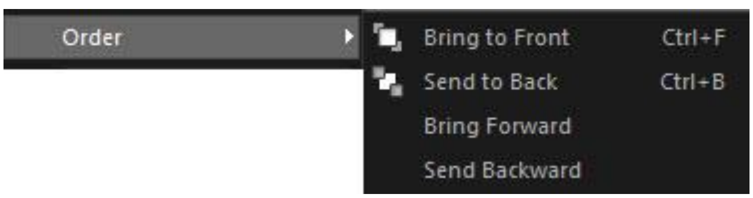
Make Same Size

Makes objects the same size. To select multiple objects **left click+Shift** or selecting a blank area of the surface - click and drag to lasso objects. The primary selected object (green handles) becomes the reference for sizing.

Action	Description
Both	sizes selected objects both in width and height.
Width	sizes selected objects in width only
Height	sizes selected objects in height only.

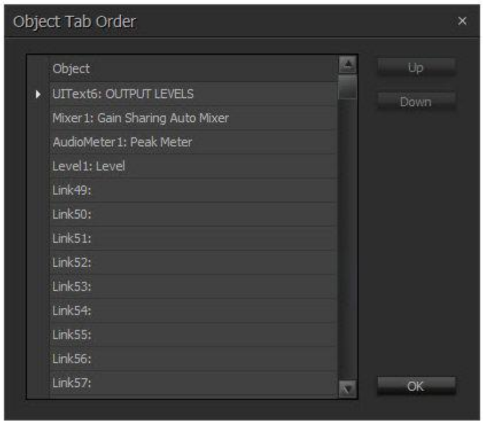
Order Menu

The Order Menu moves objects forward or back in the layout. These functions are also available on the Layout Bar.



Action	Description
Bring To Front	moves selected objects in front of all other objects. This also is available via the CTRL+F Keyboard Shortcuts
Send To Back	moves selected objects behind all other objects.This also is available via the CTRL+B Keyboard Shortcuts
Bring Forward	moves selected objects forward relative to others.
Send Backward	moves selected objects backward relative to others.

Object Tab Order

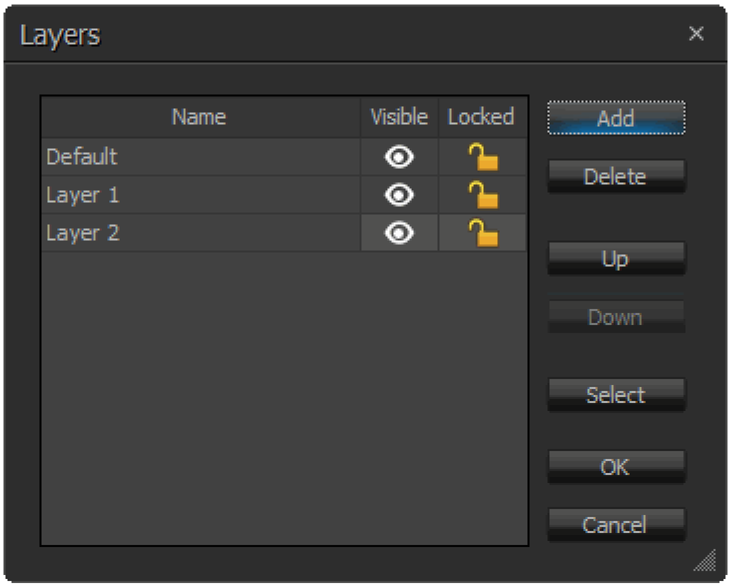


Name	Description
Up	Moves the selected Item up (Forward) out of the layer
Down	Moves the selected item down (back) into the layer
OK	Accepts changes and closed the dialog

Layers Sheet

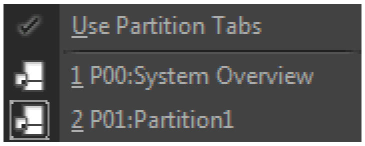
Provides an editable table of Layer properties. Layers can be used to separate a Layout into multiple parts. The Layers can be organized with regards to object types, system segments, or any other criteria. The Default Layer always remains, but other Layers may be created or removed.

NOTE: *Components cannot be selected when the current Layer is invisible.* **Lock** prevents a Layer from being changed or selected. Lock & View may also be accessed by double-clicking on the corresponding icons within the list.

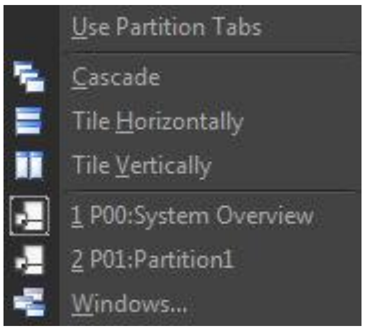


Name	Description
Name	Can be used to give a custom name to the layer
Visible	turns on/off visibility of a Layer in the Surface
Locked	prevents a Layer from being changed or selected. Lock and View may also be accessed by double-clicking on the corresponding icons within the list.
Add	Creates a new Layer
Delete	Removes the selected layer
Up	moves the position of a Layer up the list (Layers are not stacked, so this does not affect Tab Order or visual overlapping).
Down	moves the position of a Layer down the list (Layers are not stacked, so this does not affect Tab Order or visual overlapping).
Select	Will select all objects on the selected layer. Objects cannot be selected when the current Layer is invisible.
OK	Applies changes and Closes the dialog
Cancel	Does not apply changes and closes the dialog

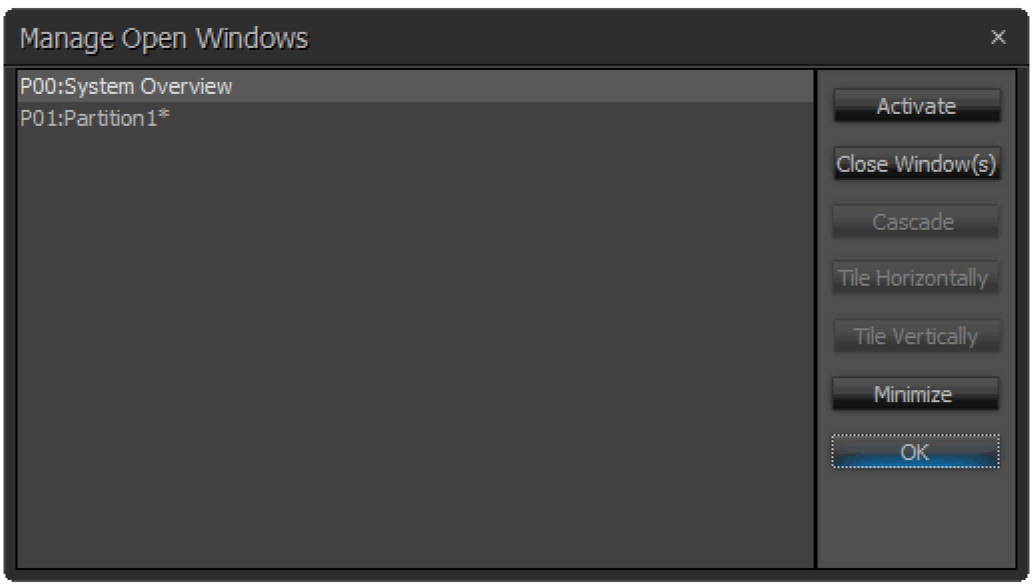
Window



The Window Menu configures how the System Overview and Audio Partition Layouts are displayed in the application window and chooses which is active (in front). By default these items are tabbed in the window and clicking on the tab or selecting that tab from the Window menu brings it to the front. If Use Partition Tabs is unchecked, the System View and Audio Partitions become windows that can be Cascaded, Tiled Vertically or Horizontally.

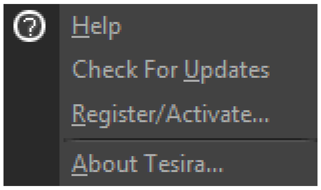


In this mode an additional **Windows** menu item appears to manage the open windows.

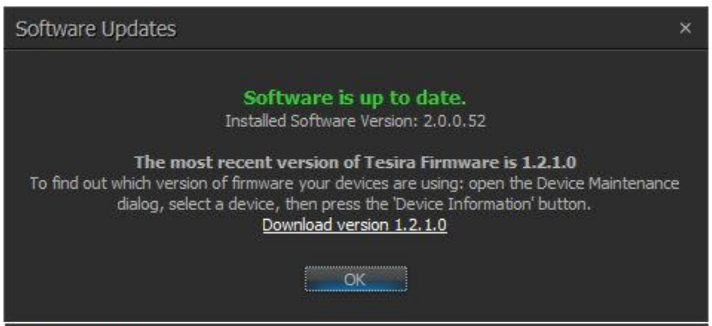


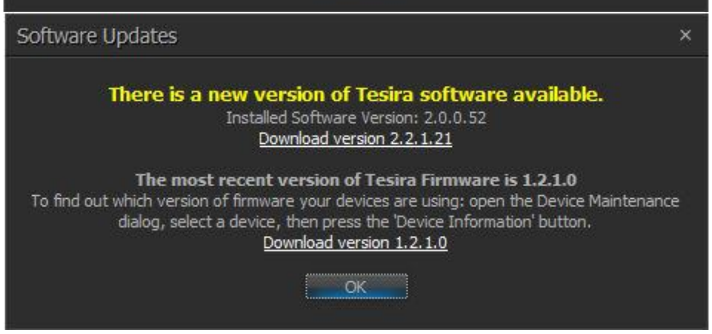
Help

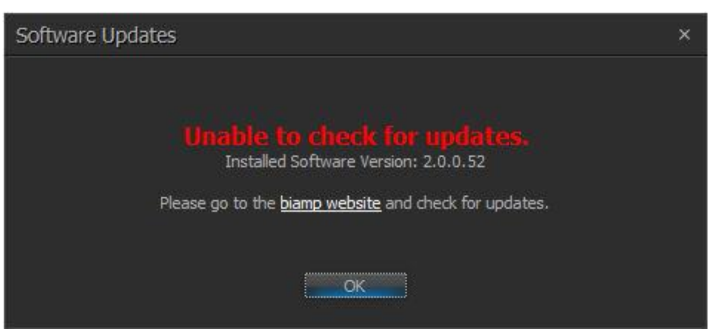
Provides **Tesira**® Help Topics and About Tesira information. Including registering and activating the software.



- Help - Opens the Help File
- Check for updates - Will confirm the most recent software is being used. Opens the Software Update Dialog. The Dialog will also advise the most recent Firmware version as well.
  - Green = Software up to Date
  - Yellow = There is a newer version of Software available - please download and update
  - Red = Unable to verify the correct version - confirm network connectivity allows access to the internet.







- Register/Activate - Allows software to be registered and Activated
- About Tesira - Opens the Tesira Splash screen and displays the version details.

Component Objects

Graphic Elements

Graphic Elements

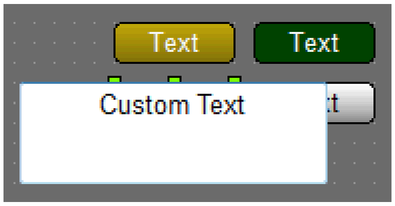
Graphical or text elements can be used in the layout to assist in labelling and layout.

Text

Text

A text Block can be used to provide labelling of signal path or items in the layout. The Property Sheet can be used to modify the color and Block text and text alignment.

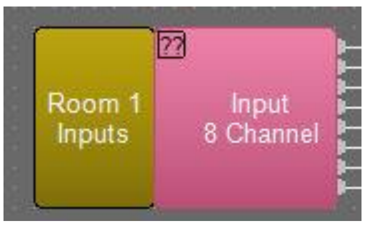
Alternatively selecting the text block and pressing the 'enter' key will allow editing of the text.



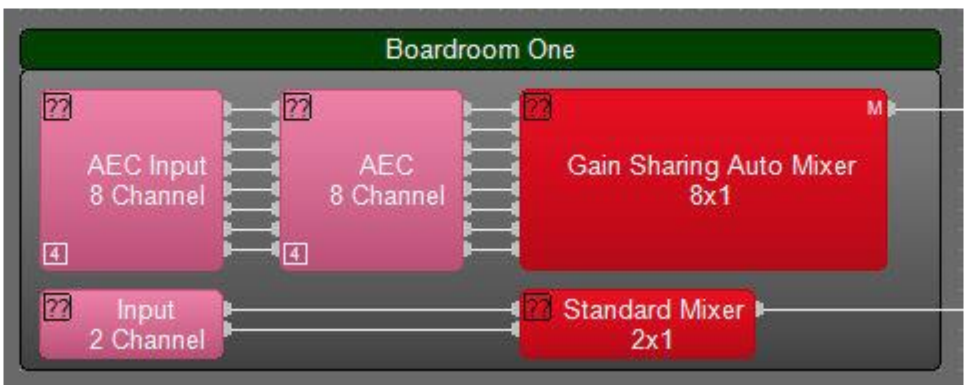
Examples

A text box can be used to define and label areas of the layout.

Naming an Input Block:



Using two text boxes - one to label the area and one placed behind the processing objects to act as a border.



Input Output

Input / Output

These Component Objects provide the audio inputs/outputs (I/O) to the system. Analog, CobraNet® (digital) I/O and AVB formats are available, allowing system designs to include A/D & D/A converters which have CobraNet and AVB capability.

AEC Input components are available for acoustic echo cancellation, a Telephone Interface is available for conferencing.

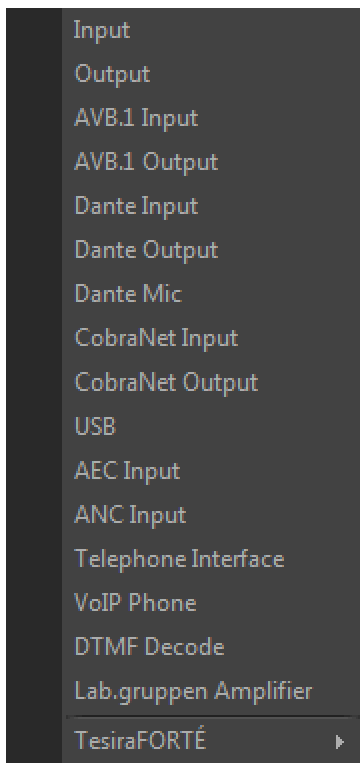
Input/Output components are available in pre-defined configurations, however, the configuration may be customized when being placed from the Object Bar.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface. Right-clicking over the object provides a pop-up menu of options.

- Analog Input
- Analog Output
- AVB.1 Input
- AVB.1 Output
- Dante Input
- Dante Output
- Dante Mic
- CobraNet Input
- CobraNet Output
- USB
- AEC Input
- ANC Input
- Telephone Interface
- VoIP Phone
- DTMF Decode
- Lab.gruppen Amplifier

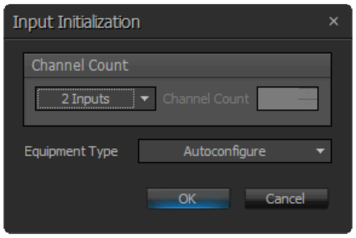
TesiraFORTÉ

- FORTÉ AI
- FORTÉ CI
- FORTÉ TI
- FORTÉ VI



Input

If an analog input is required, the **Input** processing block should be selected. When this component is selected from the Object Toolbar, an Input Initialization dialog is produced.

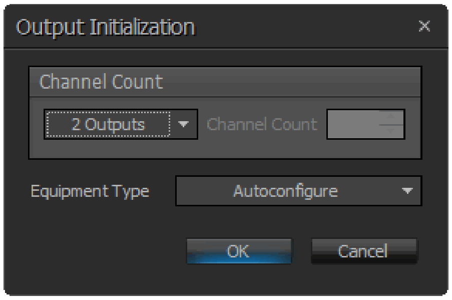


Name	Description	Range
Channel Count	Provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels can be specified.	1-24
Equipment Type	specifies what type of hardware the compiler should allocate the block to. Review the Equipment Type section for more details.	

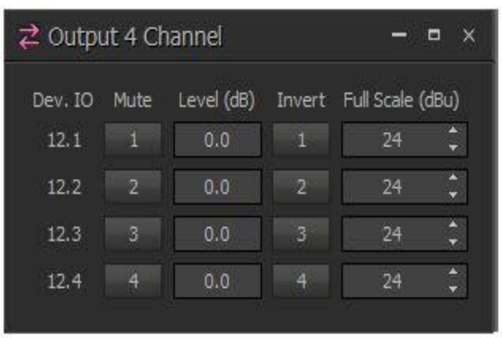
Name	Description	Range
Device IO	Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted as x.y - where x indicates which card slot and y indicates which channel on the card.	
Peak	a software indicator that flashes when the input signal is within 3dB of clipping.	
Gain	sets the amount of analog gain for that channel and is used to compensate for differing input levels (mic or line).	0-66 in 6dB steps
Phantom Power	assigns +48 Volt phantom power to the input for use with condenser microphones	
Mute	turns the input signal on/off.	On or Off
Level	adjusts the relative input volume.	-100 to +12
Invert	adjusts the polarity of the input signal.	0° or 180°

Analog Output

If an analog output is required, the **Output** processing block should be selected. When this component is selected from the Object Toolbar, an Output Initialization dialog is produced.



Name	Description	Range
Channel Count	Provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels can be specified.	1-24
Equipment Type	specifies what type of hardware the compiler should allocate the block to. Review the Equipment Type section for more details.	



Name	Description	Range
Device IO	Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted as x.y - where x indicates which card slot and y indicates which channel on the card.	
Mute	turns the input signal on/off.	On or Off
Level	adjusts the relative input volume.	-100 to +12
Invert	adjusts the polarity of the input signal.	0° or 180°
Full Scale (dBu)	sets the amount of analog gain for that channel and is used to compensate for differing input levels (mic or line).	-31 (Mic level), 0dBu to +24dBu in 6dB steps

AVB.1 Input

AVB.1 Input objects provide support for explicit AVB audio reception from IEEE 1722.1 compatible third-party devices.

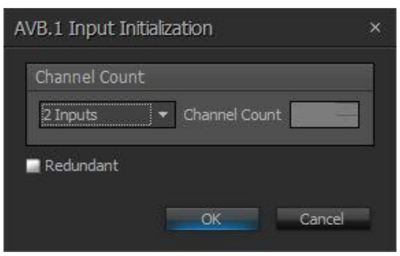
AVB.1 blocks can be allocated to a Tesira Server, Server IO and TesiraFORTÉ device, Audio Expanders (EX-MOD, EX-AEC, EX-IN, EX-OUT, EX-IO and Lab.gruppen Amplifier) do not support AVB.1 blocks.

Up to sixteen Tesira AVB.1 Input blocks can be placed into each Server, Server IO or TesiraFORTÉ device. Review the AVB Network Considerations sections for AVB stream and channel bandwidth information.

A requirement of 1722.1 is that the talker and listener streams must have the same channel count. If the channel counts differ, Audio streams will not flow correctly.

Initialization dialog

When this object type is selected from the Object Toolbar, an AVB.1 Input Initialization dialog window is displayed.



Channel Count – Determines the number of channels in the block. If Custom is selected from the drop-down list, any number of channels from 1 to 60 can be specified.

Redundant - Checking this box indicates that the AVB.1 Input is part of a redundant system. See Redundancy for more details.

Control Dialog

Peak is a software indicator that illuminates when the input signal is within 3dB of clipping.

Mute turns the input signal on/off.

Level adjusts the relative input volume. The level range is from -100 to +12dB.

Invert reverses the polarity of the input signal.

AVB 1722.1 Stream

Name allows the AVB stream name to be defined. The stream name must be unique, and must not be left blank. When a stream name is entered that is not unique, an error message is displayed.

Active is used to indicate whether the stream is actively passing audio or not. In cases where audio is expected but not present, this can trigger a fault condition based on the setting for the 'Fault when inactive' flag in DSP Properties.

AVB.1 Output

AVB.1 Output objects provide support for explicit AVB audio reception from IEEE 1722.1 compatible third-party devices.

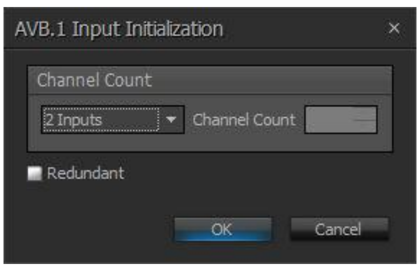
AVB.1 blocks can be allocated to a Tesira Server, Server IO and TesiraFORTÉ device, Audio Expanders (EX-MOD, EX-AEC, EX-IN, EX-OUT, EX-IO and Lab.gruppen Amplifier) do not support AVB.1 blocks.

Up to sixteen Tesira AVB.1 Output blocks can be placed into each Server, Server IO or TesiraFORTÉ device. Review the AVB Network Considerations sections for AVB stream and channel bandwidth information.

A requirement of 1722.1 is that the talker and listener streams must have the same channel count. If the channel counts differ, Audio streams will not flow correctly.

Initialization dialog

When this object type is selected from the Object Toolbar, an AVB.1 Output Initialization dialog window is displayed.



Channel Count – Determines the number of channels in the block. If Custom is selected from the drop-down list, any number of channels from 1 to 60 can be specified.

Redundant - Checking this box indicates that the AVB.1 Input is part of a redundant system. See Redundancy for more details.

Mute turns the input signal on/off.

Level adjusts the relative input volume. The level range is from -100 to +12dB.

Invert reverses the polarity of the input signal.

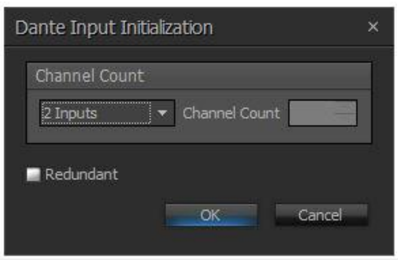
AVB 1722.1 Stream Name allows the AVB stream name to be defined. The stream name must be unique, and must not be left blank. When a stream name is entered that is not unique, an error message is displayed.

Active is used to indicate whether the stream is actively passing audio or not. In cases where audio is expected but not present, this can trigger a fault condition based on the setting for the ‘Fault when inactive’ flag in DSP Properties.

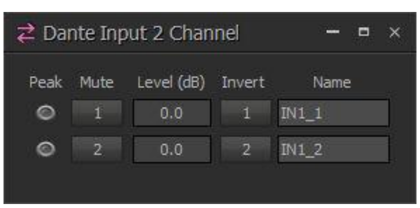
Dante Input

Channel Names can be changed offline in the property sheet. It is recommended that name changes are only done in Tesira software.

Initially all channels will be given names in the form **IN<block number>_<channel number>**, where block number is a unique integer associated with the Input block when it is created and channel number is within the block, starting with 1.
All Dante names and labels are up to 31 characters in length. Name and label comparisons are case-insensitive; “Guitar” and “guitar” are treated as the same label. Unicode and non-roman characters are not supported.
Device names should follow Domain Name System (DNS) hostname rules. Legal characters are A-Z, a-z, 0-9, and '-' (dash or hyphen). Device names must begin with A-Z (or a-z).
Channel labels may use any character except '=' (equals), '.' (full stop or period), '@' (at), \, < and >. Channel labels must be unique on a device. Channel labels do not need to be unique on the network as they are always qualified by device (channel@device).
Please review the Dante Networking section for more details



Description	Range
Channel Count	Provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels can be specified between 1 and 64.
Redundant	If the block is used in a system which is designated as redundant this check box must be enabled. See Redundancy for more details.

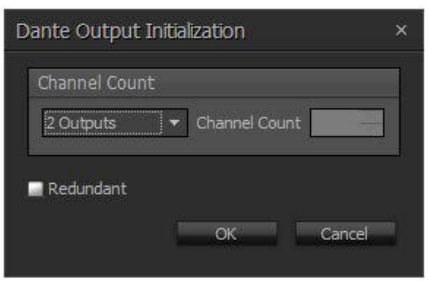


Description	Range	
Peak	A software indicator that flashes when the input signal is within 6dB of clipping	
Mute	turns the input signal on/off.	
Level	adjusts the relative input volume	-100 to +12
Invert	Reverses the polarity of the signal	0° or 180°
Name	allows an individual name of each channel.	

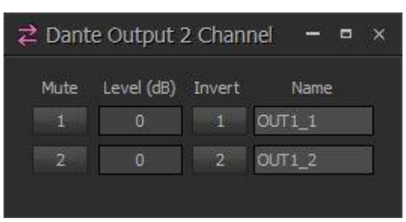
Dante Output

Channel Names can be changed offline in the property sheet. It is recommended that name changes are only done in Tesira software.

Initially all channels will be given names in the form **OUT<block number>_<channel number>**, where block number is a unique integer associated with the Output block when it is created and channel number is within the block, starting with 1.
All Dante names and labels are up to 31 characters in length. Name and label comparisons are case-insensitive; "Guitar" and "guitar" are treated as the same label. Unicode and non-roman characters are not supported.
Device names should follow Domain Name System (DNS) hostname rules. Legal characters are A-Z, a-z, 0-9, and '-' (dash or hyphen). Device names must begin with A-Z (or a-z).
Channel labels may use any character except '=' (equals), '.' (full stop or period), '@' (at), \, < and >. Channel labels must be unique on a device. Channel labels do not need to be unique on the network as they are always qualified by device (channel@device).
Please review the Dante Networking section for more details.



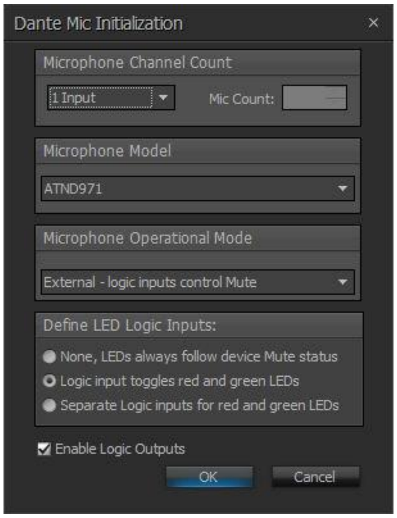
Description	Range
Channel Count	Provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels can be specified between 1 and 64.
Redundant	If the block is used in a system which is designated as redundant this check box must be enabled. See Redundancy for more details.



Description	Range	
Invert	Reverses the polarity of the signal	0° or 180°
Level	adjusts the relative input volume	-100 to +12
Mute	turns the input signal on/off.	
Name	allows an individual name of each channel.	

Dante Mic

Dante Microphone input objects provide native Tesira software support for certain Dante-enabled microphones. When this object type is selected from the Object Toolbar, a Dante Mic Initialization dialog window is displayed. Please also review the Audio-Technica Mic Networking Considerations



Microphone Channel Count – Determines the number of channels in the block. If Custom is selected from the drop-down list, any number of channels from 1 to 64 can be specified.

Microphone Model – Selects which Dante-enabled microphone will be supported by the block. The Audio-Technica ATND971 and ATND8677 are supported.

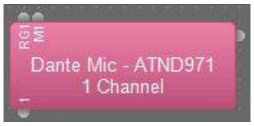
Microphone Operational Mode - The operation of the mute button on the microphone base can be specified. Two push-to-toggle modes are supplied (initially muted and initially unmuted), as well as momentary push-to-talk and momentary push-to-mute. An External setting means that the mute button will display the current mute status by its color, but it will not change the mute state when pressed.

- Initially set to Mute, button press sets to Talk
- Initially set to Talk, button press sets to Mute
- Mute unless button is being pressed
- Talk unless button is being pressed
- External – logic inputs control Mute (default)

Define Logic Inputs – Determines what type of logic inputs for controlling the LED behavior will be shown on the Dante Mic object, if any.

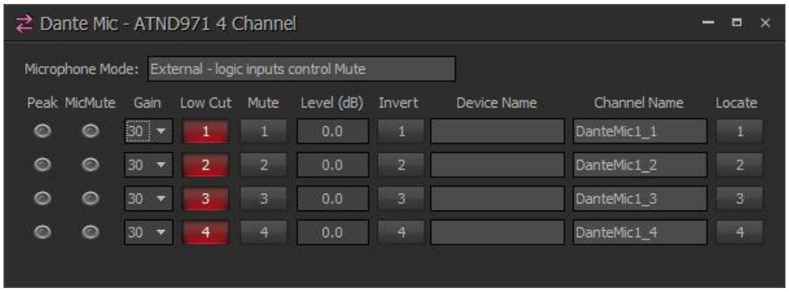
- None, LEDs always follow device Mute status
- Logic input toggles red and green LEDs (default)
- Separate Logic inputs for red and green LEDs

Enable Logic Outputs – Determines whether logic outputs will be provided on the bottom of the Dante Mic object. If enabled, one logic output per channel will be shown. If the Microphone Operational Mode is set to "External", the logic output will correspond to the physical push button on the mic base. It will be a logic high when the button is pressed and logic low when not pressed. In all other Microphone Operational Modes, the logic output will follow the mute state of the microphone: high when muted and low when unmuted.



Control Dialog

The ATND971 and ATND8677 have similar control dialogs.



Above: ATND971 Control dialog.

Above: ATND8677 Control dialog.

Peak a software indicator that flashes when the input signal is within 3dB of clipping.

Mic Mute indicates that the microphone's internal mute circuit has been activated.

Gain Sets the Microphone Preamp gain to either +30, +40, +50 dB.

Low Cut when On (default), applies an 80Hz Low Cut filter to the microphone audio.

Phantom Power (ATND8677 Only) Enables 12V Phantom Power.

Mute turns the input signal on/off in the Dante Mic software object. This has no effect on the internal mute circuit of the microphone.

Level adjusts the relative input volume from -100 to +12.

Invert adjusts the polarity of the input signal between 0° or 180°.

Device Name - The Hostname of the transmitting device. Is read only in the Tesira a Interface. Must be unique. Can be changed via Dante Controller software.

Dante Channel Name - allows an individual name of each channel. Can be changed offline in the property sheet.

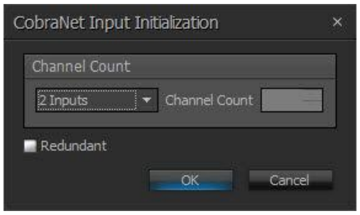
Initially all channels will be given names in the form **<block number>_<channel number>**, where block number is a unique integer associated with the Input block when it is created and channel number is within the block, starting with 1.
All Dante names and labels are up to 31 characters in length. Name and label comparisons are case-insensitive; “Guitar” and “guitar” are treated as the same label. Unicode and non-roman characters are not supported.
Device names should follow Domain Name System (DNS) hostname rules. Legal characters are A-Z, a-z, 0-9, and '-' (dash or hyphen). Device names should follow Domain Name System (DNS) hostname rules. Legal characters are A-Z, a-z, 0-9, and '-' (dash or hyphen). Device names must begin with A-Z (or a-z).
Channel labels may use any character except '=' (equals), '.' (full stop or period), '@' (at), \, < and >. Channel labels must be unique on a device. Channel labels do not need to be unique on the network as they are always qualified by device (channel@device).

Please review the Dante Networking and the Audio-Technica Mic Networking Considerations section for more details

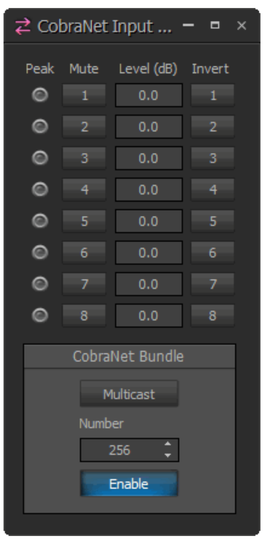
Locate - allows the user to locate the physical microphone. When pressed, the LEDs on the microphone flash.

CobraNet Input

This component provides a means of receiving a digital audio bundle from the CobraNet network. One bundle of 1 to 8 channels can be received per CobraNet Input block.



Name	Description	Range
Channel Count	Provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels can be specified.	1-8
Redundant	If the block is used in a system which is designated as redundant this check box must be enabled. See Redundancy for more details.	



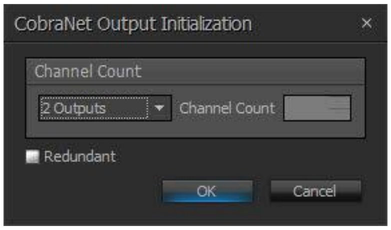
Name	Description	Range
Peak	a software indicator that flashes when the input signal is within 6dB of clipping.	
Mute	turns the input signal on/off.	
Level	adjusts the relative input volume.	-100 to +12
Invert	Reverses the polarity of the signal	0° or 180°
CobraNet Bundle	determines which bundle of digital audio channels is received from the CobraNet network.	
Multicast Off	The block is using Unicast CobraNet bundles	256 to

		65,279
Multicast On	The block is using Multicast CobraNet Bundles. Users must acknowledge the increased network bandwidth message	1 to 255
Enable	turns the CobraNet transmitter on or off.	

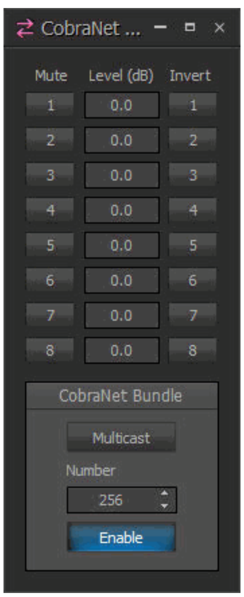
When a CobraNet Input block is placed into the Layout from the Object Toolbar, a dialog will appear, allowing the Channel Count to be specified. (see CobraNet Network Considerations).

CobraNet Output

This component provides a means of outputting a digital audio bundle onto the CobraNet network. One bundle of 1 to 8 channels can be transmitted per CobraNet Output block.



Name	Description	Range
Channel Count	Provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels can be specified.	1-8
Redundant	If the block is used in a system which is designated as redundant this check box must be enabled. See Redundancy for more details.	



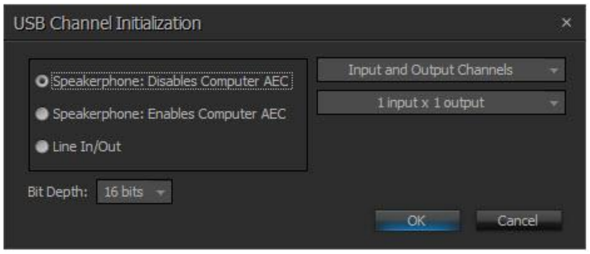
Name	Descrition	Range
Mute	turns the input signal on/off.	
Level	adjusts the relative input volume.	-100 to +12
Invert	Reverses the polarity of the signal	0° or 180°
CobraNet Bundle	determines which bundle of digital audio channels is received from the CobraNet network.	
Multicast Off	The block is using Unicast CobraNet bundles	256 to

		65,279
Multicast On	The block is using Multicast CobraNet Bundles. Users must acknowledge the increased network bandwidth message	1 to 255
Enable	turns the CobraNet transmitter on or off.	

When a CobraNet Output block is placed into the Layout from the Object Toolbar, a dialog will appear, allowing the Channel Count to be specified. Refer to the CobraNet Network Considerations for more information.

USB

USB is only available on TesiraFORTÉ devices. The USB initialization dialog allows the selection of three different operational modes. See the USB considerations section for more details. USB connections may require the Asynchronus clocking mode to be disabled. Please review the USB Clocking section if audible artifacting is heard on active USB streams.



The two Speakerphone modes provide a single audio input and output stream for use with a soft codec application on a PC. The USB Input represents the incoming audio from the soft codec and the USB Output is used to send audio to the far side.

- **Speakerphone: Disables Computer AEC** – In this mode, the TesiraFORTÉ unit will provide the Acoustic Echo Cancellation (AEC) function, and a control message is transmitted to the soft codec via the USB link telling it to disable its internal AEC. This would be appropriate for TesiraFORTÉ models that have built-in AEC (CI, TI, VI).
- **Speakerphone : Enables Computer AEC –** This mode is for situations where the soft codec will provide the Acoustic Echo Cancellation (AEC) function, which would be appropriate for TesiraFORTÉ models that do not have built-in AEC (AI).
- **Line In/Out –** This mode provides up to 8 channels of audio. Input channels, Output channels or both can be selected, and the number of channels can be specified. Combinations of 2, 4 or 6 total USB channels can operate in 24-bit or 16-bit mode, selected by the Bit Depth control. Combinations of 8 total USB channels operate in 16-bit mode only. When connected and configured, the TesiraFORTÉ device installs the chosen number of input and output channels in Windows, but they are not enabled by default. The channels can be enabled in the Windows Control Panel, and then selected as Record and Playback channels in the audio software application.

Note: There is no AEC capability on the USB inputs, since these will generally be line level sources and/or far end sources. Microphones used in distance conferencing should be connected to AEC inputs.

AEC Input

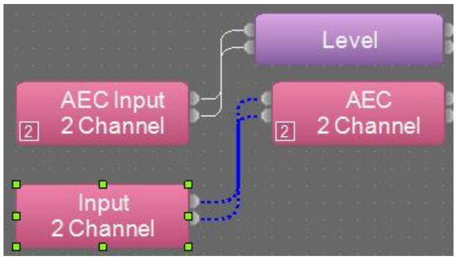
Acoustic Echo Cancelling (AEC) Inputs provide support for distance conferencing applications.
The AEC functionality is comprised of an input block, an AEC processing block and a reference block.

- **AEC Input** - contains the microphone preamp settings for the ambient sensing input and has one output connection per channel.
- **AEC Processing Block** - The AEC processing block has the signal processing functionality.
- **AEC Ref-** The AEC Reference block is used to tell the AEC process what signal to remove from the mic input.

The three blocks will have a number on the lower left, assigned by the software, which indicates which blocks are associated with each other, which is important when there are multiple AEC Inputs in the system.

Remote AEC

The input and processing block can be separated, so the AEC processing feature can be used with non-AEC inputs, such as digital audio inputs from the network. In this instance, the input block could still be used to bring analog mic or line inputs into Tesira, without the AEC processing function.

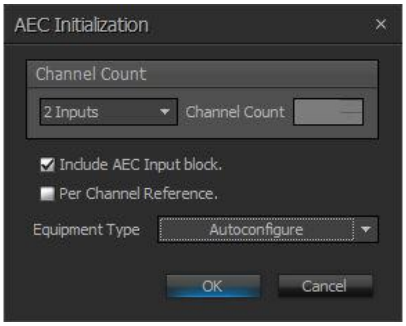


Expander AEC

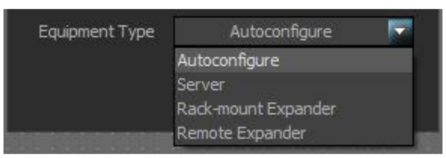
Tesira version 2.1 and later introduces Expander and EX-MOD based AEC processing. The allocation of these processing blocks to a system can be achieved through the **Equipment Type** filter in the **Initialization Dialog** or by adding Expander devices to the equipment table and using the property sheet to fix the AEC blocks in unit.

Initialization Dialog

When the AEC Input component is selected from the Object Toolbar, an AEC Initialization dialog is produced.
Users can choose Autoconfigure to allow the software to allocate it to hardware, or choose to manually assign it to a Server, Rack-mount Expander (EX-MOD with EEC-4 card), or Remote Expander (EX-AEC).

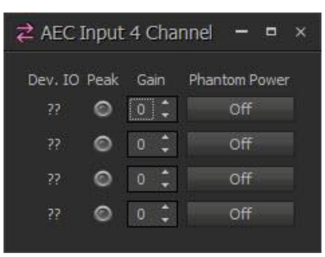


- **Channel Count** determines the number of channels in the block. If Custom is selected from the drop-down list, any number of channels from 1 to 24 can be specified.
- **Include AEC Input Block** specifies whether both the Input and AEC blocks are created (checked) or only the AEC block (unchecked).
- **Per Channel Reference** determines whether the AEC Reference block will have one common reference point for all channels on that block (unchecked) or separate reference points for each channel (checked).
- **Equipment Type** defines the type of hardware for this AEC input. Select Autoconfigure, Server, Rack-mount Expander (EX-MOD with EEC-4 card), or Remote Expander (EX-AEC).



AEC Input

Double clicking on the AEC Input block produces a control dialog window.

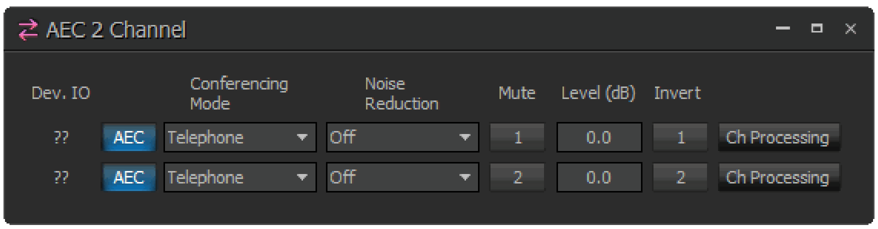


Name	Description	Range
Device IO	Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted	

	as x.y - where x indicates which card slot and y indicates which channel on the card.	
Peak	a software indicator that flashes when the input signal is within 3dB of clipping.	
Gain	sets the amount of analog gain for that channel and is used to compensate for differing input levels (mic or line).	0-66 in 6dB steps
Phantom Power	assigns +48 Volt phantom power to the input for use with condenser microphones	On or Off

AEC Processing Block

Double clicking on the AEC processing block produces a control dialog window.



- **Device IO** (x.y) indicates which physical hardware output is associated with that software channel, where x indicates which card slot and y indicates which channel on the card.
- **AEC** enables and disables the AEC processing chain.
- **Conferencing Mode** optimizes the operation of the Non-Linear Processing (NLP), which is a stage of signal processing post-AEC, and is designed to eliminate any residual echo that may remain after the AEC adaptive filter. The proper setting for this control is related to the total round-trip delay the far end experiences.

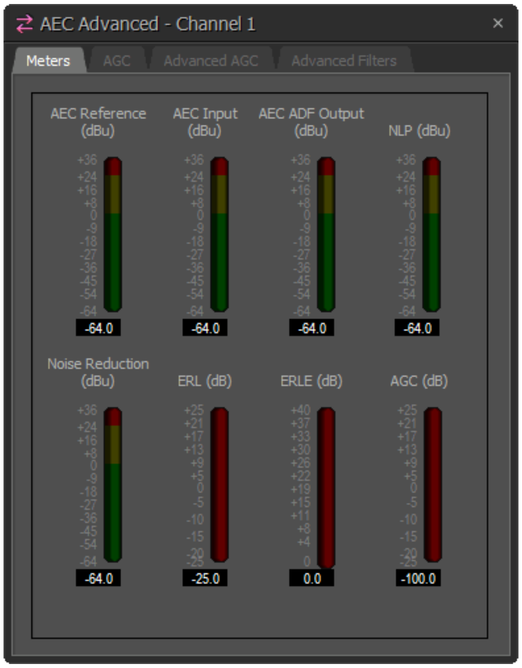
Conferencing Mode Setting	Round Trip Delay
Test	Bypassed
Telephone	<100ms
VoIP	100-250ms
Video	>250ms
Custom	See Note

Note: The Custom setting is for use by Biamp Technical Support personnel only.

- **Noise Reduction** is intended to reduce steady-state background noises, such as HVAC systems, fans, motors, or other mechanical devices, that may be picked up by the conferencing microphones and transmitted to the far end. Possible values are Off, Low, Medium, High and Custom. Use the lowest setting that achieves the desired level of background noise reduction. The Custom setting is for use by Biamp Technical Support personnel only.
- **Mute** turns the input signal on/off.
- **Level** adjusts the relative input volume.
- **Invert** reverses the polarity of the input signal.

AEC Channel Processing

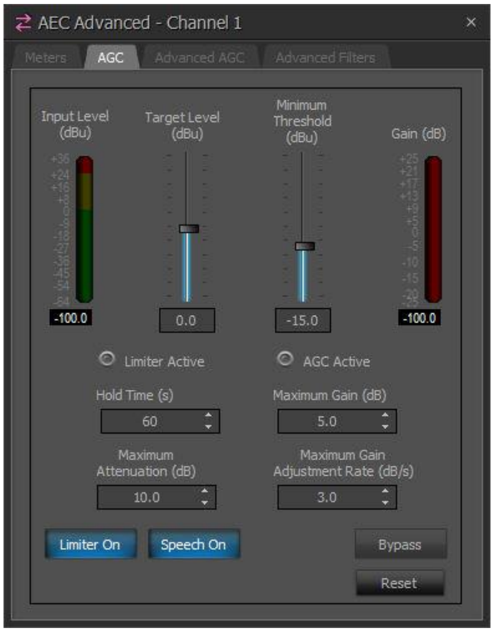
Each channel of AEC has a **Ch Processing** button, which opens up a control dialog window. The **Meters** tab gives information about how the AEC process is functioning.



- **AEC Reference** shows the signal level at the channel's AEC Reference input, and should nominally read 0dBu.
- **AEC Input** shows the signal level at the AEC filter input and should nominally read 0dBu.
- **AEC ADF Output** shows the signal level at the output of the adaptive filter (ADF) that performs most of the echo cancellation. This should report a somewhat lower signal level than the AEC Input, since some signal components will have been removed.
- **NLP** shows the signal level after the Non-Linear Processing filter.
- **Noise Reduction** shows the signal level at the output of the Noise Reduction Filter.
- **ERL (Echo Return Loss)** shows the difference in level between signal components arriving at the AEC reference and those same signal components arriving at the AEC input after having been introduced into a physical space and picked up by the microphone. Since these signal components will be attenuated somewhat by air absorption, ERL will normally indicate a positive amount of loss. If ERL is too positive or in the negative range, it may indicate a gain structure problem.
- **ERLE (Echo Return Loss Enhancement)** shows the amount of echo reduction the AEC adaptive filter is doing. It is the difference in level between echo components arriving at the filter's inputs and the residual echo remaining at the output of the filter.
- **AGC (Automatic Gain Control)** shows the amount of gain or attenuation being applied to the input signal.

The **AGC** and **AGC Advanced** tabs show the settings for the Automatic Gain Control feature built in to the SEC-4 AEC input card. The functionality is identical to the ACG component object in the Dynamic Blocks menu of the Object Toolbar.

AGC



- **Input Level** meter shows the level of the input signal.
- **Target Level** defines the signal level that the AGC block will constantly strive to output. If the input level is higher than the target level, the AGC block will subtract gain. If the input level is lower than the target level, the AGC block will add gain.
- **Minimum Threshold** is the minimum input signal level required for the AGC to make adjustments. The maximum value for the Minimum Threshold will be the Target Level. If the input signal level is lower than the minimum threshold, the AGC block will temporarily suspend gain adjustments.
- **Gain** shows how much gain is currently being added or subtracted from the input signal.
- **Limiter Active** will light when the clip limiter is actively engaged in preventing clipping.
- **AGC Active** will light when the AGC block is making a gain adjustment.
- **Hold Time** is the number of seconds that the AGC block will hold the current Gain setting while not receiving a qualifying input signal. After the Hold Time elapses, the AGC block will reset the gain to zero.
- **Maximum Gain** defines the maximum amount of gain that the AGC block will add to the signal.
- **Maximum Attenuation** defines the maximum amount of gain that the AGC block will subtract from the signal.
- **Maximum Gain Adjustment Rate** defines how quickly the AGC block can adjust the gain, specified in decibels per second.
- **Limiter On/Off** turns the clip limiter feature on or off. When the clip limiter is on, the AGC will temporarily reduce the gain applied to the input signal if that gain would have caused the signal to clip. Gain adjustments made by the clip limiter may briefly exceed the Maximum Gain Adjustment Rate as necessary to prevent clipping.
- **Speech On/Off** turns SpeechSense™ technology on or off. When Speech mode is on, the AGC analyzes the input signal to determine if it is human speech. Non-speech signals will not cause the AGC to adjust the gain when Speech mode is on.

Qualifying Input Signals

The AGC block will only adjust its gain when it receives a qualifying input signal. The definition of a qualifying input signal depends on whether Speech mode is on or off. When the input signal is not a qualifying signal, the AGC block will hold its previous gain setting until it receives a qualifying signal or until the Hold Time elapses.

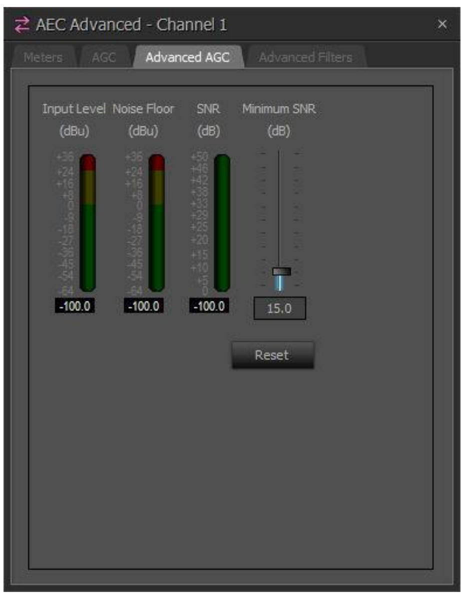
When Speech mode is ON, a qualifying signal must satisfy ALL of the following:

- Level of input signal must be above the specified Minimum Threshold.
- Input signal must be human speech.
- Signal-to-Noise Ratio must be above the specified Minimum SNR setting.

When Speech mode is OFF, a qualifying signal must satisfy the following:

- Level of input signal must be above the specified Minimum Threshold.

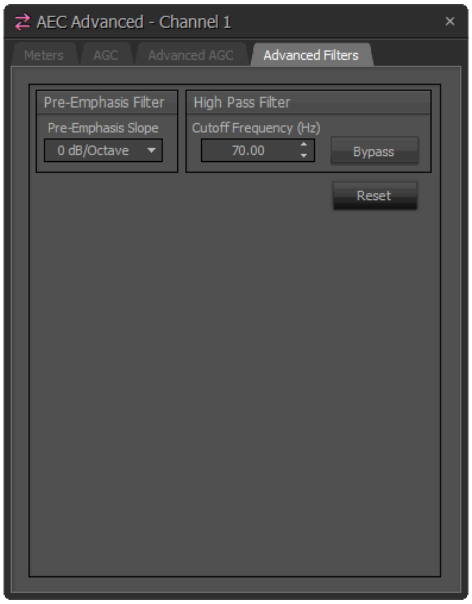
Adanced AGC



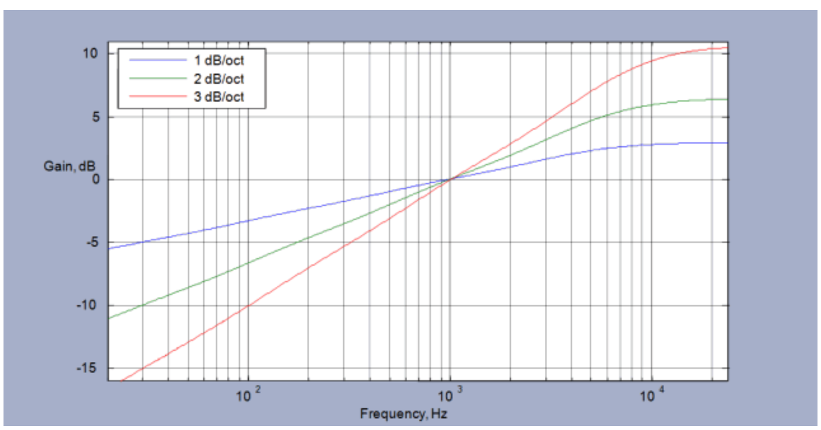
- **Noise Floor** meter shows the estimated level of the noise floor of the input signal. The Noise Floor is used in calculating the Signal-to-Noise Ratio.
- **SNR** meter shows the Signal-to-Noise Ratio of the input signal. This is equal to the Noise Floor level subtracted from the Input Level. In general, the closer the talker is to the microphone, the higher the Signal-to-Noise Ratio will be while they are talking.
- **Input Level** shows the level of the input signal.
- **Minimum SNR** determines how high the Signal-to-Noise Ratio (SNR) must be before the AGC will make gain adjustments. If the SNR is below the minimum, the AGC will temporarily suspend gain adjustments.

Advanced Filters

The **Advanced Filters** tab shows settings for two pre-AEC filter stages.



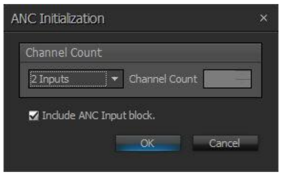
Pre-Emphasis Filter is a filter that emphasizes frequencies above 1kHz and de-emphasizes frequencies below 1kHz. It can be used to increase speech intelligibility in highly reverberant environments or in situations where the microphones are distant from the talkers.
Pre-Emphasis Slope specifies the slope of the emphasis curve in dB/octave, with 0 representing no effect or bypass and 3 representing maximum effect. The following table shows the magnitude response of the Pre-Emphasis Filter for slope values of 1, 2, and 3 dB/octave.



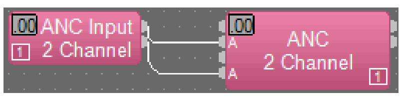
High Pass Filter is a filter that attenuates input signal components below a programmable **Cutoff** Frequency. Values in the range of 20Hz to 500Hz can be set. The filter is an elliptic 5th order design which attenuates frequencies below the cutoff at a slope of 30 dB/octave.

ANC Input

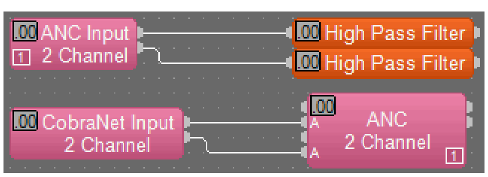
Automatic Noise Compensation (ANC) Inputs provide support for applications where automatic adjustment of zone level based on the ambient noise level is needed. When the ANC Input component is selected from the Object Toolbar, an ANC Initialization dialog is produced.



Channel Count determines the number of channels in the block. If Custom is selected from the drop-down list, any number of channels from 1 to 16 can be specified.
Include ANC Input Block specifies whether both the ANC Input and ANC processing blocks are created (checked) or only the ANC processing block (unchecked).

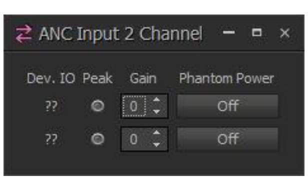


The ANC functionality is comprised of an ANC Input block and an ANC processing block. The two blocks will have a number in the lower corner, assigned by the software, which indicates which blocks are associated with each other, which is important when there are multiple ANC Inputs in the system. The ANC Input block contains the microphone preamp settings for the ambient sensing input and has one output connection per channel. The processing block has two inputs per channel, the program input and the ambient sensing input, designated **A.** Normally, the outputs of the ANC Input block are wired to the A inputs of the ANC processing block. However, the input and processing blocks can be separated, so the ambient sensing inputs can arrive via digital audio inputs from the network, for example. In this instance, the ANC Input block could still be used to bring analog mic or line inputs into Tesira.



ANC Input Block

Double clicking on the ANC Input block produces a control dialog window.

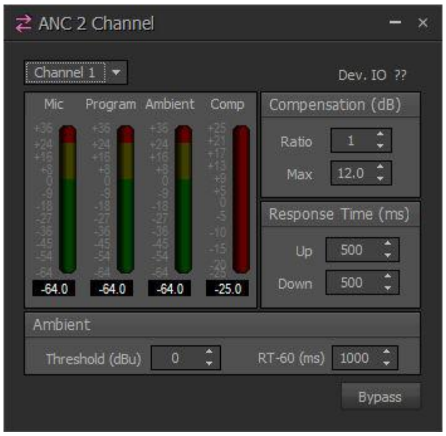


Device IO (x.y) indicates which physical hardware input is associated with that software channel, where x indicates which card slot and y indicates which channel on the card. **Peak**

is a software indicator that flashes when the input signal is within 6dB of clipping. **Gain** (0 to 66dB, in 6dB steps) sets the amount of analog gain for that channel and is used to compensate for differing input levels (mic or line). **Phantom Power** assigns +48 Volt phantom power to the input.

ANC Processing Block

Double clicking on the ANC processing block produces a control dialog window.



Use the **Channel** drop down menu to specify a channel for viewing/editing.

The **Mic** meter shows the signal level at the A (ambient sensing) input of the ANC processing block. **Program** shows the signal level at the program input. **Ambient** shows the level of the A input after any program audio signal components in the ambient sensing mic have been filtered out. The **Comp** meter shows how much gain compensation will be applied to the program input.

Under the **Compensation** heading, **Ratio** determines the how gain compensation is applied when the ambient level is above the threshold. It specifies the amount in dB the program gain is increased for every 1dB the Ambient level exceeds the ambient threshold. Values in the range of 0.25 to 1dB are possible. **Max** specifies the maximum amount of gain the ANC process can apply to the program input. Values in the range of 0 to 25dB are possible.

Under the **Response Time** heading, **Up** and **Down** determine how quickly a calculated gain change is applied to the program signal. Separate values can be programmed for how quickly gain is applied when the ambient level increases and how quickly it is removed when the ambient level decreases. Values in the range of 500 to 300,000 ms can be set.

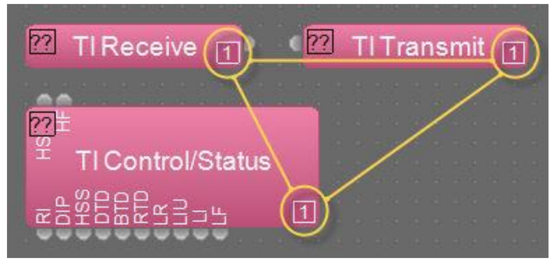
Under the **Ambient** heading, **Threshold (dBu)** specifies the level of the ambient sensing input above which the ANC process will begin to add gain to the program audio. If the ambient sensing input falls below this threshold, ANC will have no effect on the program audio. **RT-60 (ms)** is used to optimize the cancellation of program audio from the ambient sensing input. In large, reverberant spaces, it is important to let the filter know how long to listen for reflected versions of the program audio which may by picked up by the ambient sensing microphone.

The **Bypass** button stops the ANC process and returns the program signal to its normal, uncompensated level.

Telephone Interface

The Telephone Interface provides support and control for analog telephone lines. The Telephone Interface consists of three blocks,

- TI Receive
- TI Transmit
- TI Control/Status



The component objects will have a number on the lower right, assigned by the software, which indicates which blocks are associated with each other, which is important when there are multiple Telephone Interfaces in the system. In addition, a dialer block can be associated with telephone or VoIP interfaces. The [Dialer] control is available in the **Object Toolbar > Control** section.

Initialization Dialog

The Telephone Interface provides support and control for analog telephone lines. When the Telephone Interface is created from the [Object Toolbar], an initialization dialog is produced. Select the **Country** the system will be installed in, to configure the telephone line and call signaling properties appropriately.

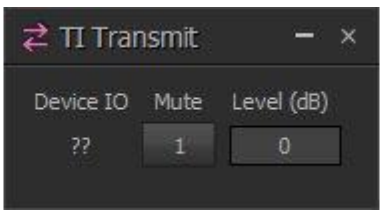
TI Receive

The **TI Receive block** is an input for received audio coming into the system via the telephone line.

Name	Description
Device IO	Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted as x.y - where x indicates which card slot and y indicates which channel on the card.
Input Mute	turns the signal off/on.
Input Level (dB)	controls the volume of the signal
Ring Tone Level (dB)	controls the volume of the generated ring tone when an incoming call is received.

TI Transmit

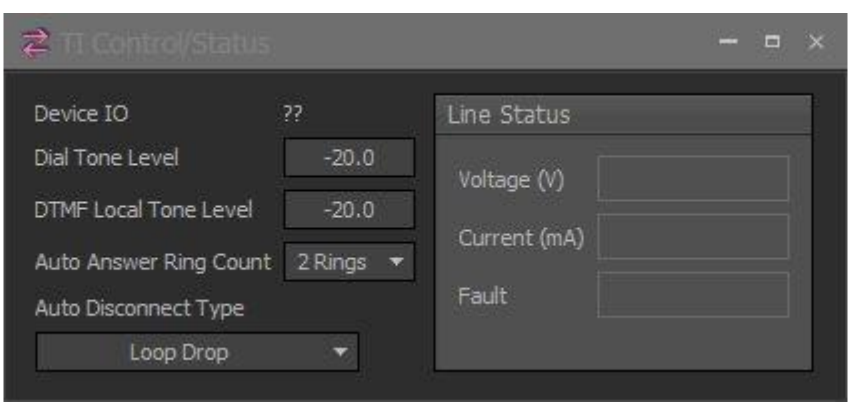
The **TI Transmit** block is an output for sending audio out of the system via the telephone line.



Name	Description
Device IO	Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted as x.y - where x indicates which card slot and y indicates which channel on the card.
Mute	turns the signal off/on.
Level (dB)	controls the volume of the signal

TI Control Status

TI Control/Status provides global line settings and status.



Device IO	Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted as x.y - where x indicates which card slot and y indicates which channel on the card.
DTMF Local Tone Level	sets the local volume of any generated DTMF tones.
Auto Answer Ring Count	sets the number of rings of an incoming call before the Telephone Interface will automatically answer the call, if Auto Answer is enabled on that line.
Auto Disconnect Type	sets what type of call termination signal the Telephone Interface will use to automatically disconnect the line when the call is finished. **None**, **Loop Drop**, **Call Progress** and **Loop Drop + Call Progress** can be selected.
Line Status	provides real-time information about the line **Voltage (V)** and **Current (mA)**. If either of these values is outside of the normal expected range, a **Fault** message will be displayed.

Note: Line Status values on different phone systems may have different readings, but expected Line values will be in the region of 7.5V 32mA Off Hook. 54V, 0mA when On Hook.

TI Control Status Logic

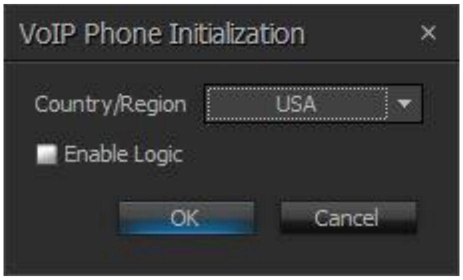
The **TI Control/Status** block has a number of logic nodes that provide additional functionality.

- **HS (Hook Switch)** will take the line off hook on a rising edge and go on hook on a falling edge.
- **HF (Hook Flash)** will initiate a hook flash on a rising edge. The amount of time the line is momentarily disconnected is controlled by the **Hook Flash Duration (ms)** setting in the DSP Properties tab in the Properties window of the TI Control/Status block.
- **RI (Ring Indicator)** is a logic output that is normally low, and goes high when the corresponding line is ringing, and through the entire duration of ringing. When ringing has stopped, RI will go back to a low state.
- **DIP (Dial In Progress)** is normally low, and goes high during any dial operation.
- **HSS (Hook Switch State)** will be low when the Telephone Interface is on hook and high when off hook.
- **DTD (Dial Tone Detect)** is normally low and will go high when a dial tone is detected on the line.
- **BTD (Busy Tone Detect)** is normally low and will go high if a busy signal is detected on the line.
- **RTD (Ring Tone Detect)** is normally low and will go high if the far end of the outgoing call is ringing. This should not be confused with the RI (Ring Indicator) node, which indicates when a ring is incoming.
- **LR (Line Ready)** will output a high state when a valid telephone line is connected (line voltage and current within expected parameters).
- **LIU (Line In Use)** will output a high state when the phone is off hook or ringing.
- **LI (Line Intrusion)** will output a high state if another extension on the active line goes off hook. This is determined by sampling the line current when the Telephone Interface initially goes off hook. If the current rises by an amount consistent with another phone off hook on the line, this logic node will go high.

- **LF (Line Fault)** is normally low and will go high when a fault condition is detected (line voltage or current outside expected ranges). This would correspond to a message in the Fault field of the Line Status indicators in the TI Control/Status block.

VoIP Phone

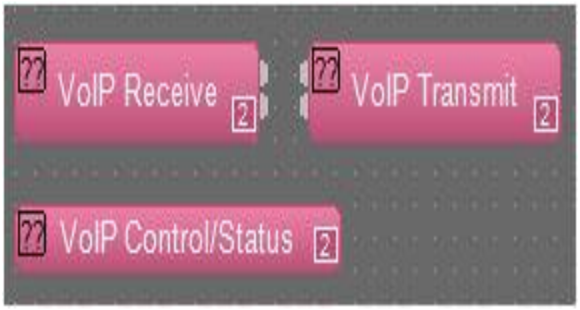
The **VoIP Phone** provides support for IP phone systems. When the VoIP phone is selected from the [Object Toolbar], an initialization dialog is displayed.



Under the **Country** drop down menu, select the country in which the system will be operating, to configure the local call progress tones accordingly. **Enable Logic**, if checked, will cause the logic support to be shown on the VoIP Control/Status block.

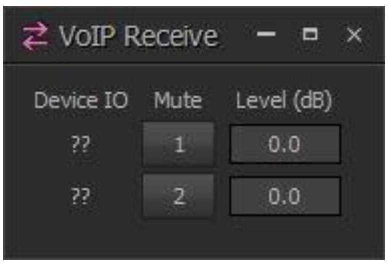
Default VoIP Blocks

The VoIP Phone is comprised of a **VoIP Receive**, **VoIP Transmit** and **VoIP Control / Status** blocks. The three blocks will have a number on the lower right, assigned by the software, which indicates which blocks are associated with each other, which is important when there are multiple VoIP Phones in the system. In addition, a dialer block can be associated with telephone or VoIP interfaces. The [Dialer] control is available in the **Object Toolbar > Control** section.



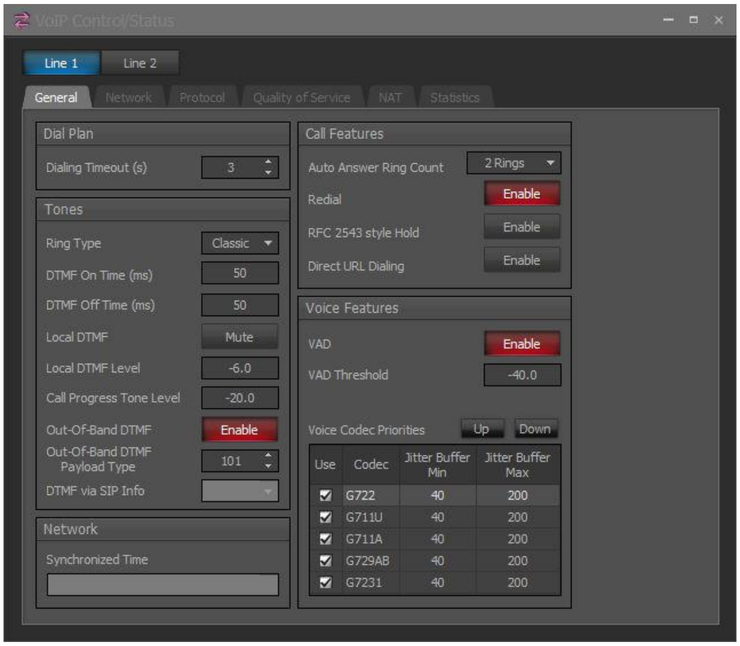
VoIP Receive and Transmit

The **VoIP Receive block** is an input for received audio coming into the system via the VoIP interface. The **VoIP Transmit block** is an output to the VoIP telephone system



- **Device IO** (x.y) Indicates which physical hardware input is associated with that software channel. For Server and Server-IO devices is formatted as x.y - where x indicates which card slot and y indicates which channel on the card.
- **Mute** turns the output signal off/on.
- **Level (dB)** controls the volume of the output signal.

VoIP Control Status



Use the **Line Select** buttons to display line 1 or 2 for editing or viewing. Several tabs show screens for setting general properties and viewing status information about the VoIP Phone when the system is connected.

General tab

Under **Dial Plan:**

- **Dialing Timeout (s)** specifies how long after the last digit is entered before the VoIP Phone will consider the dial string complete and issue a dial request.

Under **Tones:**

- **Ring Type** selects whether the standard ring tone will be heard or not (Classic or Silent) when a call comes into the VoIP Phone.
- **DTMF On Time (ms)** sets the duration of generated DTMF tones.
- **DTMF Off Time (ms)** sets the length of the pause between successive DTMF digits.
- **Local DTMF Mute** turns local DTMF generation off/on.
- **Local DTMF Level** controls the local volume of generated DTMF tones.
- **Call Progress Tone Level** controls the volume of call progress tones, such as the dial tone, busy tone and other signals.

DTMF Settings.

The VoIP card supports inband DTMF, out of band DTMF (using RFC2833) and DTMF via SIPInfo. The three DTMF modes are mutually exclusive.

- **Out Of Band DTMF** -When this button is enabled all DTMF signals will be sent to the far end via RTP Events in the RTP stream.
- **Out-Of-Band DTMF Payload Type** - This selection allows you to set the RTP Payload type reserved for Out-of-Band DTMF signals. 101 is the most typical (and default) setting for Out-of-Band DTMF however if this Payload Type is in use for

something else it can be changed to something that is available. The Payload Type is adjustable between 97 to 127.

- **DTMF via SIP Info** - When this feature is enabled all DTMF signals will be delivered using SIP protocol. Setting the SIP Info to Normal will include the DTMF digit information as well as the duration of the DTMF tone. Setting the SIP Info field to Simple will only deliver the DTMF digit information.

Note: If Out-of-Band DTMF is disabled and SIP Info is set to off the SVC-2 card will revert to using In-Band DTMF. This means DTMF tones will be produced by the SVC-2 card and sent to the far end as audio using the selected VoIP CODEC. This may be required if the far end device does not support one of the VoIP DTMF signal transfers mentioned above.

Network:

- **Synchronized Time** shows the VoIP Phone's time as obtained from the network or as set in the DSP Properties tab in Properties of the VoIP Control/Status block. This will be used for authenticating security certificates which depend on time stamps.

Call Features:

- **Auto Answer Ring Count** (Immediately, or 1, 2, or 3 Rings), **Redial Enable**, **RFC 2543 style Hold Enable** and **Direct URL Dialing Enable** can be set.

Voice Features:

- **VAD (Voice Activity Detection) Enable** and **VAD Threshold** can be set, as well as **Voice Codec Priorities**. A list of supported voice codecs is shown in descending order of priority. Use the **Up** and **Down** buttons to change the priority order of the selected codec. Remove the check from the **Use** column to remove that codec from use.
- The the **Jitter Buffer Min** and **Max** can be set to compensate for network conditions. Increasing the buffer sizes may improve call quality at the expense of additional delay. Decreasing the buffer sizes may improve delay at the expense of call quality. These settings are adjustable per codec, per line.

Network Tab

When connected to a system, the Network Tab displays read only information**.** The settings displayed here are based on the settings configured in the VoIP Property Sheet**.**

Protocol Tab

When connected to a system, the Protocol Tab displays read only information. The settings displayed here are based on the settings configured in the VoIP Property Sheet and VoIP Line Properties Protocol Tab.

Quality of Service

When connected to a system, the Protocol Tab displays read only information. The settings displayed here are based on the settings configured in the VoIP Property Sheet and VoIP Line Properties QoS Tab.

NAT

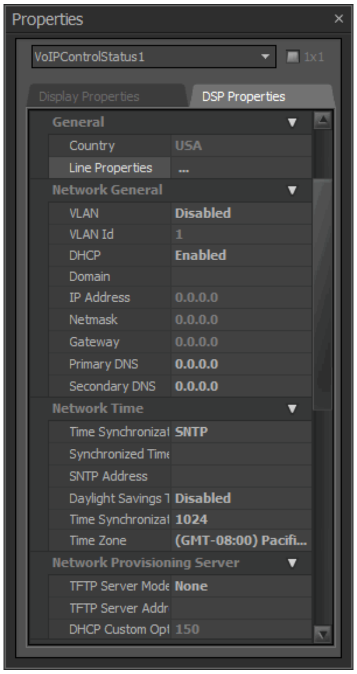
When connected to a system, the Protocol Tab displays read only information. The settings displayed here are based on the settings configured in the VoIP Property Sheet and VoIP Line Properties QoS Tab.

Statistics

When connected to a system, the Statistics Tab displays read only information. The settings displayed here include network, call and firmware information.

VoIP Properties

VoIP Property Sheet



In the **DSP Properties** tab of the Properties window of the VoIP Control/Status block, there are additional settings which are global for the VoIP Phone card.

General

Country - Allows you to specify the country the VoIP phone is operating in.

Line Properties - Opens the VoIP Line Properties dialog to allow per line configuration fo the VoIP card. Some of the settings in this dialog are only available if enabled via the Property sheet.

Network General

VLAN and VLAN Id - When enabled, the VoIP card will only respond to and transmit to packets tagged with this VLAN ID number. VLAN's can also be configured by switch port. This option should only be enabled if requested by the network administrator.

DHCP / IP & **Network Address** - If DHCP is enabled, the DHCP server should provide the **IP Address**, **Subnet Mask**, **Gateway** and **Primary/Secondary DNS**.

Domain - This setting specifies the search domain for **DNS** names. For example, if the domain is set to "example.com" and the proxy is set to "voip", the result would be "voip.example.com". This setting is only enabled when DHCP is not being used; otherwise the DHCP server can provide the domain details.

Network Time

Time Synchronization Mode - If Simple Network Time Protocol (**SNTP**) is selected the IP address of the SNTP Server is required in the **SNTP Address** field for automatic network time synchronization. If SNTP is used the **Time syncronization interval** and **timezone** can also be specified.

If **Static** is set a value should be entered in the **Synchronized Time**

Network Provisioning Server

- TFTP Server Mode, TFTP Server Address and DHCP Custom Option can be set.

Network Ethernet

- Settings for speed and duplex properties of the VoIP Phone card.

QoS
QoS Mode - Selects the Quality of Service mechanism, TOS or DiffServ.
Diffserv:

- **L2 Other User Priority** - Sets the priority of VLAN tagged frames. 0-3 is low priority, 4-7 is high priority.
- **Other DiffServ** - Specifies how QoS is to be handled to and from other DiffServ capable domains.

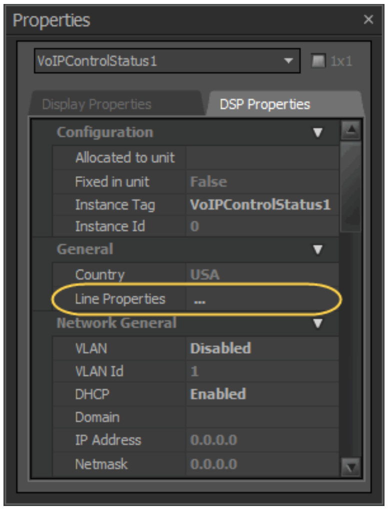
TOS

- **L3 Other Precedence -** The Layer 3 Other Precedence uses 8 levels of priority, numbered 0-7, with 0 being the lowest priority and 7 the highest for managing the priority of all traffic, other than SIP and RTP data packets.
- **L3 Other Min Delay -** All traffic, other than SIP and RTP data packets to be forwarded with minimum delay.
- **L3 Max Throughput -** All traffic, other than SIP and RTP data packets to be forwarded with maximum throughput.
- **L3 Max Reliability -** All traffic, other than SIP and RTP data packets to be forwarded with maximum reliability.
- **L3 Min Cost** - All traffic, other than SIP and RTP data packets to be forwarded with minimum cost to network bandwidth.

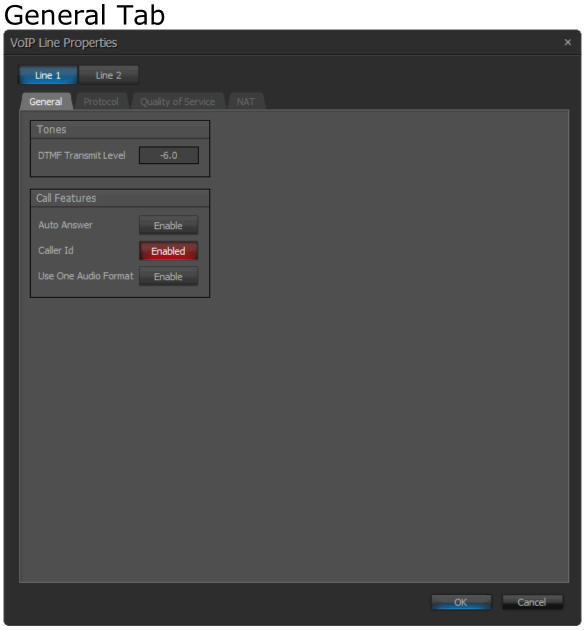
Protocol SIP
Transport can be set to UDP, TCP or TLS.
Protocol SIPS
Is available if TLS is specified in the Protocol SIP section. **Certificate Preference** can be set to Fully Verify, Trust, Keyword, or Accept All.
Root Certificate File Name, **Customized Certificate File Name**, **Certificate File Name** and **Private Key File Name** can be specified.

VoIP Line Properties

Configuring a VoIP Phone Line
Configuring a VoIP Phone line is accomplished by clicking on the Line Properties entry in the General section of the DSP Properties tab in the Properties window of the VoIP Control/Status block.
This opens a new control dialog window. Use the Line Select buttons to choose Line 1 or Line 2 for editing.



General Tab



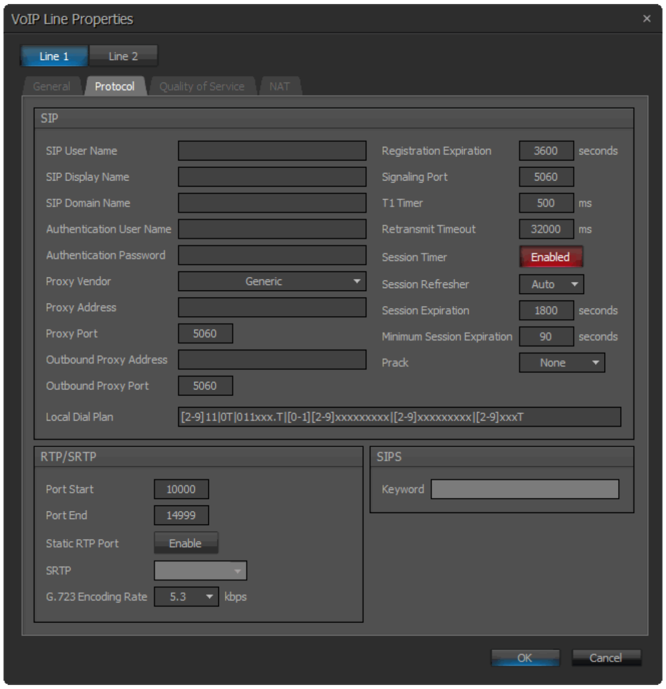
Tones

- **DTMF Transmit Level** - Sets the volume of outgoing DTMF tones. Range between -100 - 0dB. Default -6.

Call Features

- **Auto Answer** - Enables Auto Answer - The Auto Answer Ring Count in the VoIP Control/Status - General / Call features section will specify the number of rings.
- **Caller ID** - Shows the Caller ID in TTP updates and the Dialer.
- **Use One Audio Format** - Acknowledges the SIP “invite” with multiple audio format or one audio format.

Protocol Tab



Under SIP

- **SIP User Name -** is the alphanumeric string that identifies the VoIP extension on the network. It is the number or string you would need to dial to reach this extension
- **SIP Display Name** - is the string used for Caller ID name purposes.
- **SIP Domain Name** - The SIP domain name to be used.
- **Authentication User Name / Authentication Password** - the credentials needed to register and authenticate with the VoIP proxy server.
- **Proxy Vendor** - choose the entry that matches the phone system the VoIP Phone is integrating with. Possible selections are Generic, Avaya SES, Avaya SM, Avaya IP Office, Avaya CS1000, Cisco, and ShoreTel. If an exact match does not appear in the list, select Generic.
- **Proxy Address** - the network address of the VoIP proxy server
- **Proxy Port** - is the network port the VoIP Phone should use to communicate with the proxy server. Port 5060 is a standard port used in VoIP systems, but this number can be modified if need be.
- **Outbound Proxy Address / Outbound Proxy Port** - If a separate proxy server is used for inbound versus outbound traffic to specify the network address and port number of the outbound server. Most IP phone systems use a single proxy server for inbound and outbound, in which case Outbound Proxy Address should be left blank.
- **Local Dial Plan** -is a regular expression which determines dialing behavior according to the method specified in RFC 3435. Use the default Local Dial Plan string unless an alternate one has been provided for you. Default Local Dial Plan: [2-9]11|0T|011xxx.T|[0-1][2-9]xxxxxxxxx|[2-9]xxxxxxxxx|[2-9]xxxT
- **Registration Expiration** - determines the interval the VoIP line will attempt to re-register with the Proxy. Note that the proxy may override this setting with a value of its own. If an acknowledgement has not been received from the Proxy within the agreed time the VoIP card registration information kept in the proxy's database will be cleared. The default registration expiration period is 3600 seconds and should be left at this value unless specified by the network administrator. Can be set between 60 to 86400 seconds.

- **Signaling Port -** The signaling Port is used to direct incoming SIP traffic to the correct Line for communications between the VoIP card and the Proxy. The default port for Line 1 is 5060 and the default port for Line 2 is 5062. These settings should be left at this value unless specified by the network administrator
- **T1 Timer** - This timer is used when sending requests over UDP. If the response is not received within this interval, the request is retransmitted. The retransmission interval is doubled after each retransmission.
- **Retransmit Timeout** - The total amount of time the card will continue to retransmit a UDP packet that has not been responded to.
- **Session Timer** - Enables periodic refresh of SIP sessions through a Re-INVITE or UPDATE request. When disabled the Session Refresher, Session Expiration and Minimum Session Expiration options will be disregarded. If a call unexpectedly disconnects, disabling this option may help.
- **Session Refresher** - In a SIP session that utilizes a session timer, the Session Refresher is the device that will send the periodic Session Refresh requests to refresh the session.
- Refresher Options:
    - Auto - This (Default) setting allows both ends of the call to negotiate who will be the refresher. Typically this leaves the decision to the device receiving the SIP packets. This setting should be used unless specified otherwise by the network administrator.
    - UAS - The User Agent Server (UAS) is the VoIP device that responds to the SIP Request. In the case of a phone call it would be considered the "called" device. Engaging this setting will ensure that the SVC-2 card will only negotiate to a Session Timer where the UAS is nominated the refresher.
    - UAC - The User Agent Client (UAC) is the VoIP device that send the SIP Request. In the case of a phone call it would be considered the "calling" device. Engaging this setting will ensure that the SVC-2 card will only negotiate to a Session Timer where the UAC is nominated the refresher.
    - Local - This setting will ensure that the SVC-2 card will always be the refresher of a Session Refresh.
    - Peer - This setting will ensure that the SVC-2 card will never be the refresher of a Session Refresh.
- **Session Expiration** - Determines the interval the VoIP card will try to negotiate with the Proxy to keep the VoIP session alive. Note that the proxy may override this setting with a value of its own. If a Session Refresh request is not properly received by both parties within this agreed time, the session will expire and the call ended. Can be set between 90 to 65535 seconds, the default is 1800 and should be used unless specified otherwise by the network administrator.
- **Minimum Session Expiration** - If the proxy tries to override the Session Expiration value as specified in the VoIP card, the time entered in this field will be the minimum value allowed. Can be set between 90 to 65535 seconds, default = 90
- **Prack** - guarantees a reliable and ordered delivery of provisional responses in SIP. PRACK Improves network reliability by adding an acknowledgement system to the provisional Responses. Can be set to None, Supported, Required.

Under RTP/SRTP

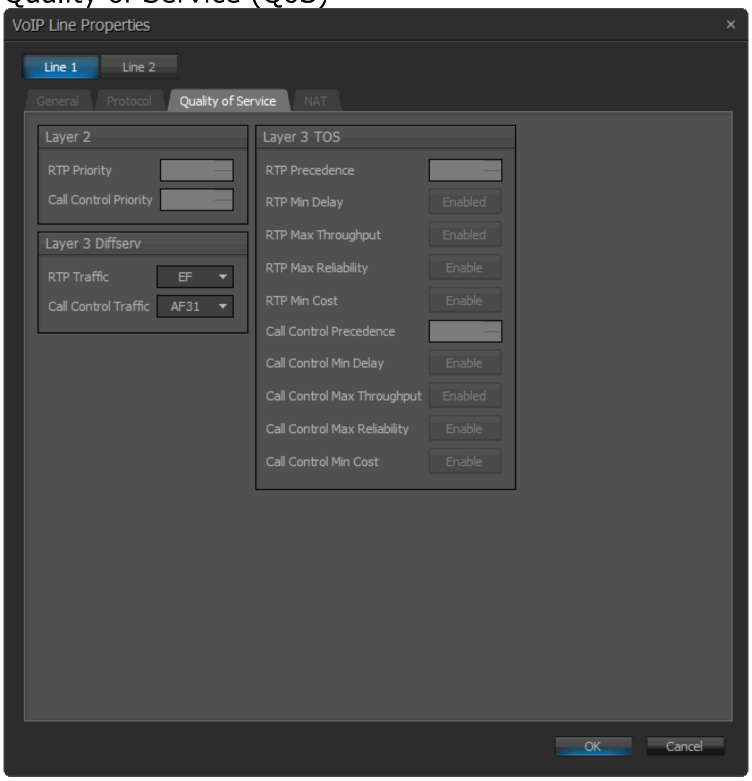
- **Port Start** - The first RTP Port used by this line. Must be between 4000 - 65534 and must be one less than the Port End
- **Port End** - The last RTP port used by this line. Must be between 4001 - 65535 and must be one more than the Port Start.
- **Static RTP Port** - Static Real-time Transport Protocol Port - The Port number used for RTP traffic

- **SRTP** - Secure Real-time Transport Protocol - Provides encryption of the RTP audio data. Available if Transport is set to TCP or TLS. Go To VoIP Control/ Status Block, open the property sheet DSP properties, Set Protocol SIP-Transport to TCP or TLS. Can be set to Disabled, Allowed, Preferred or Required
- **G.723 Encoding Rate** -Defines the G.723 bit rate. The options available are 5.3 and 6.3 kbps.

Under SIPS Protocol

SIPS Keyword - This field is used to enter the keyword used for secure SIP on a per-Line basis. To enable Secure SIP, go To VoIP Control/ Status Block, open the property sheet DSP properties, Set Protocol SIP-Transport to TLS.

Quality of Service (QoS)



Under Layer 2

- RTP Priority and Call Control Priority QoS levels can be set, if QoS Mode is set to TOS and VLAN is set to Enabled.

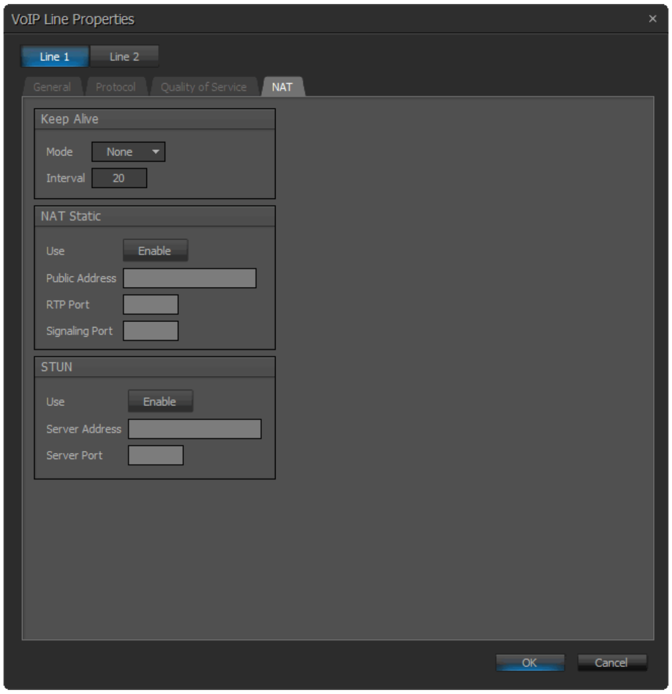
Under Layer 3 DiffServ,

- RTP Traffic and Call Control Traffic levels can be set if QoS Mode is set to DiffServ.

Under Layer 3 TOS

- If QoS Mode is set to TOS, RTP Precedence level can be set, RTP Minimum Delay, RTP Max Throughput, RTP Max Reliability and RTP Minimum Cost can be Enabled, Call Control Precedence level can be set, and Call Control Min Delay, Call Control Max Throughput, Call Control Max Reliability and Call Control Min Cost can be Enabled.

NAT



Keep Alive

- Mode can be set to None, Options, Register, or CRLF. Interval can be set from 20 to 30; default is 20.

NAT Static and STUN

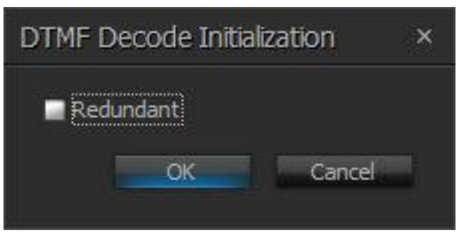
- NAT Static and STUN are mutually exclusive options; enabling one disables the other. When NAT Static is enabled, a Public Address can be entered, and an RTP Port and Signaling Port specified. When STUN is enabled, a Server Address and Server Port can be entered.

DTMF Decode

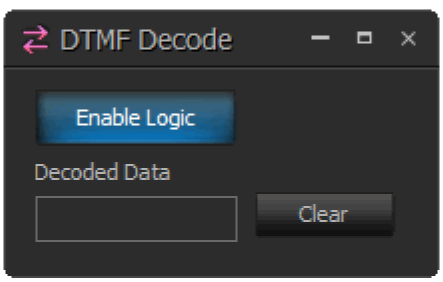
The **DTMF Decode** block allows digits to be decoded from any audio input. The input signal could originate from a telephone or VoIP receive block, or from any analog or digital input.

The Component Object has 17 logic output nodes, 16 for all the possible DTMF digits (**0**-**9**, *****, **#**, **A**, **B**, **C**, **D**) and one additional output, **Any**, which is the logical OR of all the other nodes. That is, if any valid DTMF digit is received, the corresponding output node and the **Any** node will trigger. Double clicking on the DTMF Decode block produces a control dialog window.

Initialization Dialog



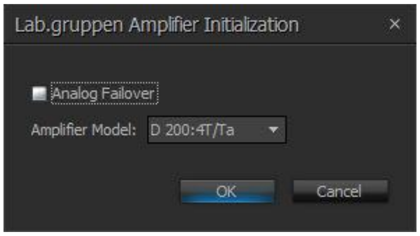
Control Dialog



Enable Logic turns the logic output nodes on the block on and off. **Decoded Data** shows the last few decoded digits. **Clear** removes any digits from this display.

Lab.gruppen Amplifier

The Lab.gruppen amplifier is intended to be configured using the Device Import/Export docking window and the lab.gruppen editing software.



Name	Description	Range
Analog Failover	Specifies if the analog inputs on the amplifier act as failover inputs	
Amplifier Model	Specifies the Amplifier model.	D 200:4T / Ta D 120:4T / Ta D 80:4T / Ta

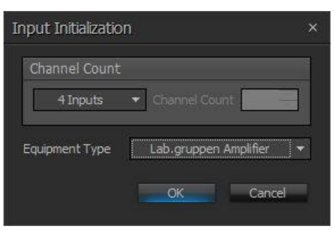
The Lab.gruppen Amplifier block has 4 audio channels, one logic input and 5 logic outputs.

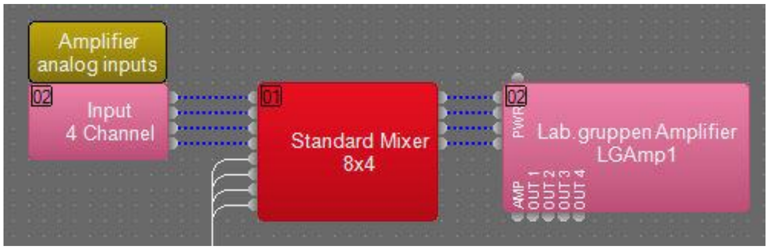
Logic In
- PWR - Toggles between ON and STANDBY.

Logic Out
- AMP - The status of the amplifier frame
- OUT 1 - The Status of Amplifier Channel 1
- OUT 2 - The Status of Amplifier Channel 2
- OUT 3 - The Status of Amplifier Channel 3
- OUT 4 - The Status of Amplifier Channel 4

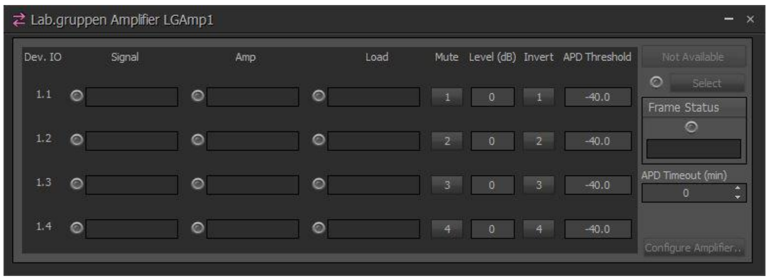
Associating analog inputs

Add a Four channel Analog Input block and set the equipment type to Lab.Gruppen Amplifier. This will allow the analog audio from the amplifier to be streamed to a Tesira SERVER, Server IO or TesiraFORTÉ device for further processing.





Dialog



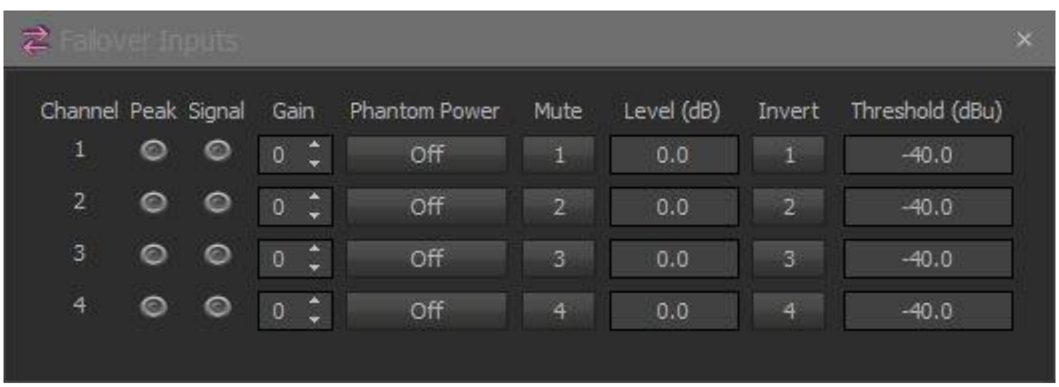
The main areas of the dialog indicates the channel and audio levels:

Name	Description	Range
Device IO	Indicates which physical hardware input is associated with that software channel.	
Signal	Signal present LED based on threshold	
Amp	Displays the amplifier channel status	
Load	sets the amount of analog gain for that channel and is used to compensate for differing input levels (mic or line).	0-66 in 6dB steps
Mute	turns the input signal on/off.	On or Off
Level	adjusts the relative input volume.	-100 to +12
Invert	adjusts the polarity of the input signal.	0° or 180°
APD Threshold	If all the incoming signals are below their threshold for the specified length of time, the amplifier will power down.	-100 to 0 Default -40
Failover	A Test Button to force a Failover from the AVB network audio to the local analog input. (Ta Model Only)	

The right side of the dialog indicates the device availability and other amplifier settings:

Name	Description	Range
Availability	Displays the current availability of the device when online	NOT AVAILABLE, STANDBY, ON
Select	LED and button. Causes the dialog and device LED's to light up green for a short time.	Off or Green
Frame Status	Will indicate the amplifier status	Green, Yellow, Red
APD Timeout	If all the incoming signals are below their threshold for the specified length of time, the amplifier will power down.	0 = disabled 1-60 minutes
Failover Inputs	available on the Lab.gruppen Ta version. Will open the Failover Input Dialog	
Configure Amplifier	Opens the Lab.gruppen Cafe software for additional amplifier configuration	

Failover Input dialog



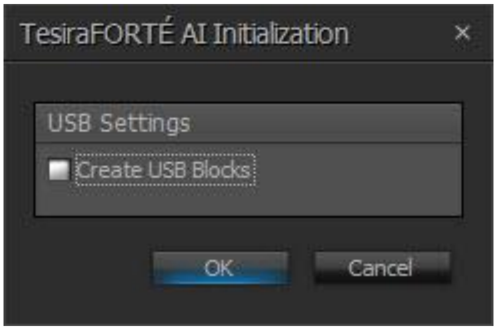
Name	Description	Range
Channel		
Peak	a software indicator that flashes when the input signal is within 3dB of clipping.	
Signal	Signal present LED based on threshold	
Gain	sets the amount of analog gain for that channel and is used to compensate for differing input levels (mic or line).	0-66 in 6dB steps

Phantom Power	assigns +48 Volt phantom power to the input for use with condenser microphones	
Mute	turns the input signal on/off.	On or Off
Level	adjusts the relative input volume.	-100 to +12
Invert	adjusts the polarity of the input signal.	0° or 180°
Threshold	Level reference for signal present meter. Default -40.	-64 to 30 dBu

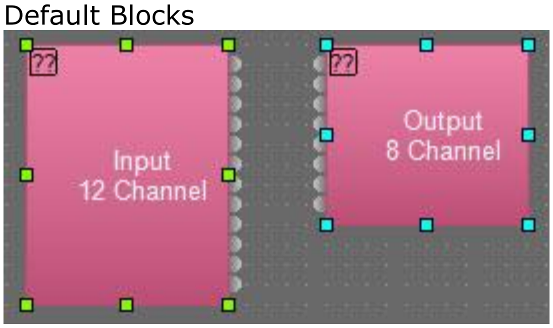
TesiraFORTÉ

FORTÉ AI

TesiraFORTÉ AI
The TesiraFORTÉ AI component objects provide the input and output blocks associated with a FORTÉ AI hardware device.



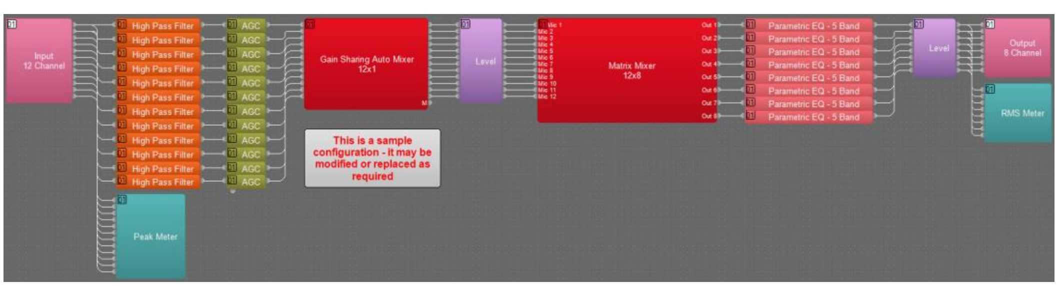
The TesiraFORTÉ Initialization dialog allows USB settings to be specified. Selecting the 'Create USB Blocks' option will open the USB Channel Initialization dialog.
Default Blocks



The TesiraFORTÉ AI will add the following default component objects:

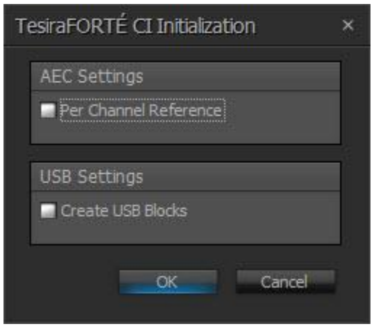
- 1 x 12 Channel Input block
- 1 x 8 Channel Output block

Optional:
Adding USB I/O Blocks - depending on the initialization settings the corresponding Input and Output block will be added.
Default Configuration
Each TesiraFORTÉ device comes pre-configured from the factory with a default system design.(see design layout below). If you wish to load your own custom designed configuration you will need to reset/initialize the unit(s) first.

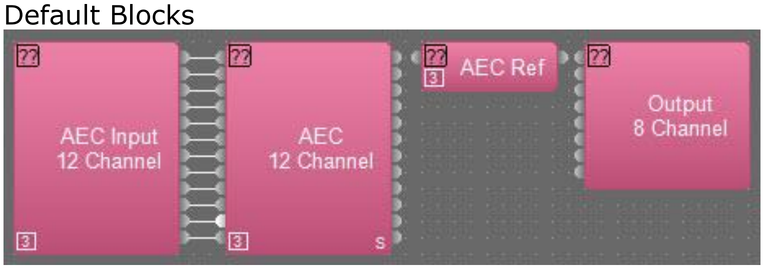


FORTÉ CI

TesiraFORTÉ CI
The TesiraFORTÉ CI Component objects add input and output blocks associated with a FORTÉ CI hardware device.



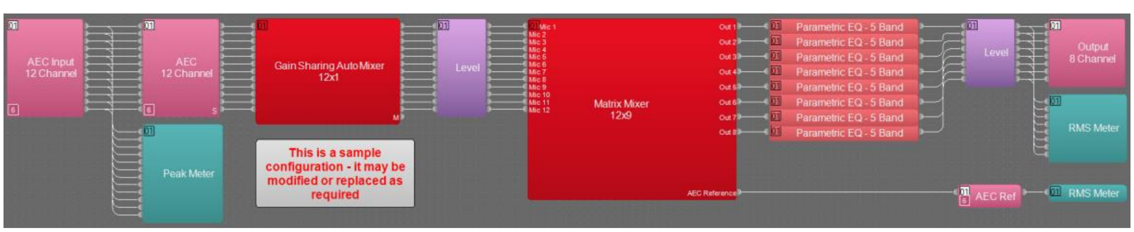
The TesiraFORTÉ Initialization dialog will allow the setting of a Per channel AEC reference and allow USB settings to be specified.
Selecting the 'Create USB Blocks' option will open the USB Initialization dialog.
Default Blocks



The TesiraFORTÉ CI will add the following default component objects:

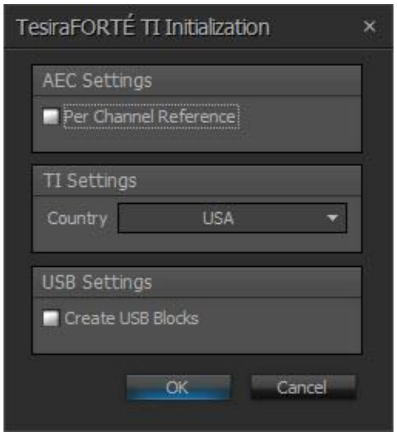
- 1 x AEC Input 12 Channel
- 1x AEC Processing Block 12 Channel
- 1x AEC Reference
- 1x Output 8 Channel

Optional:
Adding USB I/O Blocks - depending on the initialization settings the corresponding Input and Output block will be added.
Default Configuration
Each TesiraFORTÉ device comes pre-configured from the factory with a default system design. If you wish to load your own custom designed configuration you will need to reset/initialize the unit(s) first.

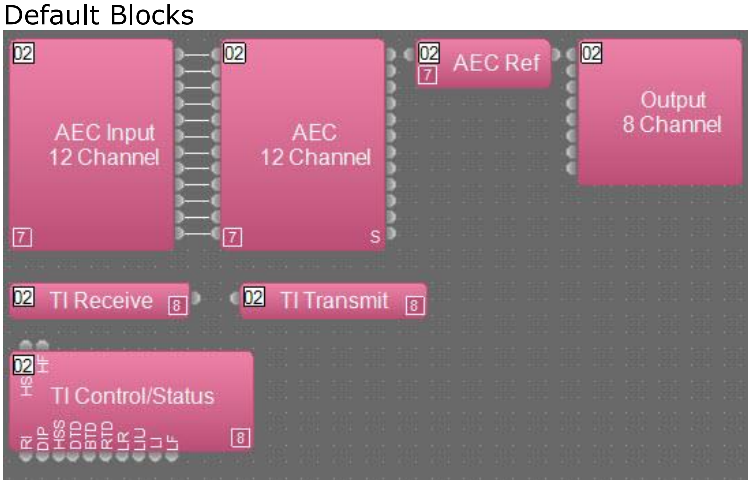


FORTÉ TI

TesiraFORTÉ TI
The TesiraFORTÉ TI Component objects adds Input and output blocks associated with a FORTÉ TI hardware device.



The TesiraFORTÉ Initialization dialog will allow the setting of a Per channel AEC reference, the Telephone Country settings and allow USB settings to be specified.
Selecting the 'Create USB Blocks' option will open the USB Initialization dialog.
Default Blocks

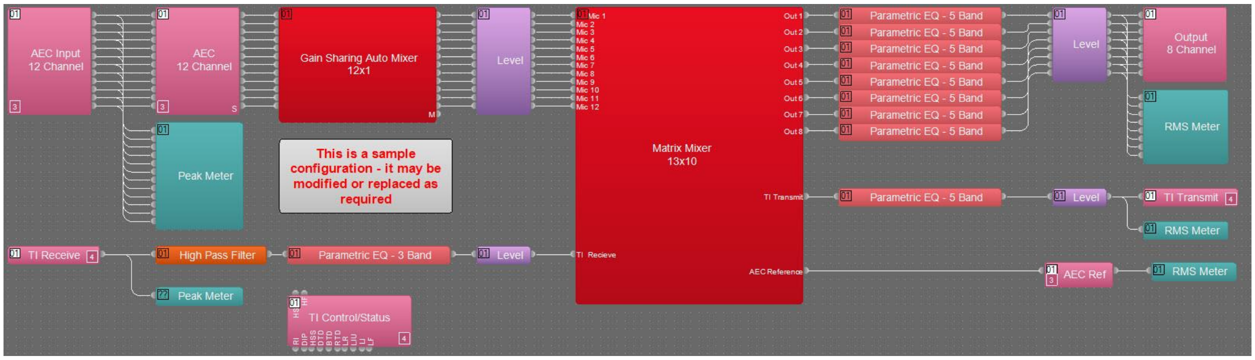


The TesiraFORTÉ TI will add the following default component objects:

- 1x AEC Input 12 Channel
- 1x AEC Processing Block 12 Channel
- 1x AEC Reference
- 1x Output 8 Channel
- 1x Telephone Interface
- 1 channel Input, Output and Control Status Block

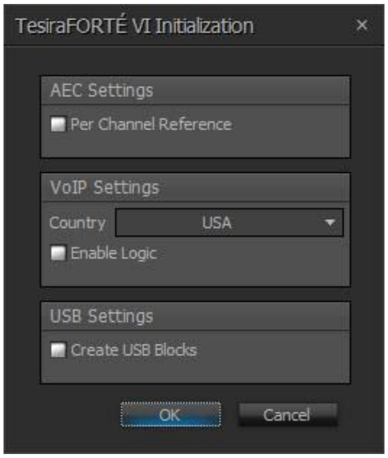
Optional:
Adding USB I/O Blocks - depending on the initialization settings the corresponding Input and Output block will be added.
Default Configuration

Each TesiraFORTÉ device comes pre-configured from the factory with a default system design. If you wish to load your own custom designed configuration you will need to reset/initialize the unit(s) first.



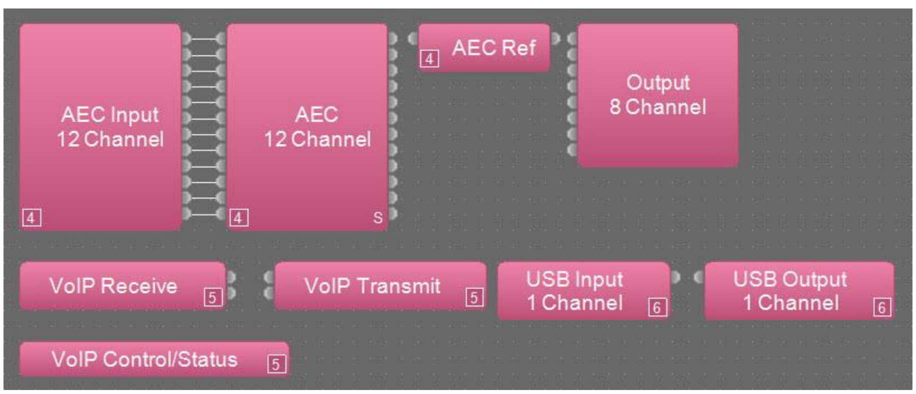
FORTÉ VI

TesiraFORTÉ VI
The TesiraFORTÉ TI Component objects adds Input and output blocks associated with a FORTÉ TI hardware device.



The TesiraFORTÉ Initialization dialog will allow the setting of a Per channel AEC reference, the VoIP Country and Logic settings and allow USB settings to be specified.
Selecting the 'Create USB Blocks' option will open the USB Initialization dialog.

Default Blocks

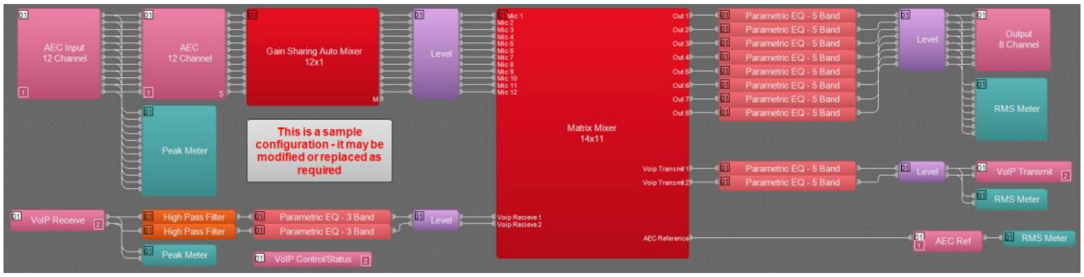


The TesiraFORTÉ VI will add the following default component objects:

- 1x AEC Input 12 Channel
- 1x AEC 12 Channel
- 1x AEC Reference
- 1x Output 8 Channel
- 1x VoIP Receive 2 channel
- 1x VoIP Transmit 2 channel

Optional:
Adding USB I/O Blocks - depending on the initialization settings the corresponding Input and Output block will be added. See the USB
section for more details
Default Configuration

Each TesiraFORTÉ device comes pre-configured from the factory with a default system design. If you wish to load your own custom designed configuration you will need to reset/initialize the unit(s) first.



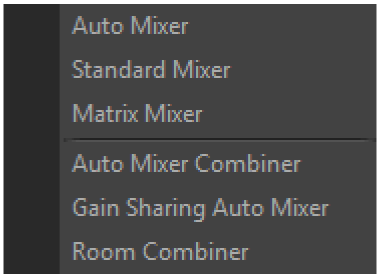
Mixers

Mixers

These Component Objects provide audio mixing functions, in five categories: Gating Auto Mixers, Gain Sharing Auto Mixers, Standard Mixers; Matrix Mixers; Auto Mixer Combiners; and Room Combiners. Auto Mixer Combiners are provided to enhance the capabilities of Auto Mixers in mix-minus and input expansion applications. Room Combiners are provided to enhance the capabilities of Auto Mixers in room combining and zone routing applications. The configuration of the various mixer blocks may be customized when created from the Object Bar

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a conventional user interface.

- Auto Mixer
- Standard Mixer
- Matrix Mixer
- Auto Mixer Combiner
- Gain Sharing Auto Mixer
- Room Combiner



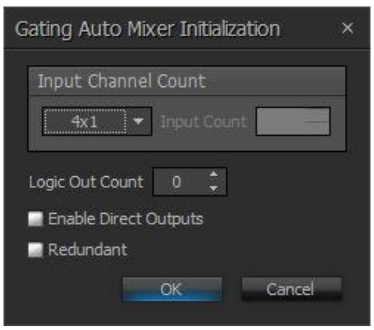
Gating Auto Mixer

The Gating Auto Mixer is useful in speech reinforcement applications that require a number of always-on microphones. The Gating Auto Mixer analyzes all microphone inputs, and attenuates or gates off any microphones that are not passing speech audio. Microphones that are passing speech audio are heard normally.

The Gating Auto mixer accomplishes this by setting a dynamic threshold just above the background noise floor. Microphones that are below this threshold are gated off. If a qualifying speech signal exceeds the threshold, the channel gates on and the signal passes. If the background noise floor changes, the threshold updates in real time.

The Gating Auto Mixer keeps track of the number of open microphones (NOM) and can limit the number that can be active at one time. It can also attenuate the output of the mixer 6 dB for every doubling of NOM.

When the Gating Auto Mixer is selected from the Object Toolbar, an initialization dialog is produced.



Input Channel Count specifies the number of channels in the block. If Custom is selected, any input count from 2 to 256 channels can be specified.

Logic Out Count determines how many logic output connections will appear on the bottom of the Gating Auto Mixer block. If a number *n* less than the number of channels in the block is specified, the logic outputs will correspond to channels 1 through *n*.

Enable Direct Outputs determines whether the block will have direct outputs for each channel, along with the always present Mix output. The signal appearing at the direct outputs can be pre- or post-NOM attenuation. (See Mic Options.)

Redundant determines if the block is used in a system which is designated as redundant. See Redundancy for more details.

Control Dialog

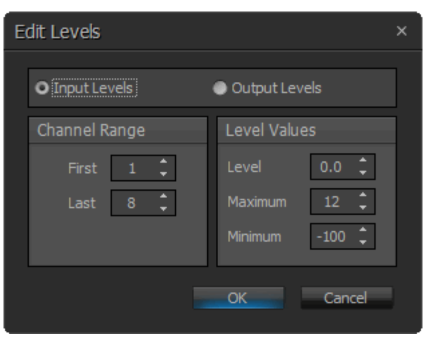
Double clicking on the Gating Auto Mixer block produces a control dialog window.

Mute In turns the input signal on/off, pre-gate detection.
Level In adjusts the channel volume, post-gate detection.
Bus determines whether that channel is routed to the Mix output bus. The signal will appear at the direct output (if direct outputs are enabled) even if the signal is unassigned from the Mix bus.
Mute Out turns the Mix output signal on/off.
Level Out adjusts the Mix output volume.

Mixers allow labeling of inputs/outputs within their dialog boxes. See Label in the DSP Properties tab in the Property Sheet.

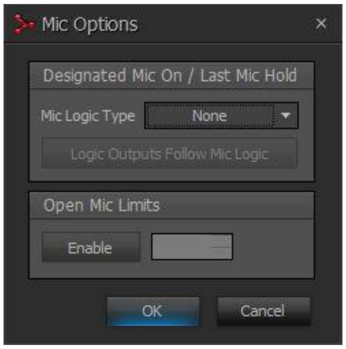
**View Grid Bird's Eye**: Clicking this button opens the mixer's bird's eye view window. This is useful in very large mixer objects as a means of navigating around the available mixer crosspoints. Light blue indicates a mixer crosspoint that is enabled, and dark blue indicates off.

**Edit Levels**: This function provides a way to set a range of input or output level attributes. Clicking the button produces a control dialog window.



First, select whether you wish to set **Input Levels** or **Output Levels**. Under **Channel Range**, specify the **First** channel and **Last** channel number you wish to modify. Under **Level Values**, specify the desired **Level**, **Maximum** and **Minimum** values. Clicking OK will set the values.

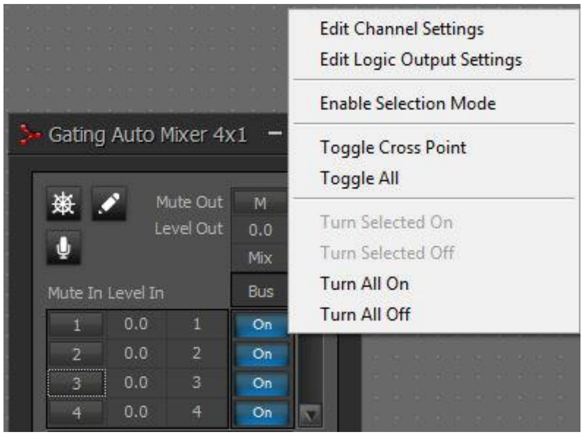
**Mic Options**: Clicking this button opens a control dialog window that shows global settings.



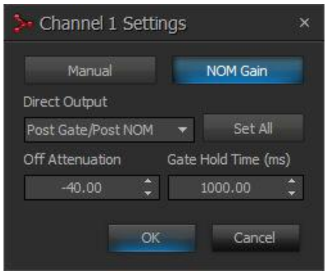
Designated Mic On / Last Mic Hold determines which microphone (if any) will stay/become active when no signal is present. **Logic Outputs Follow Mic Logic** assigns logic outputs to follow Designated Mic On / Last Mic Hold. **Open Mic Limits** enables (and designates) a maximum allowable number of active microphones.

Context Menu

In the main dialog window, right-clicking on Mix Bus crosspoints allows selection of two additional dialog boxes.



Edit Channel Settings



Channel Settings affects individual channel settings, but may be applied to all channels.
Manual turns on/off channel gating.
NOM Gain determines whether that channel is affected by NOM attenuation.

Direct Output designates channel direct output signal as Post Gate / Pre NOM, or Post Gate / Post NOM. (Direct Outputs must be enabled when the Gating Auto Mixer block is created from the Object Toolbar.)
Set All causes current Channel Settings to be applied to all channels.
Off Attenuation determines the amount of attenuation applied when channel is inactive.
Gate Hold Time determines length of time before channel becomes inactive, once signal is no longer present.

Edit Logic Settings

Logic Outputs affects individual Logic Output settings, but may be applied to all Logic Outputs. (Logic Outputs must be designated when the Gating Auto Mixer block is created from the Object Toolbar).
Logic Output selects the condition of the Logic Output. The following can be selected:

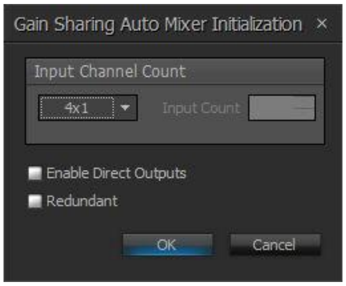
- Follow Gate
- On
- Off

Invert reverses normal operation of the Logic Output (off when channel active).
Set All causes current Logic Output settings to be applied to all channels.

Gain Sharing Auto Mixer

The Gain Sharing Auto Mixer is an automatic microphone mixing process whereby the total gain of the system remains constant. Each individual input channel is attenuated by an amount, in dB, equal to the difference, in dB, between that channel's level and the sum of all channel levels.

When the Gain Sharing Auto Mixer is selected from the Object Toolbar, an initialization dialog is produced.



Input Channel Count specifies the number of channels in the block. If Custom is selected, any input count from 2 to 256 channels can be specified.

Enable Direct Outputs determines whether the block will have direct outputs for each channel, along with the always present Mix output.

Redundant determines if the block is used in a system which is designated as redundant. See Redundancy for more details.

Control Dialog

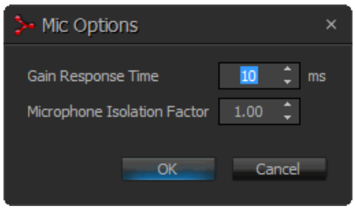
Double clicking on the Gain Sharing Auto Mixer block produces a control dialog window.

Mute In turns the input signal on/off. **Mute Out** turns the Mix output signal on/off.

**View Grid Bird's Eye**: Clicking this button opens the mixer's bird's eye view window. This is useful in very large mixer objects as a means of navigating around the

available mixer crosspoints. Light blue indicates a mixer crosspoint that is enabled, and dark blue indicates off.

**Mic Options**: Clicking this button opens a control dialog window that shows global settings.

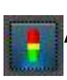


Gain Response Time is the amount of time (in milliseconds) it will take to apply a new gain value to a microphone channel.

Microphone Isolation Factor (0 to 2.00, default = 1.00) is used to tailor the gain distribution in systems with a large number of microphones, where the cumulative noise floor may make it difficult for a talker to receive enough gain. In this situation, increasing this setting from the default value of 1.00 will compress the detection level of channels below a signal level threshold, which will make it easier for a qualifying speech signal to receive full gain. Values less than 1.00 will lessen the effect of this detection level compression, which may be appropriate for systems with a small number of microphones and low cumulative noise floor.

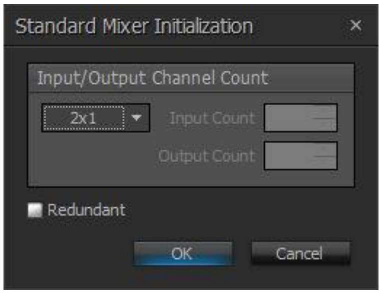
**Channel Levels**: If direct outputs are enabled, this button will be visible. Clicking it produces a control window.

Mute turns the corresponding direct output on/off and removes the channel from the mix bus. **Level** controls the output volume of the corresponding direct output and the contribution of that channel to the mix bus.

**Applied Gain Meters**: Shows the amount of Gain being applied to each channel.

Standard Mixer

The **Standard Mixer** object can be used to create various mixes of the input signals to the outputs. When the block is created from the Object Toolbar and initialization dialog is produced. **Channel Count** specifies the number of input and output channels in the block. If Custom is selected, any input count from 2 to 256 channels, and any output count from 1 to 256 channels can be specified.



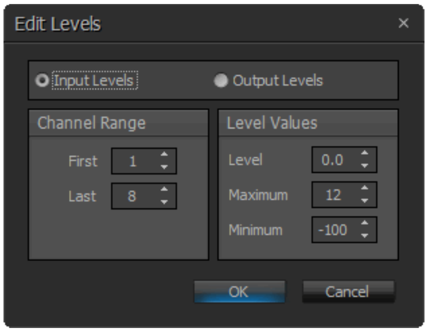
Redundant determines if the block is used in a system which is designated as redundant. See Redundancy for more details.

Control Dialog

Mute In turns the input signal on/off.
Level In adjusts the relative input volume.
Bus assigns inputs to specific outputs.
Mute Out turns the output signal on/off.
Level Out adjusts the relative output volume.

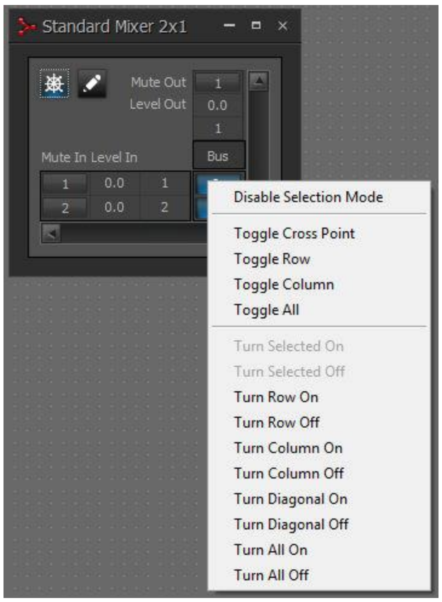
**View Grid Birds Eye**: Clicking this button opens the mixer's bird's eye view window. This is useful in very large mixer objects as a means of navigating around the available mixer crosspoints. Light blue indicates a mixer crosspoint that is enabled, and dark blue indicates off.

**Edit Multiple Levels:**This function provides a way to set a range of input or output level attributes. Clicking the button produces a control dialog window.



First, select whether you wish to set **Input Levels** or **Output Levels**. Under **Channel Range**, specify the **First** channel and **Last** channel number you wish to modify. Under **Level Values**, specify the desired **Level**, **Maximum** and **Minimum** values. Clicking OK will set the values.

Right Clicking on a bus assign crosspoint will show a selection menu that allows multiple crosspoint selection.



Matrix Mixer

- **Mute In** turns the input signal on/off.
- **Level In** adjusts the relative input volume.
- **Bus** assigns inputs to specific outputs, and right-clicking allows level adjustment.
- **Mute Out** turns the output signal on/off. Level Out adjusts the relative output volume.

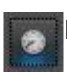
Matrix Mixers with Delay are also available. Mix Table assigns Bus to affect either level or delay settings.

Right-clicking over certain settings will provide a menu of additional options.

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

Edit Multiple Levels : This is used as a way to affect a group of crosspoints for larger system settings.

Level Assign (Visible on Matrix With Delay) - Enables Crosspoint Level Control.

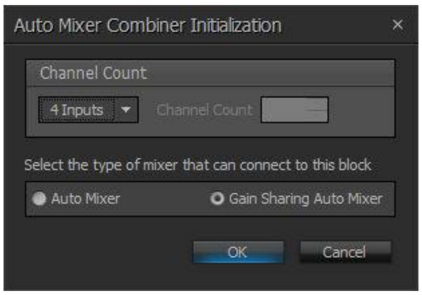
Delay Assign (Visible on Matrix With Delay) - Enables Crosspoint Delay Control.

Right Clicking on the Crosspoints will also show a selection menu that allows multiple crosspoint selection and the Editing of Level or Delay settings.

Auto Mixer Combiner

Auto Mixer Combiner blocks allow multiple, separate auto mixer blocks to share data about the level and number of open microphones, in order to enhance the capabilities of auto mixers in room combining, mix-minus, and input expansion applications. Auto Mixer Combiners combine control data only and, therefore, have no audio outputs. Inputs to an Auto Mixer Combiner come from the Mix outputs of separate Auto Mixer blocks.

When the Auto Mixer Combiner is selected from the Object Toolbar, an initialization dialog is produced.



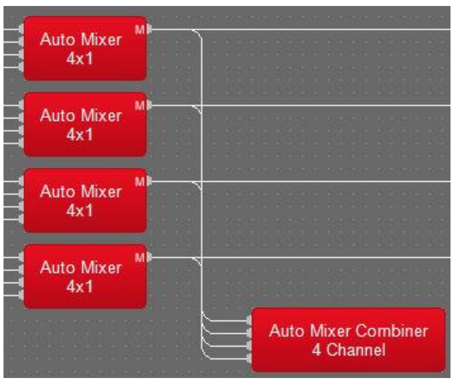
Channel Count specifies the number of channels in the block. If Custom is selected, any input count from 2 to 32 channels can be specified. **Gating Auto Mixer** or **Gain Sharing Auto Mixer** selections specify which type of Auto Mixer can be connected to the Auto Mixer Combiner block. Outputs of different types of Auto Mixer blocks cannot be combined.

Double clicking on the Auto Mixer Combiner block produces a control dialog window.

View Grid Bird's Eye opens a separate window that shows a condensed view of the crosspoint grid. It is useful when large mixers are required as a means of navigating around the available crosspoints. Light blue indicates an active crosspoint and dark blue indicates a crosspoint that is off.

Combined shows an *n* x *n* grid equal to the number of input channels. This grid allows input channels (1, 2, 3, etc.) to be assigned into specific combine groups (A, B, C, etc.). These combine groups determine proper routing of control data for the Auto Mixer blocks. If an input channel is the only input assigned to a particular combine group, it will act independently. If more than one input is assigned to the same combine group, the control data of the auto mixers connected to those inputs will be combined.

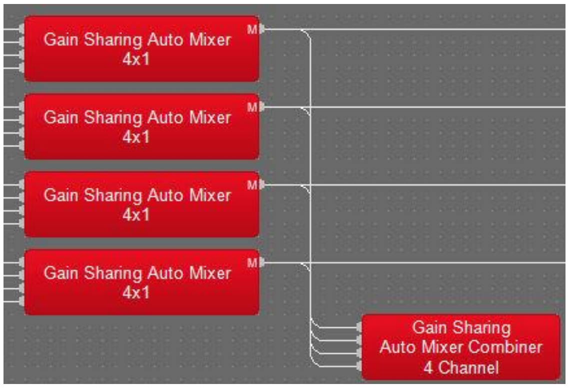
Auto Mixer Combiner



For **Gating Auto Mixers**, control data consists of NOM (number of open mics) count, ATS (adaptive threshold sensing), and last mic hold status information. Settings at the bottom of the control dialog window control aspects of this data.

The **Groups** drop down menu is used to choose a group for viewing and editing two settings: Last Mic Hold and Open Mic Limits. **Last Mic Hold** determines whether the last open microphone across all Gating Auto Mixer inputs assigned to the current group is allowed to gate off when activity on that channel ceases. **Open Mic Limits** enables (and designates) a maximum allowable number of active microphones across all Gating Auto Mixer inputs assigned to the current group.

Gain Sharing Auto Mixer Combiner



For **Gain Sharing Auto Mixers**, only the Combined grid is shown.

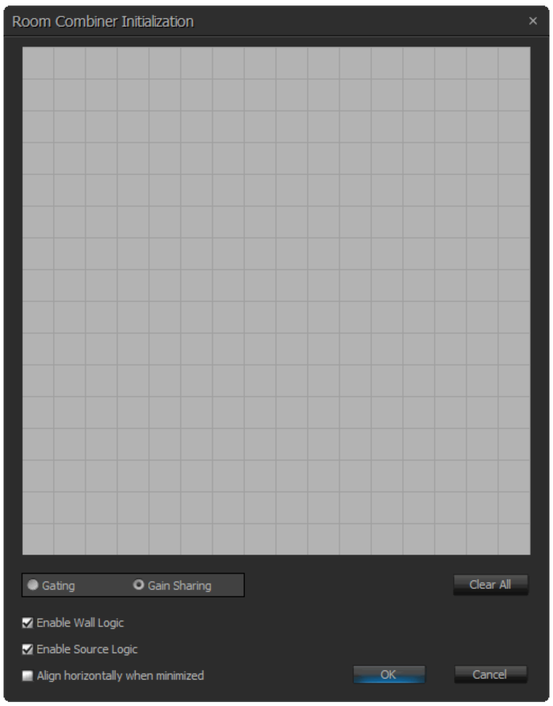
In either Gating or Gain Sharing mode, right clicking on a grid assignment button will display a menu of crosspoint enable options.

Auto Mixer Combiner blocks enhance the capabilities of Auto Mixers in room combining, mix-minus, and input expansion applications. Auto Mixer Combiners combine control data only and, therefore, have no audio outputs. Inputs to an Auto Mixer Combiner come from the Mix outputs of separate Auto Mixer blocks.

Room Combiner

The Room Combiner block provides a way of managing the signal routing and control of combinable/divisible spaces. It can support a maximum of 32 rooms in numerous configurations, with combinable levels, mutes and source tracking. Logic inputs and outputs are provided for wall state and source selection, as well as the ability to combine and control the function of auto mixers connected to the Room Combiner block's inputs.

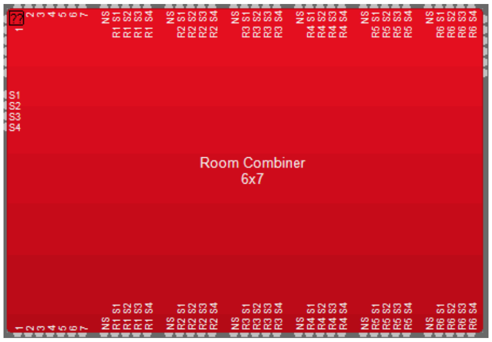
When a Room Combiner object is created from the Object Toolbar, an initialization dialog window is produced.



A 16 row by 16 column grid of blocks is used to lay out graphically the relative position of each room, so rooms that are combinable via a removable wall will share at least one border. Up to 32 unique combinable spaces can be defined. Enabled blocks in the grid that are adjacent to each other can have three types of walls: permanent, removable and none. The wall type is selected by repeatedly clicking on the border between adjacent enabled blocks. A removable wall is indicated by a thin dashed line, and the wall will have a corresponding logic node on the block. A permanent wall is indicated by a thick gray line and represents a non-removable border between those rooms. It will have no logic connection on the block. No border between adjacent enabled blocks indicates there is no wall, and the blocks will be considered part of the same space. The **Clear All** button removes all blocks from the grid.

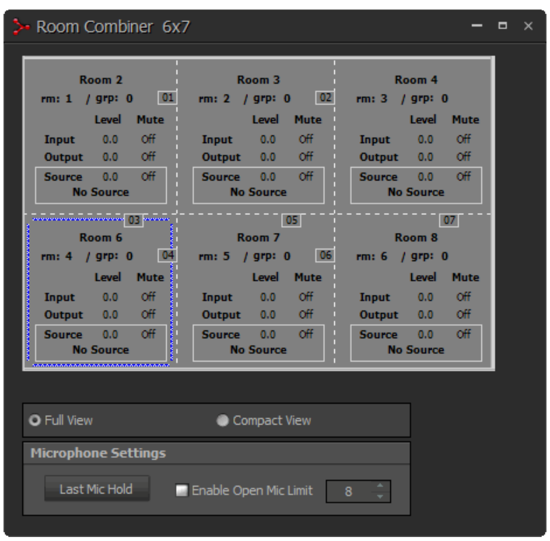
The Room Combiner block can combine the function of auto mixers if their mix outputs are connected directly to the channel input nodes on the block. Select which type of auto mixer will be used by choosing Gating or Gain Sharing.

Enable Wall Logic controls whether the Room Combiner block will have logic connections for wall state. **Enable Source Logic** determines whether logic connections for source selection will appear on the block. **Align horizontally when minimized** controls whether the room information block will appear below (unchecked) or beside (checked) the room layout view when the Room Combiner control dialog is minimized.

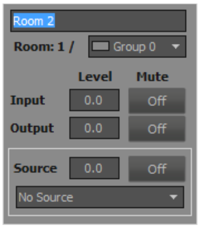


The Room Combiner block has one audio input and output per room. There are also four audio source inputs, which can be assigned to any room. If Enable Wall Logic and Enable Source Logic are checked when the block is created, the block will have a number of logic nodes on the top and bottom of the block. Numbered nodes correspond to the various removable walls in the block. A logic low signal presented to one of these nodes causes the corresponding wall to be removed and the two adjacent spaces to be combined. A logic high replaces the wall and uncombines the two spaces. Source selection logic inputs and outputs may also be shown. Four selections are provided per room, plus an Off node. When a source selection logic node receives a low-to-high transition (rising edge) that room's source will switch to the corresponding selection.

Double clicking on the Room Combiner block produces a control dialog window.



A graphical depiction of the rooms laid out in the initialization dialog is shown, with information about each room. Double clicking on a room opens its room edit dialog window.



The **Room Name** field can be used to specify a custom name for each room. The line of text below this shows the room number as assigned by the software. **Room Group** controls which combine group the room is assigned to. Rooms in the same combine group will output the audio stream of the combine group, which is a mix of the inputs of all rooms assigned to that group, and their output level and mute and source selection, level and mute will be linked. If a room is in combine group 0, it will be uncombined and operate independently. **Input Level** controls the volume of the input to the Room Combiner block. **Input Mute** turns the input signal on/off.

The following settings are linked with any other rooms in the same combine group. **Output Level** controls the volume of the channel output. **Output Mute** turns the output signal on/off. **Source Level** sets the volume of the selected source to that room. Source Mute turns the source input to the room off/on. **Source** selects which of the four source inputs is routed to that room. No source can be chosen as well.

When combining two adjacent rooms using the software control dialog, hovering on either side of a partition causes the mouse pointer to become a left or right pointing arrow, which will determine which room's source selection will become the common source of the combined space when the partition is removed. When combining rooms via logic or the text protocol (TTP), by default, the lower numbered room's source is used as the common source, although this behavior can be changed using the preferredRoom attribute in the text protocol. (See TTP)

When two rooms are combined, if either room's Output Mute or Source Mute is enabled, the corresponding controls in the resulting combined space will also be muted. Also, if Output Level or Source Level controls differ, the lower of the two levels will be used in the combined space.

Use the view selection buttons at the bottom of the Room Combiner control dialog to display the **Full View** or the **Compact View**, which shows a room information field at the bottom to enable a smaller display.

Equalizers

Equalizer Components

These Component Objects provide both graphic and parametric equalization, as well as feedback suppression. Equalizers may be connected between any components within the Layout, for applications which require room equalization, tone adjustment, or feedback control. The configuration of Equalizer objects may be customized when placed from the Object Toolbar.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking on the object. This produces a Control Dialog Box, which displays the component controls in a conventional user interface.

- Parametric Equalizer
- Graphic Equalizer
- Feedback Suppressor

Right-clicking over the object provides a pop-up menu of options.

Parametric Equalizer

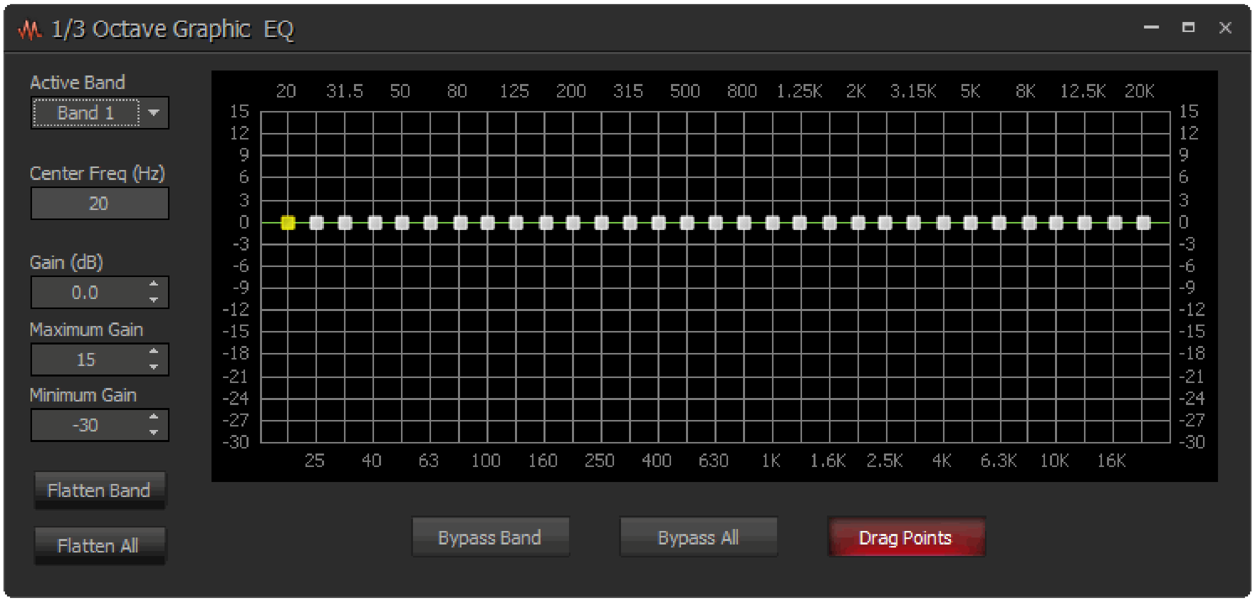
When the Parametric Equalizer component is created from the Object Toolbar, Up to 16 bands can be specified.

Active Band selects the current band to be adjusted. **Center Freq (Hz)** controls the center frequency for the current band. **Gain (dB)** adjusts the amount of cut or boost applied at the center frequency for the current band. **Maximum Gain** specifies the maximum possible setting of the Gain parameter for the current band. Values between 0 and 15dB can be set. **Minimum Gain** specifies the minimum possible setting of the Gain parameter for the current band. Values between -30 and 0dB can be set. **Bandwidth (oct)** controls the range of frequencies, above and below the center frequency, which are affected by the current band. Values between 0.01 and 4 octaves in hundredths of an octave can be set.

The settings may also be adjusted by using **Drag Points**. This allows the setting of frequency band controls shown inside the graph. Dragging the larger center dot affects both **Center Freq** & **Gain**. Dragging either smaller outer dot affects **Bandwidth**. **Flatten Band** and **Flatten All** change the gain of the current band or all bands to 0dB (flat). **Bypass Band** and **Bypass All** disable the band(s) without changing settings. **Drag Points** turns on/off the band controls, revealing the resultant curve only. **Band** highlights the current band inside the graph.

Graphic Equalizer

When the Graphic Equalizer component is created from the Object Toolbar, the type can be specified, 1/3 octave, 2/3 octave or 1 octave.

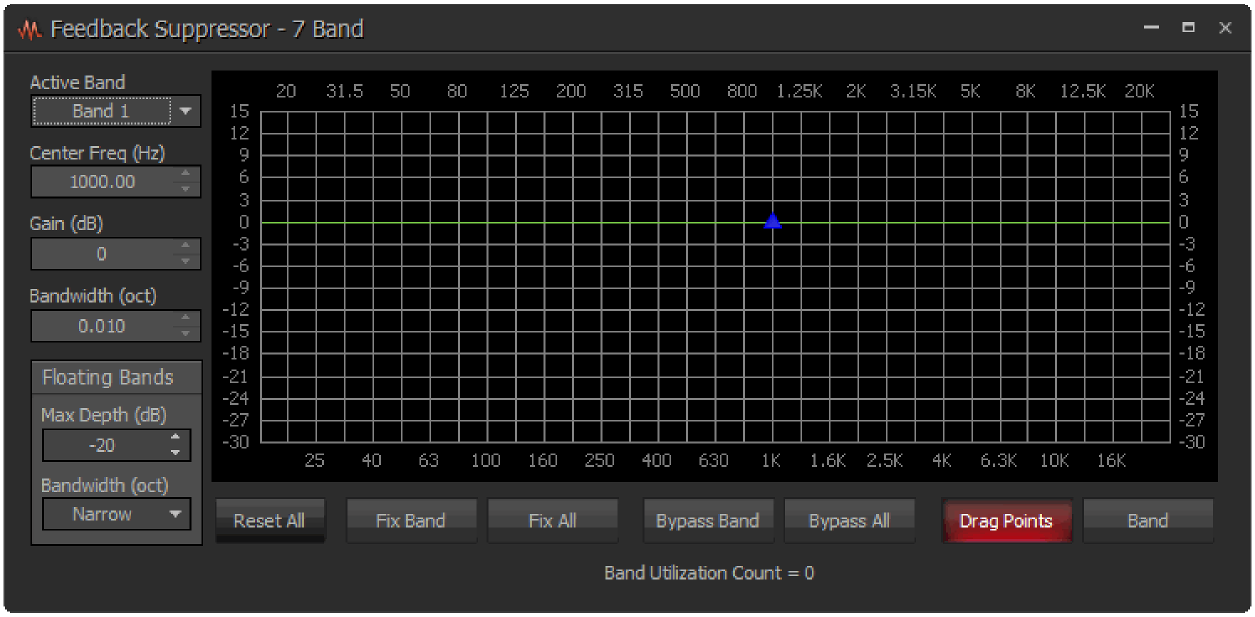


Active Band selects the band to be adjusted. **Center Freq (Hz)** displays the center frequency for the current band. **Gain (dB)** adjusts the amount of cut or boost applied at the center frequency for the current band. **Maximum Gain** specifies the maximum possible setting of the Gain parameter for the current band. Values between 0 and 15dB can be set. **Minimum Gain** specifies the minimum possible setting of the Gain parameter for the current band. Values between -30 and 0dB can be set.

Active Band and Gain may also be adjusted by dragging the band controls shown inside the graph. The selected band control becomes yellow, and dragging it up/down affects Gain for that band. **Flatten Band** and **Flatten All** change the gain of the current band or all bands to 0dB (flat). **Bypass Band** and **Bypass All** disable the band(s) without changing settings. **Drag Points** turns on/off the band controls, revealing the resultant curve only.

Feedback Suppressor

When the Feedback Suppressor component is created from the Object Toolbar, the number of bands can be specified, up to 16.



Feedback Suppressors behave like an automatic, cut-only Parametric Equalizer. They utilize 'floating' bands of equalization which detect and remove feedback frequencies.

Active Band selects the current band for which settings will be displayed. The following three parameters are read only when the band is set to floating, and are editable when the band is set to fixed. **Center Freq (Hz)** displays the center frequency for the current band. **Gain (dB)** displays the amount of cut applied at the center frequency for the current band. **Bandwidth (oct)** displays the range of frequencies, above & below the center frequency, which are affected by the current band. For the Floating Bands, **Max Depth** restricts all floating bands to a selected maximum depth (cut) and **Bandwidth** (Narrow = 1/40-octave; Wide = 1/10-octave).

Reset All temporarily returns the gain of all floating bands to 0dB (flat). **Fix Band** and **Fix All** allow the band(s) to become manually adjustable (non-floating). **Bypass Band** and **Bypass All** disable the band(s) without changing settings. **Drag Points** turns on/off the band controls, revealing the resultant curve only. **Band** highlights the current band inside the graph. **Band Utilization Count** indicates the number of floating bands currently being employed.

NOTE: *Feedback Suppressors are fairly intensive in their use of DSP resources. They are limited to a maximum of sixteen bands, however, in most applications the number of actual bands used should be significantly less. Fixed bands in a Feedback Suppressor may be copied to a Parametric Equalizer. When applicable, this may be a more DSP efficient choice.*

Filters

Filter Components

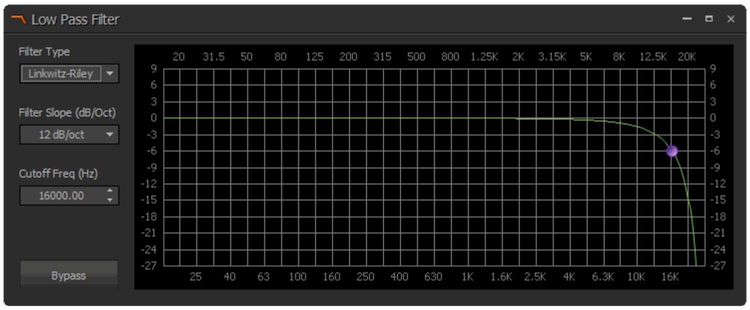
These Component Objects provide High-Pass, Low-Pass, High-Shelf, Low-Shelf, and All-Pass filters. Filters may be connected between any components within the Layout, for applications which require 'roll-offs', simple tone controls, or even phase compensation.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking on the object. This produces a control dialog box, which displays the available component controls.

- Pass Filter
- Shelf Filter
- All Pass Filter
- Uber Filter
- FIR Filter

Pass Filter

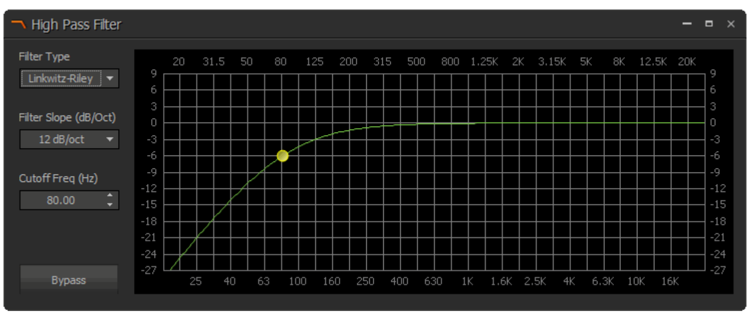
The **Pass Filter** object is for High Pass and Low Pass filter types. When this object is created from the Object Toolbar, a control dialog is presented. Under **Filter Type**, choose whether Low Pass or High Pass is desired.



In the Low Pass filter control dialog, **Filter Type** selects from Butterworth, Linkwitz-Riley or Bessel.

Filter Slope (dB/Oct) determines how quickly the filter's amplitude response rolls off above the cutoff frequency.

Cutoff Freq (Hz) specifies the point at which the filter's amplitude response begins to roll off. For the Butterworth and Bessel types, the cutoff frequency represents the -3dB point in the rolloff. For the Linkwitz-Riley type, the cutoff frequency represents the -6dB point.



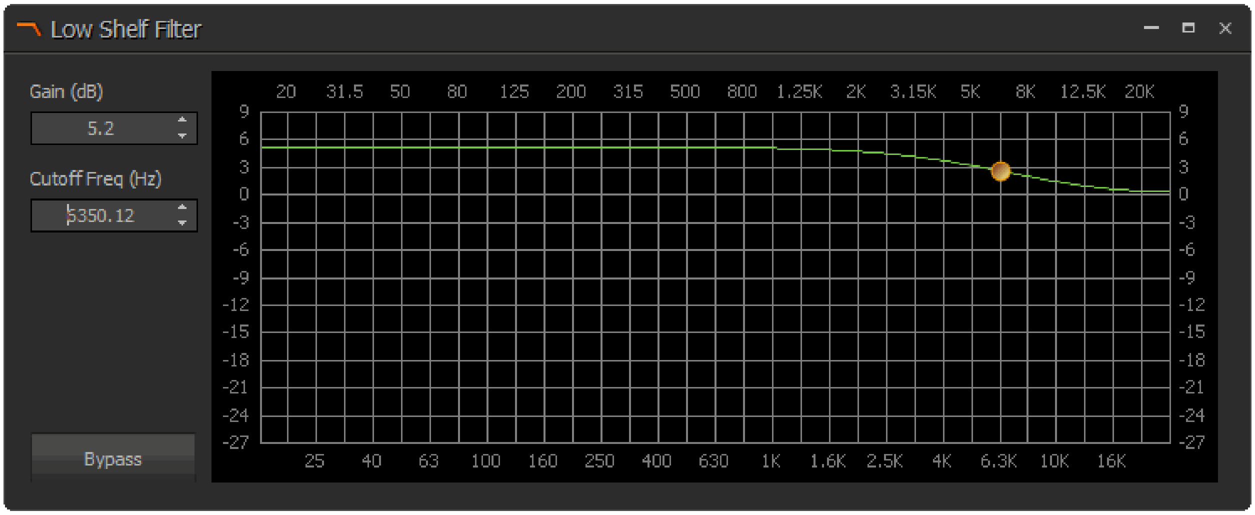
In the High Pass filter control dialog, **Filter Type** selects from Butterworth, Linkwitz-Riley or Bessel.

Filter Slope (dB/Oct) determines how quickly the filter's amplitude response rolls off below the cutoff frequency.

Cutoff Freq (Hz) specifies the point at which the filter's amplitude response begins to roll off. For the Butterworth and Bessel types, the cutoff frequency represents the -3dB point in the rolloff. For the Linkwitz-Riley type, the cutoff frequency represents the -6dB point.

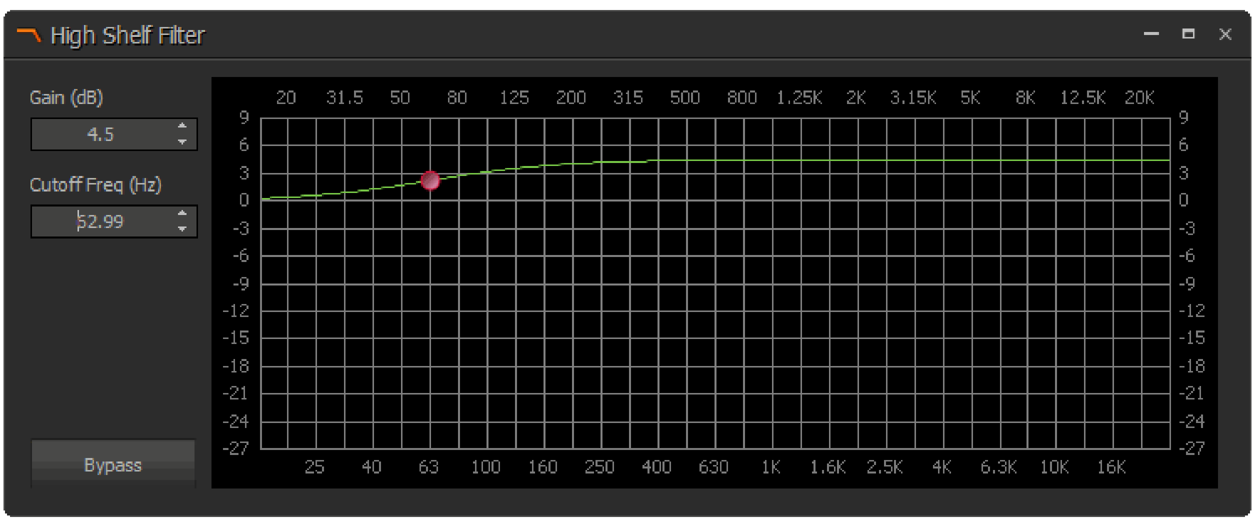
Shelf Filter

The **Shelf Filter** object is for High Shelf and Low Shelf filter types. When this object is created from the Object Toolbar, a control dialog is presented. Under **Filter Type**, choose whether Low Shelf or High Shelf is desired.



In the Low Shelf filter control dialog, **Gain (dB)** controls the amount of cut or boost below the cutoff frequency.

Cutoff Freq (Hz) specifies the point below which the filter's amplitude response begins to transition toward the value specified by the Gain setting.

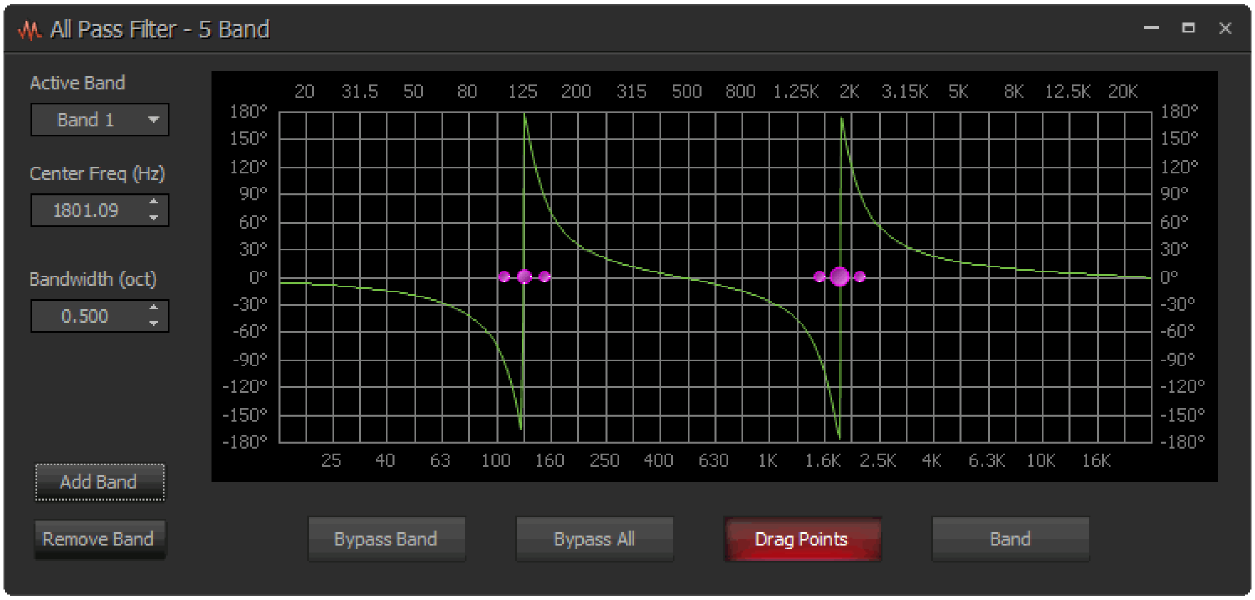


In the High Shelf filter control dialog, **Gain (dB)** controls the amount of cut or boost above the cutoff frequency.

Cutoff Freq (Hz) specifies the point above which the filter's amplitude response begins to transition toward the value specified by the Gain setting.

All Pass Filter

All Pass Filters are so named because they do not affect the amplitude response of the signal passing through it. Instead, they affect phase response only and, therefore, can be used to compensate for the phase anomalies caused by equalizers, crossovers and other filters. All Pass Filters are available with up to sixteen bands.



Active Band selects the current band to be adjusted.

Center Freq (Hz) adjusts the center frequency for the current band.

Bandwidth (oct) adjusts the range of frequencies, above and below the center frequency, which are affected by the current band. These settings may also be adjusted by dragging the band controls shown inside the graph. Dragging the larger center dot affects center frequency. Dragging either of the smaller dots affects bandwidth.

Add Band and **Remove Band** work within the designated number of bands for the filter.

Bypass Band and **Bypass All** disable the band(s) without changing settings.

Drag Points turns on/off the band controls, revealing the resultant curve only.

Band highlights the phase response of the current band inside the graph.

The **Transfer Function** object can be used in conjunction with All-Pass Filters to help visualize the effect of this block on the phase response of the signal.

Uber Filter

The Uber Filter object provides a way of having a number of different filter types available in a single block. When the block is created from the object toolbar, the maximum number of filters in the block can be specified.

Add New Filter allows a new filter to be created. High Pass, Low Pass, High Shelf, Low Shelf, and Parametric Equalizer types can be selected. New filters can be created up to the maximum number specified in the initialization dialog when the block is created. A filter can be deleted from the list, either by right clicking on it and selecting Delete from the pop-up menu, or by highlighting it and pressing the Delete key on your keyboard.

ID is the index number given to each filter stage.

Filter Type is used to change the type of filter in that position, which can be done even when the configuration is loaded and running live, unless the Locked box has been checked.

If a filter stage is **Locked**, the DSP resources for that filter type are allocated and the filter type cannot be changed once the configuration is loaded. The locking of a filter type also may allow some DSP saving. For example a Parametric EQ uses less DSP resource than a high or low pass filter.

The Filter parameters can still be modified even if Filter Type is Locked. The rest of the parameters affect the operation of each filter stage. See the Filter Components section for details on the individual filter types available.

Bypass causes that filter stage to by bypassed. **Bypass All** causes all filter stages in the block to be bypassed.

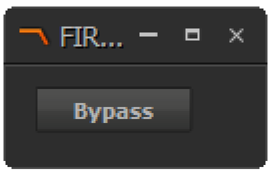
Drag Points turns off/on the ability to change filter parameters by dragging handles on each filter curve in the graph window. When enabled the uber filter has a color coded drag point.

- High Pass - Yellow
- Low Pass - Purple
- High Shelf - Red
- Low Shelf - Orange
- Parametric EQ - Blue
- Composite Curve -Green

Band turns on/off a display showing the curve of the currently selected filter, along with the composite curve.

FIR Filter

The FIR Filter object provides a way of importing a set of FIR filter coefficients and filtering an audio signal through those coefficients. The linear phase nature of the FIR Filter makes it useful for static filter applications such as inverse loudspeaker curves, crossovers, line array steering and the like. When the FIR Filter block is created from the Object Toolbar, an initialization dialog is presented which allows the user to browse to and select a coefficient file. The file can be in 16-bit, 48kHz, mono WAVE (.wav) format or ASCII text in comma separated variable (.csv) format and can contain a minimum of 4 and a maximum of 2048 coefficients. Coefficients should be between -1.0 and +1.0. Filter coefficients should be synthesized assuming a 48kHz sampling rate.



The FIR Filter block has one input and one output. The only control is a Bypass button, which bypasses the filter.

Crossovers

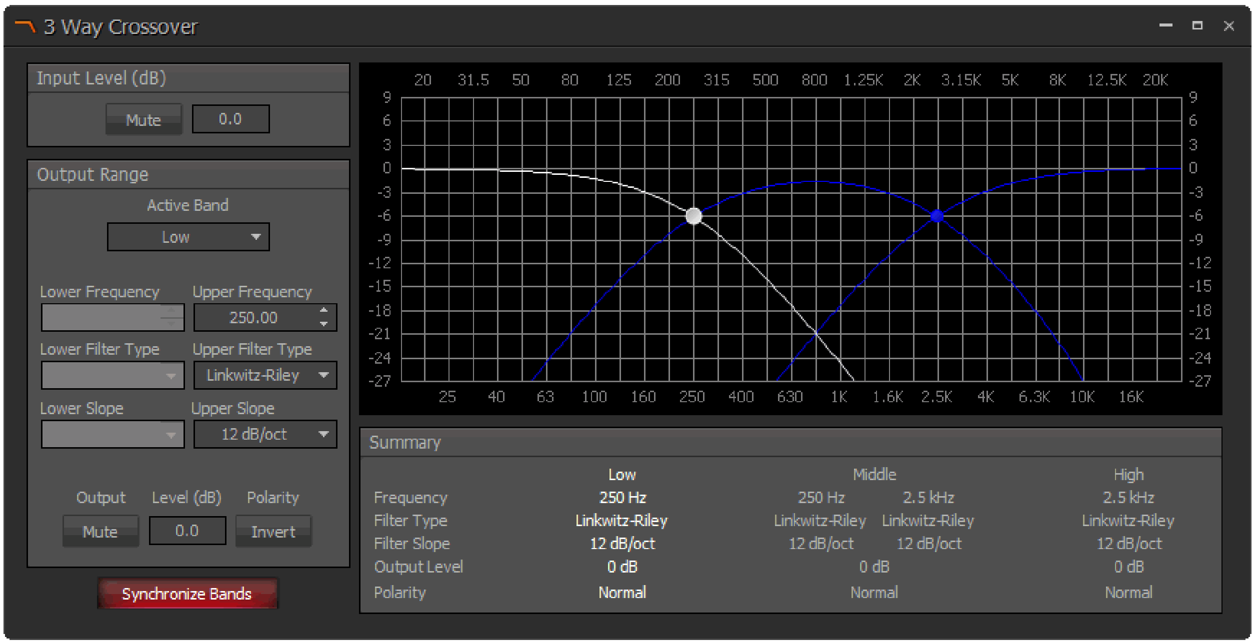
Crossover Components

Crossovers may be connected between any components within the Layout, for applications which require multiple outputs with specified frequency ranges.

- Crossovers

Crossover

This component provides 2-way, 3-way, and 4-way crossover functionality. Crossovers may be connected between any components within the Layout, for applications which require multiple outputs with specified frequency ranges. When the Crossover object is created from the Object Toolbar, an initialization dialog is produced. Under Crossover Type, select the number of crossover outputs needed, 2-way, 3-way or 4-way. Once the Crossover object is placed into the layout, all available settings can be accessed by double-clicking on the object. This produces a control dialog window, which displays the user interface.



Input Level (dB) provides Mute and Level adjustment for the signal input.

Output Range selects the Low, Low Middle, Middle High or High frequency output for editing, depending on how many outputs were chosen when the block was created.

Lower Frequency and **Upper Frequency** specifies the filter cutoff frequency or frequencies for the selected output. Output Range & Frequency may also be selected by dragging the cursors shown inside the graph.

Lower Filter Type and **Upper Filter Type** selects the filter type(s) for the selected output.

Lower Slope and **Upper Slope** selects the filter slope(s) for the selected output.

Output **Mute**, **Level** and **Polarity** are available for the selected output.

Synchronize Bands forces filter adjustments on adjacent outputs to be linked. Settings for each output are displayed across the bottom of the dialog box.

Dynamics

Dynamic Components

These Component Objects provide Leveler, Compressor, Peak Limiter, Ducker, Noise Gate, & Ambient Noise Compensator functions. Dynamics components may be connected between any other components within the Layout, for applications which require automatic control of volume levels and/or dynamics.

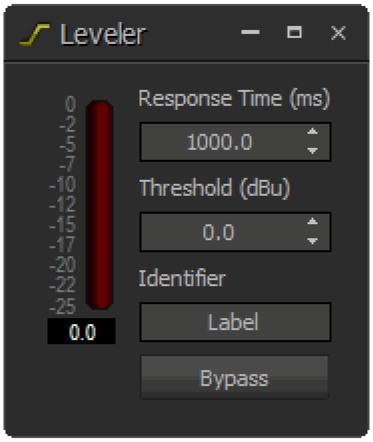
Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface.

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

Right-clicking over the object provides a pop-up menu of options. Control Dialog Boxes for some Dynamics components can be minimized to create user control surfaces.

Leveler

Levelers are a form of automatic gain controls, which affect long-term average levels. The Leveler differs from the AGC in that it only attenuates the signal, it will not add gain. When the Leveler's input signal exceeds the threshold, the leveler will attenuate the signal until it reaches the threshold point.



Response Time determines how quickly the Leveler reacts to input level changes.

Threshold determines what input level will trigger gain reduction. To maintain a consistent level, set Threshold to lowest desired level.

Identifier provides a custom label, when dialog box is minimized (see below).

Bypass disables the Leveler without changing settings.

A meter & numeric display indicate the amount of gain reduction.

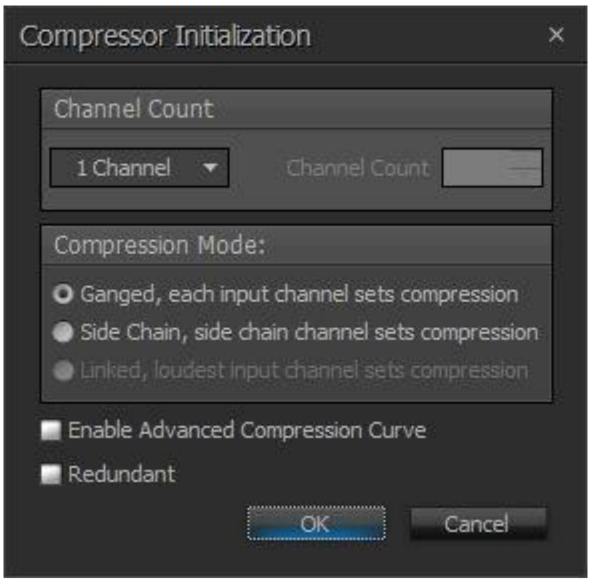
Right-clicking over certain settings will provide a menu of additional options. Control Dialog Boxes for Levelers can be minimized to create user control surfaces (see Customizing Component Objects).

Compressor

The Compressor object is used to reduce the volume of loud signals and/or otherwise reduce the dynamic range of an audio signal. When an input exceeds a programmed threshold level, the compressor will reduce the volume of the signal according to a ratio setting. Ganged, Linked and Side Chain compression modes are available, as well as an advanced mode with multi-knee capability.

Initialization Dialog

When the Compressor object is selected from the Object Toolbar, an initialization dialog is produced.



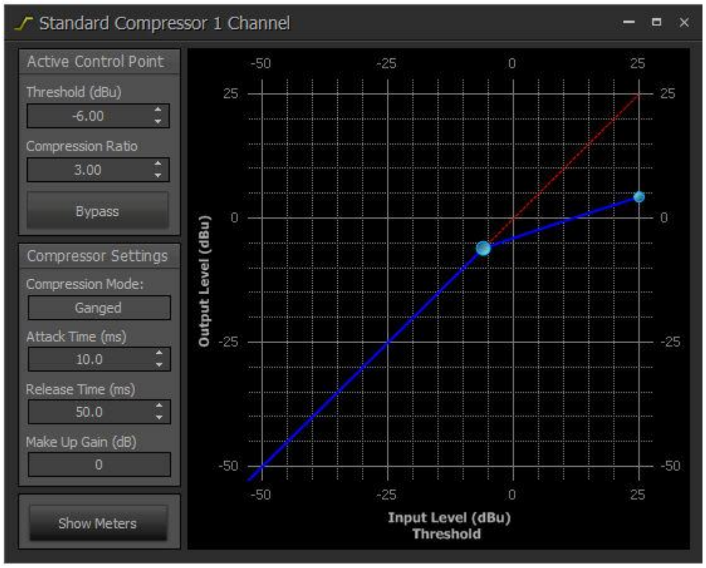
Channel Count sets the number of channels of the Compressor object. A Custom setting allows the number of channels to be set to any number between 1 and 32.

Compression Mode can be set to one of three values.

- In **Ganged** mode, the compressor settings will be applied to all channels independently and there will be no interaction between channels.
- In **Side Chain** mode, the compression on all channels is determined by the level appearing at the SC input on the DSP object (only seen in this mode).
- In **Linked** mode, (available when more than 1 channel is selected) the compression on all channels is determined by the level of the loudest channel.

Selecting **Enable Advanced Compression Curve** will result in an advanced multi-segment compression curve with up to four knees and one endpoint. Leaving this option clear will result in the typical standard compression curve with one knee (suitable for most situations).

Standard Curve



The graph of **Input Level (dBu) / Threshold (dBu)** versus **Output Level (dBu)** shows the response of the Compressor block in **Standard** one-knee form.

When the input signal exceeds the **Threshold (dBu)** setting its output is attenuated according to the **Compression Ratio** setting. A setting of 3.00 means that for every 3dB the input exceeds the threshold, the output only increases 1dB.

Bypass will bypass both the compression curve and the applied make up gain. This allows for easy comparison of the compressed signal with the original.

Compression Mode shows Ganged, Side Chain or Linked, according to the option selected when the block was created.

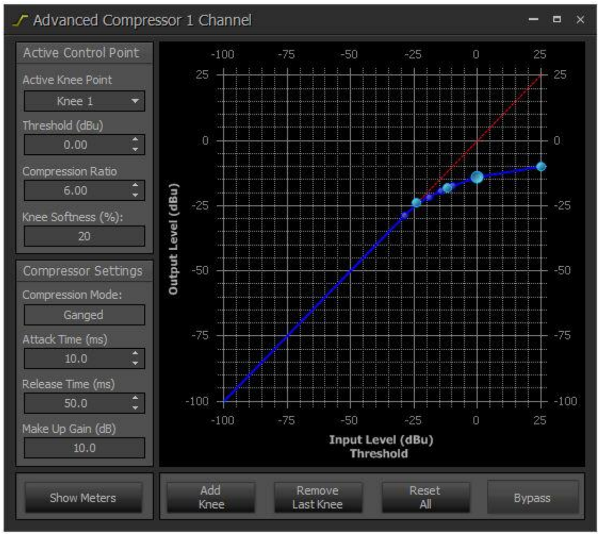
Attack Time (ms) sets the time it takes for the output level attenuation to activate once the input level exceeds the threshold.

Release Time (ms) sets the time it takes for the level attenuation to deactivate once the input level drops below the threshold.

Make Up Gain (dB) is used to restore the nominal operating level of the input signal after it has been affected (attenuated) by the compression curve. A maximum of 12dB of gain can be applied to the outgoing signal to compensate for this attenuation.

Show Meters will display the gain reduction meters for the processing block. In Ganged mode, per channel gain reduction is shown. In Side chain or linked mode one meter is shown. It is useful to look at this meter to determine how much the compressor is affecting the signal for various inputs and to help determine the amount of make up gain needed.

Advanced Curve



In Advanced mode, up to four knees (inflection points on the compression curve) can be defined, with individual Threshold and Compression Ratio settings for each. Active Knee Point selects the current knee point to be controlled.

The circle on the graph representing the active knee appears larger than other knee points.

Active Knee Point selects the current knee point to be controlled. The circle on the graph representing the active knee appears larger than other knee points. The Compressor block allows for a minimum of one end point and a maximum of four knee point. An Endpoint is always located at the far right end of the graph, with an Input Location of 25dBu. Clicking and/or dragging on a knee point will also make it the active knee point.

Threshold (dBu) indicates the input level at which a given knee takes effect. This represents the location of the knee point on the horizontal ("Input Level Threshold") axis of the graph. The thresholds for any two knees and compression ratio between them defines the response of each region of compression.

Compression Ratio determines the extent to which the signal level will be reduced when the input signal level is above the Threshold point. For instance, a compression ratio of 3.00 means that for every increase in input signal level of 3dB, the output signal level will only increase by 1dB. Changing the compression ratio of a knee point will affect the Output level of the next knee point to the right. Correspondingly, changing the Output level of a knee point will affect the compression ratio of the previous knee point to the left.

Knee Softness (%) defines the smoothness of the transition from one compression ratio to another. A radius of zero (also known as a "hard knee") creates an abrupt transition. Larger values (also known as a "soft knee") create a smoother transition, and in some cases can reduce the audible transition from an un-compressed to compressed signal, particularly when the compression ratio is high.

Attack Time (ms) is the time it takes for the compression to activate once the input signal level exceeds the threshold.

Release Time (ms) is the time it takes for the compression to deactivate once the input signal level recedes below the threshold.

Make Up Gain (dB) is used to restore the nominal operating level of the input signal after it has been affected (attenuated) by the compression curve. A maximum of 12dB of gain can be applied to the outgoing signal to compensate for this attenuation.

Show Meters will display the gain reduction meters for the processing block. In Ganged mode, per channel gain reduction is shown. In Side Chain or Linked mode one meter is shown. It is useful to look at this meter to determine how much the compressor is affecting the signal for various inputs and to help determine the amount of make up gain needed.

Add Knee button creates a new knee point to the left of the last knee point.

Remove Last Knee deletes the left-most knee point.

Reset All resets all knee points to a compression ratio of 1. For other settings, refer to the Standard Curve mode above.

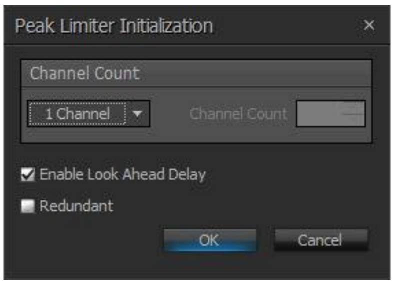
Bypass will bypass both the compression curve and the applied make up gain. This allows for easy comparison of the compressed signal with the original.

When the control dialog for the Compressor object is minimized, the gain reduction meter(s) become visible in a user control surface, which can be positioned in the layout for programmer convenience (see Customizing Component Objects).

Peak Limiter

The Peak Limiter object is used to prevent a signal from exceeding a specified peak level. It is often used in professional sound systems before the system outputs to prevent clipping in the digital-to-analog conversion stage or to prevent potentially damaging sound levels from reaching power amplifiers and loudspeakers. When the input to the Peak Limiter exceeds the threshold, the Peak Limiter will reduce the volume of the signal instantaneously so the output does not exceed the threshold. A 1ms look-ahead feature is available which allows the Peak Limiter to limit the audio more transparently, at the slight expense of some additional latency through the block. When the Peak Limiter is selected from the Object Toolbar, an initialization dialog is produced.

Initialization Dialog

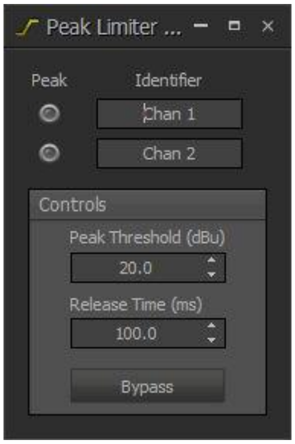


Channel Count sets the number of channels of the Peak Limiter object. A Custom setting allows the number of channels to be set to any number between 1 and 32. Channels are ganged in the sense that all channels share the same settings, but limiting is applied independently and with no interaction between channels.

Selecting **Enable Look Ahead Delay** will enable the look ahead feature for more transparent limiting, at the expense of 1ms of additional processing delay through the object. This additional processing delay will be accounted for in the Delay Equalization phase of compiling. Leaving this option clear will result in no look ahead and the standard processing delay. In this mode, limiting is still instantaneous, but may introduce some distortion, as peaks are being handled as they occur.

Redundant includes the processing block in a redundant configuration.

Control Dialog



A **Peak** indicator illuminates when the input signal is above the programmed threshold and/or when limiting is taking place (related to the Release Time setting).

The **Identifier** is a user configurable name that can be used for channel identification.

Controls

Peak Threshold (dBu) sets the point above which the Peak Limiter will engage and attenuate the signal level so the output does not exceed the threshold.

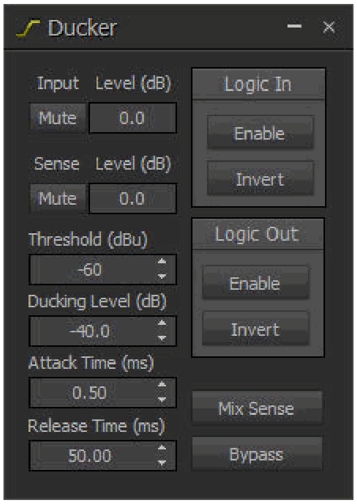
Release Time (ms) sets the time it takes for the level attenuation to deactivate once the input level drops below the threshold. This can be can be set between 1ms and 10,000ms. The attack time is not shown in the control dialog because it is instantaneous.

The **Bypass** button allows for easy comparison of the limited signal with the original.

Ducker

Ducker blocks are used to duck (attenuate) one signal when the level of another signal exceeds a specified threshold. The unlabeled input on the upper left side of the Ducker block is for the input signal (i.e., the signal to be ducked). The input labeled with an S is the sense signal, which is the signal that will trigger ducking to occur when it exceeds the threshold.

Ducking can also be triggered via a logic signal instead of an audio sense signal.



Input Level provides muting and level adjustment for the input signal, which appears as the upper input on the left side of the Ducker.

Sense Level provides muting and level adjustment for the sense signal, which appears as the lower input on the left side of the Ducker, labeled with an S.

Threshold determines what sense input signal level will trigger ducking to occur.

Ducking Level determines how much attenuation is applied to the input signal when ducking is active.

Attack Time determines how quickly the ducker attenuates the input signal when ducking is activated.

Release Time determines how quickly attenuation is released when ducking is deactivated.

Logic In allows ducking to be triggered by a logic signal, connected to the logic input on the top of the Ducker block. Normally, a logic HIGH signal will trigger ducking to occur, and a logic LOW will deactivate ducking. When the Invert button is selected, a logic LOW will trigger ducking to occur.

Logic Out activates the logic output on the bottom of the Ducker block. Normally, when ducking is activated the logic output will generate a logic HIGH signal. When the Invert button is selected, the logic output will generate a logic LOW signal when ducking is activated.

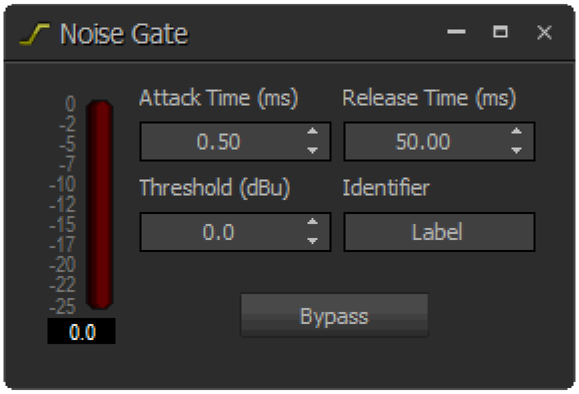
Mix Sense causes the sense signal and the input signal to be mixed together and be sent to the output. When Mix Sense is not enabled, only the input signal is sent to the output and the sense signal does not pass through the block.

Bypass disables the Ducker without changing settings.

Right-clicking over certain settings will provide a menu of additional options.

Noise Gate

Noise Gates are used to mute a signal whenever the signal level drops below a specified threshold.



Attack Time determines how quickly the gate opens when signal is present.

Release Time determines how quickly the gate closes when signal is no longer present.

Threshold determines what input signal level will trigger the gate to open.

Identifier provides a custom label, when dialog box is minimized.

Bypass disables the Noise Gate without changing settings.

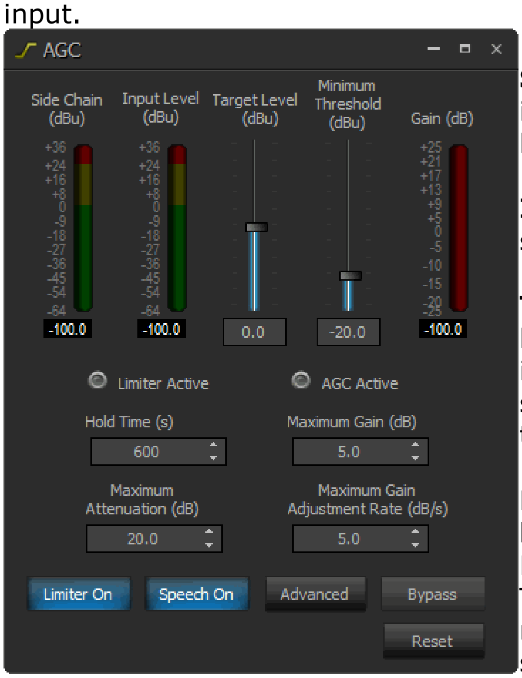
A meter & numeric display indicate the current amount of gain reduction being applied.

Right-clicking over certain settings will provide a menu of additional options. Control Dialog Boxes for Noise Gates can be minimized to create user control surfaces (see Customizing Component Objects).

AGC

The Automatic Gain Control (AGC) block is a dynamics processing block which regulates the level of an audio signal. Its function is to maintain a relatively constant output level, and it will add or subtract gain in order to bring the level of the input signal closer to the Target Level. The AGC block only adjusts gain when it receives a qualifying input signal (see "Qualifying Input Signals" below).

The AGC block can optionally utilize SpeechSense™ technology to make better decisions about when to make level adjustments. It can also be optionally triggered from a side chain input.



Side Chain meter shows the level of the side chain input. It is only visible when the Side Chain input has been enabled.

Input Level meter shows the level of the input signal.

Target Level defines the signal level that the AGC block will constantly strive to output. If the input level is higher than the target level, the AGC block will subtract gain. If the input level is lower than the target level, the AGC block will add gain.

Minimum Threshold is the minimum input signal level required for the AGC to make adjustments. The Maximum value for the Minimum Threshold will be the Target Level. If the input signal level is lower than the minimum threshold, the AGC block will temporarily suspend gain adjustments.

Gain shows how much gain is currently being added or subtracted from the input signal.

Limiter Active will light when the clip limiter is actively engaged in preventing clipping.

AGC Active will light when the AGC block is making a gain adjustment.

Hold Time is the number of seconds that the AGC block will hold the current Gain setting while not receiving a qualifying input signal. After the Hold Time elapses, the AGC block will reset the gain to zero.

Maximum Gain defines the maximum amount of gain that the AGC block will add to the signal.

Maximum Attenuation defines the maximum amount of gain that the AGC block will subtract from the signal.

Maximum Gain Adjustment Rate defines how quickly the AGC block can adjust the gain, specified in decibels per second.

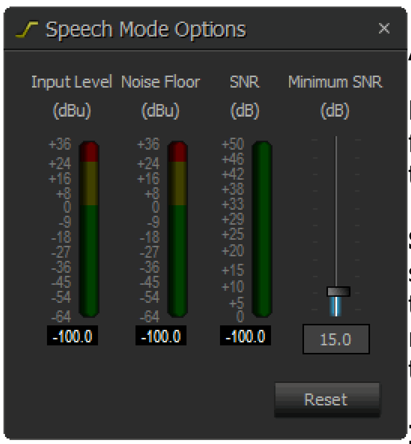
Limiter On/Off turns the clip limiter feature on or off. When the clip limiter is on, the AGC will temporarily reduce the gain applied to the input signal if that gain would have caused the signal to clip. Gain adjustments made by the clip limiter may briefly exceed the Maximum Gain Adjustment Rate as necessary to prevent clipping.

Speech On/Off turns SpeechSense™ technology on or off. When Speech mode is on, the AGC analyzes the input signal to determine if it is human speech. Non-speech signals will not cause the AGC to adjust the gain when Speech mode is on.

Advanced opens the Speech Mode Options window (see below). These advanced controls are only available when Speech mode is enabled.

Bypass disables the AGC without changing the settings.

Reset will adjust settings to the default block values.



Advanced Speech Mode Options Dialog

Noise Floor meter shows the estimated level of the noise floor of the input signal. The Noise Floor is used in calculating the Signal-to-Noise Ratio.

SNR meter shows the Signal-to-Noise Ratio of the input signal. This is equal to the Noise Floor level subtracted from the Input Level. In general, the closer the talker is to the microphone, the higher the Signal-to-Noise Ratio will be while they are talking.

Input Level shows the level of the input signal.

Minimum SNR determines how high the Signal-to-Noise Ratio (SNR) must be before the AGC will make gain adjustments. If the SNR is below the minimum, the AGC will temporarily suspend gain adjustments.

Qualifying Input Signals

The AGC block will only adjust its gain when it receives a qualifying input signal. The definition of a qualifying input signal depends on whether Speech Mode is on or off. When the input signal is not a qualifying signal, the AGC block will hold its previous gain setting until it receives a qualifying signal or until the Hold Time elapses.

When Speech mode is ON, a qualifying signal must satisfy ALL of the following:

- Level of input signal must be above the specified Minimum Threshold.
- Input signal must be human speech.
- Signal-to-Noise Ratio must be above the specified Minimum SNR setting.

When Speech mode is OFF, a qualifying signal must satisfy the following:

- Level of input signal must be above the specified Minimum Threshold.

Side Chain Input

The optional Side Chain input can be enabled when the block is initially created (or by right-clicking on the block and choosing “Edit Block Parameters”). When the Side Chain input is enabled, the audio signal connected to the Side Chain (SC) input is analyzed by the AGC block instead of the input signal. The AGC calculates the amount of gain required to bring the Side Chain signal to the Target Level, and it applies that amount of gain to the input signal.

Logic Output

The logic output on the AGC block will generate a logic HIGH signal when the AGC block is actively making adjustments. In other words, whenever the AGC Active light is lit, the logic output will go HIGH.

Routers

Router Components

These Component Objects provide typical audio routing functions. Routers may be connected between any components within the Layout, for applications which require routing of input signals to various outputs. Routers are available in pre-defined configurations, however, the configuration may be customized when being placed from the Object Bar.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface.

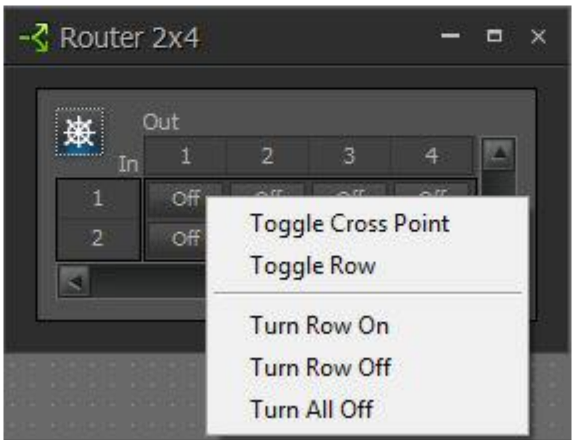
- Router
- Source Selector

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. For increased input/output assignment capability, see Standard Mixer or Matrix Mixer.

**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.



Source Selector

Source Selection blocks are N by 1 routers (where N represents the number of sources) with level control per input and optional logic input and output connections. Source Selection blocks are useful when remote control of audio source selection is required.

When the user first places a Source Selection block into a Tesira layout, this prompts an initialization window.

Source Channel Count specifies the number of input channels and generally corresponds to the number of sources from which the user can choose.
Enable Stereo creates a separate left and right channel for each input, as well as the output.

Enable Logic provides a logic input and output connection point for each channel.

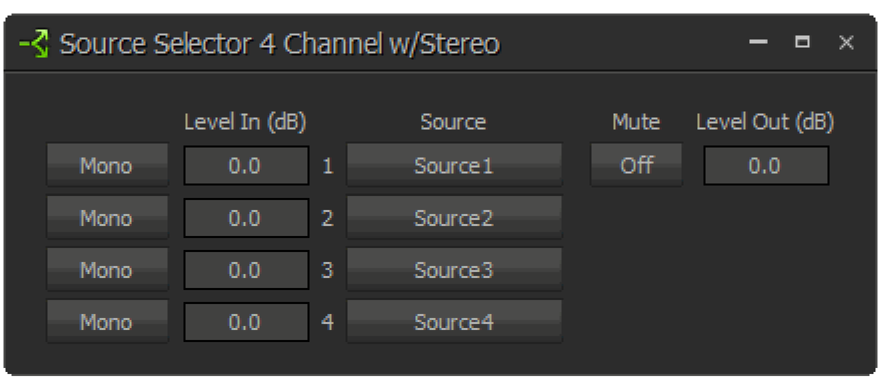
Source Selection is represented in the layout as a block with a number of audio input connections (specified by the Source Channel Count parameter), one audio output connection, and optionally, a logic input and output connection point for each channel. If logic is enabled, a low-to-high logic transition (i.e., a rising edge) presented to a logic input connection will cause the Source Selection block to switch to the corresponding audio input channel, and the corresponding logic out connection will be at a logic high. All other logic outputs will be low.

Double clicking on a Source Selection block produces a control window.

Mono is used to designate an input as a mono (single-channel) input. This button is only available if the block is in Stereo mode. When selected, the right channel of the input will be grayed out and the mono signal should be connected to the left input. If a mono source is selected, it will be sent equally to both the left and right outputs.

Level (dB) adjusts the level (-100 to 12 dB) of the source connected to that channel's input.

Source buttons are used to select the input source that is routed to the audio output connector of the Source Selection block. Only one source can be selected at a time. If the user right clicks a source selection button, this produces a dialog box that allows customization of the text that is displayed on that button. This dialog box can be minimized to create user control surfaces (see Customizing Component Objects).



Delays

Delay Components

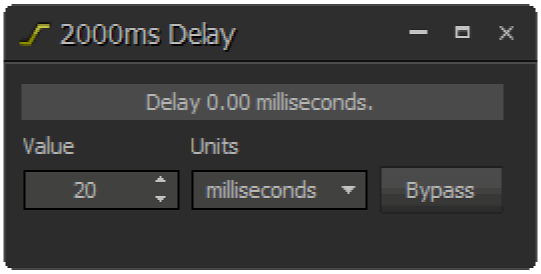
These Component Objects provide typical audio time-delay functions. Delays may be connected between any components within the Layout, for applications which require room delay and/or loudspeaker time-alignment.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface.

- Delay

Right-clicking over the object provides a pop-up menu of options.

Delay



Delay blocks will delay an audio signal for the specified amount of time. When first created, the maximum delay available in the delay block must be chosen. Delay memory is allocated based on the maximum delay setting for each delay block. Each Tesira DSP-2 card has a maximum of 150 seconds of delay memory available.

Value determines the amount of delay, based on the selected Units.

Units selects either time (milliseconds) or distance (centimeters, meters, inches, or feet). Selecting a distance unit will calculate the delay time based on the time it takes sound to travel the specified distance.

Bypass disables the Delay without changing settings.

Controls

Control Components

These Component Objects provide both internal and external control functions. Level Controls, Mute Buttons, and Invert may be connected between components within the Layout, for control of volume, muting, and polarity. Preset and Remote Preset Buttons may be placed within the Layout, and defined to recall specified Presets.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface.

- Level
- Invert
- Mute
- Preset Button
- Command String
- Dialer
- TEC-1

Right-clicking over the object provides a pop-up menu of options. Control Dialog Boxes for Level, Invert, Mute, and Preset related components can be minimized to create user control surfaces (see Customizing Component Objects).

Level

Level blocks can be used to adjust the level (i.e. volume) of the audio signals that are passing through them. Audio signals can also be muted or unmuted within Level blocks.

The channel level may be entered numerically from a keyboard, adjusted by dragging the fader with the mouse, or nudged up and down by using the up/down arrows on the keyboard.

Initialization dialog

The initialization dialog provides several options:

Gang Controls allows all of the channels within the Level block to be adjusted simultaneously from a single set of controls. Level blocks with a "G" in the upper-right corner of the block are ganged.

Use Logic adds logic inputs to the top of the Level block. This feature allows the level of a channel to be incremented or decremented using a logic signal.

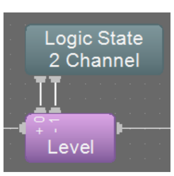
Use Ramping allows for continuous ramping of the level when incremented/decremented via a logic signal. This option is only available is "Use Logic" is enabled.

Redundant allocates the block to a Tesira Server redundant pair, See Redundancy for more details.

Control Dialog Boxes for Level components can be minimized to create user control surfaces (see Customizing Component Objects).

Logic Control

Enabling the "Use Logic" option of a Level block adds two logic inputs to the block for every audio channel. Each pair of logic outputs is labeled + and -. When one of the logic inputs receives a HIGH logic signal, it will increment (+) or decrement (-) the level of the corresponding channel.



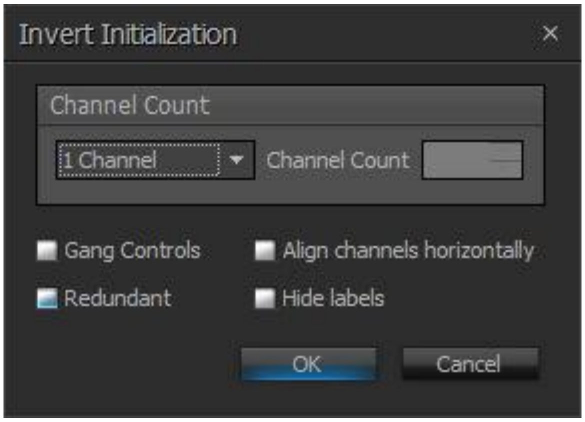
The amount that the level will be incremented or decremented is controlled by the **Step** parameter (specified in decibels).

The **Rate** parameter (specified in milliseconds) controls the behavior of the block when a HIGH logic signal is continuously applied to a increment/decrement logic input for a length of time. When a HIGH logic signal is first applied, the level will immediately increment or decrement by the Step amount. Then, if the logic signal remains HIGH, after every increment of the time specified in Rate, the level will be incremented or decremented by the Step amount again.

Invert

Initialization Dialog

Invert blocks can be used to reverse the polarity(180° phase shift) of the audio signals that are passing through them. The initialization dialog provides several options:



Gang Controls allows all of the channels within the Invert block to be muted/unmuted simultaneously from a single mute button. Invert blocks with a “G” in the upper-right corner of the block are ganged.

Align channels horizontally changes the orientation of the invert buttons when the control dialog window is minimized.

Redundant allocates the block to a Tesira Server redundant pair, See Redundancy for more details.

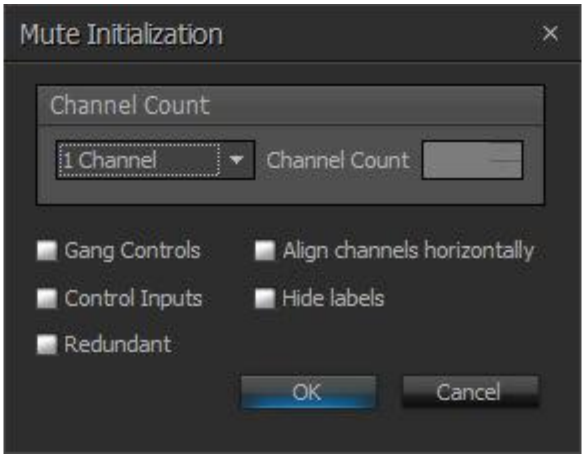
Hide labels removes channel labels when the control dialog window is minimized.

Control Dialog Boxes for Invert components can be minimized to create user control surfaces (see Customizing Component Objects).

Mute

Mute Control blocks can be used to mute and unmute the audio signals that are passing through them.

The initialization dialog provides several options:



Gang Controls allows all of the channels within the Mute Control block to be muted/unmuted simultaneously from a single mute button. Mute Control blocks with a "G" in the upper-right corner of the block are ganged.

Control Inputs adds logic inputs to the top of the Mute Control block. This feature allows signals to be muted/unmuted via logic signals. When a logic input receives a HIGH signal, it mutes the audio for that channel. A LOW signal unmutes the audio.

Align channels horizontally changes the orientation of the mute buttons when the control dialog window is minimized.

Hide labels removes channel labels when the control dialog window is minimized.

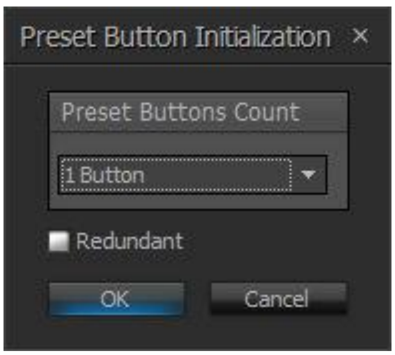
Redundant allocates the block to a Tesira Server redundant pair, See Redundancy for more details.

Control Dialog Boxes for Mute components can be minimized to create user control surfaces (see Customizing Component Objects)

Preset Button

Preset Button blocks allow for logic signals to trigger presets to be recalled. A Preset Button block has one logic input per preset button, and the preset for each preset button is selected via a drop-down menu in the control dialog window. When the logic input receives a HIGH logic signal (specifically, a LOW-to-HIGH logic transition), it recalls the preset loaded into the corresponding preset button. Presets can also be recalled by clicking on the "Recall" button next to the desired preset.

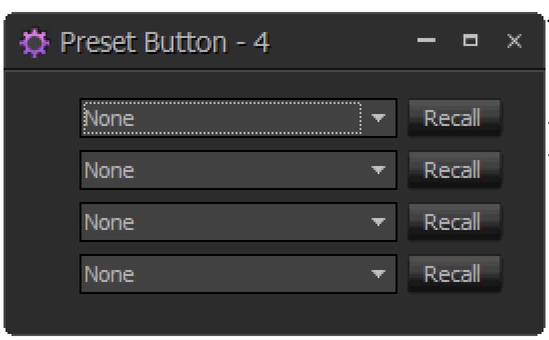
Initialization dialog



Preset Button Count allows up to 10 buttons can be allocated to a block.

Redundant allocates the block to a Tesira Server redundant pair, See Redundancy for more details.

 This icon denotes that this preset is the last preset to have been recalled in the system.



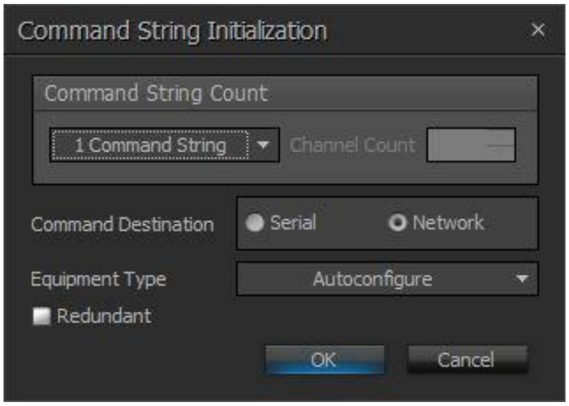
The names of presets cannot be edited within the Preset Button, instead the name of the preset can only be edited within the Preset Manager. Changes to the preset name will be reflected automatically in the Preset Button block.

Control Dialog Boxes for Preset Button components can be minimized to create user control surfaces (see Customizing Component Objects)

Command String

Command String blocks allow serial control of external devices via the Serial Control Port or the IP Network. Alternatively, if active third party control feedback is required please see the Subscriptions information in the TTP section.

When a control input node along the top of the block receives a HIGH logic signal (specifically, a LOW-to-HIGH logic transition), it will trigger a user-defined string to be transmitted from the unit's serial port or network connections. Commands can also be initiated using the command button which appears within the control dialog box.



Command String Count provides a dropdown where the number of channels can be selected. If Custom is selected from the drop-down list the number of channels between 1 and 32 can be specified.

Command Destinations can be either a serial port or Network (IP) port.

Equipment Type specifies what type of hardware the compiler should allocate the block to. Review the Equipment Type section for more details.

Redundant allocates the block to a Tesira Server redundant pair, See Redundancy for more details.

Control Dialog

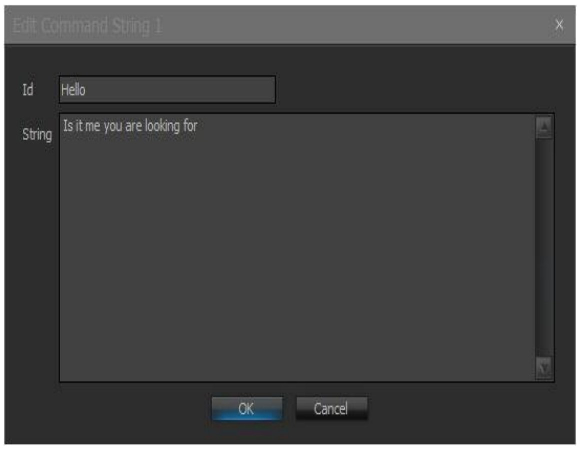
Input relates to the corresponding logic Input

Command ID Send button allows manual sending of the command string

Command String shows the data that will be sent
Edit opens the **Edit Command String** Dialog.

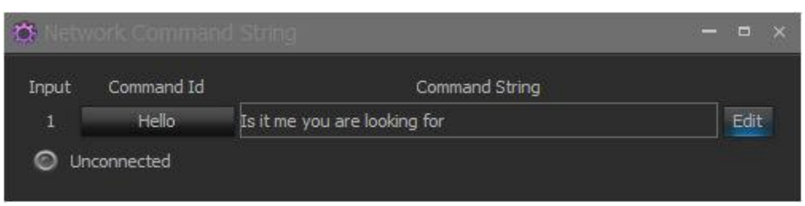
Edit Command String

Command ID - Clicking the Command ID button will trigger the transmission of the command string.



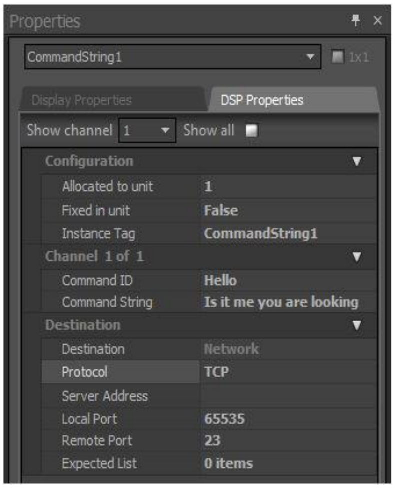
Edit Allows the Command Id name and String to be defined
Status only shown on the Network Command String. It will show the state of the connection:

- Connected
- Unconnected
- Connect Error (101)

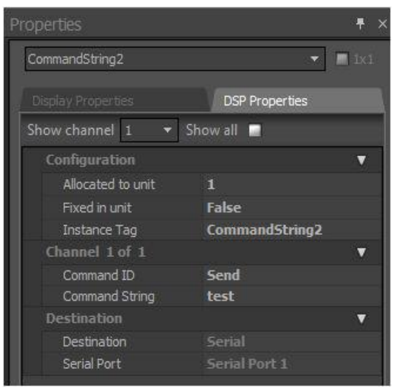


The Property sheet of the Command string block is used to define the connection settings.

Network Command String Block Properties:

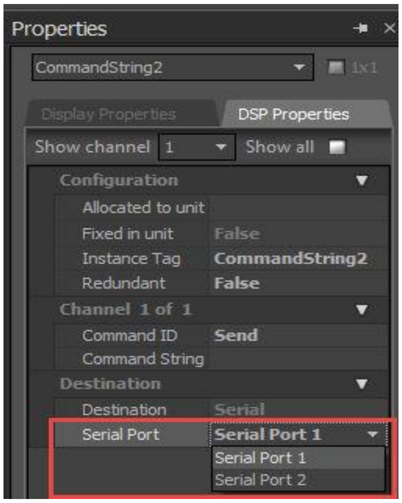


Serial Command String Properties:



Some command strings may need to contain non-printable ASCII characters. A non-printable character can be represented by a tilde (~) followed by the two-digit hex code for the desired ASCII character. For example, a carriage return character would be entered as ~0D and a line feed character would be entered as ~0A. If you need to send an actual tilde character in your command string, it can be entered as ~7E.

The Command String block will only transmit strings from serial port of the unit to which the block has been allocated. On a Server-IO and Server, it can be either of the ports depending on how they are configured. On TesiraFORTÉ and the EX-Logic there is only one serial port. The Property Sheet allows selection of which port is used.



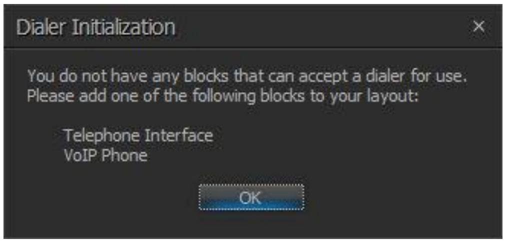
Consider fixing the allocation of Command String blocks to prevent the block from being unexpectedly allocated to the wrong device.

The Baud rate for the serial port for the Server IO, Server and TesiraFORTÉ is configured in Device Maintenance under Serial Port Settings.
The Baud rate for the serial port for the EX-Logic is configured in Expander device maintenance in the EX Logic Serial Device Settings section.

Control Dialog Boxes for Command String components can be minimized to create user control surfaces (see Customizing Component Objects).

Dialer

The Dialer block provides an interface for the dialing functions of an analog or VoIP telephone interface. If no VoIP or TI component objects are available the following dialog is shown:



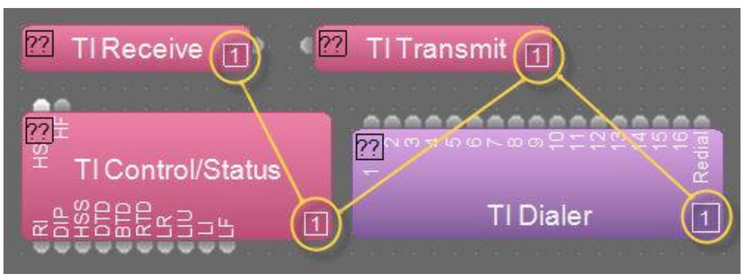
Initialization Dialog

Upon creating a new Dialer block, it must be linked to an existing set of telephone interface blocks. Linking it to an analog telephone interface creates a "TI Dialer" block which controls a single telephone line, whereas linking it to a VoIP telephone interface creates a "VoIP Dialer" block which controls two VoIP phone lines.

A dialer block allows up to 16 speed dial numbers to be stored and recalled. Double-click on a speed dial field to edit its label and number. Click on a speed dial button to dial that number. Logic inputs on the top of the block allow speed dial numbers to be triggered via logic signals, and also allow a redial to be triggered.

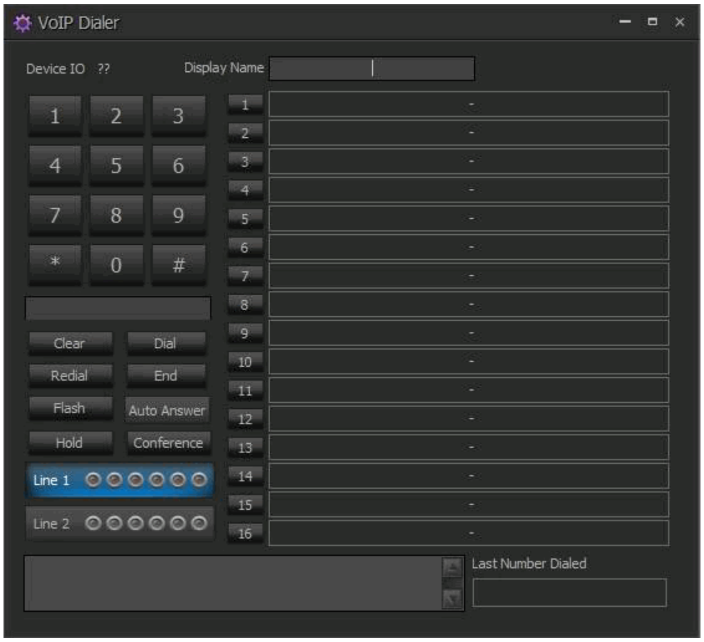
Dialer blocks which are linked to VoIP telephone interfaces provide some additional options, including putting calls on Hold and conferencing multiple calls. VoIP interfaces can support up to three simultaneous calls - one main call and two Call Appearances on each line, which can be conferenced together or independently put on hold.

The drop down will allow selection of any VoIP or telephone interface lines.



Dialer Control Dialog

The text field at the bottom left of the dialog displays the current state of the telephone interface, as well as any errors that may have occurred.



Display Name displays an identifier for the telephone line, which may be used as its Caller ID name. The name displayed here is defined in the corresponding Control/Status block for the telephone line.

Clear removes all numbers from the dialing field.

Dial triggers the telephone interface to go off-hook and dial any numbers which have been entered into the dialing field.

Redial dials the last number dialed.

End hangs up the telephone line. For a VoIP interface with multiple call appearances conferenced together, pressing the End button once will remove the currently selected call appearance from the conference and put it on hold. Pressing the End button again will hang up the call appearance.

Flash performs a hook flash, which may result in different behavior depending on whether the Dialer block is linked to an analog or VoIP telephone interface. For an analog phone line, the Flash button will trigger a traditional hook flash event. For a VoIP phone line, the Flash button will cause the current call to be put on hold, and a new call appearance to be started.

Auto Answer determines whether the telephone interface will automatically answer incoming calls. The number of rings to wait before auto-answering can be adjusted in the Control/Status block of the corresponding telephone interface.

Hold (VoIP Dialer only) puts the call on hold, which effectively mutes both the transmit and receive signals for the selected line.

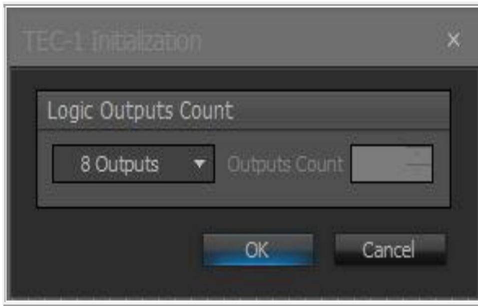
Conference (VoIP Dialer only) joins up to two call appearances on a line to a conference call. Any call appearances which are currently on hold will be made active.

Line 1/Line 2 (VoIP Dialer only) allows switching between which VoIP line is currently being controlled by the Dialer block. The six circular buttons next to each line show the call appearances for each line. Clicking on an idle call appearance button will put other active call appearances on hold and start a new call appearance. Green call appearance buttons denote active calls, red buttons denote calls on hold, and grayed-out buttons denote idle (unused) call appearances.

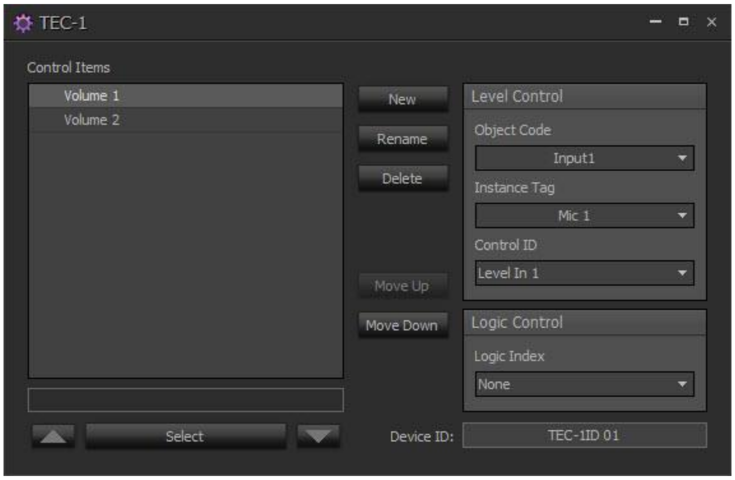
TEC-1

Tesira Ethernet Controller 1 (TEC-1) is an external remote control panel that integrates with Tesira systems via the control network, using a single CAT5 cable for control and Power-over-Ethernet (PoE). TEC-1 allows for the selection of up to 32 control items. A control item can be the initiation of a logic event (such as a preset recall or a source selection), selection of a volume assignment, or both. Volume assignments may be individual or ganged levels within the layout, including Level Control blocks, as well as levels within other component blocks (such as Input/Output blocks, Mixers, Equalizers, etc).
Please review the TEC-1 Device Maintenance for details on how to configure the device on the network.

When a TEC-1 block is first placed into the layout, an initialization dialog box appears. Logic Outputs Count specifies the number of logic connection points (0 to 32) on the TEC-1 block. These connection points are typically wired to Remote Preset or Source Selector blocks but can also be used as general-purpose logic inputs. The use of Logic Gates, Flip Flops and Logic Delays allows for different priorities and functions to be assigned.

<table>
<tr><td>
</td><td>TEC-1 blocks in the Tesira DSP layout are correlated with physical field devices by a parameter called the Device ID. The Device ID associated with a TEC-1 block is found and set in the Property sheet for the block.</td></tr>
<tr><td>
</td><td>No two TEC-1 blocks in any layout may have the same Device ID; however, multiple TEC-1 units may have the same Device ID. In that case, the units' functions are identical and governed by the TEC-1 block with the corresponding Device ID.</td></tr>
<tr><td>
</td><td>TEC-1 is represented in the layout as a block with a number of logic connection points (determined by the Logic Output Count setting when the block is created).</td></tr>
</table>

All programmed panel functions are assigned using the control dialog box, which is produced by double-clicking the TEC-1 block.



At the bottom right, the **Device ID** associated with this set of controls is shown. A list displays all defined control items and customizable Control Label.

The **New** button creates a control item in the list, and the user is able to edit the default label.
Rename allows the user to change the Control Label of the selected control item.
Delete removes the selected control item from the list.
Move Up and **Move Down** modify the order of the control items in the list, giving the user the ability to control the display order of the control items on the TEC-1 unit.

Each control item may have a Level Control assignment, a Logic Control assignment, or both.

Level Control

The **Object Code** or **Instance Tag** can be used to specify the DSP object to be controlled. The text box will expand when selected and will give a list of available blocks found in the layout. If the Object Code is selected the Instance Tag is automatically entered. If the Instance Tag is selected the Object Code is automatically entered. The Object Code is assigned by the Tesira Compiler at compilation and is not user adjustable. The Instance Tag is User adjustable and must be unique.
Control ID selects from a list of available levels within the chosen block.

Logic Control

Logic Index specifies which logic connection point, if any, on the TEC-1 block will be triggered by a logic pulse when that control item is selected.

Select Button

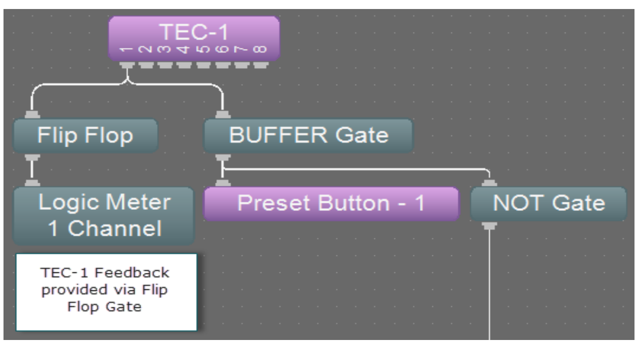
Once the TEC-1 block has been programmed with control items, the **Select** and Up/Down Arrow buttons at the bottom of the control dialog box may be used to mimic how the control will function from the physical panel. For example if a volume control is selected it will display the mode in the text field above the Select Button and the Up/Down Arrows will also control the level.

Control dialog boxes for TEC-1 devices may be minimized to create user control surfaces (see Customizing Component Objects).

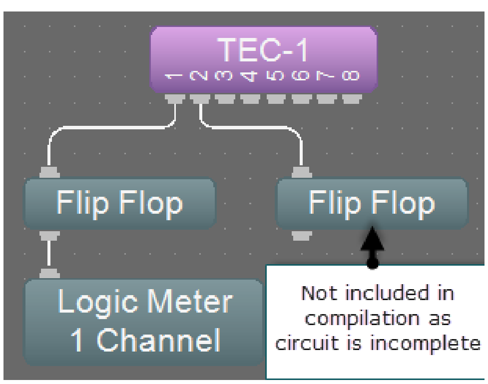
Logic Output

Certain blocks support logic feedback to the TEC-1 in order for the triangle indicator to be used to display the active function. In order for the blocks to indicate feedback they must be directly connected to the TEC-1 logic node.
The following blocks support feedback: Source Selector, Room Combiner, Flip Flop, Preset Button.

It may be that the logic circuit being used 'fans out' to multiple blocks that provide feedback logic. In this instance a BUFFER_Gate should be used to stop logic feedback on unwanted parts of the logic circuit.



The use of a Flip-Flop gate can also be used to enable feedback logic on some circuits. The Tesira compiler ignores incomplete logic circuits, so the use of another block such as a logic meter is required.



Meters

Meter Components

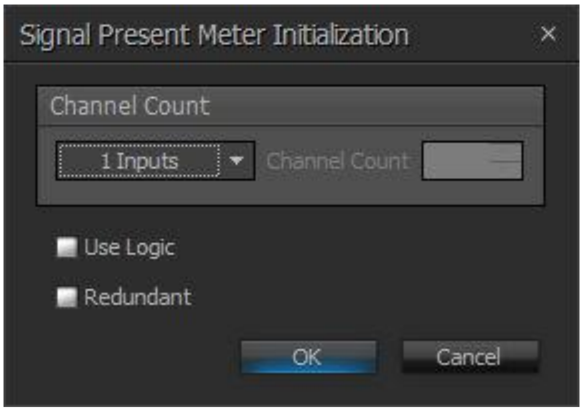
Meter Components allow for a visual representation of the level (i.e. volume) of an audio signal, and may be used for diagnostic and setup purposes, or for applications which require real-time metering.
Meter Components include:

- Signal Present Meter
- Audio Meter

Signal Present Meter

A Signal Present Meter block provides a simple visual indication when the level of an audio signal exceeds a specified threshold level. Signal Present Meters can also be configured to generate a logic signal when the audio level threshold is exceeded.

When initially inserting a Signal Present Meter into a configuration:

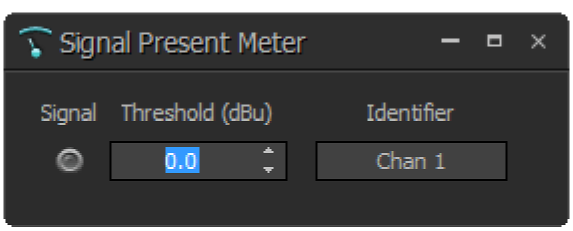


Channel Count determines how many separate audio signals the block will be able to meter.

Use Logic will create one logic output terminal on the block for each channel. See below for an explanation of the logic features of this block.

Redundant determines if the block is used in a system which is designated as redundant. See Redundancy for more details.

Control Dialog



Signal is a visual indication which will light up when the audio signal exceeds the threshold.

Threshold is the level which the audio input signal must exceed before the Signal Present Meter will activate.

Identifier provides a custom label when the dialog box is minimized.

Invert reverses the logic signal that is produced by the block. When Invert is activated, the logic output will go HIGH when the audio level is below the threshold, and it will go LOW when the audio level is above the threshold. Using Invert is equivalent to wiring a NOT gate to the logic output. Invert is only available when "Use logic" is checked in the initialization dialog.

Debouncer Delay stabilizes the logic output signal by requiring that the audio signal remain above or below the threshold for a certain amount of time before the logic output signal will change. Setting an On Delay will require that the audio signal remains above the threshold for the specified time before the logic output signal will go HIGH. Setting an Off Delay will require that the audio signal remains below the threshold for the specified time before the logic output signal will go LOW. Both an On and Off Delay can be specified simultaneously. Additionally, the Invert control does not affect the operation of the Debouncer Delay; that is, the On Delay is always triggered when the audio level exceeds the threshold, and the Off Delay is always triggered when the audio level is below the threshold.

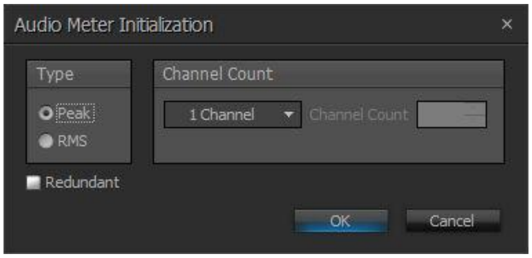
Using Debouncer Delay is equivalent to wiring a Logic Delay to the logic output. Debouncer Delay is only available when "Use logic" is checked in the initialization dialog.

The Logic Meter's Control Dialog can be minimized to create user control surfaces (see Minimizing Control Dialogs).

Audio Meter

Audio Meters allow for a visual representation of the level (i.e. volume) of an audio signal, in decibels.

When initially inserting an Audio Meter into a configuration:



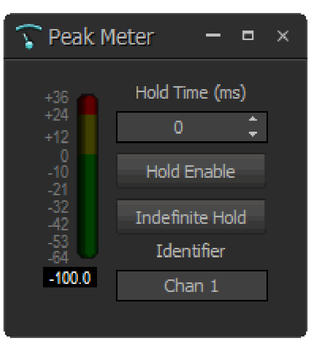
Type selects the type of audio meter, **Peak** or **RMS**.

Peak vs. RMS
A Peak Meter displays the instantaneous level of the signal as fast as it can. This is useful for metering highly transient signals, or for gauging the highest level of an audio signal regardless of how brief the peak is. An RMS meter has a slower response and displays a level that is averaged over time. The result is that short peak signals may not register as much on an RMS meter, however the response of an RMS meter is generally considered to be closer to the response of the human ear. Note that sending the same signal to a Peak Meter and an RMS Meter may not result in an identical decibel reading on both meters.

Channel Count determines how many separate audio signals the block will be able to meter.

Redundant determines if the block is used in a system which is designated as redundant. See Redundancy for more details.

Control Dialog



The meter shows the signal level in decibels (specifically, in dBu). Audio signals will clip at +24dBu.

Hold Time can be used to slow the meter ballistics as it determines the maximum speed at which the meter will decay. This can be set from 0.1ms to 1000ms. The meter will decay by up to 20dB during the selected hold time. This function is only available when Hold Enable is enabled.

For example:
The decay rate on a 1000ms hold time would be 20dB a second. So for signals at 0dB it will take the meter 4 seconds to reach -80dB
The decay rate on a 500ms hold time would be 40dB a second. So for signals at 0dB it will take the meter 2 seconds to reach -80dB

Hold Enable turns the Hold Time function on and off.

Indefinite Hold causes the meter to constantly display the highest level it recorded since Indefinite Hold was enabled.

Identifier provides a customizable text label. This is also shown when the dialog box is minimized.

The Logic Meter's Control Dialog can be minimized to create user control surfaces (see Minimizing Control Dialogs).

Generators

Generator Components

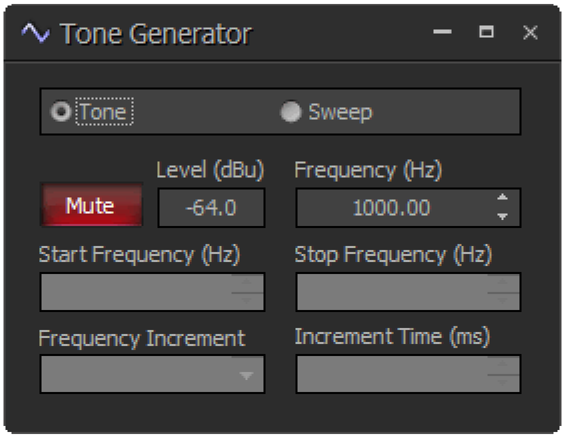
Generator Components digitally generate different audio signals for diagnostic and setup purposes, or for applications which require tones or sound masking signals. Available generated signals include sine waves, sweeps, pink noise, and white noise.

Generator Components include:

- Tone Generator
- Noise Generator

Tone Generator

A Tone Generator block can generate pure tone (sine wave) audio signals at different frequencies.



Tone causes the generator to output a signal at a fixed frequency.

Sweep causes the generator to output a signal whose frequency regularly or continuously changes.

Mute turns the generator on or off.

Level sets the audio signal level of the generated signal.

Frequency sets the frequency of the generated tone (when in Tone mode).

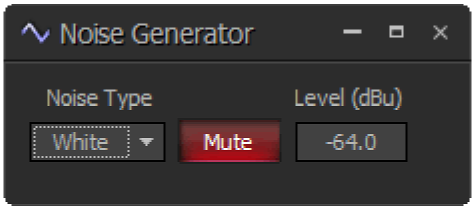
Start Frequency and **Stop Frequency** set the range of frequencies that the generated tone will sweep through (when in Sweep mode)

Frequency Increment sets the step size by which the frequency of the generated tone will be increased (when in Sweep mode). The maximum setting is 1 octave, which equates to a doubling of the frequency The minimum setting is 1/96 octave, which results in a smoothly swept tone.

Increment Time sets the time for which each frequency increment is held (in Sweep mode).

Noise Generator

A Noise Generator block can produce two types of randomly-generated noise signals: pink noise and white noise.



Noise Type selects between white noise and pink noise generation. In general, white noise contains much more high frequency energy than pink noise.

Mute turns the generator on or off.

Level sets the audio signal level of the generated signal.

Logic

Logic Components

Logic Gates and Logic Delay may be connected between component control nodes in the layout, to customize control behavior.

Supported Logic Functions Include:

- Logic Gate
- Logic State
- Flip Flop
- Fan-In OR Pulse
- Logic Delay
- Logic Meter
- Digital Input
- Digital Output
- Control Voltage

Logic Gate

Logic Gates are used to perform functions on logic signals. Logic signals can only be input and output from logic nodes, which appear on the top and bottom of component blocks.

A logic signal can only be in one of two states: HIGH or LOW (also commonly referred to as ON or OFF, 1 or 0, etc).

Logic Gates can perform various functions on incoming logic signals. The output of a Logic Gate depends on both its input and the type of Logic Gate that it is. Logic gates can be used in any combination to produce many varied behaviors tables below summarize the function of each Logic Gate.
Refer to the Logic Blocks for more details.

NOT Gate

INPUT	OUTPUT	Summary
HIGH	LOW	A NOT gate simply flips its input signal to the opposite state. Each individual input only affects its corresponding output.
LOW	HIGH	

AND Gate

INPUT 1	INPUT 2	OUTPUT	Summary
LOW	LOW	LOW	An AND gate's output will only go high whenever all of its inputs are high. Otherwise, its output will be low.
LOW	HIGH	LOW	
HIGH	LOW	LOW	
HIGH	HIGH	HIGH	

NAND Gate

INPUT 1	INPUT 2	OUTPUT	Summary
LOW	LOW	HIGH	A NAND gate's output will only go low whenever all of its inputs are high. Otherwise, its output will be high. A NAND gate is a combination of a NOT gate and an AND gate.
LOW	HIGH	HIGH	
HIGH	LOW	HIGH	
HIGH	HIGH	LOW	

OR Gate

INPUT 1	INPUT 2	OUTPUT	Summary
LOW	LOW	LOW	An OR gate’s output will go high whenever at least one of its inputs is high.
LOW	HIGH	HIGH	
HIGH	LOW	HIGH	
HIGH	HIGH	HIGH	

NOR Gate

INPUT 1	INPUT 2	OUTPUT	Summary
LOW	LOW	HIGH	A NOR gate’s output will go high whenever at least one of its inputs is low. A NOR gate is a combination of a NOT gate and an OR gate.
LOW	HIGH	LOW	
HIGH	LOW	LOW	
HIGH	HIGH	LOW	

XOR Gate

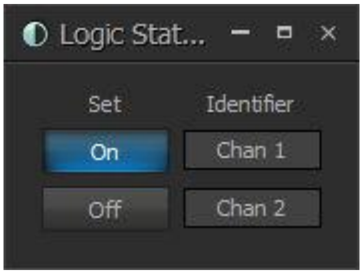
INPUT 1	INPUT 2	OUTPUT	Summary
LOW	LOW	LOW	An XOR gate’s output will go high whenever only one of its inputs is high. For XOR gates with more than 2 inputs, the output will go high whenever an odd number of inputs are high.
LOW	HIGH	HIGH	
HIGH	LOW	HIGH	
HIGH	HIGH	LOW	

BUFFER Gate

INPUT 1	OUTPUT	Summary
LOW	LOW	The BUFFER logic gate is a pass-through block where the logic input and logic output is always the same value. This block is intended to be used with the TEC-1 to allow Logic priority for feedback. The circuits that include the buffer gate will not provide feedback to the TEC-1 screen.
HIGH	HIGH	

Logic State

A Logic State block is a logic signal generator. It outputs either a low or high logic signal depending on the state of its Set button. The Set button can be toggled either by clicking it with the mouse or via presets. Below is an example of a 2-channel Logic State block with channel one set high and channel two set low. Refer to the Logic Blocks for more details.



The Logic State's Control Dialog can be minimized to create user control surfaces (see Minimizing Control Dialogs).

Flip Flop

A Flip Flop is a logic gate which transforms a momentary logic signal into a latching logic signal. Flip Flops will toggle their output whenever they receive a LOW-to-HIGH logic signal transition at their input. A Flip Flop will not respond at all to a HIGH-to-LOW transition. Refer to the Logic Blocks for more details.

Flip Flops have a Control Dialog which is used to manage their initial states.

Fan-In OR Pulse

Fan-In OR Pulse is a logic block which outputs a brief momentary logic pulse in response to a rising edge on any of its input logic signals.

A **Fan-In OR Pulse** block normally outputs a LOW logic signal, and when any of its inputs go HIGH (i.e. a LOW-to-HIGH logic transition), the Fan-In OR Pulse block outputs a HIGH logic signal for 150ms and then returns to a LOW state. A fixed de-bounce delay of 300ms is used after the initial pulse.

The **Fan-In OR Pulse** block can be useful when multiple logic signals need to be connected to the same logic input of a block, in order to trigger the same event using multiple logic signals. For example, if multiple logic signals need to trigger a Flip Flop gate or a Remote Preset block, a Fan-In OR Pulse block can be useful in aggregating those signals. Using a normal OR Gate in these cases may be problematic, because if one of the inputs is latched high, none of the other inputs will be able to trigger the event.

Logic Delay

A Logic Delay block can be used to perform time-based functions on logic signals. A Logic Delay works by only passing logic signals that have remained in a particular state for a certain amount of time.

When a HIGH logic signal is sent to the input of a Logic Delay block, **On Delay** sets the amount of time that the input signal needs to remain HIGH before the Logic Delay's output signal will go HIGH. If the logic signal goes LOW before the specified amount of time, the Logic Delay's output will not change and will remain LOW.

When a LOW logic signal is sent to the input of a Logic Delay block, **Off Delay** sets the amount of time that the input signal needs to remain LOW before the Logic Delay's output signal will go LOW. If the logic signal goes HIGH before the specified amount of time, the Logic Delay's output will not change and will remain HIGH.

Bypass temporarily defeats the Logic Delay without changing the settings in the block. This is equivalent to setting the On and Off Delays to zero.

Both an **On Delay** and an **Off Delay** can be set simultaneously for each channel of a Logic Delay block. The maximum delay time that can be set is 60,000 milliseconds (60 seconds). Delays longer than 60 seconds can be accomplished by chaining multiple Logic Delays together.

Refer to the Logic Blocks for more details.

Logic Meter

A Logic Meter allows metering of logic signals. When the input logic signal is HIGH, the Logic Meter's indicator will light. When the input logic signal is LOW, the indicator will go dark.

The Logic Meter's Control Dialog can be minimized to create user control surfaces (see Minimizing Control Dialogs).
Refer to the Logic Blocks for more details.

Identifier provides a custom label when the dialog box is minimized.

Logic Input

Logic Input blocks generate logic signals depending on the state of external switches or devices that are connected to the logic inputs of an EX-LOGIC expander.

The logic signal generated by a Logic Input block will be LOW when its corresponding logic input is "closed" (i.e., when the logic input is shorted to ground, or more specifically, when the voltage between the logic input and ground is less than approximately 0.8 volts).

The logic signal generated by a Logic Input block will be HIGH when its corresponding logic input is "open" (i.e., when the logic input is open to ground, or more specifically, when the voltage between the logic input and ground is between approximately 2.2 volts and 5 volts). Refer to the Logic Blocks for more details.

Device IO displays the Logic I/O channel(s) on the EX-LOGIC which this block represents.

Identifier provides a text label for the channel.

Invert flips the logic signal being generated by the Logic Input block. When inverted, a Logic Input block will output a LOW signal when the input is open to ground, and a HIGH signal when the input is closed to ground.

Logic Output

Logic Output blocks accept logic signals which control the state of the corresponding Logic I/O terminals on an EX-LOGIC expander.

When a Logic Output block receives a LOW logic signal, the corresponding logic output will “close” (i.e., the logic output will be shorted to ground, or more specifically, the voltage between the logic output and ground will be less than approximately 0.5 volts).

When a Logic Output block receives a HIGH logic signal, the corresponding logic output will “open” (i.e. the logic output will be open to ground, or more specifically, the voltage between the logic output and ground will be between approximately 2.7 volts and 5 volts).

Logic outputs can optionally be configured to supply up to 10mA of current (at a voltage of 4.3 to 5 volts) to a connected device, which is intended to power an LED. When creating a Logic Output block, check the “Enable Powered Outputs” checkbox to enable this feature. In general, it is recommended that only one LED be powered from each logic output, as powering multiple LED’s from a single logic output may result in a significant reduction in LED brightness. Refer to the Logic_Blocks for more details.

Device IO displays the Logic I/O channel(s) on the EX-LOGIC which this block represents.

Identifier provides a text label for the channel.

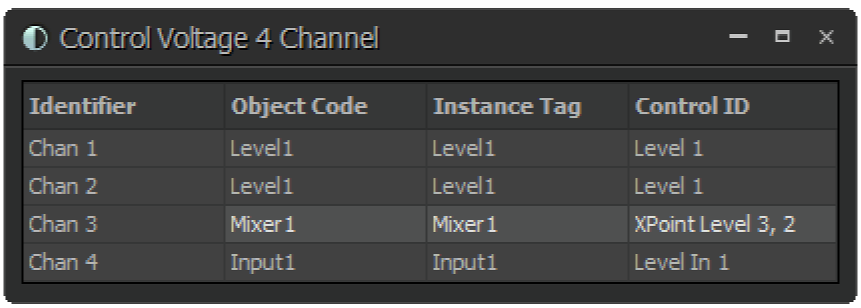
Invert flips the function of the logic signal being accepted by the Logic Output block. When inverted, a Logic Output will open its output in response to a LOW logic signal, and it will close its output in response to a HIGH logic signal.

Control Voltage

Control Voltage blocks allow analog potentiometers to adjust level controls within a Tesira configuration file. Control Voltage blocks require supporting hardware to operate, such as an EX-LOGIC.
Once a Control Voltage block has been placed, double-clicking it will produce a dialog for assigning analog controls. Each channel in Control Voltage block represents a single analog potentiometer, and must be mapped to an existing level control in the Tesira configuration file. Control is not limited to Level Control blocks; any component block which supports level control can be adjusted, including Input blocks, Mixers, Equalizers, etc.

To map a Control Voltage channel to a level control, choose either the Object Code or the Instance Tag of the block to be controlled. Then, choose the channel number to be controlled under Control ID.

External controls must be identified and associated with their corresponding component blocks within the layout (see Equipment Table). The Control Voltage block will assume a full range of 0-5 Volts returning from connected potentiometers, unless the VCB Calibration procedure is followed (see Device Maintenance).



Diagnostic

Diagnostic components

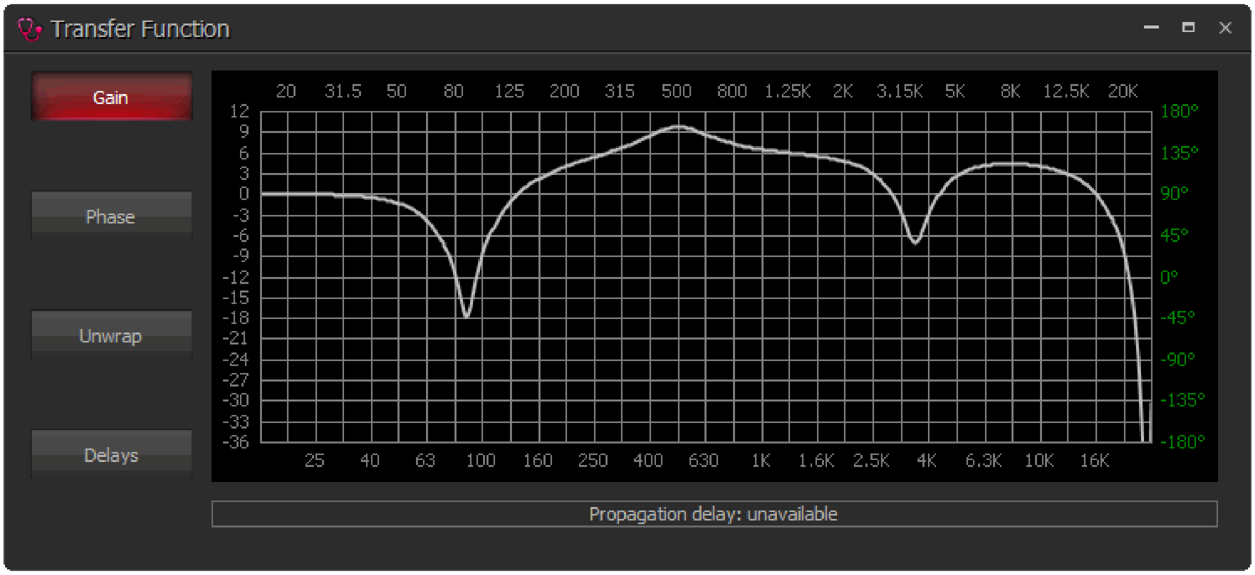
These Component Objects provide Transfer Function displays for diagnostic purposes. Transfer Function may be connected between any two component outputs on the same signal path, for a comparative analysis of processing.

Once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface.

- Transfer Function

Transfer Function

Transfer Function may be connected between any two component outputs on the same signal path, for a comparative analysis of processing.



Gain displays the difference in frequency response between the two comparison points, as a white line.

Phase displays the phase relationship between the two comparison points, as a green line.

Unwrap removes out-of-range phase rotations from the phase display.

Delays adds the effect of user-placed delay blocks (Delay and Matrix Mixer w/ Delay) into the phase display.

The Phase display does not indicate inherent propagation delay within the system. Propagation delay is indicated separately at the bottom of the dialog box (1/3mS hops, if not compiled or real-time including equalization delay, if compiled). When a Transfer Function dialog box is open, the associated signal path is indicated as a dashed orange line in the layout.

Specialty

Specialty Components

These Component Objects provide Pass-Through and Split Pass-Through functions, to aid in the organization of system connections. Pass-Through blocks allow wiring nodes to be strategically placed, so audio or control (logic) signals can be routed in different directions. Split Pass-Through blocks allow associated input and output wiring nodes to be placed in separate locations, with an implicit or 'wireless' connection being maintained between them.

Pass Through and Split Pass Through blocks simply allow custom signal routing, and provide no actual processing of their own. Therefore, they do not have Control Dialog Boxes. Instead, an Initialization Properties window appears, for component definition and customization, when these blocks are placed into the Layout.

- Pass Through
- Split Pass Through Input
- Split Pass Through Output
- Partition Connector Audio Transmitter
- Partition Connector Audio Receiver
- Partition Connector Logic Transmitter
- Partition Connector Logic Receiver
- Via
- Custom Block

Pass Through

Pass-Through blocks allow wiring nodes to be strategically placed, so audio or control (logic) signals can be routed in different directions.
Type selects whether the block is for audio or logic connections. Channel Count selects the quantity of input/output connections to be provided on the block. Signal In determines the physical location of input wiring nodes on the block. Signal Out determines the physical location of output wiring nodes on the block.

Pass-Through blocks allow custom signal routing and labeling, but provide no actual signal processing.

APT stands for Audio Pass-Through.
LPT stands for Logic Pass-Through.

Split Pass Through Input

Split Pass-Through blocks allow associated input and output wiring nodes to be placed in separate locations, with an implicit or 'wireless' connection being maintained between them.

Type selects whether the blocks are for audio or logic connections.

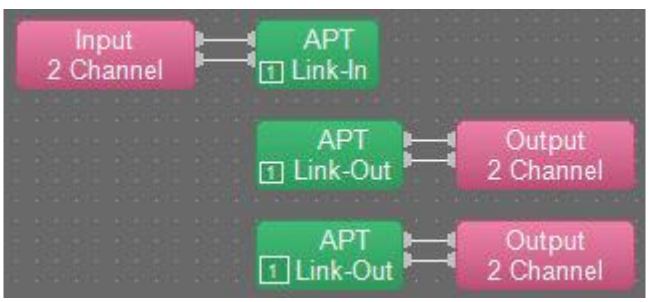
Channel Count selects the quantity of input/output connections to be provided on the blocks.

Separate 'Link-In' and 'Link-Out' blocks are placed for input and output connections. Wiring nodes appear on the left side of input blocks and on the right side of output blocks.

Link-In and Link-Out blocks are also numbered to identify their association. Their unique ID number appears in the bottom left corner of the block. More than one Link-Out block can be placed in association with an existing Link-In block (see Split Pass-Through Output), and all associated blocks will have the same ID number.

Split Pass-Through blocks allow custom signal routing and labeling,but provide no actual signal processing. APT stands for Audio Pass-Through. LPT stands for Logic Pass-Through.

Note that Split Pass-Through blocks cannot be used to send audio from one partition to another. A Partition Connector Audio Transmitter is required for this purpose.

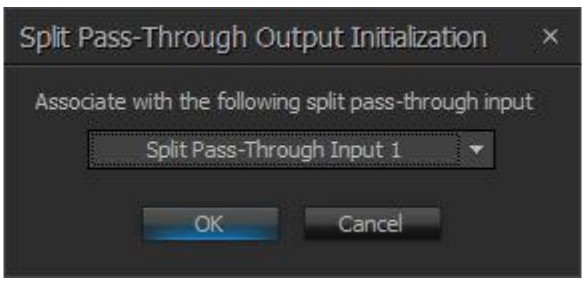


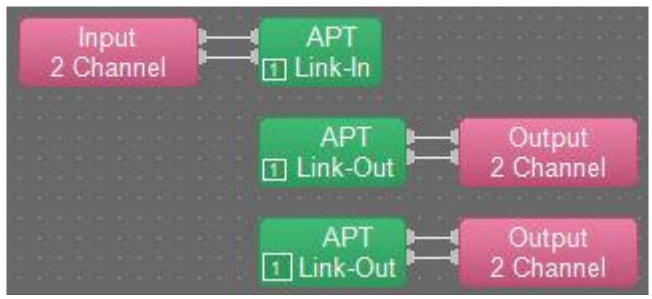
Split Pass Through Output

More than one Link-Out block can be placed in association with an existing Link-In block (see Split Pass-Through Input).

Object Code of Input Link to Associate With selects which existing Link-In block the new Link-Out block should be associated with. The new Link-Out block will include the appropriate number of output wiring nodes, and the same numbered association, as other Link-Out blocks already associated with the selected Link-In block. Additional Link-Out blocks allow a single set of input connections to be distributed to multiple sets of output connections.

Pass-Through blocks allow custom signal routing and labeling, but provide no actual signal processing. APT stands for Audio Pass-Through. LPT stands for Logic Pass-Through. This example shows an existing 2-channel Audio Split Pass-Through with an additional 2-channel Link-Out block.



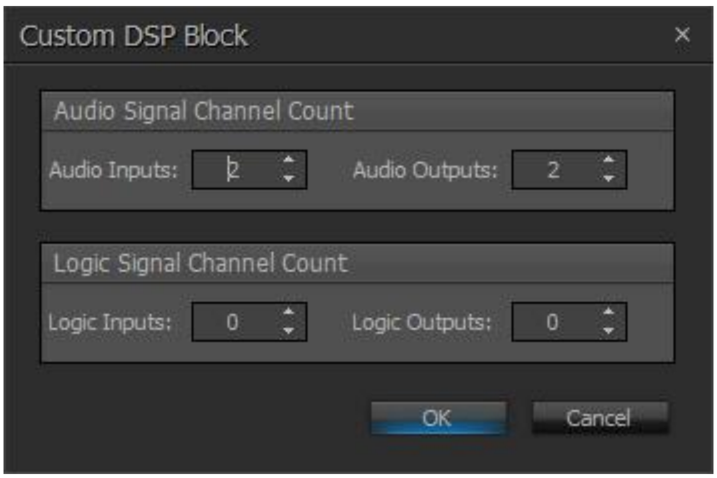


Partition Connector Audio Transmitter

Partition connectors are used to transmit audio and logic signals between different Partitions . Each partition connector consists of a transmitter and one or more receivers.

The Partition Connector Audio Transmitter transmits audio signals to a Partition Connector Audio Receiver in a different partition. The transmitter and receiver must be linked to each other, and this can be accomplished either from the Partition Connector Audio Receiver dialog box (see Partition Connecter Audio Receiver) or from the System View window.

The Partition Connector View tab of the System View window shows all available partition connector transmitters and receivers. Drawing wires from one partition connector to another will link the two connectors. Partition Connector Transmitters can transmit to more than one receiver. Below is an example of the Partition Connector View with three partitions sharing audio.

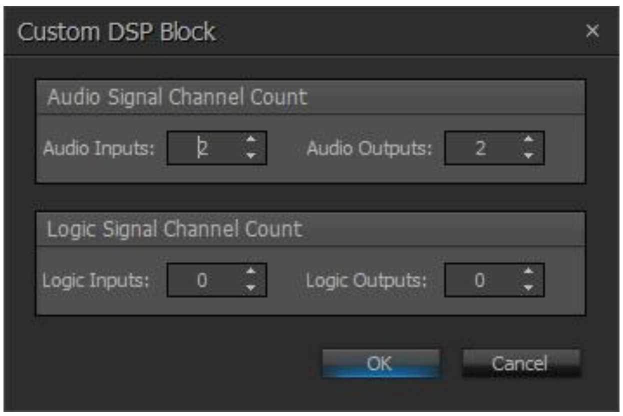


Partition Connector Audio Receiver

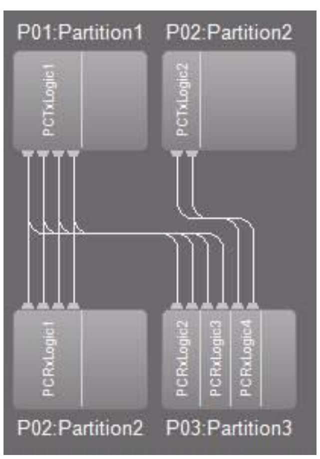
Partition connectors are used to transmit audio and logic signals between different Partitions . Each partition connector consists of a transmitter and one or more receivers.

The Partition Connector Audio Receiver receives audio signals from a Partition Connector Audio Transmitter in a different partition. The transmitter and receiver must be linked to each other, and this can be accomplished either from the Partition Connector Audio Receiver dialog box or from the System View window.

Double-click on a Partition Connector Audio Receiver block to open its Control Dialog. The left side of the Control Dialog displays the channels of this Partition Connector Audio Receiver block. The right side of the Control Dialog displays the channels of all available Partition Connector Transmitter blocks in other partitions. Click on a receiver port and then click on a transmitter port to create a link between them.



Alternately, the Partition Connector View tab of the System View window shows all available partition connector transmitters and receivers. Drawing wires from one partition connector to another will link the two connectors. Partition Connector Transmitters can transmit to more than one receiver. Below is an example of the Partition Connector View with three partitions sharing audio.

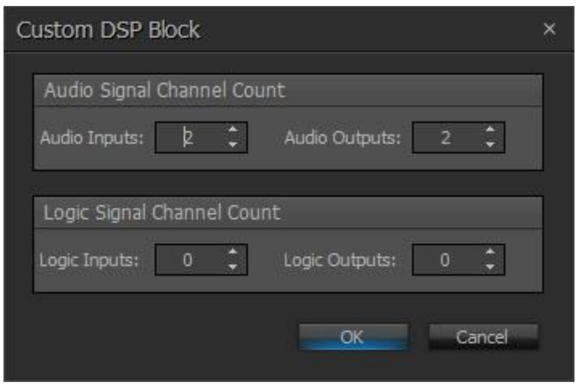


Partition Connector Logic Transmitter

Partition connectors are used to transmit audio and logic signals between different Partitions. Each partition connector consists of a transmitter and one or more receivers.

The Partition Connector Logic Transmitter receives logic signals from a Partition Connector Logic Transmitter in a different partition. The transmitter and receiver must be linked to each other, and this can be accomplished either from the Partition Connector Logic Receiver dialog box (see Partition Connecter Logic Receiver) or from the System View window.

The Partition Connector View tab of the System View window shows all available partition connector transmitters and receivers. Drawing wires from one partition connector to another will link the two connectors. Partition Connector Transmitters can transmit to more than one receiver. Below is an example of the Partition Connector View with three partitions sharing logic.

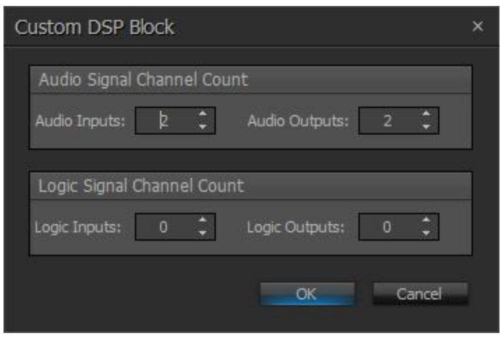


Partition Connector Logic Receiver

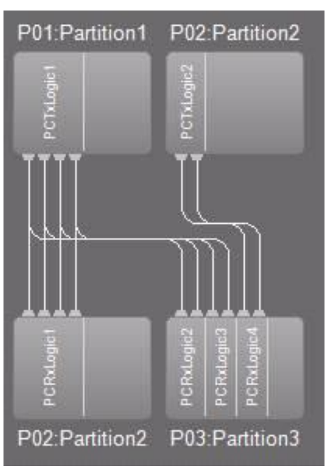
Partition connectors are used to transmit audio and logic signals between different Partitions. Each partition connector consists of a transmitter and one or more receivers.

The Partition Connector Logic Receiver receives logic signals from a Partition Connector Logic Transmitter in a different partition. The transmitter and receiver must be linked to each other, and this can be accomplished either from the Partition Connector Logic Receiver dialog box or from the System View window.

Double-click on a Partition Connector Logic Receiver block to open its Control Dialog. The left side of the Control Dialog displays the channels of this Partition Connector Logic Receiver block. The right side of the Control Dialog displays the channels of all available Partition Connector Transmitter blocks in other partitions. Click on a receiver port and then click on a transmitter port to create a link between them.



Alternately, the Partition Connector View tab of the System View window shows all available partition connector transmitters and receivers. Drawing wires from one partition connector to another will link the two connectors. Partition Connector Transmitters can transmit to more than one receiver. Below is an example of the Partition Connector View with three partitions sharing audio.



Via

The Via block is a special block whose function is to stop digital feedback loops from preventing a successful Compilation. The Via block does not prevent or attenuate analog feedback loops (e.g. between a microphone and a nearby speaker), see Feedback Suppressor for that function.

If an audio signal being output from a block can eventually get back to an input of the same block, the software will not allow the signal line to be drawn because of a potential feedback loop. If there is a Via block somewhere in the signal path, the feedback loop will be allowed.

Note that digital feedback loops can cause sudden, loud sounds that can potentially damage loudspeakers or even cause hearing loss. Via blocks should only be used if absolutely necessary, and great care should be taken to prevent the routing of audio in a feedback loop.

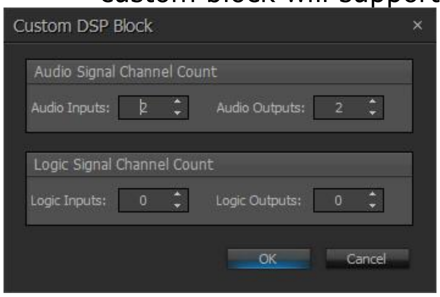
The Via block cannot accept signals directly from Split Pass Through Output or Split Pass Through Input blocks, or signals that are part of a fanout.

The use of Via blocks also disables Delay Equalization for the circuit to which it is connected

Custom Block

Multiple component blocks may be merged into a single custom block. Custom blocks can simplify the design process by encapsulating frequently-used block combinations, allowing them to be collapsed and hidden from view, easily duplicated and reused, and optionally password-protected.

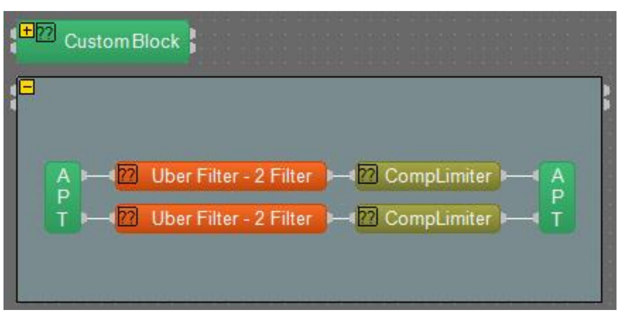
- **Audio Signal Channel Count** determines how many audio inputs and outputs the custom block will support.
- **Logic Signal Channel Count** determines how many logic inputs and outputs the custom block will support.



Custom blocks are initially created in an uncollapsed mode. The inputs and outputs of the custom block will appear as Audio Pass-Through (APT) and/or Logic Pass-Through (LPT) blocks. Insert other component objects into the custom block and wire them to the inputs and outputs. There are four ways to insert component blocks into a custom block:

1. Select the block in the Object Toolbar and then click inside the custom block.
2. Hold down the Shift key while dragging an existing block into the custom block. This will move the existing block into the custom block.
3. Hold down the Ctrl key while dragging an existing block into the custom block. This will insert a copy of the existing block into the custom block.
4. Select one or more blocks, then right-click and select "Create Custom Block". A new custom block will be created and the selected blocks will be inserted into it.

When finished with the internal wiring of the custom block, press the yellow collapse button in the top left corner to collapse the custom block. Press the collapse button again to uncollapse the custom block.



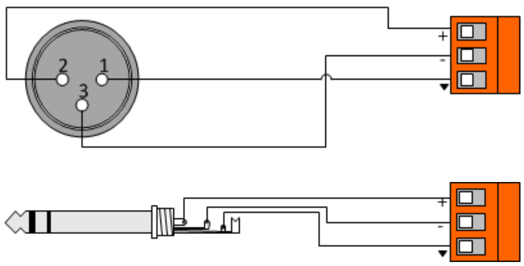
Password Protection

Custom blocks can be password-protected by specifying a password in the Object Property Sheet of the custom block. A password-protected custom block cannot be uncollapsed without first entering the password. Additionally, individual component objects within a password-protected custom block cannot be included in presets until the custom block has been uncollapsed.

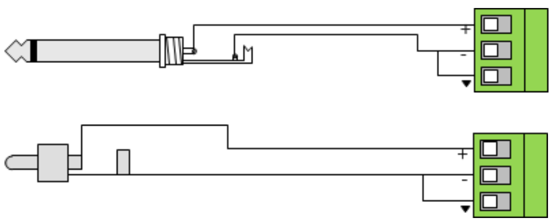
System Design

Audio Wiring Diagrams

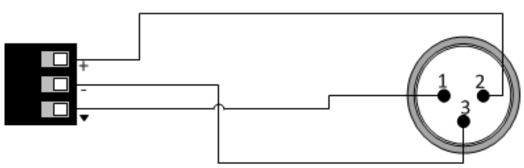
Balanced Input Connection



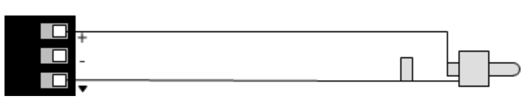
Unbalanced Input Connections



Balanced Output Connection



Unbalanced Output Connection



Compilation and Partitions

Intro

As part of implementing a Tesira system there is a requirement to configure and layout the component objects in order to achieve audio and logic rules to meet a design brief. Once the components have been placed in the layout and wired (connected) together the Tesira compiler is tasked with allocating adequate DSP resources to the proposed configuration. A number of design tools are provided to facilitate this. The following items are reviewed in this section:

- Partitions
- Compilation
- DSP resource
- Hardware Allocation
- Equipment Type

Partitions

Partitions allow a configuration file to be segmented into different sections. Each partition is an independent entity that is displayed on a separate tab in Tesira software, and partitions can even be updated with new configuration files without interrupting the audio that is being processed on other active partitions.
Defining the boundaries between partitions is completely up to the system designer, and there are virtually no restrictions on how the boundaries can be constructed. Common methods for defining partition boundaries include:

- Basing partition boundaries on physical boundaries. For example, different rooms in a facility could occupy their own partitions. This method is particularly useful, since the configuration of each room could be updated without interrupting the audio in other partitions. Additionally, since different physical spaces often don't need to be delay equalized, allocating them to different partitions makes it easier to de-couple different acoustic spaces.
- Using partitions for organizational purposes. For example, all input processing could be located in one partition, mixing and routing in another partition, and output processing in another partition. For large and complicated configurations, this can simplify navigation of the system.
- Basing partition boundaries on hardware boundaries. For example, each Tesira server in a system could occupy its own partition. This method can also help with organization and navigation for systems in which different servers share very few responsibilities with one another.

Restrictions and Limits

There are very few restrictions on how partitions can be applied to a configuration. A single partition may span multiple servers, and each server may contain multiple partitions.
Each Tesira configuration may be divided into a maximum of 32 partitions.

Sharing Audio Between Partitions

All partitions in a configuration can optionally share audio and/or logic signals with one another through the use of Partition Connectors . Partition Connector Transmitters and Receivers can be linked to one another either via the Partition Connector Receiver control dialog or via the System View window. The relationship between different partitions is summarized on the System Overview tab.

As described above, partitions can be updated independently without affecting the audio running in other partitions. However, it is important to be aware that making a change to one partition can occasionally necessitate an update in another partition. This generally occurs in partitions that share audio with one another. The reason this may happen is that partitions that share audio usually need to be delay equalized with each other. Therefore, if a change is made in one partition which requires an update to the delay equalization requirements in another partition, then both partitions will require an update. For this reason, Tesira offers three delay equalization modes to control how independent partitions are by default. See Delay Equalization for a description of these modes.

Once a system design is created (components placed & connected), the system can be compiled by selecting Compilation Options from either the System Menu or the Standard Toolbar. Compile provides system design analysis and calculates DSP processing requirements. Compile also makes initial determinations of quantity/type of Tesira® devices needed, AVB and CobraNet® channel assignments, allocation of DSP resources, and I/O channel number assignments. In addition, Compile will provide indication of system design errors.
A system design file must be compiled before it can be downloaded to Tesira devices.

Compilation

The Compiler is a feature of Tesira software that analyses and validates the layout, calculates I/O and DSP processing requirements and makes an initial determination of the number and type of Tesira hardware needed. It attempts to find the minimum hardware and cost solution that would realize the design. It also attempts to minimize the number of DSP cards and network connections. Additionally, a Compile will provide indication of system design errors via a compilation report in the Output View. Finally, a compile will perform automatic Delay Equalization on all audio paths, according to the Delay Equalization settings found under the General Settings section of the Document Settings control dialog window, which is accessed via the Options item in the Tools menu.

Compiling Single or Multiple Partition Configurations

Compile is initiated from the Compile or Compile All buttons in the Standard Toolbar, or from the Compile Active Partitions, Compile Uncompiled Partitions, Recompile All Partitions or Perform Global Optimization under the Compilation section in the System Menu.

- Compile Active Partitions - compiles only the currently active partition.
- Compile All Uncompiled Partitions- compiles all Uncompiled partitions.
- Recompile All Partitions - Compiles all partitions, even if they have been already compiled but does not optimize I/O.
- Perform Global Optimization - Compiles all uncompiled partitions and analyzes the compilation results to find an equivalent, equipment allocation of a lower cost, if one exists. By default, the compiler finds an optimum equipment list for each partition in the system individually, but the accumulated result of this may not be cost optimal across the entire configuration, particularly in systems having many partitions. Perform Global Optimization may add or remove hardware from the Equipment Table if necessary. Devices in the Equipment Table with serial numbers assigned (physical devices) will not be removed by this optimization. Likewise, devices having DSP objects with fixed allocation will not be removed. If Global Optimization finds an improvement, the equipment table will be updated, and the entire layout will be recompiled. If no improvement is found, the previous compilation results are kept.

Compilation Order of Partitions

Tesira software compiles each partition individually, and partitions to be compiled will be processed in numerical order by default.
Occasionally, compiling partitions in numerical order can lead to undesired results or a compilation failure. For instance, the compiler might fully allocate I/O blocks to a unit in partition one so that there is no room for more I/O blocks in that unit. If subsequent partitions have I/O blocks that are fixed in the same unit, the compilation may fail.
For this reason, Tesira software allows to change the compilation order of partitions. This allows partitions to be compiled in a desired order, and may help to resolve issues like the one described above. It can also be helpful to compile partitions which have large mixer blocks before other partitions, to prevent splitting of large mixer blocks over several DSP cards.

A compilation summary can be viewed in the Output View Toolbar.

DSP resource

Once Compiled the DSP Resources being used by a given layout will be shown in the DSP Resource Tab. The resource shown is based on the DSP cards being used. Depending how the component objects have been assigned to the hardware via the equipment table or via the initialization dialog Equipment type filter will dictate the required DSP resources.
The DSP Resources Tab will give detailed information on the DSP being used in a compiling layout. It will show:

DSP Usage by Partition

The DSP resources Usage by Partition table sums the required resources for each block of a partition if the block would be included in a compile. This calculation is made live as lines are connected/disconnected.

DSP Usage by Device

The DSP resources chart describes usage per device. If some partitions are not compiled the Usage by Device chart will only show information for the partitions that have been compiled. When the compiled partition set is altered (line added/removed), the DSP resource allocation for that partition is removed until the next compilation.

Hardware DSP Chip resources

- A TesiraFORTÉ device uses 1 of 1 DSP chips. (1.00 resources per device)
- A Server IO has up to 3 DSP-2 Cards indicated as 1-3 of 3 used (shown as 1/3 DSP-2 to a maximum of 3/3 used)
- A Server has up to 8 DSP-2 Cards indicated as 1-8 of 8 used (shown as 1/8 DSP-2 to a maximum of 8/8 used)
- Tesira EXMOD, EX-In, EX-Out, EX-IO do not have any DSP resources and are not included in any DSP resource calculation.

Context menu DSP resource

When selecting one or many component objects in the layout, the right-click context menu has a DSP Resource indicator which provides indication of DSP resources required. THe component object is only required to be placed in the layout, and does not need to be part of a valid compilation.

Delay Equalization and DSP resource

Delay Equalization maintains synchronization between signals that have different propagation times. As the function of Delay Equalization is to insert Delay in the audio path it can impact the DSP resources available to the compiler. For very complex designs, there are some adjustments that can be made to optimize the Delay Equalization and DSP resources. Most compilation will succeed without requiring any user adjustments. Please review the Delay Equalization section for more details.

DSP Usage variations

If using the same system layout file on different PC's the DSP allowance slider in Document Settings will stay with the file and display the same on both machines.
There will also be variations of DSP usage if the exact same software version is not used. The DSP usage may also change between two versions of the software. If a previous layout was compiled in a different version of software the DSP algorithms used for blocks may have changed slightly.

In some scenarios, there may be slight differences in the DSP usage totals of the partition and device summary on the DSP Resource Tab. For example - component objects such as large mixers may need to be broken across several DSP's in a DSP-2 card. The partition table doesn't factor this in, but a full compile does, so there might be a slight discrepancy in this particular case.

Hardware Allocation

The compiler will allocate I/O blocks to Tesira hardware according to the type of input or output and the Equipment Type setting (for applicable I/O blocks). If an I/O block cannot be placed into any available hardware unit in the Equipment Table, the Compiler may add devices to the Equipment List in accordance with the Device Selection settings in the Application Settings>Compile Options control dialog window, which is accessed via the Options item in the Tools menu. If Smart Select is turned off, the user can specify what type of device the compiler may add to the equipment list to accommodate the required I/O or DSP processing.

I/O Blocks and DSP objects can be manually allocated to units in the Equipment Table by using the **Allocated To Unit** and **Fixed In Unit** settings in the DSP Properties tab in the Properties window of each block. If an allocation is made and Fixed In Unit is set to True, the compiler will attempt to use that allocation in the compilation process. The programmer may wish to do this if certain pre-determined I/O allocations are desired.

Equipment Type

The Initialization dialog for processing blocks that can be added to configurable devices.

Equipment type	Details
Autoconfigure	Will add a device to the equipment table that achieves the functions based on lowest cost.
Server	Will add the I/O block to a Tesira Server Class device
Rack-Mount Expander	Will add the I/O block to a Rack-Mount Expander class device
Remote Expander	Will add the I/O block to a Expander class device

Forté	Will add the I/O block to a TesiraFORTÉ class device
Lab.gruppen Amplifier	Will add the Input block to a Lab.gruppen Amplifier class device

The Equipment type available will adjust depending on the channel size or type of the block. For example a Lab.gruppen Amplifier will only support an Input block up to 4 channels.

Device and channel Assignment

Once a unit allocation has been made for each I/O block, it will be assigned a Device I/O port, which is the physical connector on the hardware device that corresponds to each software I/O object. If Fixed In Unit is set to **True**, the Device I/O port can be user specified in DSP Properties in the Object Properties window. The port is specified by its slot number first, then by the channel on the card in that slot. For example, a Device I/O of 4.2 would be a port located at channel 2 of the I/O card in slot 4.

Tesira SERVER, SERVER I/O and TesiraFORTÉ devices are viewed by the Compiler as either a:

- **Logical device** : A device specified in the Tesira equipment table that has no serial number allocated. Software can add and remove cards depending on the design and layout.
- **Physical device**: A device specified in the Tesira equipment table that has a serial number allocated. This is a true hardware device so the software must match the physical I/O available.

Due to the complexities and scale of Tesira the compiler will only guarantee **not** to move channel I/O around if you have a physical unit (I.E. assigned S/N in the equipment table) and have 'fixed in unit' on the software I/O blocks set to **True.**
The compiler rules may have been adjusted between software and firmware version.
This means that after a firmware update a recompile may be required. It may also mean that if a Physical device is not specified there might be a change to I/O channel assignments.
If you have logical devices or the physical device I/O is not fixed in unit, the compiler may reallocate I/O channels every time you **compile** or **recompile all**. The recompile all invokes compilation of each partition in a specified order and will un-compile and recalculate. This may mean that if you reorder partitions or add I/O to certain partitions the channel I/O numbers will likely change / move. So in Tesira, in order to never have the I/O change the software must have a physical device and the I/O must be fixed in unit.
The compiler will place cards into a configurable chassis (SERVER, SERVER I/O and EXMOD) in the same order specified in the Biamp order form (or as it is done in the field). If these orders always coincide, the substitution of a physical device for a logical one will never shift any cards and therefore the compilation will stay valid. If some additional cards are discovered in the physical device, the compilation will stay valid.
While the card order for the compilation process is irrelevant as the compiler can work with cards placed in any order, It is strongly recommend that any I/O cards should be in the factory specified order.
The Server I/O will support cards in different slots. The cards themselves will not work if placed in an inappropriate slot; the compiler will only allow the correct card in the appropriate slot.

Slot .	Server IO Card Type
14	SNC-1
13	AVB-1 DAN-1,or SCM-1

12	DAN-1,SCM-1 or card from below
11	DAN-1,SCM-1 or card from below
10	SVC-2 or card from below
9	STC-2 or card from below
8	SOC-4 or card from below
7	
6	
5	
4	
3	SNC-4 or card from below
2	SEC-4 or card from below
1	SIC-4 or card from below

Note: The Server I/O has a hard limit of a maximum of

- 3 audio Network cards with 2 cards of the same type (AVB-1, SCM-1 and DAN-1) See Supported Network Topologies for more details.
- 6x Telephony cards (Mix of STC-2 and SVC-2)
- 12x SIC-1, SEC-4, SNC-4 or SOC-4

Component Objects

Component Object Properties

Component Object Properties are the control settings available for each component. These represent the same settings normally found on equivalent analog audio products. Component Object Properties can be copied & pasted (as DSP Data) between components of the same type, using the Edit Menu. However, once a Component Object is placed into the Layout, all available settings can be accessed by double-clicking over the object. This produces a Control Dialog Box, which displays the component controls in a more conventional user interface.

There are twelve Component Object categories, with some categories providing several component variations. Each component type will have a unique Control Dialog Box. For more information on Control Dialog Boxes for specific components, select the desired category from the list below.

Input and Output Components
Mixer Components
Equalizer Components
Filter Components
Crossover Components
Dynamic Components
Router Components
Delay Components
Control Components
Meter Components
Generator Components
Logic Components
Diagnostic Components
Specialty Components

Component objects used in a compiled system can also be controlled via external control systems. Please refer to the System Control section for more details

Customizing Component Objects

Component Objects can be customized in several ways. First, when using the Object Toolbar to place components into the Layout, certain components will present the user with a pop-up 'Initialization Properties' window of configuration options. Several pre-defined configuration (input/output) options may be available for the component, as well as a 'custom' selection that allows the user to more specifically configure the component. Some components (such as Auto Mixers) allow activation of additional features (such as Logic Inputs & Direct Outputs). Multiple channels of Level, Invert, & Mute may be 'ganged' onto a single control. Multi-channel Invert and Mute Buttons may be set for 'horizontal alignment' and/or 'no labeling' when minimized as a user control. The configuration of certain components can be further edited, even after they have been placed in the Layout and the design has been compiled.

Actual component settings can be customized through Control Dialog Boxes (see Component Object Properties).
Control Dialog Boxes for certain Combiners, Dynamics, Controls, Meters, & Telephone Interface components can be minimized to create customized control surfaces (room

combiners, meters, level controls, mute buttons, preset buttons, telephone dialers, etc.). These control surfaces can then be made user accessible, with other system components hidden on Layers which are invisible to the user, and which are Password protected. (See Layout Property Sheet.)

Multiple component objects may be merged together using the Custom Block specialty component. Custom Blocks can simplify the design process by integrating frequently used component combinations, and can provide password protection for intellectual property such as unique processing and component settings.

Live Control of Component objects

Component objects have a control dialog to allow real time adjustment of any audio controls. To open the control dialog, right click the object to open the context menu and select Control Dialog, or double click the component object in the layout.

Min Max volume range

Most component objects that use a volume control also allow a minimum (min) and maximum (max) setting to be specified. When offline this can be adjusted in the DSP properties of the Object Property Sheet. The control dialog will also have text boxes that allow user adjustment, on some dialogs such as output blocks or mixers, there is an 'expand/collapse' button that reveals more controls.

The following blocks support this functionality:

- **Input and Output:** Analog Input, Analog Output, CobraNet Input, CobraNet Output, Dante Input, Dante Output, USB Input and Output, AEC Input (processing Block), Telephone Interface Receive and Transmit, VoIP Phone Receive and Transmit
- **Crossovers:** Crossover Input and Output
- **Mixers:** Gating Auto Mixer, Gain Sharing Auto Mixer, Standard Mixer, Matrix Mixer, Room Combiner
- **Equalizers:** Parametric Equalizer, Graphic Equalizer
- **Dynamics:** Ducker (Input)
- **Router:** Source Selector
- **Controls:** Level
- **Generators:** Tone Generator, Noise Generator

Minimizing Control Dialogs

To aid system control during commissioning certain DSP object will show a compact control dialog when the Minimize icon on the top right of the control dialog is selected. Processing objects that support this functionality include: Level Mute Preset Button Signal Present Meter Audio Meter Dialer

Fader showing Normal and Minimized View:

Double clicking the original control block in the layout will cause a minimized control dialog to display a highlight surround. Double clicking the grey area or using the right click context menu will allow the dialog to be restored.

Mixer, Router and Matrix Cross-points

When a Mixer, Router or Crosspoint dialog is used moving the mouse over the crosspoints will highlight the current cross point location. The labels cell will change color following the mouse location and tooltip will appear to show the channel Input and Output name.

Delay Equalization

Delay Equalization

Some signal paths through a Tesira system can take longer to process than others. Tesira includes a mechanism called Delay Equalization, which maintains synchronization between signals that have different propagation times.
For example, consider a simplified Tesira system with two inputs and two outputs. The signal from the first input passes through many processing blocks and several AVB hops before it gets to the first output. The signal from the second output travels directly to the second output without going through any processing blocks or AVB hops. Clearly, the signal from the first input will take longer to process than the signal from the second input.

Without delay equalization, the signal from the second input would be heard slightly before the signal from the first input. This can be undesirable in many cases, particularly if both outputs can be heard simultaneously in the same acoustic space. Delay equalization solves this problem by inserting a small delay on the second input signal such that it reaches the output at the same time as the first input signal.

Delay Equalization Groups

Most input and output blocks are assigned to a Delay Equalization Group. Each I/O block's Delay Equalization Group can be found in its Property Sheet. Each I/O block is initially assigned to the Default group, but can also be assigned to the None group (meaning it will be excluded from delay equalization calculations) or a user-defined group.

Signals originating from input blocks that are in the same group will be synchronized up until the point they encounter a block with multiple inputs (e.g., mixers, routers, duckers, room combiners). Signals which feed output blocks that are in the same group are assumed to feed the same acoustic space, and will be synchronized from when they output a mixer/router block until they reach the output blocks.

Assigning an input block to the same group as an output block is allowed, but internally they will be treated as different groups. Input signals are only synchronized with other input signals, and output signals are only synchronized with other output signals.

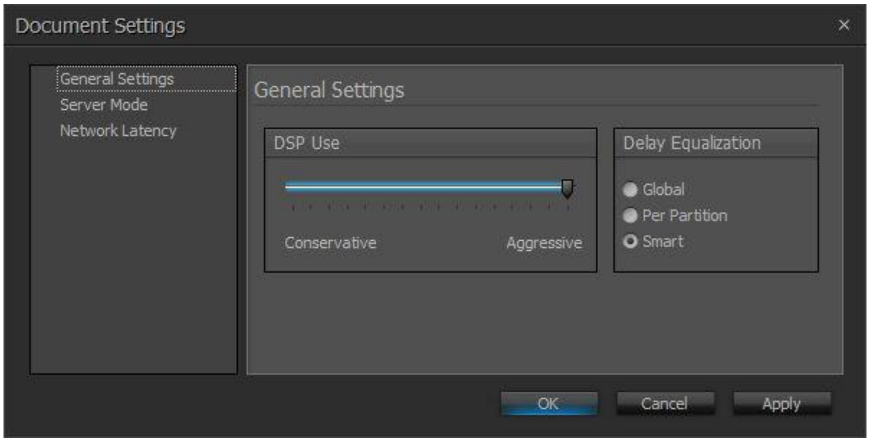
User-defined Delay Equalization Groups can be added in the Delay Equalization Groups dialog. Once a user-defined group is created, any I/O block can be assigned to it via its Property Sheet.

Delay Equalization Modes

Tesira software can operate in one of three Delay Equalization Modes, which define how Delay Equalization Groups in different partitions interact with one another.

In some cases, making changes to one partition can affect the delay equalization requirements of another (otherwise unchanged) partition. In this case, both partitions will need to be updated when the configuration is sent, which means that audio will be interrupted in both partitions. A warning will be displayed during compilation if this situation arises. Managing Delay Equalization Groups and/or selecting the appropriate Delay Equalization Mode can prevent this situation from occurring when it is undesirable.

The Delay Equalization Mode for a configuration file can be selected in the Document Settings dialog.



Delay Equalization Modes only affect the behavior of I/O blocks that are assigned to the Default Delay Equalization Group; I/O blocks that are assigned to user-defined Delay Equalization Groups will always be synchronized, regardless of the currently selected mode. Similarly, I/O blocks that are assigned to the "None" group will not be synchronized in any mode. There are three modes: Global, Per Partition, and Smart.

Global Mode

The Global Delay Equalization Mode ensures that all I/O blocks in the same Delay Equalization Group (except the "None" group) are synchronized, even if they are in different partitions.

Per Partition Mode

The Per Partition Delay Equalization Mode ensures that signals in different partitions are not synchronized. In this mode, I/O blocks in different partitions will not be synchronized with one another when they are all assigned to the Default group. In other words, this mode only allows delay equalization to operate within single partitions. The only exception is for blocks that are assigned to user-defined Delay EQ groups, which will be synchronized in this mode.

Smart Mode

The Smart Delay Equalization Mode is the middle ground between the Global and Per Partition modes. In this mode, when I/O blocks in different partitions are in the Default group, they will be synchronized only if the signals associated with those I/O blocks are connected to a common mixer/router block. If the I/O blocks don't interact with one another, they will not be synchronized. Note that this mode only affects blocks in the Default group; blocks in the same user-defined group will always be synchronized, regardless of whether they interact with one another.

Design - Lines and Labeling

There are many areas of designing a system file where labelling and drawing lines is essential. A number of useful features are available to system designers:

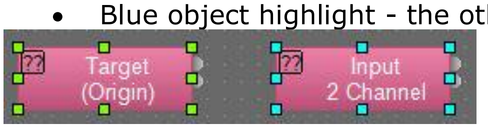
Selecting component Objects

A single component object or line can be selected with a left mouse click in the layout and will show as highlighted.

Target object

When more than one block object is selected one of the units will be assigned as the target is the block that is always the origin for the lines. This target object is the unit referenced for alignment or wire drawing tasks.

- Green object highlight - the target object.
- Blue object highlight - the other objects that are selected.

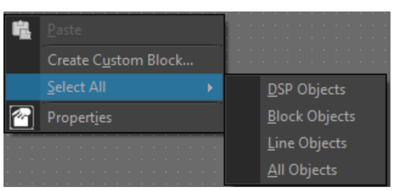


Selecting Multiple Component Objects

This can be done by

- Left click and drag over the required objects. a bounding box will appear - all items (objects and lines) will be selected.

- Ctrl + Left Mouse Click - Will allow selection of one or more objects.
- Right click the layout. A Surface Context Menu is shown - the option to Select all DSP, Block, line or all objects is available.



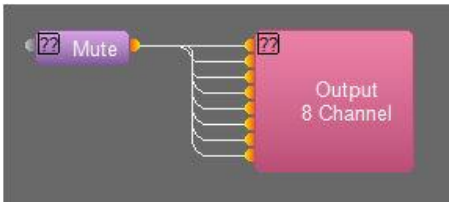
Lines

Lines represent the flow of audio signals or logic signals through the layout. They propagate from system input to system output, or from logic sources to logic receivers.
Inputs can receive only one line apiece. Lines can fan out (aka "Y" or "wye") from a single output to multiple inputs on one or more blocks.

To connect the ports of 2 component blocks, click on one or more ports of the first block and drag to the ports of the second, then release.

Ports

Ports on component blocks are large and easy to select. Each port inhabits a 10x10 pixel space making grabbing connectors painless. Ports “light up” when selected, ports which are receiving line connections will "light up" as they are connected.

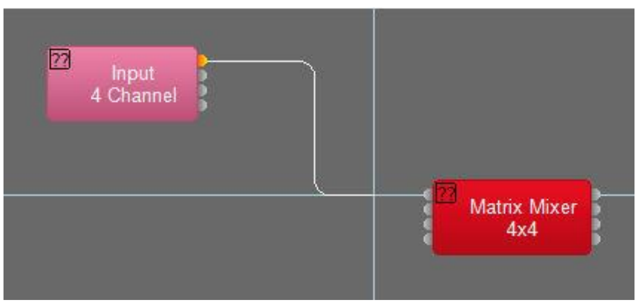


Crosshairs

When connecting components, crosshairs make it easy to align your lines with ports. "Display crosshairs" is enabled in **Tools > Options > Application Settings... > Display > Lines**.

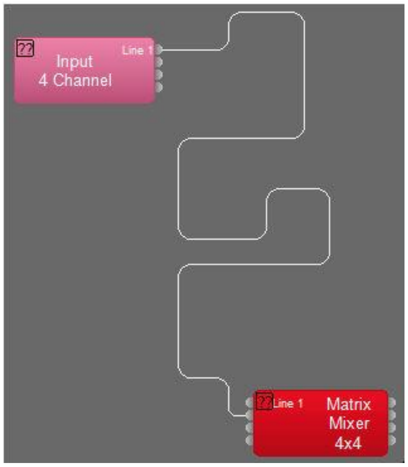
Magnification

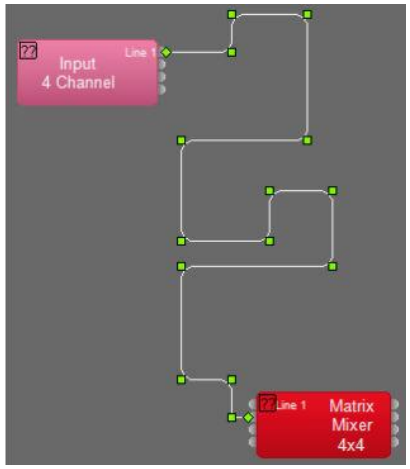
To zoom in and out *while drawing a line* use the + and - keys or scrolling the mouse wheel. When not drawing a line you can zoom in and out by holding the *Ctrl* key and scrolling the mouse wheel.



Bends

While drawing a line, clicking will place a flex point on the line. This allows you to manually determine the line path. If you don't intervene, Tesira will automatically place the turns for you when you drag from one component block port to another.

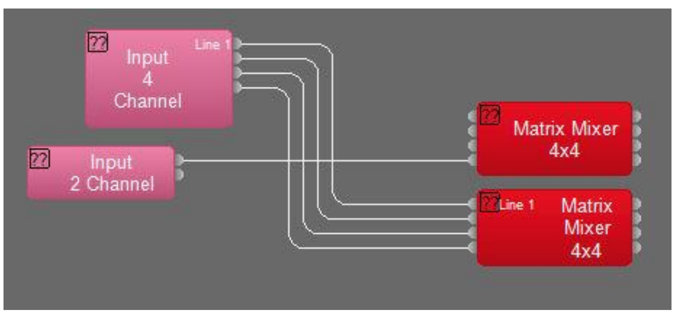




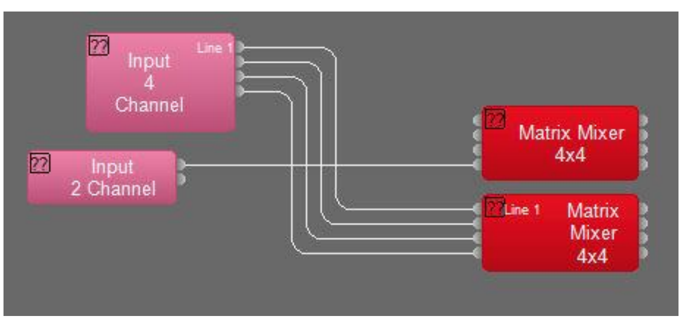
Add bump-out or flex point

Adding a flex point to an existing line created in Orthogonal line mode is now enhanced with the automatic creation of a "line jumping" bump-out bend in the line.

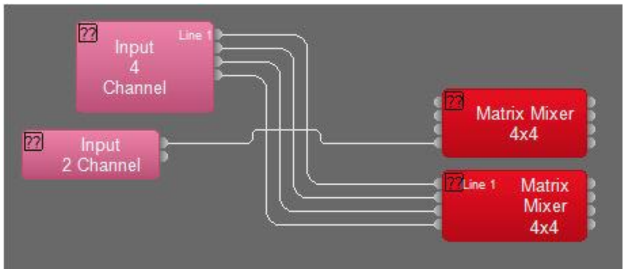
- The bump-out is created with just a single click on the line while holding the *Alt* key.
- To remove a bump-out, grab and drag the bump-out to realign it with the line. The line will "heal" itself.
- On a non-Orthogonal lines, the old behavior of adding a single flex point on the line is retained.



- Hold the *Alt* key and left-click on the line where you'd like to add a bump-out.



- A bump-out is created.

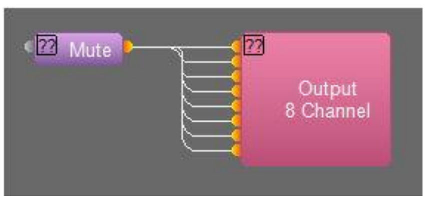


- Click and drag on line segments to position and size the bump-out to add visual clarity to the diagram.

Fan out

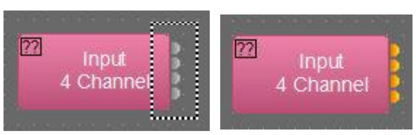
To connect one output to multiple inputs on a single block, click and hold the *Shift* key while drawing the line. As you approach a block with multiple inputs the line will fan out and automatically connect to all of the available ports.

To move the fan out lines as a group, hold the *Shift* key while you click on the lines and drag them to the desired location, place the cursor over a point which has all of the line segments of interest for the move.



Draw multiple lines.

- Left click and drag over the required ports. a bounding box will appear - all ports will be selected (highlighted orange.



- Select one of the ports. A hand will appear and the port will go white.

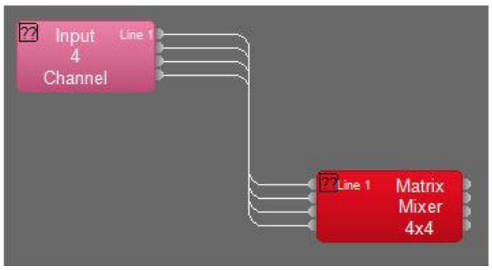
- Left click once and move the mouse to draw lines.

Parallel Bends and Folded Bends

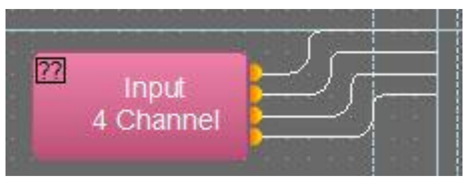
Tesira software defaults to drawing lines in parallel on the horizontal and stacked on the vertical axes.

It is easy to draw lines with parallel paths using the *right* and *left arrow keys* while drawing lines.

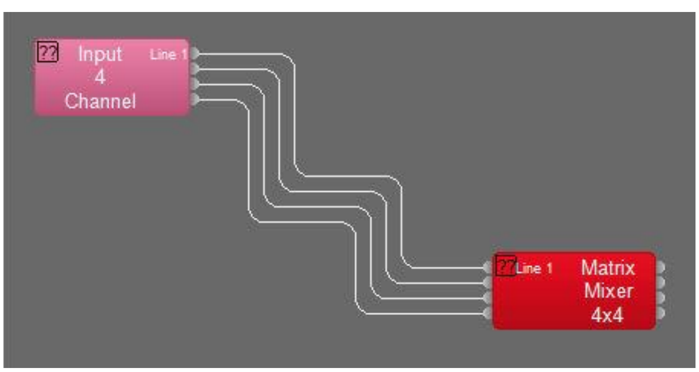
In **Tools > Options > Application Settings... > Display > Lines** there is an option called "Collapsed Line Spacing", the value here determines the spacing between the lines when the arrow keys are used. This works for both audio and logic lines. The left and right arrow keys controls the direction of spread.



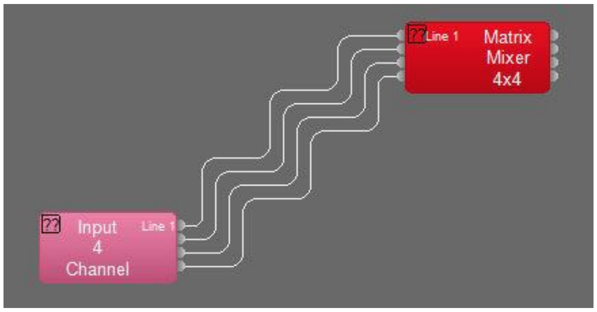
While the lines are being drawn the left and right arrows will offset the lines for clearer horizontal or vertical separation. the Application Settings **Collapsed Line Spacing** determines the horizontal separation of vertical lines in Orthogonal line drawing mode when the left arrow key is held



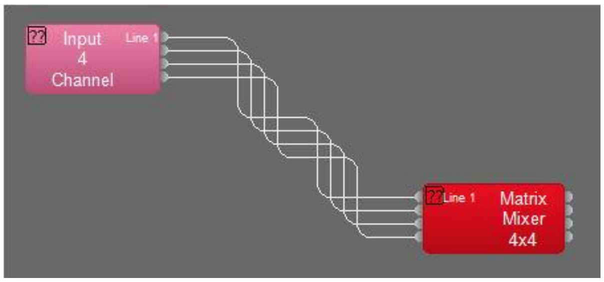
Standard lines are drawn in parallel on the horizontal and stacked on the vertical axes.



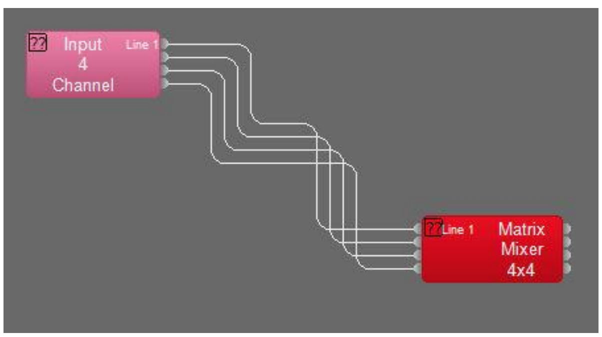
Use the right arrow key for parallel descending lines.



Use the left arrow key for parallel ascending lines.



Using the left arrow key with descending lines produces folds.



In combination: the right arrow key used for 2 descending bends, then the left arrow key for the 3rd bend (the 4th bend can only fold as shown).

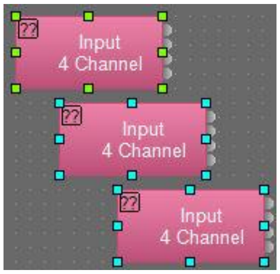
Target object

When more than one object is selected one of the units will be assigned as the target is the block that is always the origin for the lines. This target object is the unit referenced for alignment or wire drawing tasks.

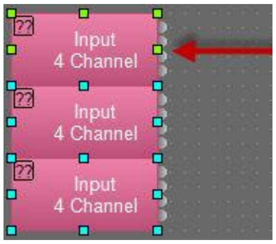
- Green object highlight - the target object.
- Blue object highlight - the other objects that are selected.

Aligning and packing blocks

- Highlight three input blocks:



- Pack left the blocks. The top block is used as the target:



Connect audio lines

Connect Audio *and* Connect Logic

In addition to the ability to select multiple ports and pull lines manually to the next block or group of blocks, you can opt to select two or more component blocks and use the right-click "Connect Audio" wiring command. The keyboard shortcut is *Ctrl+RightArrow*.

"Connect Logic" will give similar behavior, specific to logic ports on devices. The keyboard shortcut is *Ctrl+UpArrow*.

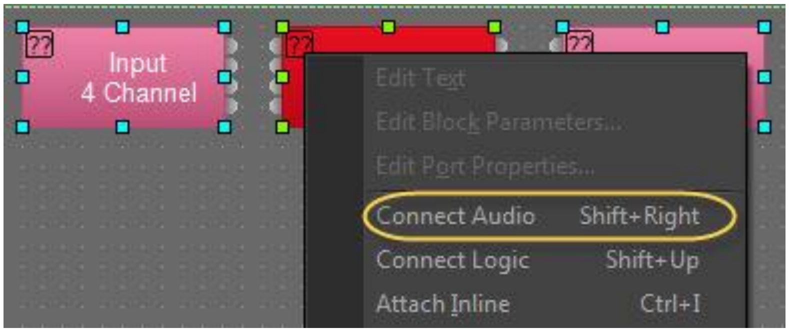
Lines can originate from either direction. Select the blocks to be connected. Note that the block with *green* handles is always the *origin* for the lines. Holding the *Alt* key while clicking on other selected blocks will allow you to choose which block has the green handles. Right-click and select "Connect Audio" or "Connect Logic" to connect the ports.

When designing, it is possible to have connection lines draw automatically to and from a target object.

- Add audio objects

- Select objects so the middle block is the target

- Right Click and select 'Connect Audio' (or shortcut key 'Shift+Right arrow)

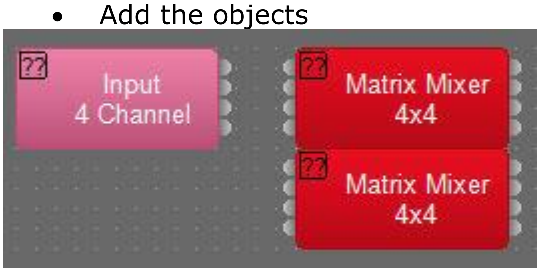


- Lines are drawn between the three objects.

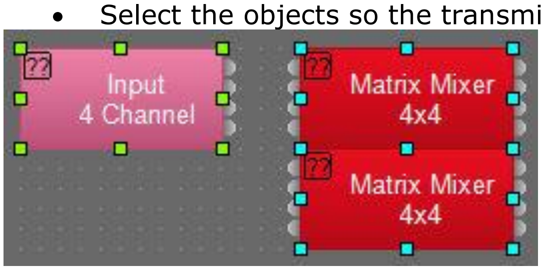
Fan from transmitting object

If it is desired that one sending object fans out to multiple receivers

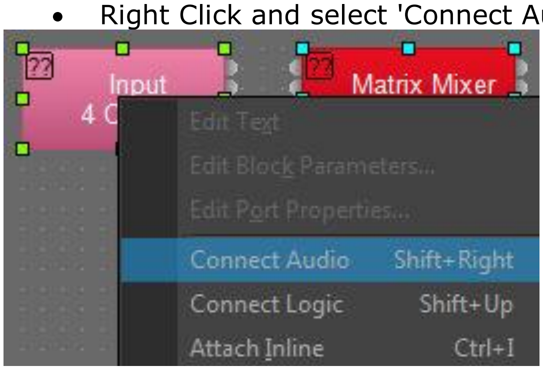
- Add the objects



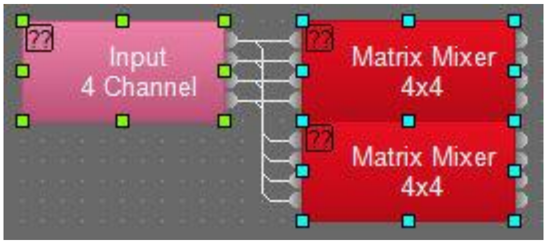
- Select the objects so the transmitter (the input- 4 channel in this case) is the target



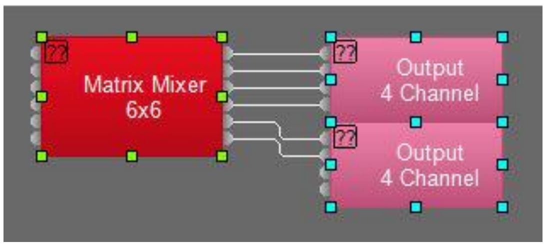
- Right Click and select 'Connect Audio' (or shortcut key 'Shift+Right arrow)

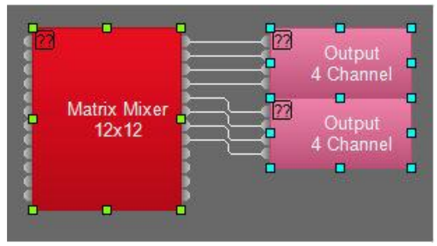


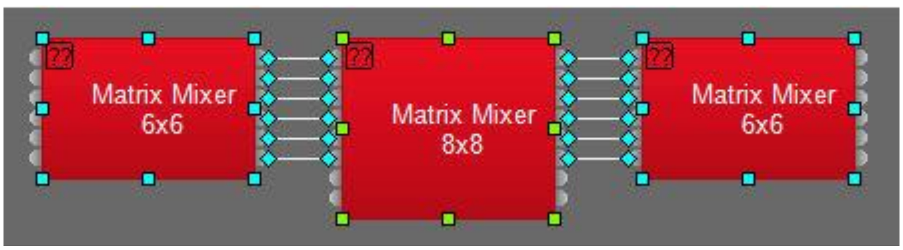
- The Target object will connect to the upstream objects.



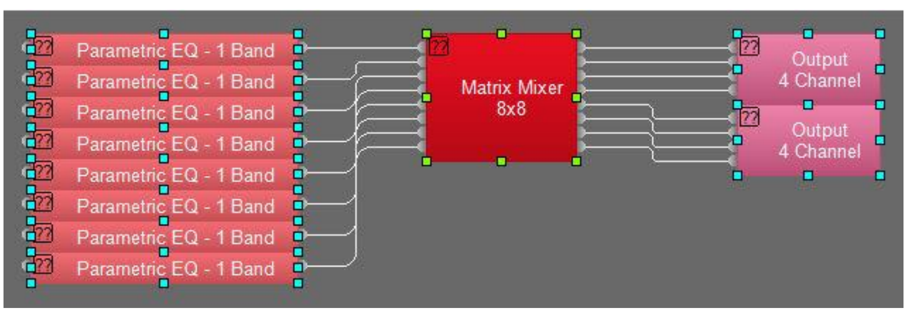
Connect to one or more blocks with matching number of ports.



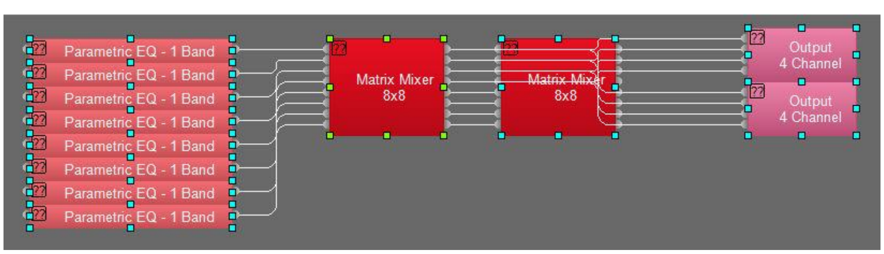




When connecting to a mismatched number of ports it will make a "best attempt" to connect to available ports.

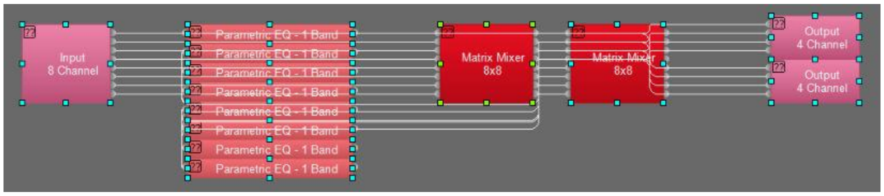


Connect components to input and output ports simultaneously.



Note: If you attempt to connect across more than 1 component in either direction it will assume you are Y-ing the output side of the green handled object. This may have unintended consequences.

- Here the matrix mixer has connected to the (8) parametric EQ blocks on the input ports, and Y-ed its outputs ports to the 2nd matrix mixer and to the (2) 4-channel output blocks.



Note: If you attempt to connect across more than 1 component in either direction it will assume you are Y-ing the output side of the green handled object. This may have unintended consequences.

- Here the matrix mixer has connected to the 8-channel input block on the input ports, and Y-ed its outputs ports to the 2nd matrix mixer, the (2) 4-channel output blocks, and back over to the (8) parametric EQs.

Attach Objects inline.

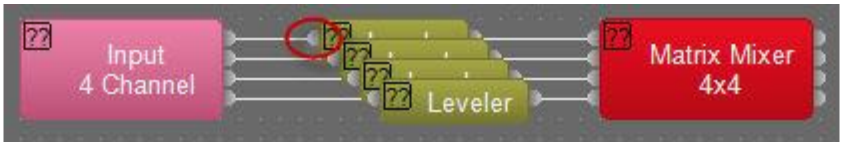
Easily insert a component on an existing line by placing the new component's ports on the line and selecting the right-click option "*Attach Inline*" or the keyboard shortcut *Ctrl+i*. The ports of the component must overlap the line for the *Attach Inline* feature to work.

This feature applies to both audio and logic components. Audio lines can only connect to audio ports, and logic lines to logic ports.

It is possible to select any number of blocks before attaching.

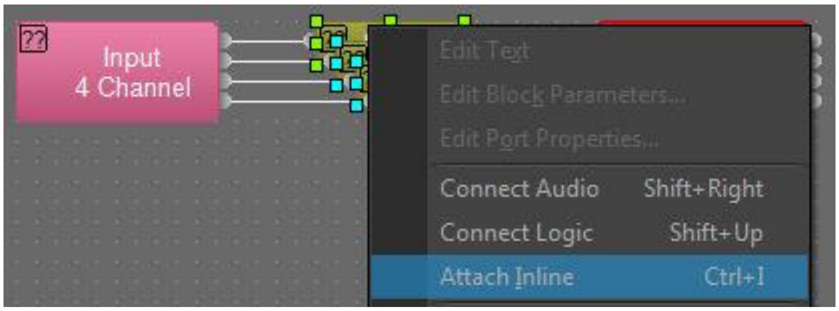
- Make enough space between objects to add additional processing objects.

- Place Additional processing objects. Confirm the I/O nodes are covering the lines that they are required to be connected to.



- Select the processing objects to be inserted

- Right Click and select 'Attach Inline' (shortcut Ctrl+I)



- Attaching Inline can be done using a single or multiople blocks

Apply line labels to ports

When designing systems labeling can be used to assist in explaining signal flow around a design.

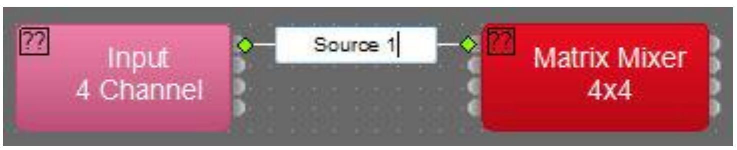
Text applied to lines can now be quickly pushed to the connected ports for easy labeling of connectors.
To add text to lines click on the line and hit the Enter key, or right-click and choose "Edit Text".
Select a line with text and right-click on it with the mouse, choose "Apply line label to ports". Labels will be pushed to the blocks at either end of the selected line segment.

- Multiple lines can be selected simultaneously.
- Lines without labels will propagate blank text to the ports.
- Labels will be retained by the blocks if the line is deleted or renamed

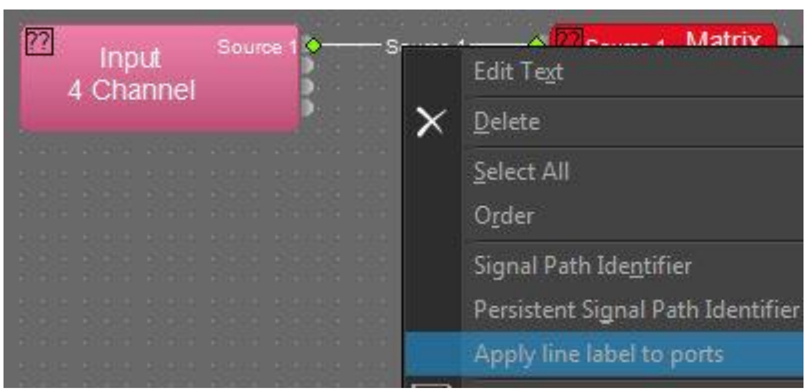
Labeling lines

- Select a wire

- Press the 'Enter' Key - Type the name in the box that appears. Press 'Enter' to apply



- The line label can be applied to the ports using the context menu.



- The Ports and line are now labelled.

- If any items are attached 'Inline' the line labels will persist.

- 

Labeling Ports

- Hover over the port. A 'hand' will appear.

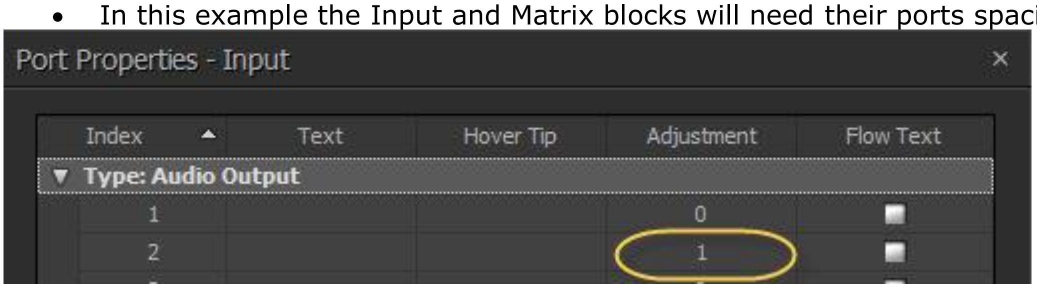
- Press 'Shift+Left Click'. A text box will appear. Label the signal flow.

- The Arrow can be used to enable 'Flow Text' and send the label name to other ports in the signal flow. The Flow Text feature is also accessible via the 'Port Properties' in the context Menu. Matrix and Mixer blocks will not continue the naming.

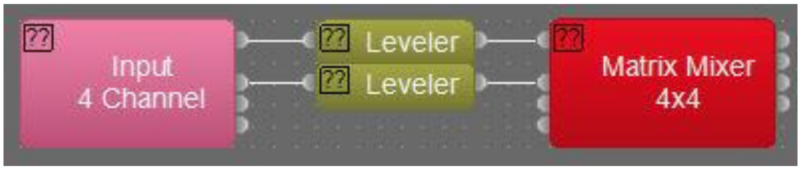
Adjusting Port Spacing

- When adding blocks that do not have offset ports it may be desired that straight lines are used. an Adjustment can be specified in the Port Properties to space the ports further apart.

- In this example the Input and Matrix blocks will need their ports spacing out.



- Adjusting the Input port and Matrix mixer port so they are spaced 1 port apart will result in all port in this signal flow being aligned.



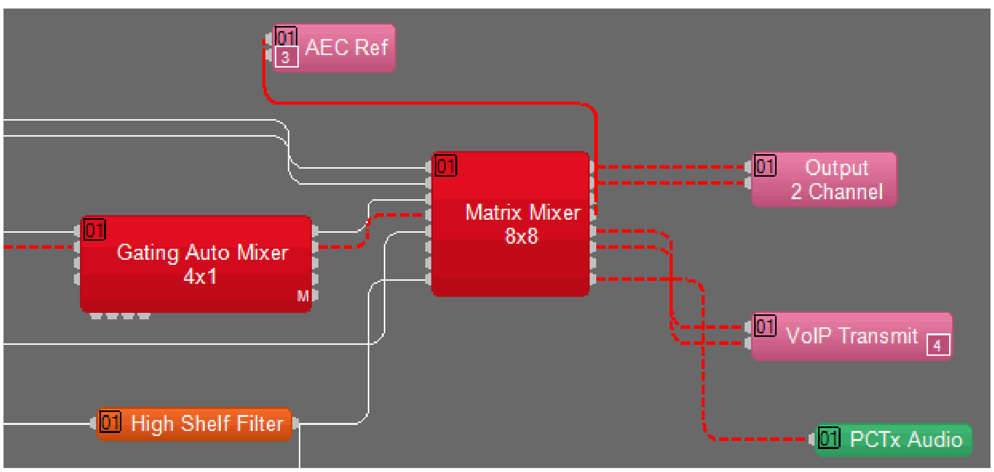
Signal Path Identifier

Provides a temporary color-coded identification of all audio signal paths (Lines) which are associated with a selected Line Object.

Right click the audio line and the context menu will appear. Persistent Signal Path Identifier is available in:

- Normal Mode (follows subsequent line selections)
- Locked Mode (remains on original line selection)
- Off (temporary selection).

If the selected Line Object includes identifying text (see Line Property Sheet), that text will be temporarily imposed on all lines being indicated by Signal Path Identifier**.**



Logic Simulation

Similar to the Persistent Signal Path Identifier, the *Persistent Logic Simulation* allows you to test logic behavior while offline. Right-click on a logic port to activate the simulation mode.

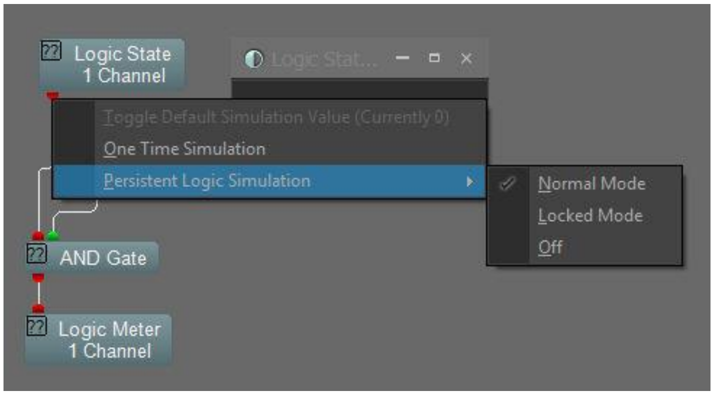
Logic Simulation is available in:

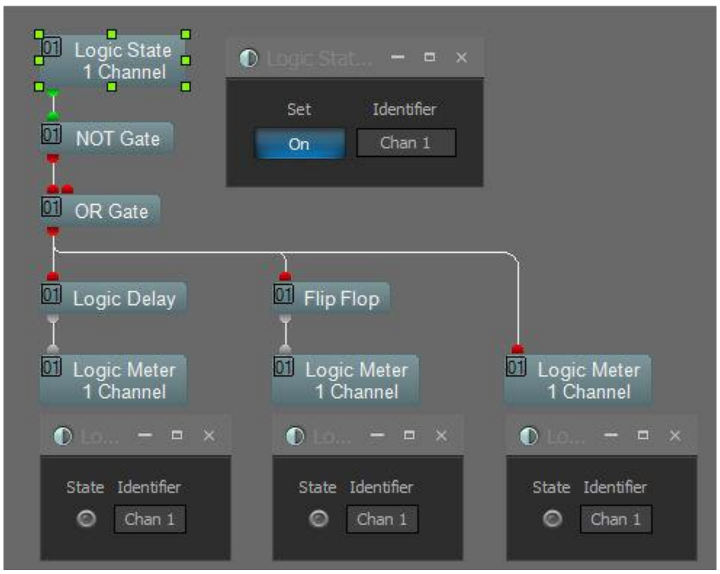
- Normal Mode (follows subsequent port selection, use the Ctrl + Left Click shortcut to select the port)
- Locked Mode (remains on original port selection)
- Off (temporary selection).

Simulation output does not trigger Presets or activate Logic Meters, rather, the logic ports "light up" in red and green to show the behavior of the logic gates in the path.

Logic State controls are used as triggering events for the logic gates in simulation mode.

Logic delays, flip flops and pass-throughs do not pass logic simulation.

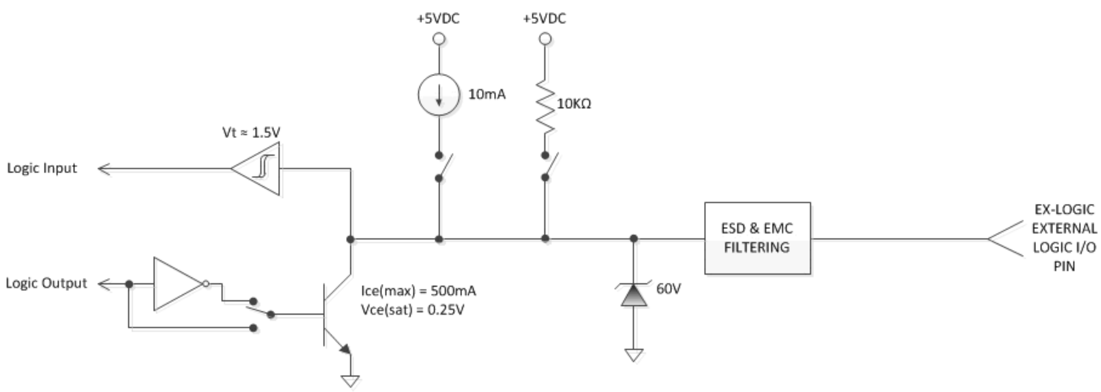




Logic IO Wiring

	500ma transistor (pull down)	10ma current source (pull up)	10k pull up resistor
Logic input	Always off	Always off	Always on
Logic output	Obeys logic state	Always off	Always on
Logic with powered output	Obeys logic state	Always on	Always off
Voltage Control	Always off	Always off	Always off

Internal wiring

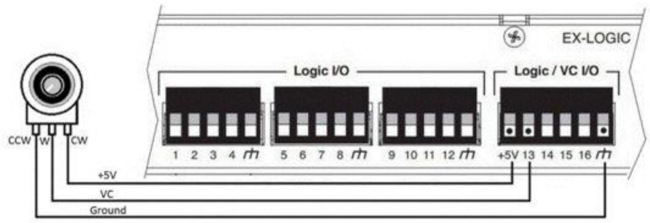


Wiring Volume Control

All potentiometers that are connected to the EX-Logic must be calibrated to ensure proper operation. See the TesiraFORTÉ, Server and SERVER IO Serial port settings EX Logic Serial Port Settings for more information.

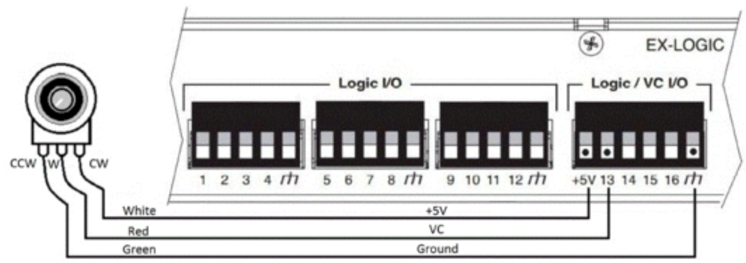
Wiring a generic potentiometer

Wiring a potentiometer to an EX-Logic can be a bit confusing since it has three contact points. Not wiring it properly might cause the potentiometer to either work in reverse or not at all. Looking at the pot from the knob side with the contact points pointing at you, you will have the Low or CCW (Counter Clockwise) terminal on the left, the W (Wiper) terminal in the middle and the High or CW (Clockwise) terminal on the right. Connect the CCW terminal to the Ground terminal, the W terminal to the analog input and the CW terminal to the +5V terminal of the EX-Logic as shown in the diagram below.
A Linear-taper potentiometer in the 5K – 50K range is recommended. If a logrithmic (audio-tapered) potentiometer is used, the control will not seem to work "evenly" across the range. The inputs do have noise filtering but for longer cable runs, lower-value potentiometers in the 5K-10K range will have greater noise immunity.

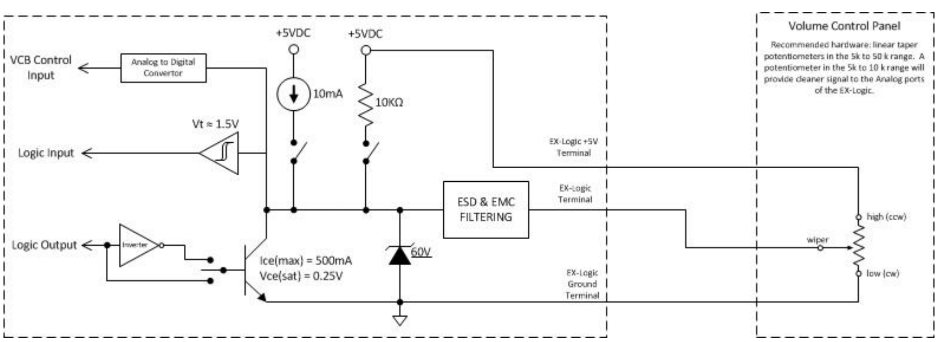


Wiring the RP-L1 or RP-L2

The Biamp RP-L1 and RP-L2 are wall plates with 25KΩ linear taper potentiometers for volume control. The RP-L1 has only one potentiometer while the RP-L2 has two. These potentiometers come prewired from the factory with pigtails for ease of installation. These pigtails are color coded green for Low or CCW (Counter Clockwise) terminal, red for W (Wiper) terminal and white for High or CW (Clockwise) terminal. Connect the CCW (green) wire to the Ground terminal, the W (red) wire to the analog input and the CW (white) wire to the +5V terminal as shown in the diagram below.



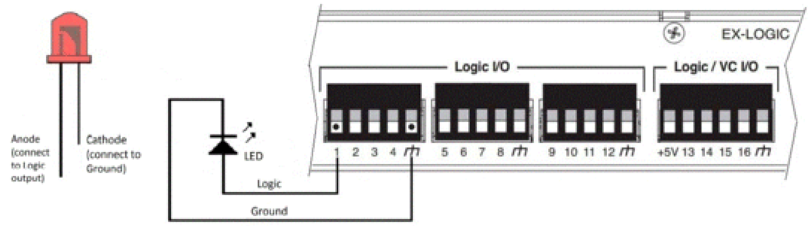
Internal Circuit



Wiring LED

Using EX-LOGIC as a power source

When configured as a logic output, each of the 16 GPIO pins can be configured as a current source to drive an LED directly (5V / 10mA maximum per output). To wire an LED in this mode, connect the LED’s anode (long lead) to the EX-Logic output and the cathode (short lead) to the ground terminal on the EX-logic as shown in the diagram below:
Using 'Enable Powered Output'



When using EX-LOGIC as a power source, the corresponding Logic Output blocks in your Tesira configuration must be created with the "Enable Powered Outputs" option selected.

Using External Voltage Source

Additionally, the GPIO pins can be configured as simple contact closure outputs capable of sinking up to 300mA at 40V. In this mode, an external power supply is required (e.g. to drive a high brightness LED or multiple LED's). **Please note:** When LED's are wired to the EX-Logic using this method, the "Enable Powered Outputs" should remain unchecked in the Logic Output Initialization window in Tesira to avoid damaging the EX-Logic.
To wire an LED to an EX-logic using an external power supply, make the following connections as shown in the diagram below:

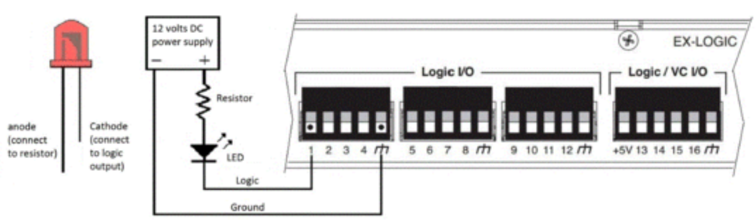
1. Connect the negative terminal of the power supply to the ground terminal on the EX-Logic.
2. Connect the positive terminal of the power supply to the resistor. Resistor value can be calculated using the following formula:

R = V/I

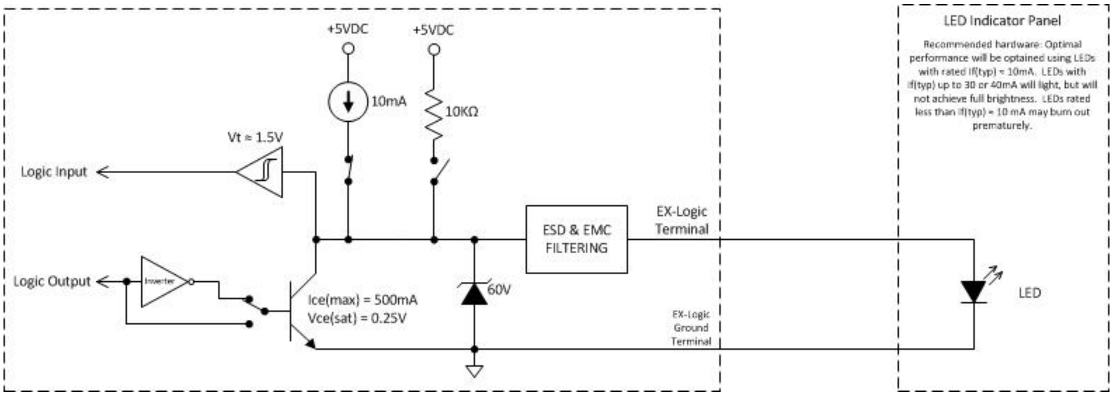
where:

R = value of resistor in kOhms
V = supply voltage – the LED turn on voltage
I = required current in mA

3. Connect the resistor to the LED's Anode (long lead).
4. Connect the LED's Cathode (short lead) to the EX-Logic logic output.



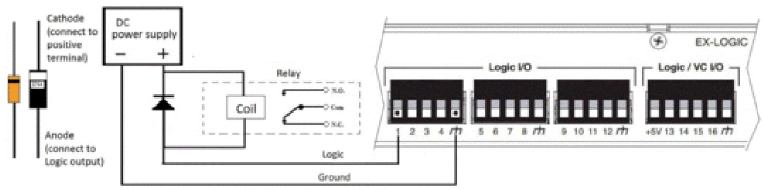
Internal circuit



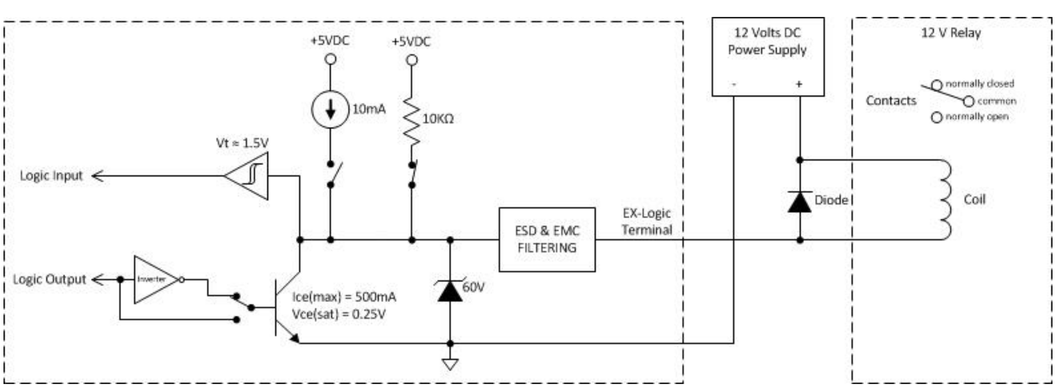
Wiring Relay

Wiring a relay is in essence very similar to wiring an LED. The only difference is that a resistor is not needed but a diode needs to be connected between the terminals of the relay coil to suppress high voltage transients that are generated when the relay turns off (Note that some relays have this diode already built-in). To wire a relay to the EX-Logic, do the following connections as shown in the diagram below:

1. Connect the negative terminal of the power supply to the ground terminal on the EX-Logic
2. Connect the positive terminal of the power supply to one of the relay's coil terminals
3. Connect the other relay coil terminal to the EX-Logic output
4. Connect the diode's Anode to the logic output of the EX-Logic
5. Connect the diode's Cathode to the positive terminal of the power supply



Internal Circuit



Logic Blocks

<table>
<tr><th>Device</th><th>Object</th><th>Logic I/O</th><th>Operation</th></tr>
<tr><td>Logic Input</td><td></td><td>Up to 16 channels</td><td>Open on logic input = Logic 1
Closure on logic input= Logic 0

Optional: Behavior of logic control may be inverted in control dialog.</td></tr>
<tr><td>Logic Output/LED Driver</td><td></td><td>Up to 16 channels</td><td>In Logic mode
Logic 1 on logic output = open
Logic 0 on logic input =closure

In LED Driver mode
Logic 1 on logic output = Up to 24V/500mA
Logic 0 on logic input = Short to ground

Optional: Behavior of logic control may be inverted in control dialog.</td></tr>
<tr><td>Logic Delay</td><td></td><td>1 logic input and 1 logic output per delay, up to 32 per block</td><td>1 on logic input must be present for specified ON time before logi output changes to 1.

0 on logic input must be preset for specified OFF time before logic output changes to 0.</td></tr>
<tr><td>OR Gate</td><td></td><td>2-32 logic inputs and 1 logic output</td><td>1 at either or both logic inputs causes a 1 at logic output. Otherwise, logic output is 0.

Truth table:<table><tr><td>Input 1</td><td>Input 2</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>AND Gate</td><td></td><td>2-32 logic inputs and 1 logic output</td><td>1 at both inputs causes a 1 at logic output. Otherwise, logic output is 0.

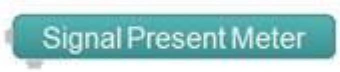
Truth table:<table><tr><td>Input 1</td><td>Input 2</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>NOT Gate</td><td>
</td><td>1 logic input and 1 logic output per gate, up to 32 per block</td><td>1 at logic input causes a 0 at logic output.
0 at logic input causes a 1 at logic output.

Truth table:<table><tr><td>Input 1</td><td>Input 2</td></tr><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table></td></tr>
<tr><td>NOR Gate</td><td>
</td><td>2-32 logic inputs and 1 logic output</td><td>0 at both logic inputs causes a 1 at logic output. Otherwise, logic output is 0. (Logically the same as an OR gate followed by a NOT gate.)

Truth table:<table><tr><td>Input 1</td><td>Input 2</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>NAND Gate</td><td>
</td><td>2-32 logic inputs and 1 logic output</td><td>1 at both logic inputs causes a 0 at logic output. Otherwise, logic output is 1.
(Logically the same as an AND gate followed by a NOT gate.)

Truth table:<table><tr><td>Input 1</td><td>Input 2</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>XOR Gate</td><td>
</td><td>2-32 logic inputs and 1 logic output</td><td>1 at either but not both logic inputs causes a 1 at logic output. Otherwise, logic output is 0.
Truth table:<table><tr><td>Input 1</td><td>Input 2</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

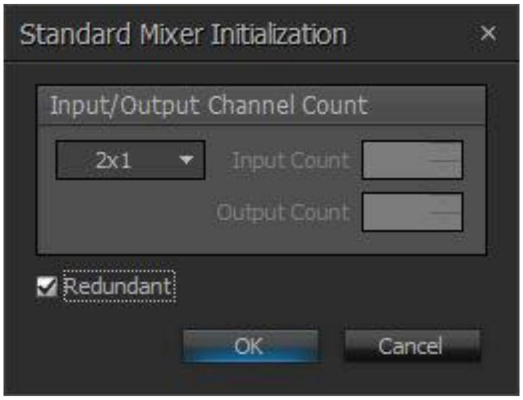
			0 1 1 1 0 1 1 1 0
Flip Flop Gate		1 logic input and 1 logic output per gate, up to 32 per block	0 to 1 transition at logic input causes logic output to toggle from its present state to the opposite state.
Logic Meter		1 logic input per meter, up to 32 per block	Logic 1 on input node turns corresponding indicator on Logic 0 on input node turns corresponding indicator off
Logic State		1 logic output per state, up to 32 per block	Setting a State On causes the corresponding logic output to be 1. Setting the State Off causes the corresponding logic output to be 0
Fan-In OR Pulse		2-32 logic inputs and 1 logic output	Transition from logic 0 to logic 1 on any of the inputs generates a 150ms logic pulse on the output. A de-bounce delay of 300ms is used after the initial pulse
Signal Present Meter		1 logic input per meter, up to 32 per block	Audio signal exceeding threshold on sense input causes logic output to be 1. Audio signal below threshold causes logic output to be 0.

Redundancy

TESIRA SERVER hardware devices support redundancy. Each SERVER pair must be configured with identical hardware including IO cards and DSP cards. The Tesira software must be operating in the **Tesira Server** Only or **Both Tesira Servers and TesiraFORTÉ** Software mode. Redundancy is not supported in **TesiraFORTÉ only** mode.

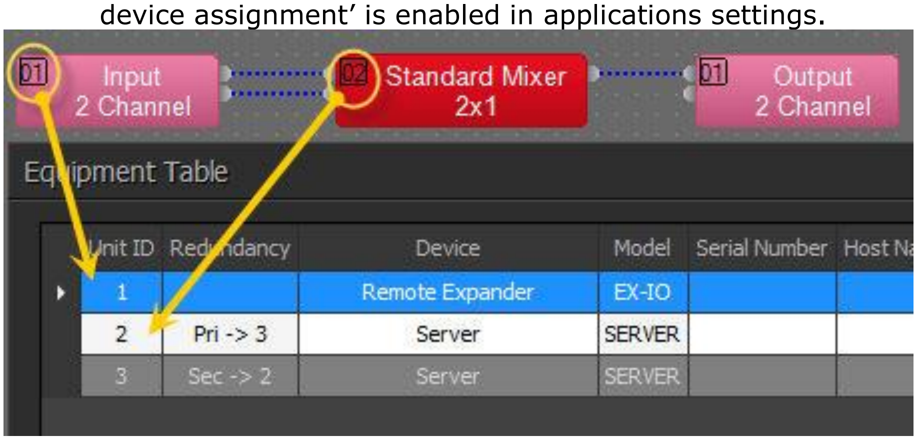
Programming

- When adding or editing DSP or logic component objects enabling the 'make redundant' option in the initialization dialog. If 'smart select' is enabled in the tools menu when a compilation occurs the redundant pair will be added to the equipment table.



- it is also possible to manually add a Server device to the Equipment table and selecting the make redundant button.

- The component objects associated with a redundant pair will display the Primary device ID number in the properties sheet and at the side of the block if 'display device assignment' is enabled in applications settings.



- A column in the device grid of the Connect to System dialog will show for each device whether that device was reportedly in an active or a standby state at the time when device discovery was done.

Topology

Network Primary and redundant servers must have unique (different) IP addresses and hostnames

Terminology:

- **Redundancy Pair:** A pair of identical frames loaded with identical IO and DSP cards in a Tesira system used to implement failover. One device in the pair can take over the functions of the other in the event of a failure.
- **Stand-Alone Device:** A device in a system that is not one of the devices in a redundancy pair.
- **Normal Operation:** A system is in a stable state devices in the system are not starting up, shutting down, or failing over, and all devices are present on the networks and operational.
- **Active Device:** At any time during the normal operation of a Tesira system with redundancy pairs, one device in each redundancy pair is active. This is the device that is actually providing the functions of the pair.
- **Standby Device:** In a redundancy pair in a normally operating Tesira system, whichever device is not active is in standby, which means it is in a state in which it is waiting to take over the functions of the pair in the event of a failure.
- **Device State**: Once redundant pairs are in **Normal Operation** they are in either the active state or the standby state. In the active state the device is attempting to operate normally. In the standby state the device is attempting to be ready to switch to the active state and begin handling the functions of a redundancy pair. Stand-alone devices always run in the active state.
- **Failover:** The process of the Standby Device taking over and becoming the active unit. If the active unit is operational enough, it will become the Standby Device
- **Automatic Failover:** A failover that occurs automatically when a failure is detected by a running Tesira system.
- **Manual Failover:** A failover that is triggered manually through TTP.
- **Failover Event:** The loss of functionality or connectivity that triggers an automatic failover.
- **Control Tracking/State Mirroring:** The maintaining of all controllable states in both the active and standby devices in each redundancy pair at all times as far as possible. This permits the system to continue with minimal disruption after a failover.
- **Primary Unit:** The member of a redundancy pair designated as the default active unit. This is the unit that will be active when the system first starts running, and it will remain active until a failover occurs or the device is reset/initialized or powered down.
- **Secondary Unit:** The member of a redundancy pair that is not the primary unit.
- **Logical Link** - The SERVER SNC-2 card has four logic IO connectors. In order to support failover monitoring a cable must be wired between both SERVER Logic IO ports.

Active Device

Generally, any active device is fully functional as far as possible:

- All audio processing is occurring normally, and audio signals are being transmitted and received by the device per the system configuration.
- All logic processing is occurring normally, and logic signals are being transmitted and received by the device per the system configuration.
- The system-level and block-level states in the device are current and authoritative.

- The SERVER is proxying expanders per the system configuration.

Standby Device

As an audio processor, a standby device will generally accept inputs and process them, but it will not produce outputs:

- Monitors the active device for failures.
- Keeps track of current control states (state mirroring).
- Can respond to TTP requests, forwarding them as appropriate.

Failover

When the active device in a redundancy pair fails, the standby device will take over providing the functions of the pair. Detection of the failure will take a small but indeterminate amount of time, and the same goes for the failover process. Between the time when the failure occurs and the time when the failover process has completed, there may be control errors and/or minor audible artifacts. Once the failover is complete, the normal functioning of the system will resume.

Device Roles

Each device in a Tesira system has an assigned role either stand-alone, primary or secondary.

Stand Alone Devices

Stand-alone devices do not participate in redundancy pairs.

Redundant Devices

Each redundancy pair includes one primary and one secondary device.

- The primary device is the default active device, which means that when the device is first started or configured, it will go into the active state unless the other device is already in that state.
- The secondary device is the default standby device, which means that when the device is first started or configured, it will go into the standby state unless the primary device is unavailable.

Control Tracking/State Mirroring

In an operating redundancy pair the standby device will track the state of the active device at all times, though there will typically be a slight control lag (typically under half a second). Therefore upon failover, the states of the blocks hosted by the pair, the portion of the presets maintained by the pair, and the system states maintained by the pair will all be either in synch or close to in synch, thus helping to minimize disruption of the functioning of the system.

TTP Control

Either device in a server redundancy pair will be able to handle TTP commands for system control. The Third Party control system is tasked with monitoring and maintaining the connection to the Active or Standby unit, as well as adjusting control priority in case of failover. The third party control system is also tasked with any-route or re-establish of controls or subscriptions when something causes the publishing Tesira device to stop publishing.

The following redundancy specific DEVICE commands are available:

- **DEVICE get knownRedundantDeviceStates -** permits control systems to determine the active device in a redundancy pair. This attribute is able to be subscribed to and will publish any change, thus permitting a control system to be notified when a failover occurs.
- **DEVICE manualFailover <device Id> -** swaps the active and standby devices in a redundancy pair. It is intended to be used during testing or if a control system is required to swap the active and standby devices in a redundancy pair. There is no ability to trigger a manual failover from the Tesira Software.

Please review the TTP **Device Services section for more details**

Hardware Requirements

Identical Frame Requirement:

Tesira redundancy is at the SERVER device level only. A redundancy pair consists of two SERVERs and failover will be from one device to the other. For each redundancy pair, it is a requirement that the frames are identical as they must have identical capacities and capabilities:

- they must be loaded with identical IO cards distributed identically into slots
- They must have the same quantity of DSP-2 cards

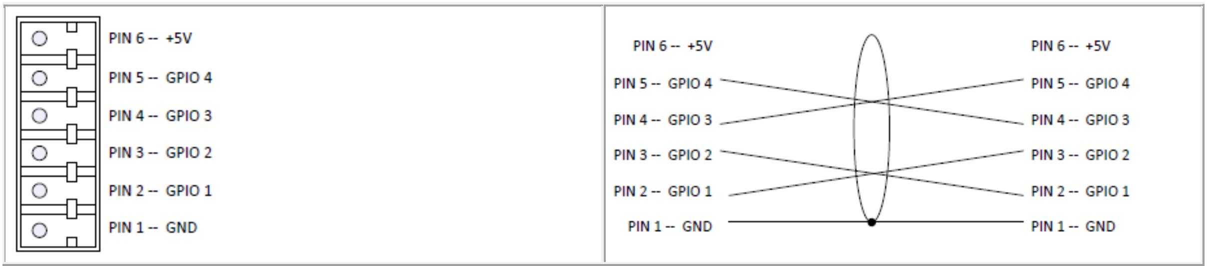
Audio Network Card behavior

- **AVB cards-** The Tesira compiler will specify the connections used as part of its calculations. The Tesira SERVER Failover pair will manage and maintain connections (no additional user setup required.)
- **1722.1 Streams** - AVB.1 Input and AVB.1 Output blocks can be assigned to a redundant Server but in a failover scenario the user must re-establish the stream. See the AVB 1722.1 Explicit Streams sections for more details
- CobraNet Cards - The User specifies the bundles and channels being used. The Tesira devices will the track the bundle number specified. The Tesira SERVER Failover pair will manage and maintain connections (no additional user setup required.)
- **DAN-1 cards -** Dante Controller Software must be used to define how the audio flows will ultimately be routed on the network. This limits the capabilities of the redundant SERVER to manage the failover process. A standby Tesira Server mutes incoming and Outgoing Dante audio. The following additional setup items must be implemented:
    -
    - A Dante-enabled device sending audio to a redundant pair must explicitly fan out each channel to go to both the primary and secondary servers. Then all those connections must be made on the Dante network (via Dante Controller or some equivalent). A design choice must be made to make all those signals multicast, or fan out inside the sending device. The bandwidth usage in these cases may be different.
    - A Dante-enabled device receiving audio from a redundant pair must explicitly accept channels from both servers in the pair, and mix them channel-by-channel.

Special Monitoring Cable:

It is necessary to connect the devices in a redundancy pair together via a special cable between their GPIO ports. This monitoring cable is always required between server devices in a redundant pair. A Cable of 28AWG and up to 100foot/30 meters can be used. As the

GPIO is being used for the Special Monitoring Cable it cannot be used for any other Logic or Control Voltage.



Front-Panel Indicators
Devices in a redundancy pair indicate whether they are operating in an active state or a standby state via the LEDs on their front panels. The activity light is amber on the standby unit and green on the active unit.

Redundancy Pair Device Startup

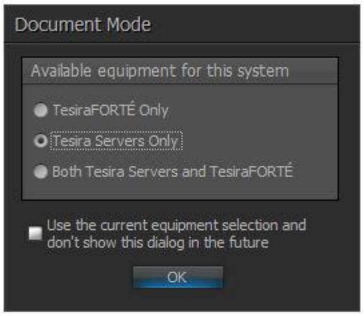
When a device in a redundancy pair is starting up or coming online, it will need to determine whether to start up in an active state or a standby state.
A redundant pair will check to see if the other device in the pair is already running in the active state. If so, the device will start in the standby state. If the other device in the pair is already running in the standby state, the device will start in the active state. If the other device's state cannot be determined, a primary device will immediately start in the active state. a secondary device will wait for the primary device to come up. As soon as the primary device comes up, the secondary device will then know what running state it should be in.
If the primary device does not become available within a reasonable time, the secondary device will become the active device.

NOTE: When both primary and secondary server devices are coming online at the same time, there is no attempt to determine which device was most recently active and therefore has the most up-to-date persistent state. The Primary Device will be the default active unit in such scenarios. If there is a need to preserve the running state from the secondary device, the secondary device must be started first, once active, start the primary device, which will come up in the standby state. After initial state synchronization, a manual failover can be triggered to switch the roles of the devices without any loss of state data.

Rebooting a Failed Device

When a device has experienced an actual failure (as opposed to a simulated failure via a manual failover), the device will reboot itself to attempt to clear temporary failure states.

Software Modes



When a new Tesira document is created, the user is prompted to choose which Document Mode the software will operate in. By checking "Use the current equipment selection...", it is possible to make the software accept the answer and not ask again. To reverse this behavior, see Application Settings.

The selected Document Mode controls which types of Tesira devices the compiler will place into the Equipment Table to implement the system. It also affects which I/O objects can be placed into the layout.

- **TesiraFORTÉ Only** mode provides the TesiraFORTÉ family blocks and USB I/O blocks. Also, since TesiraFORTÉ devices with AVB can make use of remote and rack-mount expanders, standard Input and Output blocks are available. The TesiraFORTÉ device I/O blocks are associated with each other such that deleting one I/O block causes all I/O blocks for the device to be removed from the layout and the device removed from the Equipment table. Some I/O blocks like ANC are not available in this mode since they cannot be allocated to TesiraFORTÉ devices.
- **Tesira Servers Only** mode provides all blocks except TesiraFORTÉ family blocks and USB I/O blocks.
- **Both Tesira Servers and TesiraFORTÉ** mode provides all blocks. In this mode, TesiraFORTÉ family blocks can be created as a group but the linkage between blocks is weaker. Individual I/O blocks can be deleted from the layout without all I/O blocks for the device being removed. Removing blocks has no effect on the Equipment table.

The Document Mode can be changed after a file is created, with certain limitations. For example, if a TesiraFORTÉ family device or USB I/O blocks are present in the layout, converting to Server Only mode is not possible. It is always possible to convert to **Both** mode. To do a backward conversion, you would have to copy the desired blocks onto the clipboard and paste them into a different document opened in a separate instance of the application.

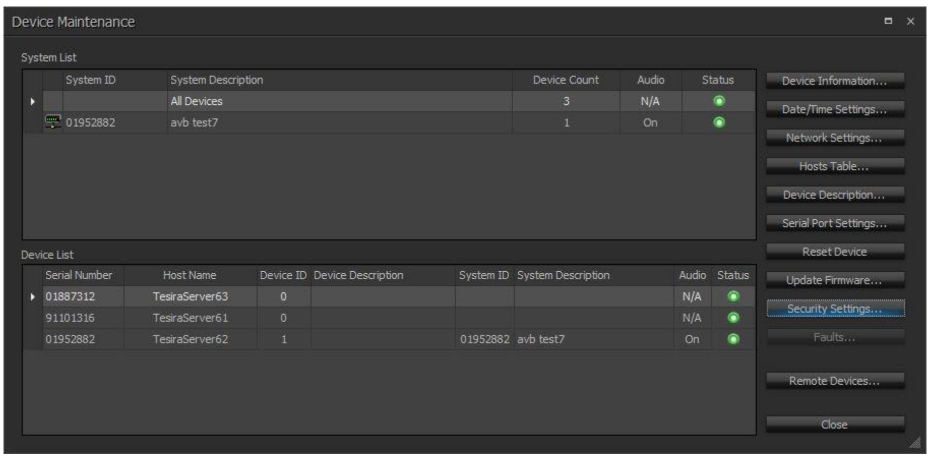
System Security

System Security

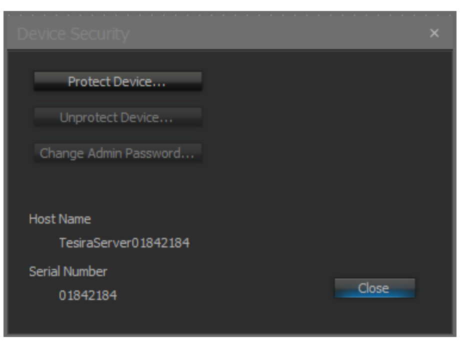
Allows a system to be password protected. Enabling Security also affects Connecting to the system via TTP. Please review TTP Security for more details.

Protection can be applied using two methods depending if the system is Unconfigured or Configured:

Unconfigured System
If a Tesira Server or Server I/O is unconfigured (Device ID 0, System ID and System Description blank as shown in device maintenance), this can initially be done by selecting the device and clicking **Security Settings**.



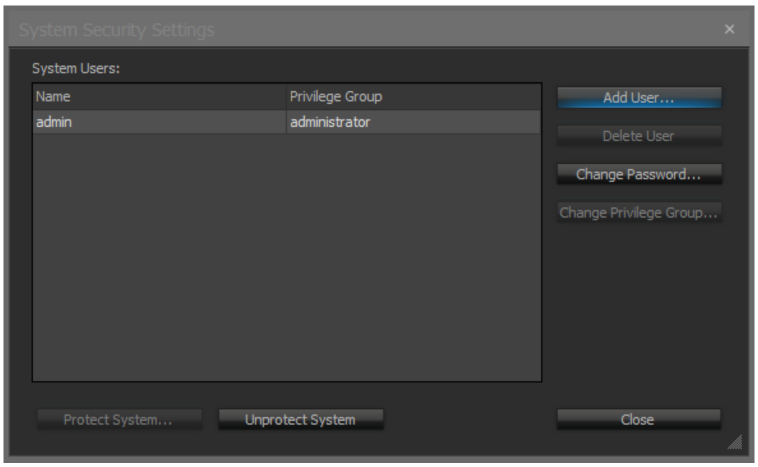
The Device Security dialog will allow the creation of an administrator password, which retains all permissions to the server. This is applied by clicking the Protect Device button. The admin password can also be changed or the system unprotected from this dialog.



Set the Admin Password as required. Other levels of system access must be configured using the Tools>System Security dialog when connected to the system when it is configured.

NOTE: When two protected Tesira Servers are configured as part of the same system, the administrator passwords must match.

Configured System
If the Tesira Server, Server I/O or TesiraFORTÉ units are already configured (part of a system), Protection can be applied using the **System > Security > Manage System** Security dialog. As in the previous case, one must first assign an administrator password that retains complete access to the system. With that accomplished, users of various types can be added to the system. With that accomplished, users of various types can be added to the system. Subsequent logins to the system at the administrator level will use the password with 'admin' as the user name



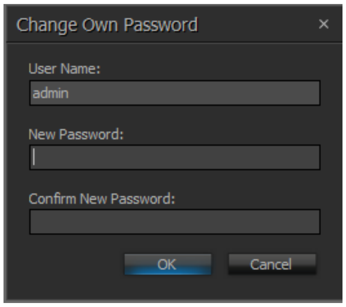
By clicking on **Add User**, other users can be created with one of a few system access (privilege) levels:

- Observer – Can only read system parameters in real time, i.e. meters, mixer control dialogs, etc.
- Controller – Can read and write system parameters in real time, i.e., set matrix crosspoints, set levels, etc.
- Supervisor – Can do above, plus:
  - Change presets
  - Start or stop audio
- Designer – All above, plus:
  - Edit signal flow and system partitions.

One must be logged on as administrator to:

- Create or Edit users
- Change the device's configurable settings in Device Maintenance
- Reset/initialize the device
- Update the device's firmware
- Add the device to a system or remove the device from a system
- Unprotect a protected unconfigured device
- Unprotect a protected system
- Change a remote device's configurable settings via the Tesira server which is acting as a proxy when the remote device is part of the same system
- Reboot the device through TTP
- Change Password

To change a password, select **System > Security > Change Own Password** while logged in to the system.


Change Own Password
User Name:
admin
New Password:
Confirm New Password:
OK
Cancel

System Limits

The following Tesira Device system limits need to be considered.

Server and Server IO

Any Tesira system which does not include TesiraFORTÉ devices is subject to the following maximum limitations:

- 128 Servers and Expander devices (in any combination) maximum
- 128 TEC-1 Control devices
- 128 devices per network segment (i.e. per subnet)
- 64 Expanders and Control devices (e.g. TEC-1) (in any combination) total per Server maximum

Please review the Audio networking limits and details in the following locations:

- AVB Network Considerations
- CobraNet Network Considerations
- Dante Network Considerations

TesiraFORTÉ Devices

Any Tesira system which does include at least 1 TesiraFORTÉ devices is subject to the following maximum limitations:

- 64 Server and Expander class devices combined
- 64 TEC-1 Control Devices
- 64 Devices per network segment (i.e. per subnet)
- 32 Expanders proxied per FORTE*

* Proxy limits would remain at 64 devices for Server(IO) in a mixed system.

Please review the Audio networking limits and details in the following locations:

- AVB Network Considerations
- USB Network Considerations

System Connect Considerations

Hostnames and IP Addresses

Network devices use IP addresses to identify and communicate with one another, but they also often make use of hostnames as human-readable shortcuts for their IP addresses. This is why, for example, you can type www.biamp.com into your web browser, and your computer knows to communicate with a web server at IP address 216.119.79.7. Hostnames can be utilized on both small, local networks and large networks like the Internet.
Tesira servers and expanders support the use of hostnames to identify themselves on the Control Network. Tesira devices are not supplied with a default IP address from the factory; instead they are configured to obtain an IP address automatically from a DHCP server, and they are supplied with a unique hostname which is derived from the device type and its serial number.

Since Tesira devices are configured to obtain an IP address automatically from a DHCP server, connecting a new Tesira device to a network will result in one of two situations: if a DHCP server is present on the network, the Tesira server will be assigned an appropriate IP address automatically; or, if a DHCP server is not present on the network, the Tesira server will assign itself an IP address in the "Link Local" IP range (169.254.1.0 through 169.254.254.255). In either case, all new Tesira servers connected to the same network should obtain IP addresses that are in the same subnet, and therefore they should all be able to communicate with one another. Additionally, a computer running Tesira software which is connected to the same network (and is also set to obtain an IP address automatically) should obtain a compatible IP address and be able to discover any new Tesira devices.

Once a new Tesira device has been discovered by a computer running Tesira software, its hostname can be changed, and its IP address can be changed to a static address, if desired. The hostname and IP address of a server can also be found via the server's front panel display.

Resolving Hostnames

Tesira devices and Tesira software communicate with each other by hostname, not IP address, and therefore it is important that both Tesira devices and computers running Tesira software are capable of resolving these hostnames to IP addresses.

The most common way for network devices to resolve hostnames to IP addresses is by using a DNS server (Domain Name Server). A DNS server can be configured with the hostnames and IP addresses of each device on its network, and the server distributes that information to any network device that asks for it. A computer and a Tesira server can be configured to use a DNS server simply by supplying them the IP address of the DNS server in the appropriate field.

If there is not a DNS server installed on Tesira's Control Network, Tesira devices also support the Multicast DNS (mDNS) protocol to resolve hostnames. mDNS works in a similar way to DNS, except that it is peer-to-peer, and therefore doesn't require a DNS server. Tesira devices will respond to mDNS queries for their IP addresses from other devices on the network. Note that mDNS only operates on a single subnet, and therefore devices on different subnets will not be able to resolve each other's hostnames using mDNS.

Finally, each Tesira server maintains a Hosts Table file, which can contain a user-definable table of hostnames and their corresponding IP addresses. The Hosts Table is useful when a Tesira system spans multiple subnets, and the Tesira servers in each subnet need to resolve each other's hostnames without the help of a DNS server. Note that the Hosts Table is only useful if IP addresses are static; if they are dynamic (because they are being assigned by a DHCP server), then the information in the Hosts Table will become incorrect when a Tesira server is assigned a different IP address.

Note that this means that systems which span multiple subnets must be addressed in one of two ways:

- Devices have static IP's, Hosts Table must be used if there is no DNS server available
- Devices obtain IP's from a DHCP server, DNS server must be available

Discovering Tesira Devices

Tesira software will attempt to discover Tesira devices on the network in response to four different actions:

- When the "Connect to System" button is pressed (or **System > Network > Connect To System** is chosen).
- When the Equipment Table is opened.
- When the Device Maintenance window is opened.
- When the "Send Configuration" button is pressed (or **System > Network > Send Configuration** is chosen).

Device discovery can be disabled by selecting "Disable Device Discovery" in the Applications Settings dialog (**Tools>Options>Applications Settings>Device Discovery**). This can be used to prevent the software from wasting time by trying to discover devices when you know that it will be unable to successfully discover any devices.
Tesira software will try to discover devices using all available network interfaces on your computer by default. The time it takes for software to discover devices can usually be shortened by excluding any network interfaces which are not connected to a Tesira network (in **Tools>Application Settings>Device Discovery>Interfaces**).

Local Servers

Tesira servers on a local network are discovered automatically by the software. Each server maintains a list of other Tesira devices that it knows are on its local network, and reports that list to Tesira software during device discovery.

Remote Servers

Tesira servers which are not on the local network are not able to be discovered automatically by Tesira software. Instead, the software must be given the hostname or IP address of at least one server on the remote network (in **Tools>Options>Application Settings>Device Discovery>Device List**). Once software is able to discover one server in the remote network, that server will report a list of all other Tesira devices that exist in that network.

For each remote network that includes a Tesira server, the Device List in software must include a hostname or IP address of at least one server in each subnet. Since each server reports the presence of all other servers in its local network, it is not necessary to manually input every server into the Device List; only one per subnet is required.

Expanders

Tesira expander devices are discovered by proxy through server devices. When a Tesira server is discovered, it reports the presence of any expander devices that it knows about to Tesira software. Tesira expander devices cannot be discovered if there is not a Tesira server on its local network.

System Connect Dialog

When Tesira devices have been successfully discovered, the software will display the System Connect dialog.

The System Connect dialog is divided into two lists. The top list is the System List, which shows all of the configured systems which were discovered. The bottom list is the Device List, which shows information about individual devices which were discovered. When “All Devices” is selected in the System List, the Device List displays every individual server which was discovered. When a configured system is selected in the System List, the Device List displays only the discovered servers which participate in that system.

To connect to a system, select the system in the System List and press “Connect To System”. Pressing the “Send System Config” button will attempt to send the currently open configuration file to the devices identified in its Equipment Table. The Device Maintenance button opens the Device Maintenance dialog.
The System Connect window does not display discovered expander devices, which are only viewable via **Device Maintenance>Remote Devices**.

Using templates

This option is available via the File Menu. Templates facilitate taking a previously used design file that is operating one Tesira system and saving it in a form that removes the equipment details and system ID details.

Please note a template file is not just a normal layout with the equipment table cleaned up.

If a user shares a layout that has been used in a system, that layout file is locked to that system. Any changes to that layout or the equipment table will effect the system and can not be used independently. This is often the desired workflow. That is, two users working on the same system may want to share a file and work on the same system.

If a user wants to take a layout from one system and create a new system based on that layout they MUST convert the file to a template. If they don't, when they try to configure the new system with the non-template layout file, the old system will be taken down (assuming they are on the same network).

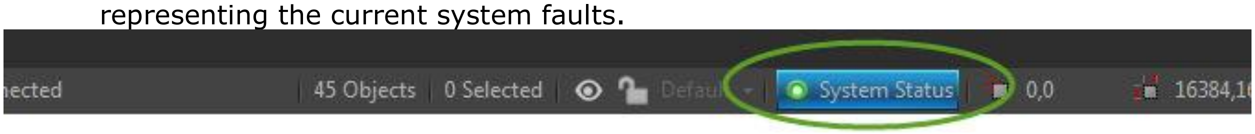
System Fault Reporting

System Status

Tesira System Status is reported via the front panel of Tesira Hardware or within the Tesira Software.

For details of Hardware fault status indicators please refer to the Hardware Status Indicators section.
The software reports the system status in the following locations:

- The Faults section of Device Maintenance
- When connected to a system the Status Bar will display a System Status LED representing the current system faults.



- When sending configuration to a system that requires audio to be stopped momentarily.

Please review the Fault Reporting section for full details of the errors reported in a Tesira system.

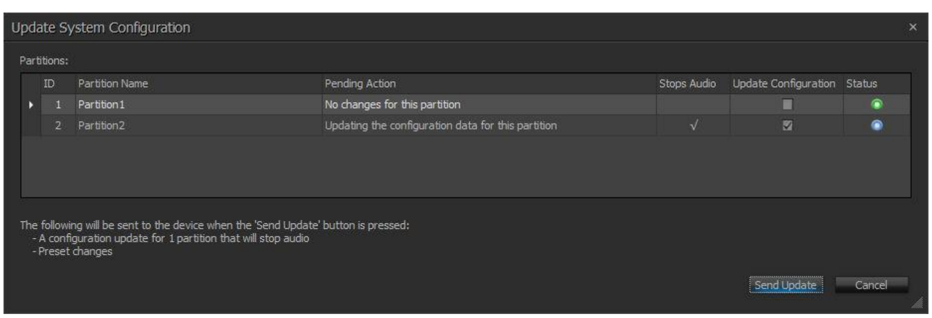
Fault Types

Green	Blue	Yellow	Red
No Faults	Updating Configuration	Minor device fault active	Major device fault active

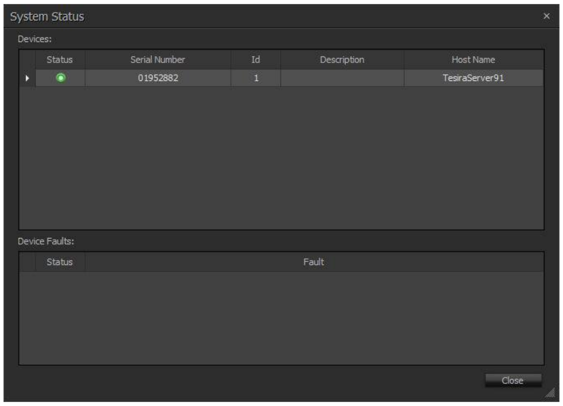
This status is a composite of all server and remote devices in the system. If any device in the system has a fault, this will display the severest fault level (i.e. if the system has a major and minor fault, this will be red). The control will update itself as faults occur or are corrected.

Examples

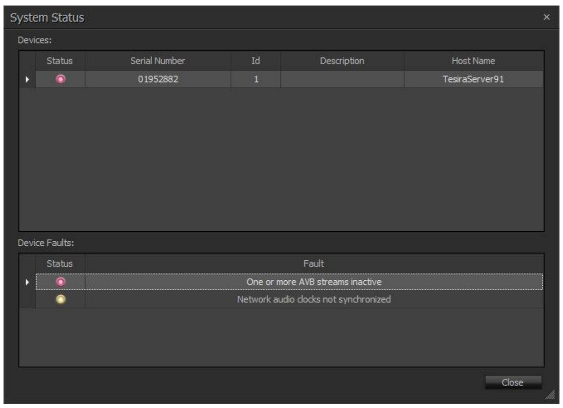
Sending configuration that requires audio to be stopped briefly in Partition 2:



System Status with no faults:

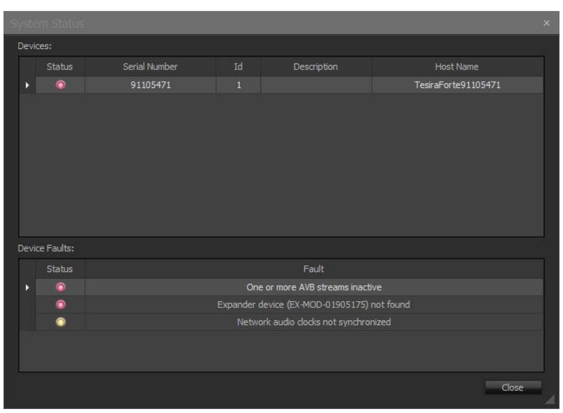


System Status with audio delivery faults:



Expander fault reporting

Tesira Server with an EX-Mod proxied to it that is not on the network.



Fault Reporting

The Following fault reporting is possible from Tesira Server, Server IO, TesiraFORTÉ and expander class devices:

- A Server, Server IO or TesiraFORTÉ device front panel will show local faults and faults for all expander class devices for which it is acting as a Proxy.
- The Tesira Software interface indicates faults of all devices in the Tesira system.
- Expander devices will also display specific faults via the Tesira Software Interface.

Server, Server IO and TesiraFORTÉ Front Panel

The front panel provides a submenu that contains a combined list of the hostname of the local server device and hostnames of all remote devices proxied by the local server which are currently in fault. If there is a fault detected locally within the server-class device, its hostname will be at the top of the list. If no faults are detected for the device (local or remote), then its hostname will not appear in the device list.

Software Reporting

When connected to Tesira Software the System Status LED in the Status Bar shows overall fault status of the system. This LED shows the most severe fault for all server and remote devices in the system. Its purpose is to give a quick view to the user whether any device in the system has a fault, including remote devices.
Clicking on the System Status LED will open the System Status dialog. This list will contain servers and remote devices found in the equipment table.
The System Status dialog will show all discovered devices regardless of their fault status. Even devices without faults will be shown here as long as they are discovered. Server-class devices are discovered explicitly by Tesira Software whereas remote devices are discovered by their proxies. If any device, Server, Server IO, TesiraFORTÉ or remote expander, is no longer discovered, it will not be shown in this list.

Major Faults

Fault Description
Expander device <hostname> not found
Expander device <hostname> failed firmware update
Unable to communicate with expander device <hostname>
Expander device <hostname> is incompatible
Unable to communicate with an audio input card
Unable to communicate with an audio output card
Unable to communicate with an audio input/output card
Unable to communicate with a VoIP card
Unable to update firmware on a VoIP card
IP Address Conflict on a VoIP card
Unable to communicate with an AEC card
Unable to communicate with an ANC card
Unable to communicate with a CobraNet card

One or more DSP cards failed to boot
Unable to communicate with the IO card in slot <card slot>
Unable to communicate with the DSP card in slot <card slot>
DSP Application error on card in slot <card slot>
Invalid IO card configuration
Wrong card type installed in slot <card slot>
One or more AVB streams inactive
Device network parameter changed
Server device (<hostname>) not found
AVB Disabled
Unable to communicate with the DAN-1 card in slot <card slot>
DAN-1 card in slot <card slot> has both cables unplugged; one or more Dante flows inactive
one or more Dante flows inactive
DSP Tasks consuming more cycles than software compiler estimated
Unable to communicate with the USB interface
DAN1 card in slot <card slot> failed to sync clock to backplane
Lab.gruppen amplifier has a frame error
Lab.gruppen amplifier has a channel error
Unable to communicate with backpanel GPIO controller
No functional DSP cards found
Active redundant device not monitored by standby device
Redundancy cable link not detected
DAN1 card incoming clocks not synchronized; card audio muted
Device needs recovery from fatal error or firmware update failure
Device requires FPGA update
FPGA power on self test failed

Minor Faults

Fault Description
Unable to communicate with front panel display
Expander device (<hostname>) firmware update in progress
Network audio clocks not synchronized
CommandString incompatible with serial port 'usage'
Cooling fan malfunction. Check filter and fan.
DAN1 card isn't Dante network clock master
Unable to find proxy device
Lab.gruppen amplifier has a frame warning

Lab.gruppen amplifier has a channel warning
Dante Mic fanout detected. Remove fanout using Dante Controller
Unsupported device routed to Dante Mic channel. Route an appropriate device using Dante Controller.

Network Considerations

System Network Considerations

General

Tesira devices support the use of many network-based features. Tesira devices utilize the Control Network to communicate both with one another and with computers running Tesira software. Additionally, Tesira supports several protocols for transmitting audio data over standard data networks, including Audio-Video Bridging (AVB) and CobraNet. Since all of these network-based features use standard data networks, they all must follow basic networking rules.

Individual network cables cannot exceed 100 meters in length, with the exception of fiber optic cables which can extend significantly further.

Network Interfaces on Tesira devices will auto negotiate so network connections between two devices (without a network switch) or via a network switch should only ever require straight-through Ethernet cables.

Control Network interfaces are assigned both an IP address and a hostname. Tesira server, Server IO and TesiraFORTÉ devices are configured by default with a unique hostname derived from a combination of the device type and serial number. Tesira server, Server IO and TesiraFORTÉ devices are not configured by default with a default IP address, instead they are set to obtain an IP address automatically from a DHCP server. The hostname and IP address of a Tesira server can be modified via Tesira software.

Control Networks can span across subnets, but doing so requires the use of either an external DNS server to resolve hostnames to IP addresses, or the use of a Host Table. When a Host Table is used, server IP addresses must be assigned statically to ensure the IP address does not change.

Control Network

The Control Network is used for communications between Tesira devices, as well as communication from Tesira servers to computers running Tesira software.

Server-class devices

All Tesira server-class devices include an SNC card for connection to the Control Network. Server Devices use the SNC-2 Card. Server-IO Devices use the SNC-1. The SNC card has a Primary and Secondary network port, and currently the Secondary network port is inactive. The Control Network port is a Gigabit network port, and is compatible with 10Mbit/s, 100Mbit/s, and Gigabit networks.

Control Network interfaces are assigned both an IP address and a hostname. Tesira servers are configured by default with a unique hostname derived from a combination of the device type and serial number. Tesira servers are not configured by default with a default IP address, instead they are set to obtain an IP address automatically from a DHCP server. The hostname and IP address of a Tesira server can be modified via Tesira software.

Control Networks can span across subnets, but doing so requires the use of either an external DNS server to resolve hostnames to IP addresses, or the use of a Host Table. When a Host Table is used, server IP addresses must be assigned statically to ensure the IP address does not change.

TesiraFORTÉ Class Devices

All TesiraFORTÉ class devices include a dedicated connection to the Control Network. The Control Network port is a Gigabit network port, and is compatible with 10Mbit/s, 100Mbit/s, and Gigabit networks.
All TesiraFORTÉ AVB devices also include a dedicated AVB Port to facilitate sharing of AVB audio. More details

Expander-class devices

Audio Expander-class devices (EX-IN, EX-OUT, EX-IO, EX-MOD,) have a single network port, which often shares responsibility for both AVB and Control Network communications. For this reason, Expander devices must be able to access both the Control Network and the AVB Network.

EX-LOGIC and TEC-1 devices have a single network port which shares the Control Network communications so these devices will require access to the Control Network Only.

Hardware	IP	VoIP	POTS	USB	AVB	AVB - 1722.1	CobraNet	Dante	RS-232
Tesira Server	Yes	With SVC-2	With STC-2	No	With AVB-1	With AVB-1	With SCM-1	With DAN-1	2 ports
Tesira Server IO	Yes	With SVC-2	With STC-2	No	With AVB-1	With AVB-1	With SCM-1	With DAN-1	2 ports
TesiraFORTÉ	Yes	Vi only	Ti only	Yes	No	No	No	No	1 port
TesiraFORTÉ AVB	Yes	Vi AVB only	Ti AVB only	Yes	Yes	Yes	No	No	1 port
Audio Expander	via Server or Forté Proxy	No	No	No	Yes	No	No	No	No
EX-Logic	via Server or Forté Proxy	No	No	No	No	No	No	No	1 port

AVB Network

AVB audio data can only pass through network switches which support the AVB protocol. The network to which AVB-1 cards are connected may include network switches which are not compatible with AVB, however any AVB data must always have a valid network path available from transmitter to receiver which passes only through AVB-compatible network switches. All AVB-1 cards in a single Tesira system must be connected to the same network; multiple AVB networks within one Tesira system are not supported.

Tesira Server IO and Server Devices

The Tesira AVB-1 card allows server-class devices to share digital audio with devices which support the Audio Video Bridging (AVB) protocol, as defined by IEEE 802.1. Each AVB card can transmit up to 420 channels and receive up to 420 channels of audio. Audio channels must be grouped into up to 64 AVB streams, and each stream can be comprised of up to 60 channels. As the number of streams being used increases, the maximum channel count of an AVB-1 card may decrease.
See the table below for an example of maximum channel counts in situations where all streams contain the same number of channels:

Channels per stream	Maximum channel count
60	420 (7 streams)
32	416 (13 streams)
16	400 (25 streams)
8	352 (44 streams)
4	256 (64 streams)
2	128 (64 streams)
1	64 (64 streams)

TesiraFORTÉ AVB Devices

Each TesiraFORTÉ AVB device can transmit up to 128 channels and receive up to 128 channels of audio. Audio channels must be grouped into up to 64 AVB streams, and each stream can be comprised of up to 60 channels. As the number of streams being used increases, the maximum channel count may decrease.
See the table below for an example of maximum channel counts in situations where all streams contain the same number of channels:

Channels per stream	Maximum channel count
60	128 (2 streams)
32	128 (4 streams)
16	128 (8 streams)
8	128 (16 streams)
4	128 (32 streams)
2	128 (64 streams)
1	64 (64 streams)

AVB Expanders

AVB Expanders such as the EX-IN, EX-OUT, EX-IO, EX-MOD and Lab.gruppen Amplifier have a single network port which shares responsibility for both AVB and Control Network communications. Therefore, for Tesira systems which use AVB Expanders, the AVB Network and Control Network must be the same network. AVB Audio Expanders (EX-MOD, EX-AEC, EX-IN, EX-OUT, EX-IO and Lab.gruppen Amplifier) do not support AVB.1 blocks.

Each AVB Expander chooses a Tesira server to act as its proxy server. All Control Network communications to and from the expander are routed through its proxy server. For this reason, an AVB Expander must always reside on the same subnet as its proxy server. The proxy server for each expander can be assigned manually in the Equipment Table; otherwise it will be assigned automatically by the Tesira software.

AVB Redundancy

The AVB-1 card features a Primary and Secondary network port. The secondary port is not to be used for daisy chaining. The Secondary port can optionally be used to support a redundant network configuration. When connected, the secondary port is a mirror of the primary port and will send out any data as sent out of the primary port, and receive any data just as the primary port would do. If either of the ports fail, the other port will continue sending and receiving AVB data.

Note
Audio Expander devices do not currently support AVB Redundancy, and will not receive audio from the Secondary port of a Server IO or Server. AVB.1 Input and AVB.1 Output blocks can be assigned to a redundant Server but in a failover scenario the user must reestablish the stream. See the Redundancy and the AVB 1722.1 Explicit Streams sections for more details

AVB Latency

Tesira software offers two latency settings for AVB data: 2ms and 1ms. These latency times represent the amount of time it will take to transmit an AVB signal from one device to another. However, if the total network latency from one device to another ever exceeds the latency value selected in Tesira software, packets may be dropped and audio may experience dropouts.

The table below shows approximate maximum switch hop values, assuming Gigabit network switches are being used; the exact values will depend on the network switches used and network traffic levels.

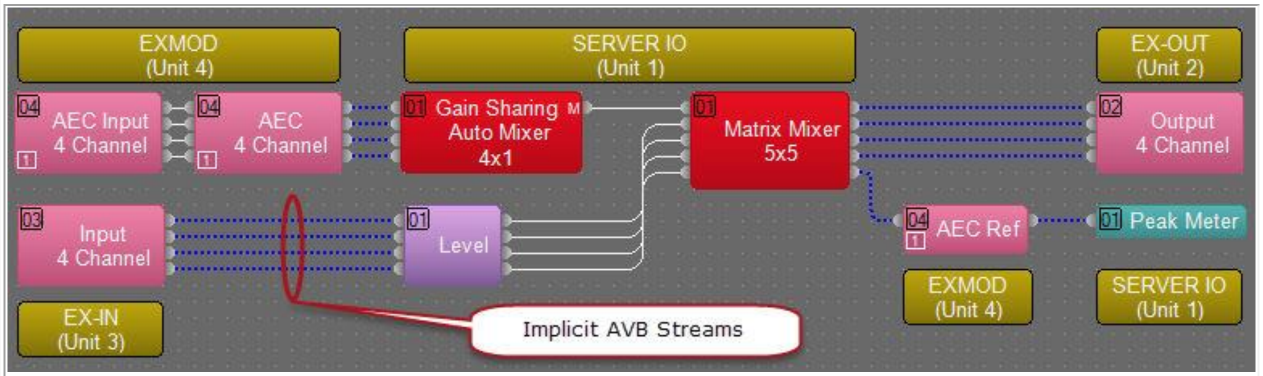
Latency setting	Switch hops
1 ms	3
2 ms	10

Table 2 - Approximation of how many Gigabit network switch hops are allowable for a given AVB latency setting. Actual values will depend on the network switches used and network traffic levels, and may be significantly higher or lower.

AVB 1722.1 Explicit Streams

Implicit AVB links

The Tesira compiler supports implicit AVB audio paths between a Server, Server IO, TesiraFORTÉ and Tesira audio expanders (EX-AEC, EX-IN, EX-OUT, EX-IO and EX-MOD). These links are shown as blue dotted lines where the Tesira software determined the best allocation of channels and streams within the Tesira AVB network. There is no need for users to define talkers and listeners, as they are defined as part of a valid compilation process, but AVB streams are restricted to Biamp Tesira devices.

Example

Above: Layout showing AVB Links from expander devices through a Server IO. Implicit AVB links have been defined by the Tesira compiler and are shown as blue dotted lines.

Explicit AVB links

With the release of Biamp Tesira 2.3 firmware and software Biamp introduces support for 1722.1 enabled AVB, also referred to as "explicit AVB streams". IEEE 1722.1 provides the Audio Video Discovery, Enumeration, Connection management, and Control (AVDECC) protocol for AVB devices. 1722.1 The user must explicitly route streams from talker device to listener device. Explicit routing is achieved via the use of 3rd party routing software such as Riedel AVB Manager or Pivitec AVDECC Controller. Explicit routing allows AVB transport between AVB devices produced by a wide variety of manufacturers. AVB products will be tested and guaranteed to be interoperable with one another through the AVnu certification process. A requirement of 1722.1 is that the talker and listener streams must have the same channel count. If the channel counts differ, Audio streams will not flow correctly. AVB.1 blocks can be allocated to a Tesira Server, Server IO and TesiraFORTÉ device, Audio Expanders (EX-MOD, EX-AEC, EX-IN, EX-OUT, EX-IO and Lab.gruppen Amplifier) do not support AVB.1 blocks.

Up to sixteen Tesira AVB.1 Input and Output blocks can be placed into each Server, Server IO or TesiraFORTÉ device. Review the AVB Network Considerations sections for AVB stream and channel bandwidth information.

Note

Please see support.biamp.com for a full list of AVB controllers. At the time of writing the following controllers are available:

- **Riedel AVB Manager** is available for download from avb.riedel.net.
- **Pivitec AVDECC Controller** is available from pivitec.com.

Each 1722.1 block placed in the layout will result in the creation of a 1722.1 enabled AVB stream in the AVB-1 card. The blocks are denoted “AVB.1” to reflect their 1722.1 support. Please see the AVB.1 Input and AVB.1 Output component objects.

These streams will be advertised via ADP, and third party controllers will be able to interact with and connect those streams using AECP and ACMP. Third party AVDECC controllers create connection on a per-stream basis.

Tesira AVB.1 input and output blocks will allow a maximum of 60 channels per block, and thus a maximum of 60 channels per stream. Streams may be given unique names within the Tesira software by the user, these names will be visible to the AVB routing software.

Once a Tesira configuration has been compiled and loaded to a Tesira AVB capable system, the participating streams will be advertised to the AVB network and become available to be connected to other devices by the AVB routing software.

AVB routing software will provide some version of an AVB matrix routing interface. Each participating AVB device on the network advertises its AVB talker streams which are available to be routed to other AVB listening devices. The user selects the intersection of the transmitting stream channels of one device and the receiving stream channels of another device and the AVB matrix negotiates the connection. Connection of output to input of the same device is not supported.

The routing software will provide clock master and slave information and other diagnostic capabilities for the AVB network participants. See the supporting documentation for your nominated AVB Manager software for further details on what is available and how to use the features of the products.

Implicit and Explicit AVB channels and streams will count against the total available channel and stream count (bandwidth) for AVB traffic across the AVB network and on Tesira AVB ports and must be considered by the installer in larger AVB systems.

CobraNet Network

The Tesira SCM-1 card allows server-class devices to share digital audio with CobraNet-enabled devices. Each SCM-1 card can transmit up to 32 channels and receive up to 32 channels of audio. Each set of up to 32 channels must be grouped into no more than 16 bundles, and each bundle is identified by a unique bundle number.

Bundle Size	**Latency**								
	5.33m			2.33ms			1.33ms		
	RX	TX	RX/TX	RX	TX	RX/TX	RX	TX	RX/TX
8	32	32	32/32	32	32	32/32	32	32	16/16
7	32	32	32/32	32	32	29/29	32	32	14/15
6	32	32	32/32	32	32	29/29	32	32	12/13
5	32	32	32/32	32	32	25/27	32	32	12/13
4	32	32	32/32	32	32	24/24	32	32	12/12
3	32	32	32/32	32	32	20/21	32	32	9/11
2	27	32	28/29	27	32	16/16	27	32	6/7
1	16	16	16/16	16	16	9/10	16	16	4/4

CobraNet networks must always be separate from the Control Network AVB Network and Dante Network. This may be accomplished either physically, by using separate network switches; or logically, by assigning CobraNet links to a separate virtual LAN (VLAN).

Conductor Priority

CobraNet devices have a mechanism for electing a device to act as the master synchronization clock for the network, also known as the conductor. When Tesira systems share audio with other devices using CobraNet, a server-class Tesira device must be elected as the CobraNet conductor. Tesira servers will automatically try to become the CobraNet conductor (by setting a very high conductor priority), but if for some reason the conductor is not a Tesira server, then the audio system may suffer from clock synchronization problems. The only exception to this conductor requirement is for Tesira systems which don’t use AVB at all, in which case a non-Tesira device may be the CobraNet conductor.

Primary and Secondary Connection

The secondary port found on the SCM-1 card is not to be used for daisy chaining – this is for CobraNet redundancy only.

Dante Network Considerations

Dante Networking

The Dante DAN-1 card, based on Audinate's Brooklyn II Module, allows Tesira Server or Server IO devices to share digital audio with other Dante-enabled devices, both from Biamp and other manufacturers. Each DAN-1 card can transmit up to 64 channels of audio and receive up to 64 channels of audio using up to 32 flows in each direction.

Each input and output block of channels can be defined with between 1 and 64 channels of audio (for a total of 64 inputs x 64 outputs allocated to all blocks per DAN-1 card). Each Dante channel will have an explicit name. Dante DAN-1 card *Hostnames* are maintained in the corresponding Server's Device Maintenance dialog.

Dante Microphone

Up to 32 Audio-Technica Dante microphones can be associated to a DAN-1 card. Please review the Dante Mic and Audio-Technica Mic Networking Considerations and Audio-Technica Dante Mic Hardware sections for more details on this device.

Dante

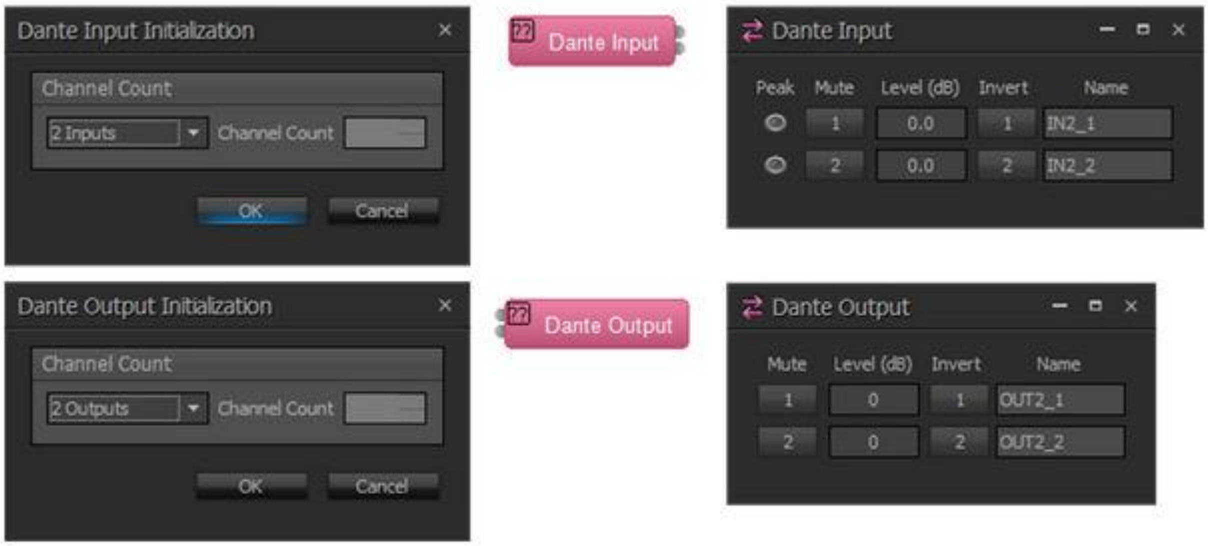
Dante is a proprietary digital media networking solution, developed by Audinate and licensed by Biamp, which operates on 100Mbps and Gigabit networks using standard Internet Protocol (IP) over Ethernet. A Dante stream distributes audio plus integrated control data over the network. It allows for transporting low latency uncompressed audio over standard IP Ethernet networks with sample accurate synchronization, automatic device and channel discovery, and easy to use signal routing.

Many of the properties of the Dante streams (or channels) are configurable only through Audinate's *Dante Controller* software. Most importantly, routing of audio signals from transmit to receive between devices is accomplished in *Dante Controller*.

- Channel naming can be done in *Tesira* software only when offline, the information is sent to the DAN-1 card on upload to be offered to the Dante network.
- Once online routing and channel assignment can only be done in *Dante Controller* software.
- Warning - Users are *strongly* discouraged from renaming transmit and receive nodes while online using *Dante Controller* as the names may become corrupted or lost in translation back to the *Tesira* software.

Firmware Updates

Normal Dante card firmware updates will be handled with a regular Tesira firmware update that is processed through Tesira Device Manager.
If a Dante card firmware update or recovery from crash is needed you will use Audinate's ***Firmware Update Manager*** software, refer to the Dante card Failsafe Recovery Section.



Audio

The standard bit rate for the Tesira and thus the DAN-1 is **48kHz / 24-bit**.
Dante automatically converts among the PCM word sizes when necessary. Dante won't connect incompatible audio devices. In practice, if *Dante Controller* software allows a connection to be made, it should pass audio.

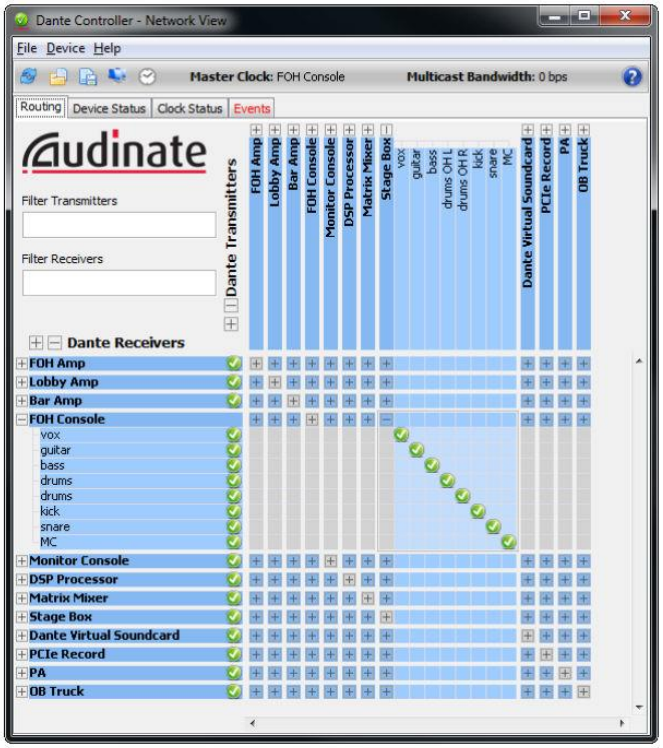
Dante Controller and channel routing

The Dante audio network's signal routing must be done via Audinate's free *Dante Controller* software on either a Mac or PC. It can be downloaded from http://www.audinate.com. Further information can be found on the Audinate website under *Support > Documentation > User Guides.*

In order to connect two Dante devices, the user must specify both endpoints using *Dante Controller*. Unlike CobraNet and AVB, Dante provides per-channel routing, so each Dante receiving channel can conceivably come from a different Dante-enabled device. Dante supports multicasting, or fan-out on the network, allowing more than one receiving device and channel per transmitting channel. There is not a limit to the number of receiving channels for a multicast stream.

Dante Controller software operates in real time, and reflects the current state of the network to which it is connected. For this reason, audio routes cannot be pre-configured before deploying a Dante network. Additionally, you can monitor device status and clock status via *Dante Controller*.

Once the system has been set up the *Dante Controller* software can be shut down or removed. The routing information is stored in the Dante-enabled devices themselves. Since Dante channel routing is done after the Tesira layout has been compiled it is not possible to predict streams and bandwidth requirements as can be done with AVB.



Flows: Unicast and Multicast

Each DAN-1 card can transmit up to 64 channels of audio and receive up to 64 channels of audio. A set of channels from a particular device is encapsulated in packets called a *flow*. A flow is a standard container for up to 4 channels which is created automatically when you configure Dante routing. In connections to the same receiver, no new flows will be created until all four channels in the most recently created flow are filled. The DAN-1 card can have 32 x 32 simultaneous flows, with a total of 64 x 64 channels.
If a *transmitter* runs out of the available flows, multicast is necessary to reduce the number of transmitted flows. You can check the number of transmitted flows using the *Dante Controller* software (under Transmit Flows in the Transmit tab of the Device View). A notification will appear if there are not enough available flows.
Also, it's possible for *receivers* to not have enough flows in special cases, such as when single channels are received from a large number of devices. In such a case, multicast will not reduce the number of flows, so it's necessary to reconsider the routing itself.
If there are not enough flows available for transmission, use the *Dante Controller* software to configure multicast, and reconfigure the network so that less flows are used. Be careful to keep the number of multicast flows (channels) to the minimum, because multicast flows increase the load that the switch is subjected to. Up to eight channels can be grouped into a multicast flow, further increasing their efficiency.
Finally, to help manage the multicast traffic on the network it is recommended to enable IGMP snooping on your switches.

Network Connections

The DAN-1 card connects via standard CAT-5e or higher network cabling to a network switch. It is a separate connection from the SNC-1 or SNC-2 network control port on the Tesira, and requires its own cable. It can share the same network switch hardware as the SNC-1 or SNC-2 network control port.

Unlike AVB (Audio Video Bridging) or CobraNet, Dante does not require special switch hardware, protocols, or VLANs, allowing it to operate with current "off-the-shelf" network hardware along with standard network traffic. As a rule of thumb, a separate, dedicated Dante network is recommended for critical, high channel-count applications.

- Audinate states that Dante is fully compliant with AVB IP protocols; however, Dante is not AVB and cannot communicate with or pass audio to or from AVB devices. Certain Dante hardware is sold as "AVB ready" meaning that firmware *may* be upgraded at some future date to support AVB when compliance standards have been completed.

Wireless LAN (Wifi) is *not* supported. While possible in principle, the practical limitations of current wireless technology (802.11a/b/g/n) render reliable performance unachievable.

Any switch that supports Diffserv (DSCP) QoS with strict priority and 4 queues, and which has Gigabit ports for inter-switch connections should be appropriate for use with Dante.
QoS is recommended for Gigabit switches on networks that share data with services other than Dante. A Gigabit interface is required for channel counts above 32 x 32 48kHz/24bit. Dante supports the use of mixed 100Mbps and Gigabit hardware, audio with mixed sample rates and bit depth, and allows the design of network zones with different latencies.
For low channel count (<32) applications, a 100Mbps switch may be used as long as it supports proper QoS, and QoS is active. The use of 100Mbps switches without QoS is not recommended or supported.
Although power management should be negotiated automatically in switches that support EEE, it is a relatively new technology, and some switches do not perform the negotiation properly. This may cause EEE to be enabled in Dante networks when it is not appropriate, resulting in poor synchronization performance and occasional dropouts.
If you use managed switches, ensure that they allow EEE to be disabled. **Make sure that EEE is disabled on all ports used for real-time Dante traffic.** If you use unmanaged switches, do not use Ethernet switches that support the EEE function, because you cannot disable EEE operation in these switches.
Single-link network limitations:

- Gigabit: 512 x 512 48kHz/24bit audio channels can be sent over a single link, giving a total of 1024 bi-directional channels. For 96kHz/24bit audio the channel capacity is halved.
- 100Mbps (Fast Ethernet): 48 x 48 48kHz/24bit audio channels can be sent over a single link, giving a total of 96 bi-directional channels. For 96kHz/24bit audio the channel capacity is halved.

The number of channels that can traverse one link in a network is proportional to the link speed. A link will always slow down to the lowest speed connector on that link; for example if a Gigabit port on switch A is connected to a Fast Ethernet port on switch B, the link speed will be 100Mbps Fast Ethernet. This is good, because it allows you to mix link speeds in a network without having to do anything complicated.
Audio is transmitted over the network in UDP/IP Packets. A single IP packet may contain audio samples from several audio channels, and may contain multiple audio samples for each channel.
Audio packets can be transmitted using either unicast or multicast addressing. By default they are sent using unicast, but the user can change this to multicast using the Dante Controller. Multicast and unicast can be used at the same time on a Dante device. Channels are individually selectable for multicast transmission.

Device Discovery

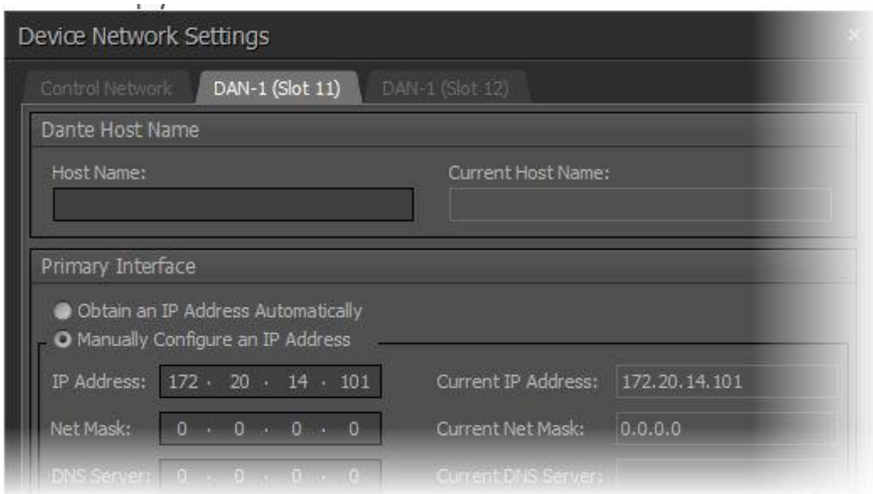
When connected to an IP/Ethernet network a Dante-enabled device will automatically configure its own IP address and advertise itself to other devices on the network. Dante-enabled devices will automatically discover one another over the network and learn each other's capabilities (number of input and output channels, sample rates and bit depths supported, etc.). Networked Dante devices and channels can be given "friendly" names to make sense to the user.
Dante devices obtain IP addresses automatically by default - so there should be no need to specify static IP addresses unless it is a specific requirement for your network.

- You can configure static IP addresses for one or both of the Ethernet ports (for supported devices) via the Network Config tab of the Device View for the device in Dante Controller.
- If your network has a DHCP server, Dante devices will receive their IP configuration using the standard DHCP protocol.
- On a network without DHCP, a Dante-enabled device will automatically assign itself an address using 'Bonjour' Zero Config auto addressing protocol by Apple. Devices will automatically assign themselves an address in the range 169.254.*.* (172.31.*.* for the secondary / redundant network, if present).

The secondary port found on the DAN-1 card is not to be used for daisy chaining – this is for Dante redundancy only. Dante offers a full-time redundancy option with permanent primary and secondary audio transmission. Redundancy requires a second distinct IP network. Cross-connecting the two networks will cause errors seen by Tesira as run time faults. *Dante Controller* must be used to identify issues in Dante streams.
Since Dante uses IP and not Layer 2 addressing, with Audinate's *Dante Netspander* software installed on a qualified rack-mount server Dante digital media can be transported across up to 40 subnets and across IP routers for large scale installations.
Dante devices are connected via a network switch, which most often means a "star" topology – all devices are connected to a single central point, which minimizes the number of "hops" through which data must pass. This also avoids the scenario in which the failure of one device causes the entire "daisy chain" to break.
Because Dante works with standards based networking technology, using fiber is simple. Use a switch that supports fiber connections to send Dante data over a fiber optic cable. Ethernet is not copper or fiber based, it is independent of the cabling medium. Many organizations may have fiber already in place from other projects and this can simply be re-used on a Dante network.

Setting Tesira Dante card Network settings

The Network settings of the Dante cards can be adjusted in **Device Maintenance > Network.** The Primary and Secondary interface IP address can be specified independently.



Selecting the Interface Status button will show the relevant settings being used.

Naming rules for Dante

In Dante, devices and audio channels are identified by names and labels which can be customized. Names can be assigned to channels while offline in the Tesira software and they will be offered to Dante Controller for use. It is *strongly* recommended that naming is done while offline in Tesira software only, to protect against names being lost or corrupted in the event of a power cycle or reboot of the Tesira device.
Initially all channels will be given names in the form

- **IN<block number>_<channel number>**, where block number is a unique integer associated with the Input block when it is created and channel number is within the block, starting with 1.
- **OUT<block number>_<channel number>**, where block number is a unique integer associated with the Output block when it is created and channel number is within the block, starting with 1.

All Dante names and labels are up to 30 characters in length. Name and label comparisons are case-insensitive; “Guitar” and “guitar” are treated as the same label. Unicode and non-roman characters are not supported.
Tesira DAN-1 hostnames will be unique, following the convention **TesiraServer*nnnnnnnn*-Slot*nn*** where the Tesira's Serial Number and Card Slot Number are appended to the string "TesiraServer".

Device names should follow Domain Name System (DNS) hostname rules. Legal characters are A-Z, a-z, 0-9, and '-' (dash or hyphen). Device names must begin with A-Z (or a-z). Channel labels may use any character except '=' (equals), '.' (full stop or period), or '@' (at). Channel labels must be unique on a device.
Channel labels do not need to be unique on the network as they are always qualified by device (channel@device). Tesira channel labeling conventions can be seen on the Dante Input and output block pages. Tesira device labels will be modified in Tesira software in the Device Network Settings dialog.

Device Names and Channel Labels

Dante routing is performed using the device names and channel labels. A receive channel can be subscribed to the name of a transmit channel at a device. Example: "Analog L@my-transmitter" describes a channel labelled "Analog L" on a device named "my-transmitter". Device names must be unique on a Dante network. Channel labels must be unique on the device.

If a device or channel is renamed, Dante routing considers it to be a different device or channel. If a new device or channel is then given the old name, Dante routing will route from the new device in place of the previous device. Example: The power supply on "stage-box" fails and "stage-box" needs to be replaced. The old "stage-box" is removed, and a new box is plugged in and named "stage-box". Dante receivers previously subscribed to the old "stage-box" will now automatically restore their subscriptions to the new "stage-box".

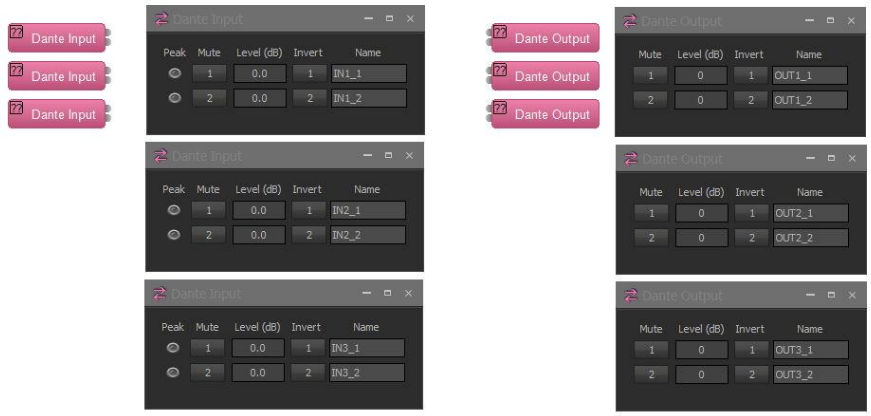
Device names must be unique on the network. If you attempt to rename a device using Dante Controller to a name that is already in use on the network, Dante Controller will notify you and reject the name change. Example: There is an existing device on the network called "MY16-slot1". If user attempts to rename another device to "MY16-slot1" Dante Controller will notify the user that the name is already in use. The device will not be renamed.

If a new device is added to the network with a name that already exists, a name conflict is detected, and one of the devices will rename itself by appending (2) to its name. This device will not be able to transmit audio until it is renamed.

Note: A device that has been renamed with (2) appended (e.g. "MY16-slot1(2)") will *not* be able to transmit audio until it is renamed. The device name must be changed by the user to be a valid non-conflicting name before the device can become fully functional.

Tesira device names will be defined by default as "TesiraServer*nnnnnnnn*" where the string TesiraServer is appended by its serial number. As with the channel names, the device names can be changed to better reflect the use case associated with the device.

In Tesira inputs and outputs will be assigned names in the order that the blocks are created. The names denote which block and channel within the block is associated with a given stream on a given device. The first input block would begin with "IN1_1", the second input block will begin with "IN2_1", etc. Output channels will be allocated in the same manner, beginning with "OUT1_1" and so on. The input and output blocks are a Tesira software convention which are not "seen" by Dante Controller, it only cares that the Tesira DAN-1 has 64 in and 64 out available.



On Card Redundancy

Dante offers a full-time redundancy option with permanent primary and secondary audio transmission on each DAN-1. Card Redundancy requires a second distinct IP network, either using a second switch network (recommended) or via a VLAN isolating the network traffic.

Dante redundancy requires that both the primary and secondary interfaces on any redundant card are connected using the same link speed.

If the secondary network is connected to a device that supports redundancy, it is enabled automatically. Audio data is transmitted on both the primary and secondary networks simultaneously. In the event of a failure on one network, audio will still continue to be received via the other network.

Dante devices that do not support redundancy must be connected to the primary network only.

Dante Controller must be connected to the primary network.

The secondary port found on the DAN-1 card is for Dante redundancy only, it is not to be used for daisy chaining devices.

Cross-connecting the two networks will cause errors seen by Tesira as run time faults. *Dante Controller* must be used to identify issues in Dante streams.

Device Redundancy

In addition Tesira systems offer device to device SERVER redundancy. Dante Controller Software must be used to define how the audio flows will ultimately be routed on the network. This brings other setup requirements. Please review the Redundancy section and Audio Network Card behavior section for more details.

Faults

If there is a major fault where all of the channels in a Dante block aren’t passing audio Tesira will report "One or more Dante flows inactive".
Cross-connecting primary and secondary Dante networks will cause network faults and errors.
Further diagnosis of faults requires the use of *Dante Controller*.

Clocks

An extremely high-quality clock is provided by the Tesira backplane, the card bus references that clock for Master Clock duties within the network unless a higher priority clock is available, such as AVB (when present) or Dante. If these higher priority clock sources are present then Tesira's clock will sync to their clocks.

Dante clocking guarantees that all devices are synchronized to within 1 microsecond or less, and that all devices can play out audio at the level of sample accuracy.

As with AVB, Dante uses a distributed Master Clock election protocol that automatically selects the best clock for the network, based upon information advertised by each Dante device. This information includes the quality of its clock, clock source, link speed and other parameters, and results in the best clock being elected as the Master Clock. One device is elected as the Master Clock to which other devices are synchronized. By default this selection takes place automatically, with no need to manually assign a Master Clock.

In the event the Master Clock drops offline audio will continue to flow and a backup Master Clock will take over. If the Master Clock fails for any reason, a new Master Clock will be chosen from the existing slaves within a few seconds. The transition from one clock master to the other does not result in any loss of audio. Slave devices “free run” during the period of master clock transition.

Most of the time, you do not need to be involved in the Master Clock selection process. Dante guarantees that the Master Clock device will be by default the strongest candidate.

To force a specific device to become the Master Clock use the Dante Controller to set a Dante device to be "Preferred Master". If more than one device is selected as the "Preferred Master", the device with the lowest MAC address will be chosen during a clock election. **Tesira will advertise its DAN-1 cards as "Preferred Master" devices by default.**

If AVB cards are present in a Server they will impose their clock on the Dante network. The DAN-1 in that chassis will show “preferred master” and “slave to external word clock” in Dante Controller. Other Dante devices will be slaved to that “preferred master”. In the event a Dante device sees multiple "preferred master" devices, the Dante devices will negotiate between themselves to determine the correct "preferred master" device.

In a system with AVB and Dante cards the AVB network must provide the clock. (Tesira will negotiate this automatically.)

Dante cannot be used as a "bridging" protocol between 2 or more Tesira AVB systems.

- AVB and Dante have different election processes to determine the clock master device in their respective networks. Dante allows an external clock to be imposed upon it, via the “preferred master” and "slave to external word clock" settings. For a Tesira system with Dante and AVB, the Dante network clock master *must* be provided by a Tesira chassis with an AVB card.
- The AVB clock "Grand Master" Tesira Server does not need to have a Dante card installed. The Dante DAN-1 card with the lowest MAC address in an AVB-enabled Tesira will become the "preferred master" for the Dante network - referencing its clock from the AVB network.

- *If an AVB card is imposing its clock on a DAN-1 card, and the Dante network is also imposing its clock on the card via another "preferred master" Dante device with a lower MAC address, a conflict will result as we cannot force non-Biamp Dante devices to submit to the Tesira media clock. Faults will be seen in this case.* ***This should be considered during the design phase of a system.***

Dante devices each contain a very high quality VCXO clock, and are synchronized with one another over the network using the IEEE 1588 Precision Time Protocol (PTP). This synchronization requires the use of Diffserv (DSCP) QoS with strict priority and 4 queues in the Dante network's switches.
The source of the Master Clock can be:

- The internal VCXO clock generated within a piece of Dante enabled equipment, or
- An external clock source which is internally connected to the Dante device (e.g. AVB in a Tesira Server).

There are certain circumstances in which the automatic Master Clock selection may be inappropriate. For example, a system may have a device that is periodically connected and disconnected, e.g., an input to the network from a stage box or mixing console. This device may not be always present and thus would be a poor choice for a Master Clock. Using the Preferred Master setting in *Dante Controller*, you may designate as a Master Clock a device (or devices) that is always present for the entire time that the network is required to function.

Slave to External Word Clock

A Dante device with "Slave to External Word Clock" set will use the external word clock from its host equipment to tune its onboard VCXO. A Dante device with this attribute set will become the PTP Master Clock, unless there is another Dante device present with "Preferred Master" set.

Preferred Master

Sometimes it may be necessary to force a particular device to provide the PTP Master Clock. A Dante device with "Preferred Master" set will always be chosen as the PTP Master Clock. If more than one device has "Preferred Master" set, the device with the lowest MAC address will be chosen. Tesira enables this by default.

If a device set as a Preferred Master is added to a Tesira Dante system, and that device's Dante MAC address is lower than that of the Tesira Server, it will become the Master Clock device. Since Tesira's clock protocol requires it to be the Master Clock for the system this scenario will cause a (major) system fault. The fault can be resolved by unchecking the Preferred Master selection for the offending device.

In a redundant network, the clock synchronization protocol operates over both primary and secondary networks. Each network will have a designated PTP Master Clock; usually this will be the same device on both networks. If this is not the case (e.g. if a non-redundant device is designated Preferred Master) then one device will bridge the clock synchronization information from the primary to the secondary network, ensuring that all devices derive their clock from the same source. Redundant PTP Slave clocks will synchronize their local clocks based on information from one of the networks they are connected to. In event of a failure on one network a redundant device will continue to receive clock synchronization information over the other network.

Troubleshooting clock issues:

Supported devices are constantly monitored by Dante Controller to establish the accuracy and stability of their clock synchronization with the Dante network master clock. If a device clock is exhibiting significant instability, it becomes at risk of losing sync with the master clock, and Dante Controller can a display a 'Clock Instability Detected' pop-up, identifying the device.

In Dante Controller a "Clock Instability Detected" popup indicates a network configuration or hardware issue that is causing inconsistent packet timing. For example:

Issue	Resolution
Energy Efficient Ethernet ('Green Ethernet') functionality is active on a switch.	EEE is a power-management system for Ethernet switches, and can easily interfere with clock synchronization. Audinate recommends that you avoid unmanaged switches with EEE functionality, and fully disable EEE on any managed switches.
There is a 100 Mb switch or link where a Gigabit connection is required.	If your devices require Gigabit connections, make sure there are no 100 Mb links or switches in the chain. Audinate recommends always using Gigabit switches for network backbones.
One or more of your switches are incorrectly configured, or are not suitable for Dante networking	Ensure that you are using switches that support QoS, and Dante traffic is properly prioritized.
Network stress from other sources.	If you are running traffic from other sources across the network, it may be causing bandwidth issues that are interfering with Dante packet timing.
Excessive multicast traffic.	Using multicast flows where they are not actually necessary can overload a network, particularly if there are any 100Mbps switches or links present. Consider switching some subscriptions to unicast to take the pressure off the slower nodes in your network. The Dante multicast audio bandwidth for the network is displayed in the Dante Controller menu bar. As a rule of thumb, total bandwidth utilization (including multicast and unicast) on any given link should not exceed 70% of the supported bandwidth for that link. Utilization above 70% of supported bandwidth can adversely impact clock synchronization (especially if there is also non-Dante traffic on the network). It is also recommended (for this particular issue, and in general) that you ensure all your Dante devices are using the latest firmware, and that you are using the latest version of Dante Controller.

Latency

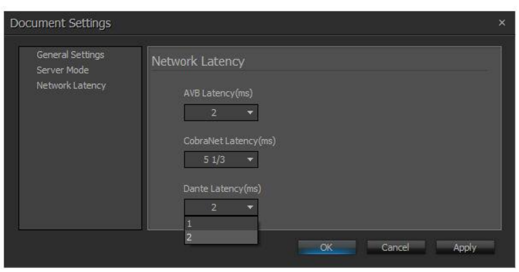
In Dante, variation in latency in the network is compensated for at the receiver. Each receiver has an Rx latency setting. This setting defines the latency between the timestamps on the incoming audio samples and when those samples are played out. The typical default latency for a Dante device is 1 ms. This is sufficient for a very large network, consisting of a Gigabit network core (with up to 10 hops between edge switches) and 100 megabit links to Dante devices. Smaller, Gigabit-only networks can use lower values of latency (down to below 200μs). Recommended latency settings are displayed in Dante Controller.

Dante uses a network-centric approach to synchronization using standard VoIP QoS to prioritize clock sync and audio traffic over other network traffic allowing synchronized playout over different audio channels, devices, and networks over multiple switch hops.

In order to bring latency down to the lowest possible values, a gigabit network should be used. This allows greater freedom to build a high performance, flexible network that maintains fantastic latency performance. Dante offers sub-millisecond latency for all products.

Latency in a Dante system is deterministic and guaranteed. Receivers that are listening to the same audio transmitter using the same latency value are guaranteed to be sample aligned. A Dante receiver introduces an additional latency before playing out audio to account for delay variation in the network or end device. The user sets this latency with Dante Controller and the value selected should be based upon the size of the network.

Latency can be configured to be different between different devices in the same network, and does not have to be the same for all connections on the network. Dante allows you to configure low latency connections for critical audio paths, while at the same time running higher latency connections for a broadcast or recording feed where latency is less critical. In principle the lowest latency between two nodes connected directly using Gigabit Ethernet is achieved if a single audio sample is collected and then transmitted in its own IP packet. Dante latency has been measured as low as 83.3µs for a gigabit implementation. The Biamp DAN-1 card supports 1 ms and 2 ms latency. The minimum latency available for a device connected to a 100 Mbps network port is 1 ms.



Adding new devices to a network does not affect the latency of devices already in the network. The latency of hardware devices does not depend on the number of audio channels routed, however some devices (e.g. the Dante Virtual Soundcard) may need to use higher latency to reliably process high channel counts. Routing additional audio channels does not change the latency of audio already passing through the network.

Dante DSCP / Diffserv priority values when configuring QoS

Some switches require special configuration to recognize and prioritize specific DSCP values. The table below shows how Dante uses various Diffserv Code Points (DSCP) packet priority values.

Priority	Usage	DSCP Label	Hex	Decimal	Binary
High	Time critical PTP events	CS7	0x38	56	111000
Medium	Audio, PTP	EF	0x2E	46	101110
Low	(reserved)	CS1	0x08	8	001000
None	Other traffic	BestEffort	0x00	0	000000

Dante Virtual Soundcard Firewall Configuration

The Dante Virtual Soundcard communicates over UDP using the following ports:

- Dante Clock Synchronization: 319, 320
- Dante Audio Routing: 4440, 4444, 4455
- Dante Control and Monitoring: 8700-8704, 8800
- Dante Multicast and Unicast Audio: 4321, 14336-14600

Dante card Failsafe Recovery

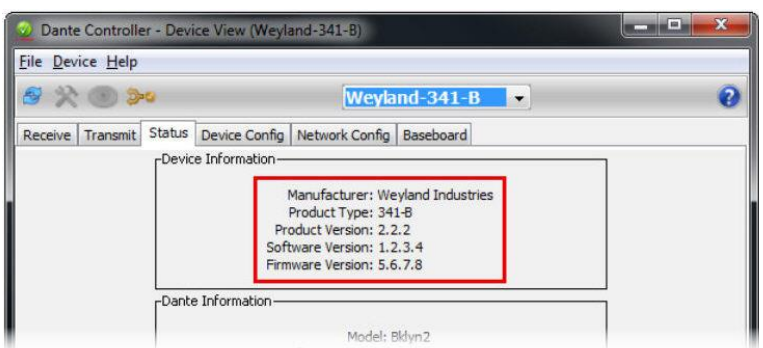
The Tesira's DAN-1 cards run Dante firmware provided by Biamp. If Dante firmware updates are required they are made using Biamp's Tesira software *Update Firmware* tool. The latest Tesira software version, firmware revisions for Biamp Dante-enabled devices, and user guide are available at support.biamp.com.
Dante updates are embedded in the Tesira system firmware updates. If you are up to date with Tesira system firmware then you are also up to date on the DAN-1 Dante firmware.

When to use it

Audinate's *Firmware Update Manager* is **only** used in the event of a "crash" of the DAN-1 card during a firmware update while using the Tesira software's Device Maintenance > Update Firmware tool. A crash is recognized by a fault where Tesira software cannot communicate with the DAN-1 card.

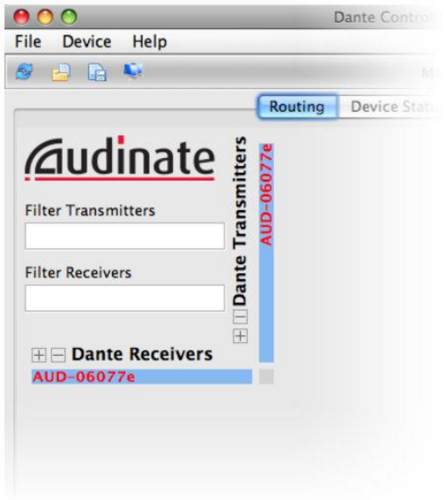
Audinate provides *Firmware Update Manager* software for performing Dante device recovery *in the event of a crash*.

Firmware Update Manager has two operating modes: ***Update Dante Firmware*** or ***Failsafe Recovery*** (roughly analogous to a boot kernel recovery).
In the event of a fault where Tesira software cannot communicate with the DAN-1 card you will proceed to the *Failsafe Recovery* option.



Dante Failsafe Recovery

In the instance where a Dante DAN-1 card has failed during an update it may go into **failsafe mode.** A system fault in Tesira indicating it cannot communicate with the [DAN-1](DAN-1) card will occur. There is a failsafe recovery file on the Dante card which allows a boot level recovery of the device. *Failsafe Recovery* mode is *only* required if a device has been corrupted and has entered failsafe mode. A Dante device that is in failsafe mode will appear with its default device name in red in the default network grid view of *Dante Controller* (versions 3.1.x and above).



Dante firmware recovery / update files are not included with the *Firmware Update Manager* - they are provided separately by Biamp. The latest firmware recovery files for the DAN-1 card can be found at support.biamp.com.

DAN-1 card firmware update files are part of Tesira system firmware updates and are made using the *Update Firmware* tool in the Tesira software's *Device Maintenance*.

Always ensure the update file you apply is the correct file for your device. Applying the incorrect type of update file for your device will render the device unusable, and it will have to be restored to its original firmware version using *Failsafe Recovery*. It is possible to 'force' an update file onto the module, using the 'Override Device Matching' option on the 'Select Firmware Update File' screen in *Update Dante Firmware.*

Before starting the update process, ensure that:

- All Dante devices that you wish to update are connected to the Dante network via their *primary* Ethernet ports only.
- The computer on which you are running *Firmware Update Manager* is connected *to the Dante network only, via one network interface only* - unplug any other network connections, and disable Wi-Fi for the duration of the updating process.
- You have saved the appropriate firmware update files to a local drive

To avoid corrupting the firmware on your device, you should not quit the application if a firmware update is already in progress.

Update Dante Firmware

This feature should **not** be used with Biamp Dante-enabled devices.

The Tesira's DAN-1 cards run Dante firmware provided by Biamp. If Dante firmware updates are required they are only performed using Biamp's Tesira software. Dante updates are embedded in the Tesira system firmware updates. If you are up to date with Tesira system firmware then you are also up to date on the DAN-1 Dante firmware.

Note: The latest recovery firmware revision for Biamp Dante-enabled devices and full instructions for using *Firmware Update Manager* will be found at support.biamp.com. Standard firmware updates are made using the Tesira software's *Device Maintenance > Firmware Update* tool.

Failsafe Recovery

Note: The latest recovery firmware revision for Biamp Dante-enabled devices and full instructions for using *Firmware Update Manager* will be found at support.biamp.com. Standard firmware updates are only made using the Tesira software Device Maintenance > *Firmware Update* tool.

All Dante hardware devices use software loaded from Flash memory as part of their operation. As with any storage device, it is possible to corrupt the data on the Flash part, potentially rendering it unusable, if there are interruptions while writing to the Flash device. By far the most common way for this to occur (as with other Flash devices) is by losing power to the Dante-enabled device in the middle of a firmware upgrade.

To provide recovery from this event, Dante hardware devices have a special protected part of Flash that will run in the event the main part is corrupted. When this is run, the device is in failsafe mode, and will wait for a recovery image to be provided by the *Firmware Update Manager*.

Dante device that is in failsafe mode will appear with its default device name in **red** in the default network grid view of *Dante Controller* (versions 3.1.x and above).

To restore a device that has entered failsafe mode, use the Dante *Firmware Update Manager* 'Failsafe Recovery' option. Your computer's default Ethernet interface *must* have a link local IP address 169.254.xxx.xxx or you will not be able to proceed with failsafe recovery.

Once the recovery image is loaded, be patient while the recovery file is written and the device reboots - this can take up to 2 minutes. You will then need to power cycle your Dante-enabled audio equipment after the Dante device or card automatically reboots.

Dante Mic Networking Considerations

The Audio-Technica ATND971 and ATND8677 enabled microphone makes use of a dedicated Dante mic Component processing object in Tesira software.

Audio-Technica Dante mic Networking Details

The Audio-Technica Dante enabled microphones have some unique requirements which must be observed for use on the Dante network. Currently the ATND971 and ATND8677 are supported by Tesira.

Latency Settings

Users are required to use Dante Controller to change the network latency settings of the Audio-Technica Dante mic as needed to match Tesira, as such, they are beholden to the range of latencies allowed in Tesira (1 or 2 ms for Dante as configured in the Network Latency section of the Document Settings in the Tools Menu). Tesira software provides no interface for changing the audio latency for an Audio-Technica Dante mic. An interface for changing the latency for the Dante mic is found in Audinate's Dante Controller software.

Slave

An Audio-Technica Dante mic will always be a "slave to external word clock" device in the Dante network, referencing the Tesira DAN-1 clock.

IP Addresses

The network IP address of the Audio-Technica Dante mic can be modified using Audinate's Dante Controller software.

Tesira Hardware Requirements and Compiler Implications

Dante mic Input Blocks may only be placed into Server and Server-IO's that contain a DAN-1 card. Block functionality depends on the ability to communicate with the ATND971 and ATND8677 mic over the Dante network.

The compiler will track each Dante mic Input block channel as a Receiver channel used on one of the DAN-1 cards in that server, in the same way it tracks channel usage for Dante Input and Output Blocks. Each Dante mic block channel will be assigned to a Dante DAN-1 card during compilation. The network settings of the assigned DAN-1, and the network settings of the Dante mics on the network, will dictate which mics may be assigned to that block channel from Dante Controller.

Number of microphones per Dante Card

The Dante API is used to implement a conmon (Con(trol)/Mon(itoring)) subscription between the Tesira Server device and the Audio-Technica Dante microphone. A maximum of 32 Dante mic channels can be assigned to any individual DAN-1 card. As the Dante mic processing block is allowed to be up to 64 channels in size, the compiler will allocate this to two DAN-1 cards in a single Server-IO chassis.

Locate

This feature allows the user to locate the physical microphone. When pressed, the LEDs on the microphone flash. Pressing and releasing the User Switch on the microphone will stop the flashing of the LEDs and return the value of Locate to false.

Block incompatibility with other Dante devices

It is expected that the user will route the correct Dante transmitter channel to the Dante mic input block channel in the Tesira Server. However, when using Dante Controller software the user has the ability to route any Dante transmitter channel to the Dante mic Input Block receiver in the Tesira Server or Server IO, including one from a non-Audio-Technica Dante device. The firmware will dynamically discover the manufacturer name and model for the Dante devices transmitting to the Dante mic Input block channels and validate accordingly. If any non-Dante mic (ATND971 or ATND8677) transmitters are discovered, or if the Dante mic component object contains a different mic selection that the mic model being used the firmware will take the following actions:

- Raise a minor fault "Unsupported device routed to Dante mic channel. Route an appropriate device using Dante Controller". Please refer to the System Status and Fault Reporting section for more details.
- Mute audio for the affected channel(s)
- Inhibit using Tesira’s Dante API to send further commands to the device. Note that Block attributes may still be changed, however, the changes will not have any effect until a supported device’s output flows to that block channel.
- Publish “<invalid device>” to deviceName attribute subscribers. The “Device Name” field in the software control dialog will display this string. Please refer to the System Status and Fault Reporting section for more details.

The user recovers from the “<invalid device>” state by using Audinate’s Dante Controller to re-route that channel’s source from a device compatible with the block.

Note
Connecting one Audio-Technica Dante mic to multiple Tesira Dante mic inputs is not allowed.

Tesira firmware only supports routing any single Audio-Technica Dante mic transmitter to one Dante mic block channel. Fan-out (routing one transmitter to many receivers) on the Dante network is not supported for the Audio-Technica Dante mic.

- Tesira firmware will mute duplicate input channels and deactivate their control functions.
- Also, a minor fault will be raised: “Unsupported Dante mic fanout detected”. Please refer to the System Status and Fault Reporting section for more details.

The user recovers from this state by removing the fanout by rerouting in Dante Controller. The user may also look in the <deviceName> fields of the control dialog to see which channels are offending.

USB Network Considerations

TesiraFORTÉ USB complies with the USB Audio 1.0 class (driverless) specification and can be used to connect to a PC or any device able to run ASIO drivers. The TesiraFORTÉ USB should be accessible from any host operating system that provides USB Audio 1.0 class drivers. This includes Windows, Mac and Unix Operating systems.

The USB Initialization dialog configure a TesiraFORTÉ USB device, plug it into a PC/Mac and have the device show up with the channels that were specified on the USB blocks within Tesira design software.
The USB interface specifically supports usage with

- Soft Codec support using Skype or Microsoft Lync client software - Via the Speakerphone option in the initialization Dialog
- ForTheRecord court recording software - Via the Line In/Out option in the initialization Dialog

Supported channel configurations

Audio stream rates are defined as either 24-bit/48kHz (up to 6 total channels I/O) or 16-bit/48kHz (up to 8 total channels I/O). The maximum bit rate is dependent on the total number of input and output channels used by the device.

The following channel and bit Depths are supported. The Sampling Rate is fixed at 48kHz.

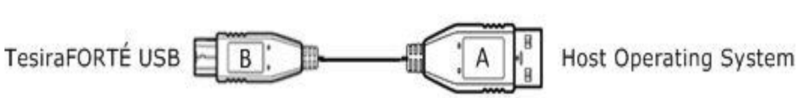
- Speakerphone with AEC and without AEC

In	Out	Bit Depth
1	1	24

- Audio In and Out

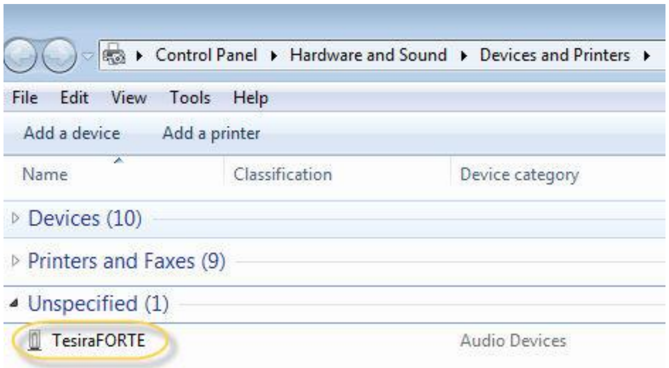
In	Out	Bit Depth
2	2	24
2	0	24
0	2	24
2	6	16
0	6	24
4	4	16
4	0	24
0	4	24
8	0	16
0	8	16

Wiring

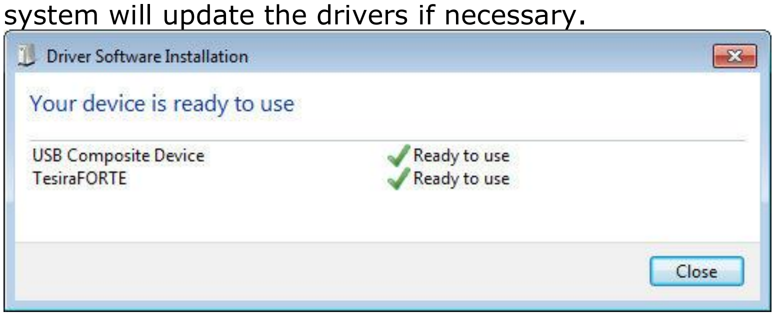


Connecting to Host device

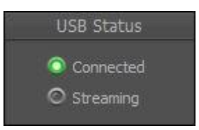
If a TesiraFORTÉ configuration has active USB blocks the first time the USB cable is connected there may be some drivers and initialization required. If a valid connection is in place, the TesiraFORTÉ device will appear as an audio device in the Host software.



Tesira software has a USB option in the I/O block menu, available when a TesiraFORTÉ device is used in a configuration. Every time the USB option changes the Host operating system will update the drivers if necessary.



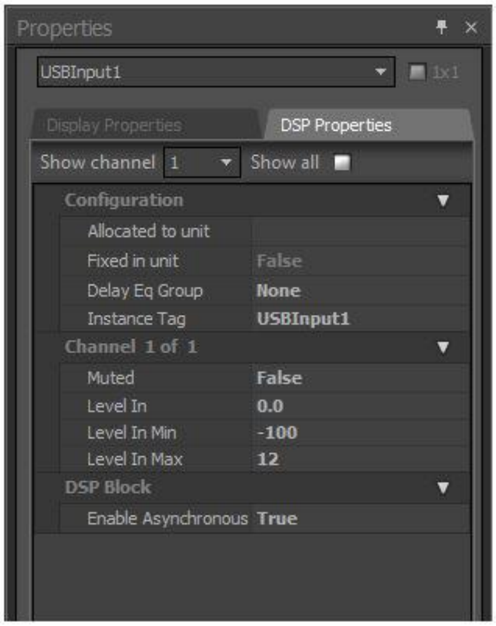
The Input or output control dialog will give indication of the Connected and Streaming Status



USB Clocking and synchronization

TesiraFORTÉ is an isochronous audio endpoint which can use its own internal clock (or AVB media clock) to synchronize data packets with the PC host (Asynchronous), or it can adaptively synchronize to a host clock, making use of playback compensation (sample interpolation) to correct for clock mismatch (Adaptive).

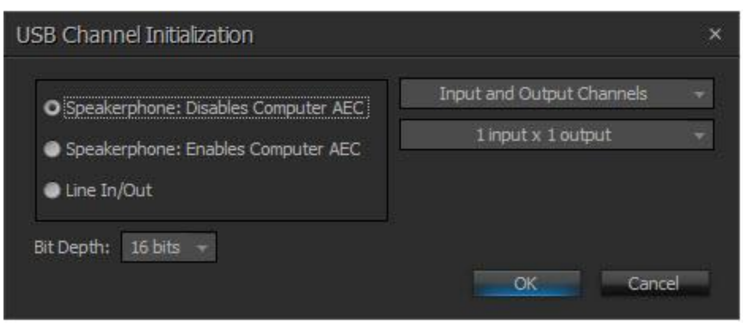
A setting in the USB Input object's DSP Properties, **Enable Asynchronous (True/False)** controls which method is used. **True** is the default value and should be appropriate for most uses. **False** selects the adaptive synchronization mode which would only be needed for USB hosts which must act as the master clock. Note that this synchronization method could result in audible distortion due to the sample interpolation used to maintain sync with the host.



Initialization options

The initialization dialog of the USB component object provides three options for how the connection is to be used.

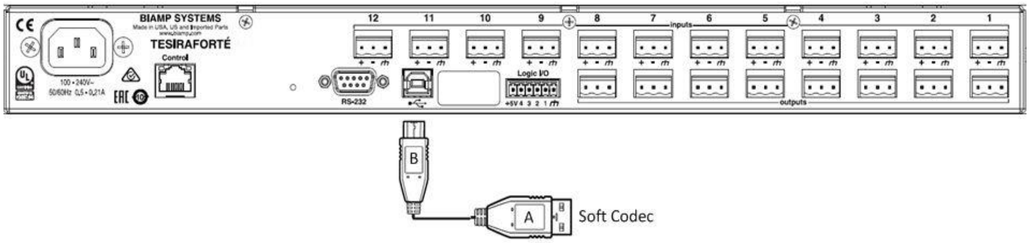
Note: There is no AEC capability on the USB inputs, since these will generally be line level sources and/or far end sources. Microphones used in distance conferencing should be connected to AEC inputs.



Speakerphone

The two Speakerphone modes provide a single audio input and output stream for use with a soft codec application on a PC. The USB Input represents the incoming audio from the soft codec and the USB Output is used to send audio to the far side.

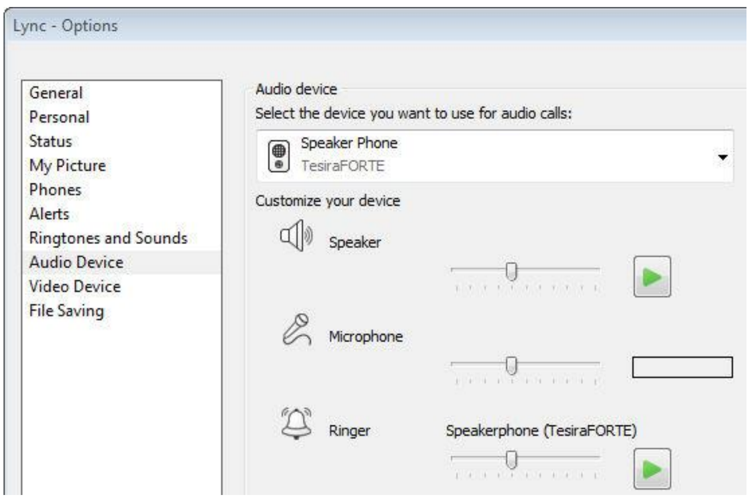
- **Speakerphone: Disables Computer AEC** – In this mode, the TesiraFORTÉ unit will provide the Acoustic Echo Cancellation (AEC) function, and a control message is transmitted to the soft codec via the USB link telling it to disable its internal AEC. This would be appropriate for TesiraFORTÉ models that have built-in AEC (CI, TI, VI).
- **Speakerphone : Enables Computer AEC** – This mode is for situations where the soft codec will provide the Acoustic Echo Cancellation (AEC) function, which would be appropriate for TesiraFORTÉ models that do not have built-in AEC (AI).



When configured for speakerphone the soft codec being used will have the option to select the TesiraFORTÉ device as the audio input and output.

TesiraFORTÉ Skype audio Options:

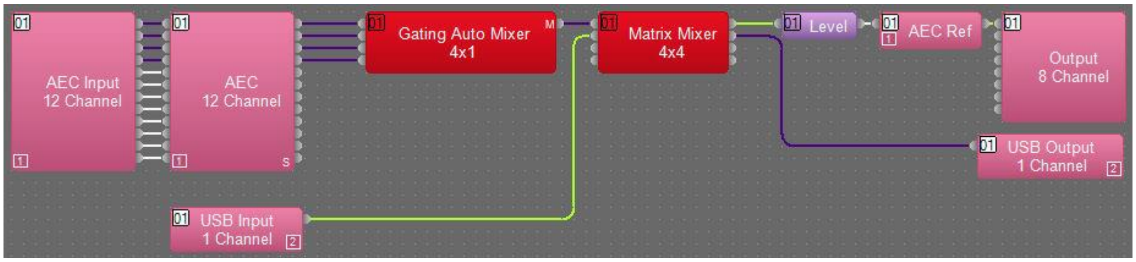
TesiraFORTÉ in Lync Audio Options:



Basic Layout

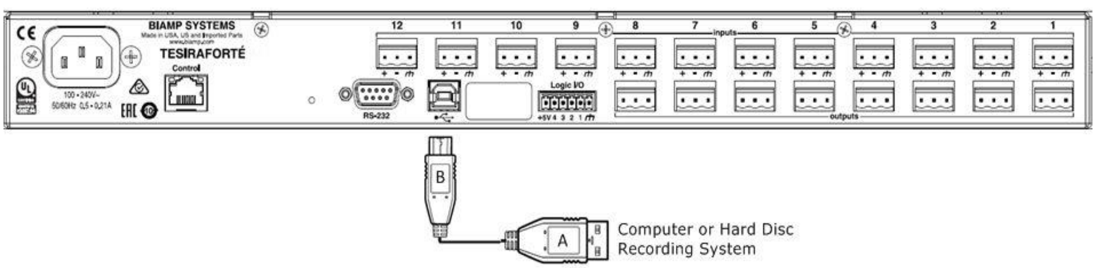
For an application using four microphones with no local Voice lift.
In the Layout below:

- the USB input and output blocks are configured to use Speakerphone.
- The Matrix mixer sends the USB input to the Room Output via the AEC reference block. Please note: as there is no local microphone reinforcement the AEC block has been placed 'inline'. The Green lines indicate the Soft Codec routing being used
- The Level Control is used as the master room audio - as the AEC Ref is inline, the AEC Ref and output signal level will track and AEC performance will remain optimized.
- Four Microphones are being used and the Matrix Mixer is sending the microphone signals to the USB output. The Purple lines indicate the Microphone routing being used.



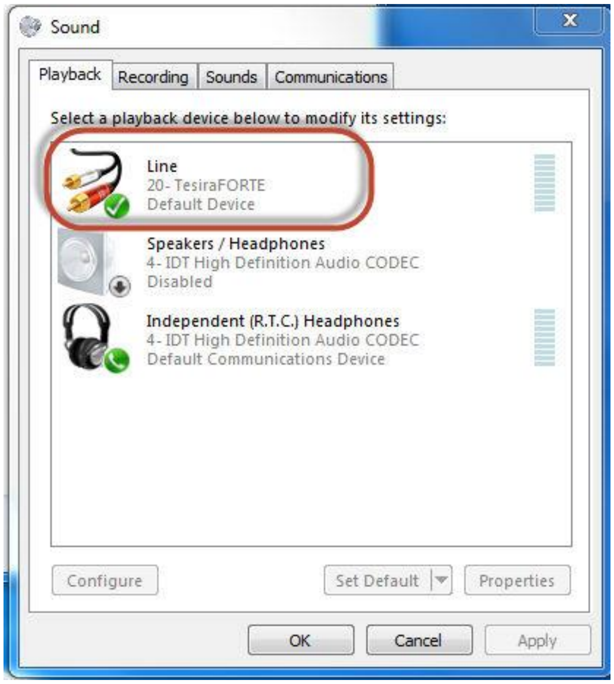
Line In/Out

This mode provides up to 8 channels of audio. Input channels, Output channels or both can be selected, and the number of channels can be specified. Combinations of 2, 4 or 6 total USB channels can operate in 24-bit or 16-bit mode, selected by the Bit Depth control. Combinations of 8 total USB channels operate in 16-bit mode only. When connected and configured, the TesiraFORTÉ device installs the chosen number of input and output channels in Windows, but they are not enabled by default. The channels can be enabled in the Windows Control Panel, and then selected as Record and Playback channels in the audio software application.

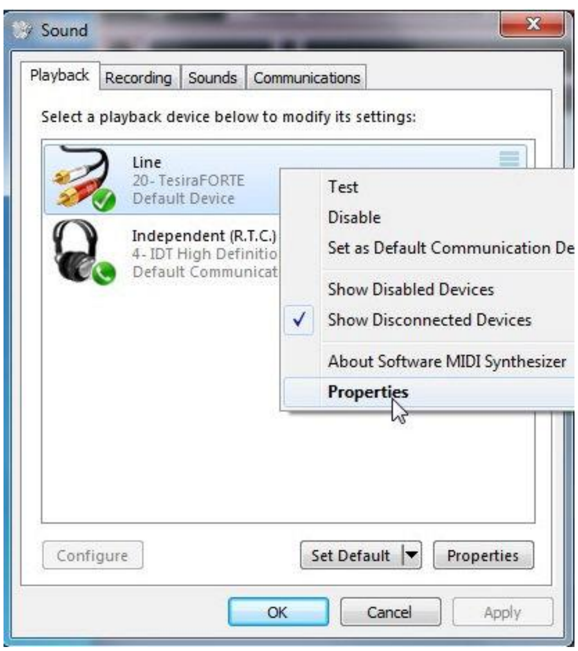


Channel settings

In Windows, right-click on the sound icon and select “Playback Devices”. This causes the Sound window to display, as shown. A Tesira device will present itself under Playback as “Line-#TesiraFORTE-Ready”.



Setting Speaker Properties - When the TesiraFORTÉ presents itself as a line in Windows, this corresponds to the USB input block in your layout. Right-click the line icon and select Properties

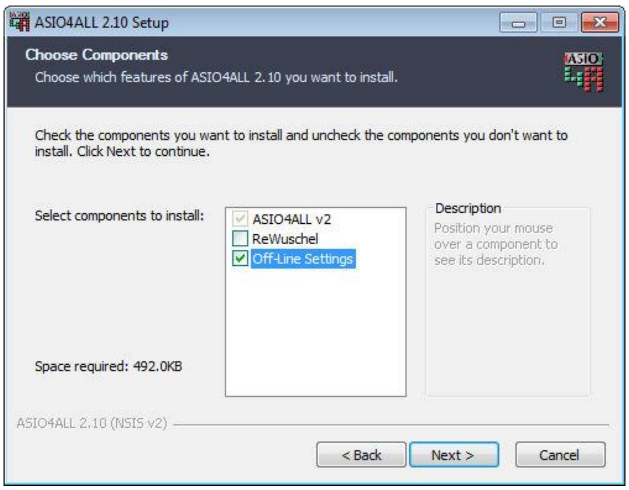


Within the properties window, select the Levels tab and note that the sound level may be set to some default value. Sometimes the sound from the PC is attenuated before arriving at the Tesira, and it may not be audible. You may either use this control or USB Input block on the Tesira layout to adjust the levels of the incoming audio.

Multi-channel audio

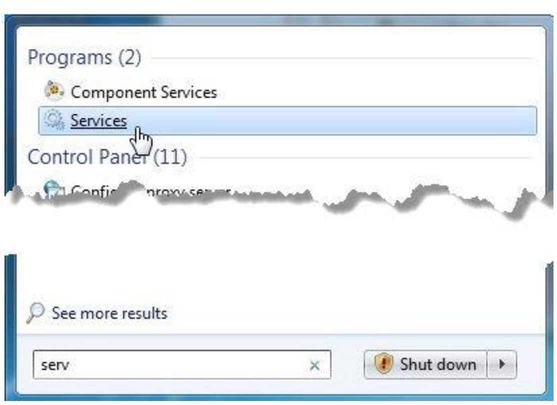
By default, Windows does not send as output or take as input the 8 channels of audio that the TesiraFORTÉ is capable of processing. It only handles two by default. Third party software is required to use more than two channels simultaneously. One technology that helps with this is the ASIO driver, which overcomes these limitations and is supported by a number of PC applications. Several ASIO drivers exist on the market. We have tested using ASIO4ALL Version 2.10 software, which may be downloaded at http://www.asio4all.com/

When installing ASIO4ALL Offline Mode must be enabled as part of the installation options. This is particularly important if using ForTheRecord Software to enable proper operation of all channels.

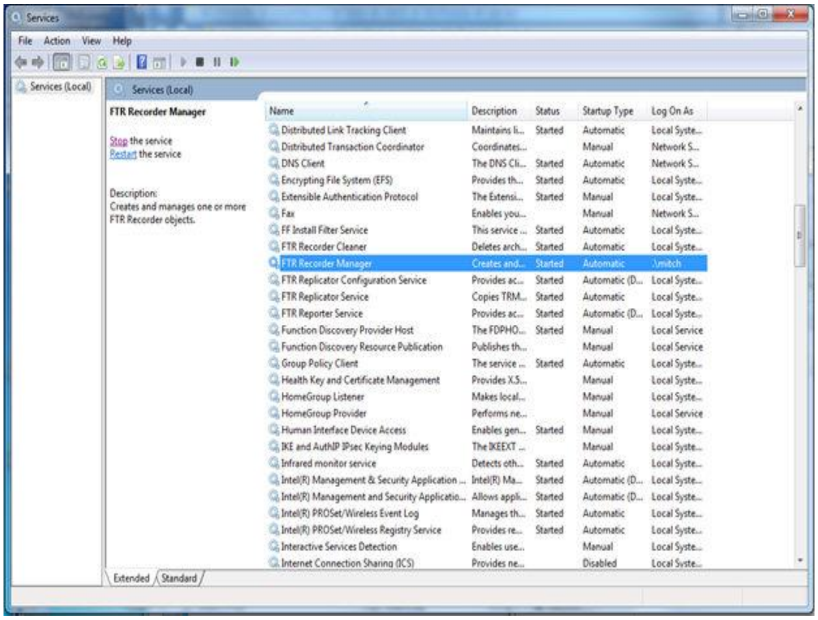


ForTheRecord settings

Special Configuration for **ForTheRecord** users
Besides the installation process for ASIO4ALL there are a few additional steps that a user must take to get ForTheRecord software to integrate properly with a Tesira Forte.
Install FTR Mixer - In order to record from more than 4 channels of audio from a Tesira, the FTR Mixer program is required. This is a requirement from ForTheRecord for any device that provides more than 4 channels of audio.
Set the FTRManager to logon as the local user - Go to the Windows Service application by typing Services using the search menu. This is available on the start menu from Window 7 or from the Windows-S key in windows 8.

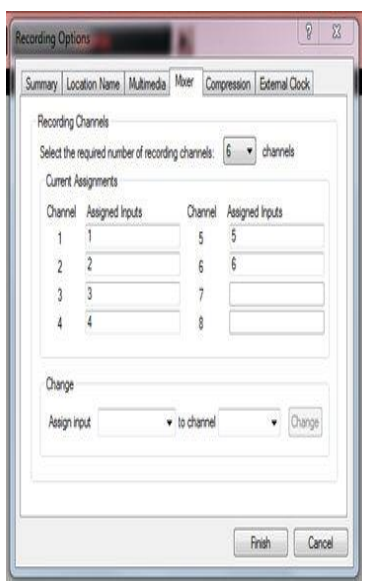


Select the FTR Manager service and right-click to view its properties



Within the FTR Manager Service's properties dialog, select the Logon tab and enter the username and password of the user that uses the FTR software. If you are on a domain, you must provide the domain and username. The user must also have an account that requires a password

After setting up the property, click okay and restart the computer. Stopping and starting the service isn't sufficient. After the system comes back, it will be able see multiple channels of audio in ForTheRecord software within the Tools->Options->Recording menu on the Mixer tab, if ASIO4ALL is selected as the sound card on the multimedia tab.



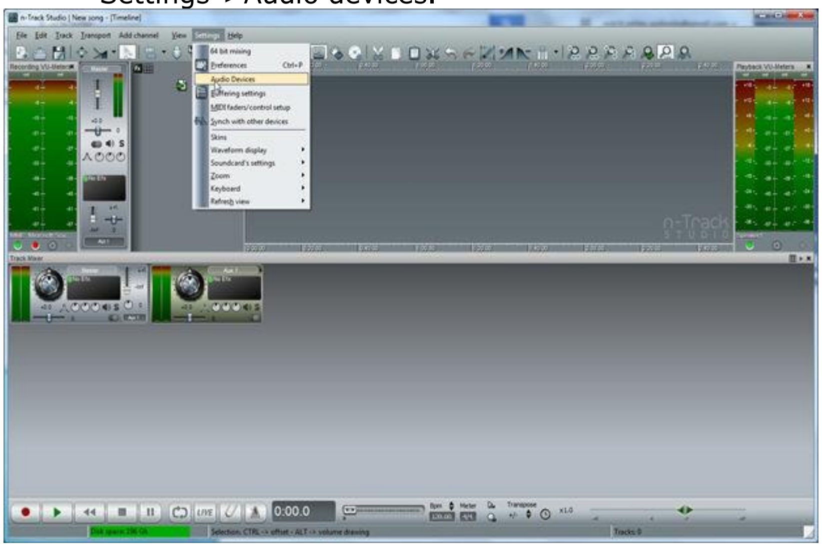
Other Audio interfacing options

you can also use an audio application that is compatible with ASIO audio that can *additionally* access the ASIO Control Panel. Before proceeding please install ASIO4ALL Version 2.10 software, which may be downloaded at http://www.asio4all.com/

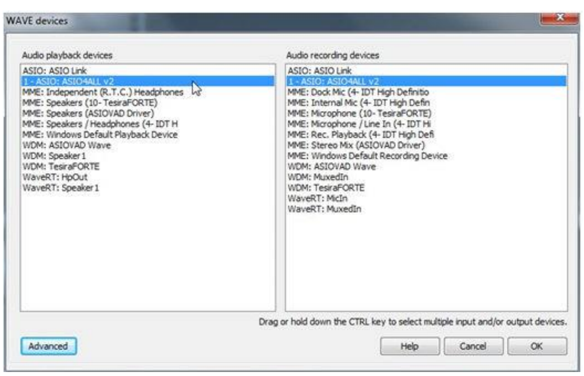
Using a Tesira with n-Track Studio 7

Ntrack Studio seven is available from http://ntrack.com/ n-Track Studio 7 is used to send audio through multiple channels using the ASIO driver.

- After installing the software described above, go into n-Track Studio and select Settings->Audio devices.

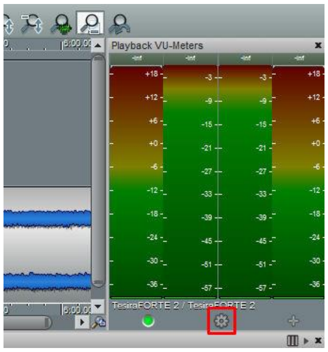


The list of devices you see will vary between computers, but select the ASIO4ALL devices. Although WDM: TesiraFORTÉ appears as a device, do not select this, as it only allows two channels to be visible. An ASIO driver is needed to access all Forte channels.



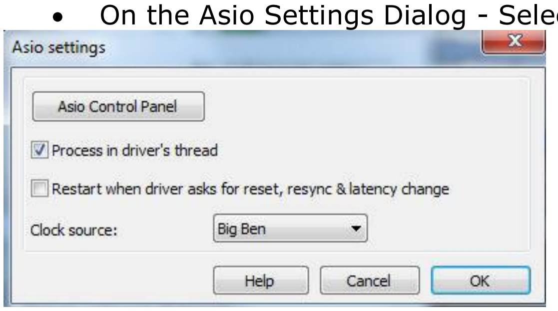
After using selecting ASIO4ALL as an audio device for playback or recording, put together the tracks, make sure that you've selected the Tesira FORTE as an IO device using the ASIO Control Panel. Various music applications provide access to the ASIO Control panel. The following steps describe how to access it using n-Track:

- On either the recording or playback VU meters, click on the gear icon at the bottom.

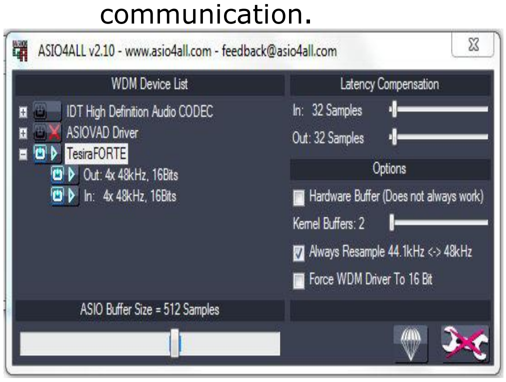


- In the audio format dialog that appears next, click on the button labeled ASIO settings.

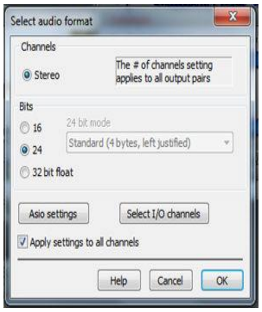
- On the Asio Settings Dialog - Select Asio Control Panel.



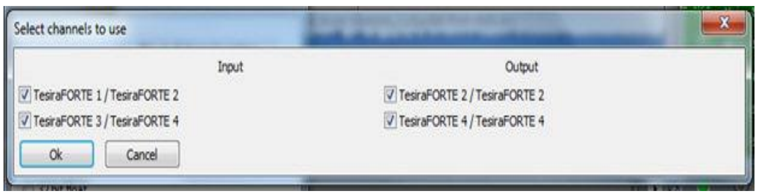
- Clicking the ASIO Control Panel reveals the ASIO control panel, which allows the user to select the TesiraFORTÉ as the Input/Output device used to which ASIO enables communication.



- The channels available should match the last layout sent to the Forte.
- Ensure that the TesiraFORTÉ device is selected, and exit out of the dialog windows on the screen. It is necessary to exit all dialogs to get subsequent settings to refresh.
- Click the Gear icon again, but this time click “Select I/O Channels” when the audio format dialog comes up.



This opens the channel selection dialog, which allows you to enable or disable channels on the TesiraFORTÉ for use with n-Track. Also note that Output on your PC corresponds to Input on the TesiraFORTÉ and vice versa. Select the channels on which you'd like to play or record audio on the TesiraFORTÉ.



Associating Audio with a Forte Channel

In the n-Track GUI, click the channel selection menu, which is to the left of the audio waveform representing a track. You'll see the tracks that you've enabled for output or input as part of the selection. From this point, you should be able to play to Tesira or record from it (or both), depending on your application.

Supported Network Topologies

Tesira devices support the use of many Ethernet network-based features to facilitate integrating Audio in an AV environment. Depending on the Tesira hardware, network port functions are separate or combined. For example Tesira Server, Server IO and TesiraFORTÉ AVB devices have dedicated Control Network and AVB network ports, while expanders use one Ethernet port and the control and AVB traffic is combined.

Individual network cables cannot exceed 100 meters in length, with the exception of fiber optic cables which can extend significantly further. All ports on Tesira devices are 100/1000Mb and are auto-sensing, so point-to-point or connections via network switches can use straight or crossover cables. All Ethernet cabling must be Cat5E or better.

A Tesira system will support the following communication and Audio protocols:

- Control Network Interface – Ethernet network used to relay configuration data between Server, Server I/O, TesiraFORTÉ and expander class Tesira devices.
- Audio-Video Bridging (AVB) Interface – Server, Server I/O, TesiraFORTÉ AVB and Expander class devices are able to share audio using AVB capable bridges (switches). Server class devices require the use of the AVB-1 card. Expander class devices support AVB functionality natively.
- CobraNet – Server and Server I/O devices using SCM-1 cards are able to share 32 x 32 channels of audio between any other CobraNet enabled devices.
- Dante - Server and Server I/O devices using DAN-1 cards are able to share 64 x 64 channels (32x32 flows) of audio between any other Dante enabled devices.
- Telephony Interfaces – Server I/O devices allow up to 6 telephony cards in any combination per device

Tesira supports the simultaneous use of AVB and CobraNet or Dante, however some network topologies are not supported. The following network topologies are supported:

- AVB only, single AVB network
- CobraNet only, single CobraNet network
- Dante only, single Dante network
- AVB & CobraNet, single AVB network & single CobraNet network
- AVB,CobraNet and Dante, single AVB network & multiple CobraNet networks

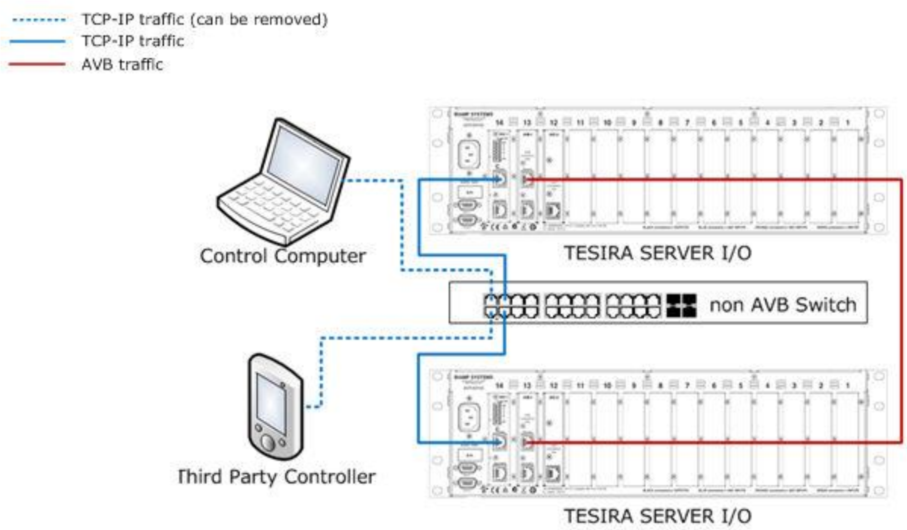
Note: While the Hardware supports SERVER IO devices to have 1 AVB and 2 CobraNet or Dante cards - this is not documented here. Please contact Biamp technical support if you require more details.

Backplane Timing

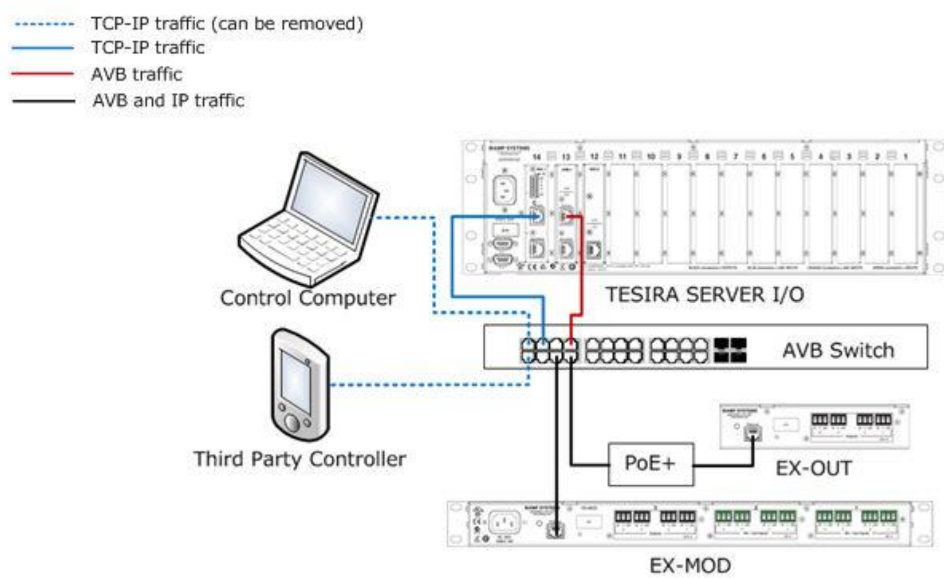
Tesira devices will assume they will be tasked with Backplane Clock timing. There is always a single network card used to provide a common clock to all servers -the network audio (AVB-1, DAN-1 or SNC-1) card in slot 13 of a Server IO or Slot 2 of a Server, and so that card must always be on the same network for all server, Server IO and TesiraFORTÉ AVB devices in a Tesira system sharing audio.

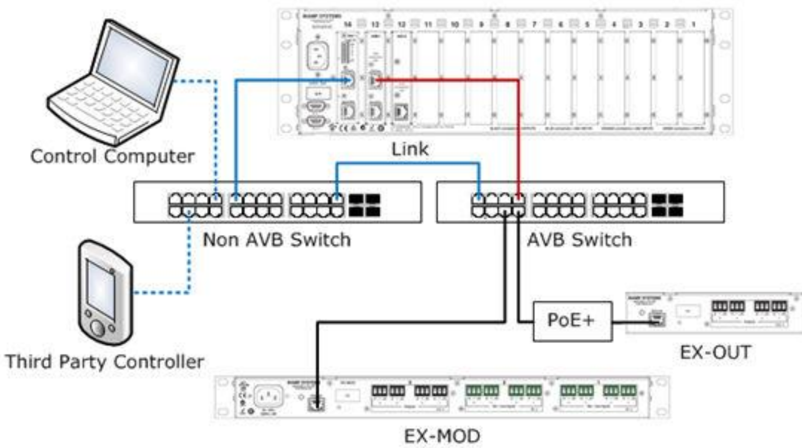
Single Network Wiring Topologies

Up to two Server, Server IO or TesiraFORTÉ AVB can be configured without the use of an AVB switch. The Control Network will use a switch to facilitate the use of a configuration PC and Control system. The Control Computer can be removed once the Tesira devices have their configuration loaded.

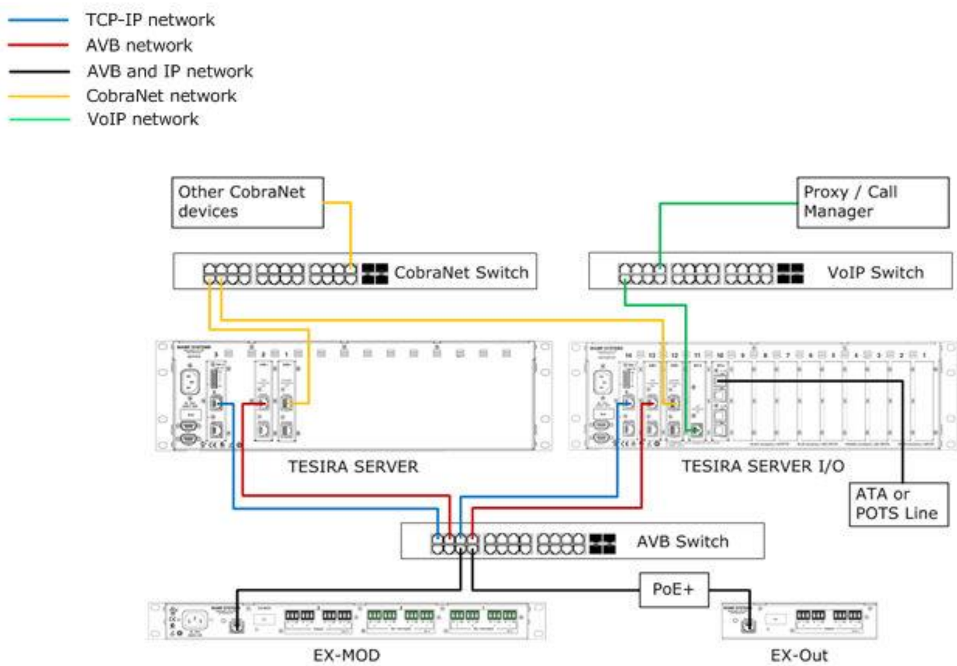


When using Tesira Server, Server IO,TesiraFORTÉ AVB and Expanders the use of an AVB Bridge (Switch) is required. The supported topology can be either to use one AVB switch and interface all Tesira control and AVB connections to it. Alternatively there is also the ability to use a separate non AVB control switch so long as there is a link between the Control and AVB switches.





Larger systems may use CobraNet interfaces for legacy devices, or telephone connections for conferencing applications. CobraNet and VoIP can be separate physical switches or separate "port based" (i.e. non-tagged) VLANs with the same switches.



Multiple Network Wiring Topologies

Multiple AVB networks are not supported under any circumstances. Multiple CobraNet or Dante networks are supported within limits only when AVB is also being used.
The diagrams below illustrate examples of supported network topologies.

- Red lines indicate AVB Network connections
- Blue lines indicate Control Network connections
- Black Lines indicate AVB and control Data networks
- Yellow lines indicate CobraNet Network connections.
- Purple lines indicate Dante Network Connections

AVB Only - Single Network

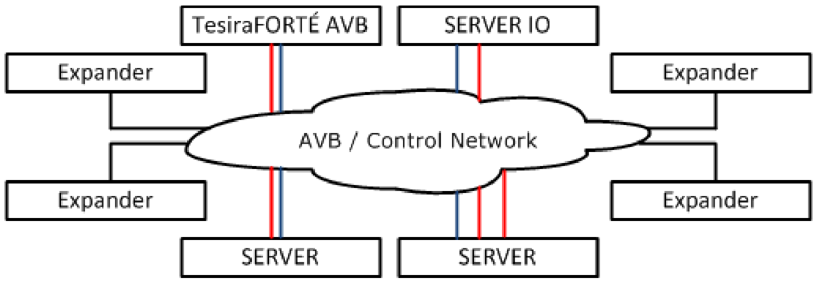
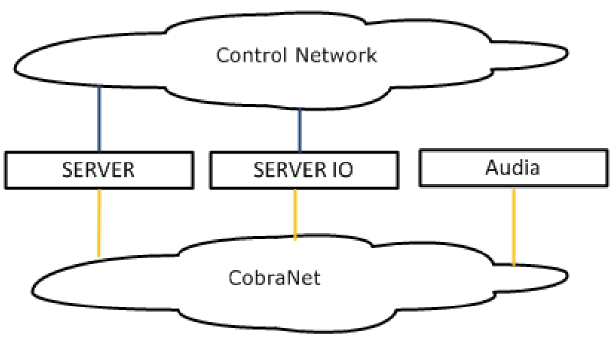


Figure 1 - AVB only, single AVB network

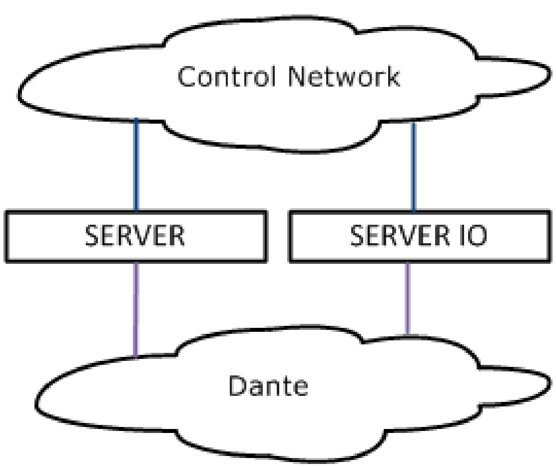
CobraNet or Dante only- single network

When no AVB cards are use in the Server or Server IO, each device will support up to two CobraNet or Dante card connection to a single network from a single device.
Note: If connecting more than one Dante or CobraNet card to a network from a single device please contact Biamp technical support for more details and programming assistance.

CobraNet topology

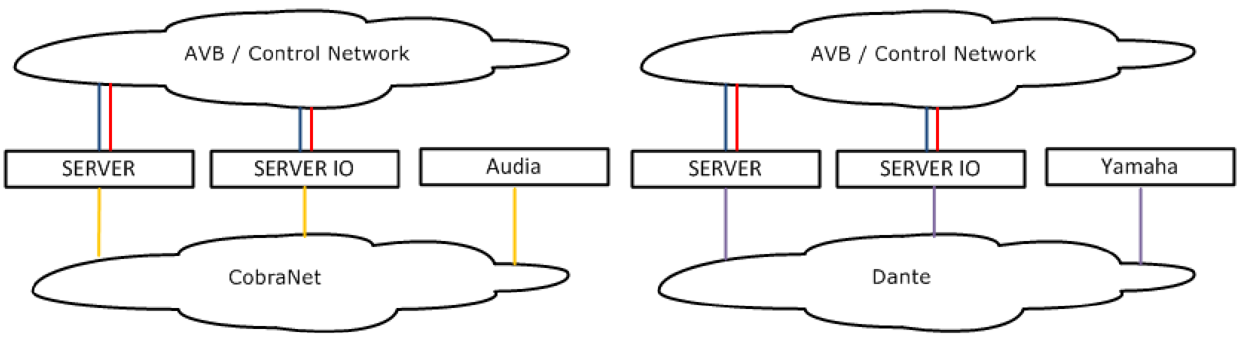


Dante Topology

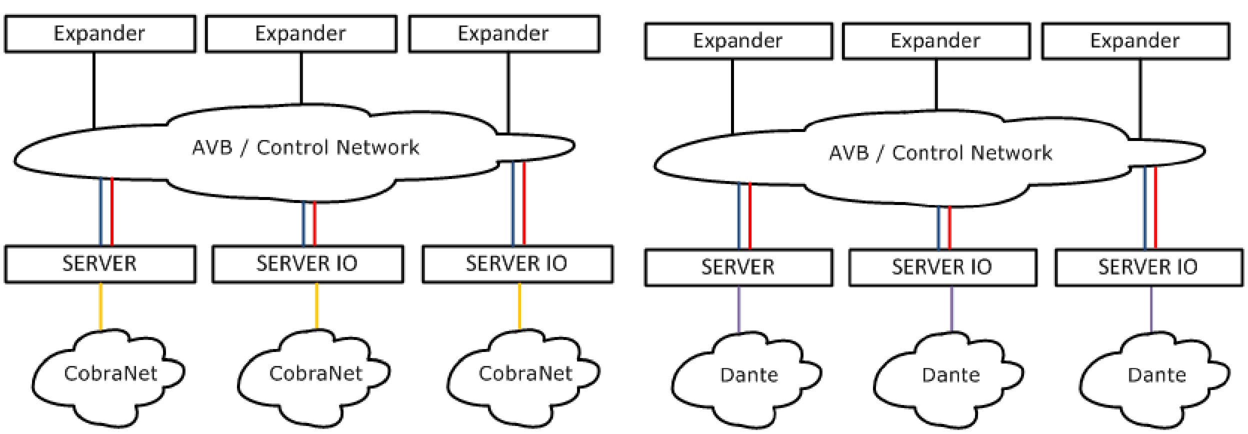


AVB and CobraNet or Dante- single networks

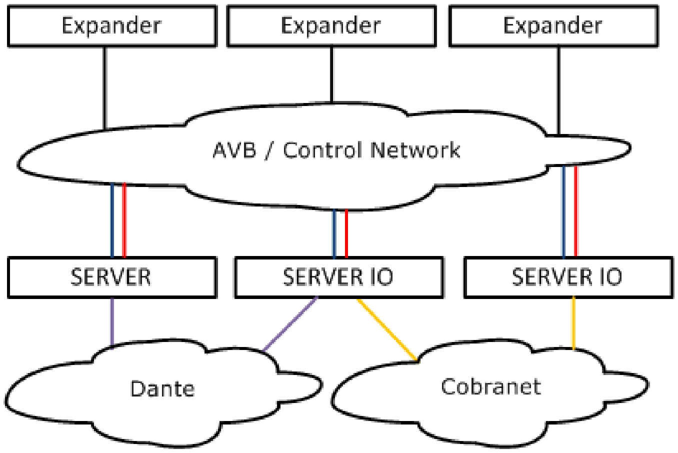
When AVB cards are use in the Server or Server IO, each device will support up to two CobraNet or Dante connections to a single network from a single device.
Note: If connecting more than one Dante or CobraNet card to a network from a single device please contact Biamp technical support for more details and programming assistance.



AVB and CobraNet or Dante - Multiple Networks



AVB, CobraNet and Dante - Single Networks



Hardware

Hardware Front Panel Status lights

Server and Server I/O Front Panel

led	Off	Green	Yellow	Red
Power	Unit not powered	Unit is powered	Not applicable	Not applicable
Alarm	No device fault active	Not applicable	Minor device fault active	Major device fault active
Activity	Not applicable	The device is active part of active system	Not applicable	Device is inactive part of a system (Audio is stopped)
Status	Not applicable	Device configured. Ready to participate in the system	Device unconfigured. Ready to receive configuration	Device not ready to receive its configuration
AIS (Alarm in System)	No fault is active in any device in the system	Not applicable	Minor fault is active in a device in the system	Major fault is active in a device in the system

TesiraFORTÉ Front Panel

led	Off	Green	Yellow	Red
Power	Unit not powered	Unit is powered	Not applicable	Not applicable
Alarm	No device fault active	Not applicable	Minor device fault active	Major device fault active
Activity	Not applicable	The device is active part of active system	Not applicable	Device is inactive part of a system (Audio is stopped)
Status	Not applicable	Device configured. Ready to participate in the system	Device unconfigured. Ready to receive configuration	Device not ready to receive its configuration
AIS (Alarm in System)	No fault is active in any device in the system	Not applicable	Minor fault is active in a device in the system	Major fault is active in a device in the system

Ex-AEC, Ex-In, Ex-Out and EX-IO and EX-MOD Front Panel

The front panel of the 4 channel expander displays four multicolor LEDs that provide information about the status of the expander.

led	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No fault is active in the device	Not applicable	Minor fault active in the device	Major fault active in the device
Activity	Not applicable	The device is an active part of an active system	Not applicable	Device is part of an inactive system (Audio is stopped)
Status	Not applicable	Device has received configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device not ready to receive its configuration

Lab.gruppen amplifier Front Panel

Chassis Front Panel LED indicators

LED	Off	Green	Yellow	Red
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an active part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration
Power	Device is powered Off	Power State is ON	Power State is STANDBY	Power State is SLEEP

Amplifier channel Front Panel LED indicators

LED	Off	Green	Yellow	Red
Load	Amplified Output Load Status UNKNOWN	Amplified Output Load Status OK	Amplified Output Load Status WARNING	Amplified Output Load Status ERROR
Amp	Amplified Output Amp Status UNKNOWN	Amplified Output Amp Status OK	Amplified Output Amp Status WARNING	Amplified Output Amp Status ERROR
Signal	Amplified Output Signal Status UNKNOWN	Amplified Output Signal Status OK	Amplified Output Signal Status WARNING	Amplified Output Signal Status ERROR

Mute	Power state is standby	Audio is not muted	Audio is muted	Not applicable

Ex-Logic Front Panel

The front panel of the Ex-Logic expander displays four multicolor LEDs that provide information about the status of the expander.

led	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No fault is active in the device	Not applicable	Minor fault active in the device	Major fault active in the device
Activity	Not applicable	The device is an active part of an active system	Not applicable	Device is part of an inactive system
Status	Not applicable	Device has received configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device not ready to receive its configuration

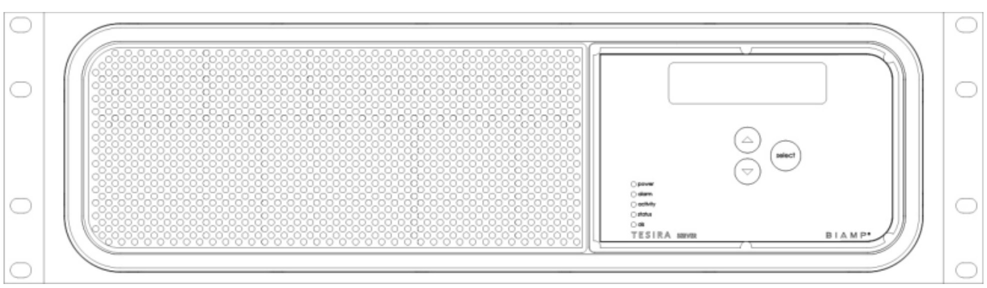
Server Hardware

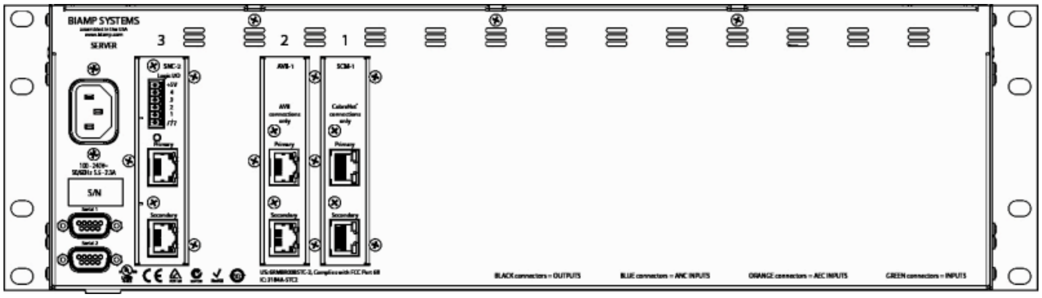
Server

The Tesira SERVER delivers highly scalable processing that can grow over time with the needs of the end customer — with up to eight DSP-2 cards in a single chassis.
The SERVER supports a maximum of (2) AVB-1 cards. The SERVER ships with (1) AVB card and can additionally have either: a second AVB-1 card or (1) SCM-1 CobraNet card, (1) Dante card or (1) analog Tesira I/O card of any type added in the second I/O card slot.
The SERVER is built for maximum flexibility, enabling I/O devices to be located at end points. Any I/O device can harness available processing.
An integral network card provides redundant network connection for configuration setup and control of the Tesira network as well as GPIO connection.
The Tesira SERVER acts as the host device for Tesira EX-MOD and Expander Hardware and EX-Logic devices.
SNC-2a cards are required to support redundancy.

Features

- Supports up to 8 DSP-2 cards
- Configured with a SNC-2 (prior to march 2014) or SNC-2a by default to enable network communications.
- 420 x 420 channels of I/O per AVB-1 card
- Optional 32 x 32 CobraNet audio networking using the SCM-1 card
- Supports Dante - Optional 64 x 64 Dante audio networking using the DAN-1 card
- Configuration and control networking over Ethernet
- Supports network redundancy
- Local GPIO connections - used for Logic IO or if part of a Redundant pair
- Front panel LCD display for device and system information
- Signal processing via intuitive software allows configuration and control for: signal routing and mixing, equalization, filtering, dynamics, delay and much more
- Extensive expansion devices (Input, Output, Logic, etc) available as part of the Tesira digital audio networking platform
- CE marked, UL listed, RoHS compliant





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration
AIS (Alarm In System	No faults active in any device in the system	Not applicable	Minor fault is active in a device in the system	Major fault is active in a device in the system

Server-IO

The Tesira® SERVER-IO is a digital network server for use with the Tesira digital audio networking platform. It is factory configured with (1) DSP-2 card and is capable of handling up to (3) DSP-2 cards per chassis. The SERVER-IO allows a maximum of (1) AVB-1 card per chassis. The SERVER-IO allows a maximum of (2) SCM-1 CobraNet cards per chassis. The SERVER-IO allows a maximum of (2) DAN-1 Dante cards per chassis.
AVB-1 cards can only reside in slot 13. SCM-1 CobraNet and DAN-1 Dante cards can only reside in slots 11, 12, and 13. All other card types can reside in slots 1-12.

- When configured with an AVB-1 card, the SERVER-IO will support the addition of up to (12) total I/O cards including up to (2) CobraNet SCM-1 cards and up to (2) Dante DAN-1 card. In this case, slot 13 can *only* be populated by an AVB-1 card. If an AVB-1 card is present and (1) DAN-1 card is present, only (1) SCM-1 card can be present in the chassis.
- When configured without an AVB-1 card, the SERVER-IO will support the addition of up to (13) total I/O cards including up to (2) CobraNet SCM-1 cards and up to (2) Dante DAN-1 card. In the event an SCM-1 CobraNet card is present, slot 13 can *only* be populated by a SCM-1 CobraNet card.

The SERVER-IO can support up to (12) standard Tesira I/O cards for up to (48) channels of audio I/O (e.g. mic and line level, VoIP, and telephone interface). The on-board DSP features two new Biamp algorithms, SpeechSense™ and AmbientSense™, both of which enhance speech processing by more accurately distinguishing between human speech and noise. The DSP also provides extensive audio processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay as well as control, monitoring and diagnostic tools; all configured through the Tesira design software.
An integral Ethernet network card SNC-1 provides redundant network connectivity for configuration and control of the Tesira network as well as GPIO connection.
The Tesira SERVER-IO acts as the host device for Tesira EX-MOD and Expander Hardware and EX-Logic devices.

Allowable I/O configurations may be confirmed using the "Tesira Server-IO Order Form" available at www.Biamp.com on the Tesira downloads page.

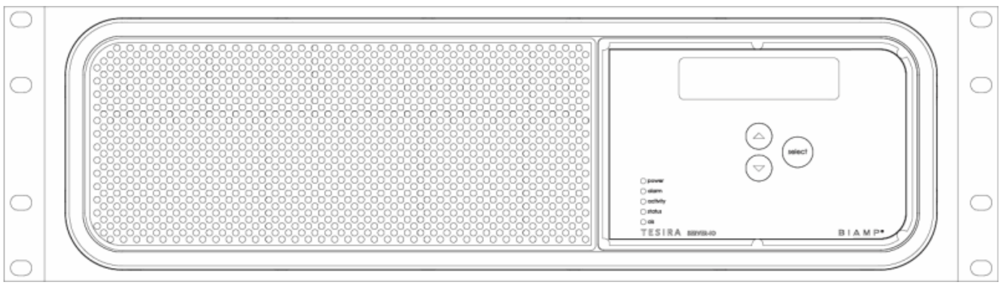
BENEFITS

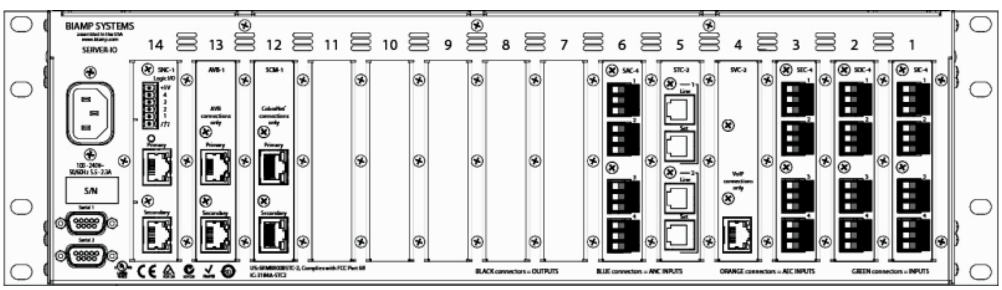
- Offers flexibility to have scalable DSP and I/O in the same device
- Enables I/O to be distributed from a central location
- Customizable I/O configurations for easy right-sizing of system design
- Control networking can run on separate (existing) Ethernet network

FEATURES

- Supports up to 3 DSP-2 cards
- Configured with a SNC-1 by default to enable network communications.
- Up to 12 I/O cards with a maximum of 48 channels of audio including:
    - SEC-4 - 4 Channel Acoustic Echo Cancellation card (also includes AGC and ANC)
    - SAC-4 - 4 Channel Ambient Noise Compensation card
    - SIC-4 - 4 channel Mic Line Input Card
    - SOC-4 - 4 channel Mic Line Output Card
    - STC-2 - 2 Line Telephone Interface card
    - SVC-2 - 2 Line VoIP Interface card

  - **Note - Up to 6 telephony cards (STC-2, SVC-2) can be used in any combination, per SERVER-IO**
- Up to 420 x 420 channels of digital I/O using AVB-1 card (only 1 allowed per SERVER-IO chassis)
- Supports optional 32 x 32 CobraNet audio networking using SCM-1 card (up to 2 allowed per SERVER-IO chassis)
- Supports up to 128 x 128 channels of Dante audio using DAN-1 cards (up to 2 allowed per SERVER-IO chassis)
- System configuration and control via Ethernet or serial connection
- Local General Purpose Input/Output (GPIO) connections
- Front panel OLED display for device and system information
- New processing algorithms: SpeechSense and AmbientSense
- Signal processing via intuitive software allows configuration and control for:
- signal routing and mixing, equalization, filtering, dynamics and delay and much more
- Rack mountable (3RU)





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

AIS (Alarm In System	No faults active in any device in the system	Not applicable	Minor fault is active in a device in the system	Major fault is active in a device in the system

Server Cards

AVB-1

The Tesira AVB-1 is a modular digital audio networking card for use with Tesira SERVER and SERVER-IO devices. The AVB-1 allows a Tesira system to send and receive digital audio over an Ethernet network utilizing approved AVB switches. AVB (Audio Video Bridging) is an open standard protocol compliant with IEEE standards.

In addition to allowing up to 420 x 420 channels of interconnectivity over AVB from any other compliant device, Tesira uses AVB as the interconnect between Server, Server-IO TesiraFORTÉ devices and EX-MOD, EX-AEC, EX-In, EX-Out and EX-IO remote expander devices.

Please refer to the AVB Network Considerations and AVB 1722.1 Explicit Streams sections for more information.

DAN-1

The Tesira DAN-1 is a modular Dante digital audio networking card for use with Tesira SERVER and SERVER-IO devices. Dante is a proprietary media networking solution developed by Audinate and is fully compliant with IEEE standards on 100Mbps and Gigabit networks. The DAN-1 allows a Tesira system to send and/or receive digital audio from other Dante endpoints over an Ethernet network utilizing standard network switches.

The DAN-1 allows up to 64x64 channels of interconnectivity with any other Dante device, including non-Biamp hardware. Audinate's Dante Controller software is required to complete stream assignments for a Tesira configuration. Sample rates are 44.1/48kHz by default, other rates may be available as determined by Dante Controller and connected devices. A Dante endpoint can simultaneously send and receive streams from more than one device. A secondary Dante port is available for network redundancy.

Tesira DAN-1 hostnames will be unique, following the convention **TesiraServernnnnnnnn-Slotnn** where the Tesira's Serial Number and Card Slot Number are appended to the string "TesiraServer".

- A single DAN-1 card is allowed per SERVER.
- Up to (2) DAN-1 cards are allowed per SERVER-IO.

Please refer to the Dante Networking section for more information.

DSP-2

The DSP-2 cards are used in Tesira Server class devices.

Each Tesira Server allows the installation of up to 8 DSP-2 cards.

Each Tesira Server I/O allows the installation of up to 3 DSP-2 cards.

SAC-4

The Tesira SAC-4 is a modular analog input card for use with Tesira SERVER and SERVER-IO devices. Each SAC-4 provides four channels of microphone or line level audio input with Ambient Noise Compensation. The ANC processing features the new Biamp algorithm, AmbientSense, which intelligently distinguishes ambient noise from program material or announcements, thus greatly improving performance over traditional ambient noise compensation.

SCM-1

The Tesira SCM-1 is a modular digital audio networking card for use with Tesira SERVER and SERVER-IO devices. The SCM-1 allows a Tesira system to send and receive digital audio using the CobraNet networking standard. Each SCM-1 CobraNet card allows for interconnectivity of 32 x 32 channels of digital audio. Implementing the SCM-1 allows a Tesira system to share audio with Biamp Audia® and Vocia® systems as well as other devices operating on a CobraNet network.

Please refer to the CobraNet Network Considerations section for more information.

SEC-4

The Tesira SEC-4 is a modular analog input card for use with Tesira SERVER or SERVER-IO devices. Each SEC-4 provides 4 channels of mic or line level audio input with Acoustic Echo Cancellation. The SEC-4 utilizes the next generation of the proprietary Sona™ algorithm and also features two new Biamp algorithms, SpeechSense and AmbientSense which enhance speech processing by more accurately distinguishing between human speech and noise.

SIC-4

The Tesira SIC-4 is a modular analog input card for use with Tesira SERVER and SERVER-IO devices. Each SIC-4 provides 4 channels of mic or line level audio input. The inputs are electrically balanced and provided on plug-in barrier strip connectors. Software control of each input includes gain with clip indicator, +48V phantom power, mute, level and signal invert.

SNC-1

The Tesira SNC-1 is a Network Controller card for Tesira SERVER-IO devices. A SERVER-I/O will always be configured with a SNC-1 card as it is used to facilitate communications between external equipment and a Tesira SERVER-I/O. The SNC-1 is not compatible with the Tesira SERVER devices which use SNC-2 cards.

SNC-2

The Tesira SNC-2 is a Network Controller card for Tesira SERVER devices. A Tesira SERVER will always be configured with a SNC-2 card as it is used to facilitate communications between external equipment and a Tesira SERVER. The SNC-2 is not compatible with the Tesira SERVER-IO devices which use SNC-1 cards.

SOC-4

The Tesira SOC-4 is a modular analog output card for use with Tesira SERVER or SERVER-IO devices. Each SOC-4 provides 4 channels of line level audio output. The outputs are electrically balanced and provided on plug-in barrier strip connectors. Software control of each output includes mute, level, signal invert and full-scale output reference.

STC-2

The Tesira STC-2 is a modular telephone interface card for use with Tesira SERVER and SERVER-IO devices. The STC-2 allows a Tesira system to connect directly to up to two standard analog telephone lines. Being more than just a normal "hybrid," each channel includes line-echo cancellation, noise suppression, caller ID decoding, ring detection/validation, DTMF decoding, and call progress tone decoding.

The device is supported via software using the Telephone Interface Component Object. Each STC Input is also able the be associated with the DTMF Decode and Dialer blocks.

Up to 6 STC-2 cards are supported by each Server I/O.

SVC-2

The Tesira SVC-2 is a modular Voice over Internet Protocol (VoIP) card for use with Tesira SERVER and SERVER-IO devices. The SVC-2 allows a Tesira system to connect directly to IP-based telephone systems.

The device is supported via software using the VoIP Phone Component Object. The VoIP card is also able the be associated with the DTMF Decode and Dialer blocks.

Up to 6 SVC-2 cards are supported by each Server I/O.

TesiraFORTÉ

TesiraFORTÉ Hardware

TesiraFORTÉ offers fixed input, output, and DSP hardware configurations per device. All models offer 12 analog inputs, 8 analog outputs, and USB connectivity.

- TesiraFORTÉ AI - Analog Interface
- TesiraFORTÉ CI - Conference Interface, featuring 12 channels of AEC input
- TesiraFORTÉ TI - Telephone Interface, featuring 12 channels of AEC input and a single analog telephone line via RJ-11 port
- TesiraFORTÉ VI - VoIP Interface, featuring 12 channels of AEC input and 2 channels of SIP VoIP connectivity via RJ-45 port

All TesiraFORTÉ models can be ordered with or without AVB capability. AVB-capable TesiraFORTÉ devices are able to utilize Tesira EX-MOD and Expander Hardware devices. All TesiraFORTÉ devices can be used with Tesira EX-Logic and TEC-1 devices.

FORTÉ AI

The TesiraFORTÉ AI is a digital audio server with 12 analog inputs and 8 analog outputs and includes up to 8 channels of configurable USB audio. USB audio allows TesiraFORTÉ to interface directly with USB audio hosts, as well as to take full advantage of today's most sophisticated conferencing solutions.
TesiraFORTÉ AVB AI adds Audio Video Bridging (AVB) digital audio networking. The AVB model can be used as a standalone device or can be combined with other TesiraFORTÉ devices and Tesira servers, expanders, and controllers. TesiraFORTÉ AI also provides extensive audio processing, including but not limited to: signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools; all configured through the Tesira configuration software. TesiraFORTÉ AI is best-suited for small- to medium-sized rooms that require high-quality audio solutions using voice lift and mix-minus, such as conference rooms or council chambers.

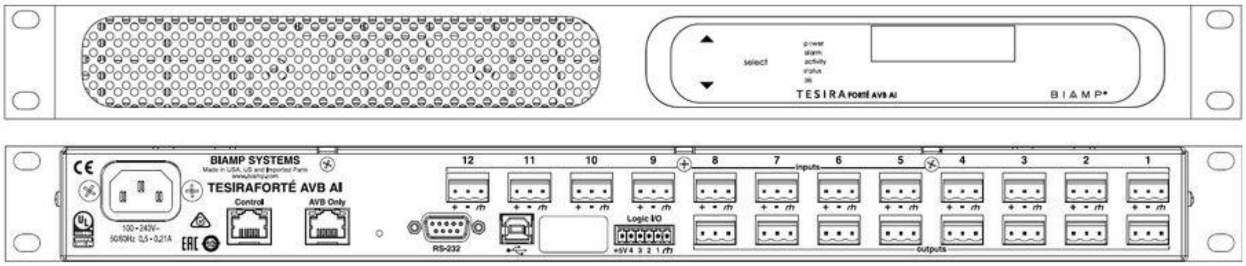
Benefits

- Allows integrators to choose which model works best for the installation environment.
- Application-specific models make system design, configuration, and installation easier and faster.
- Included default configuration file allows for plug-and-play usage. Highly scalable and cost-effective solution that can grow over time with
- the needs of the customer.
- SpeechSense™ technology enhances speech processing.
- Integrates directly with soft codecs and other USB audio hosts.

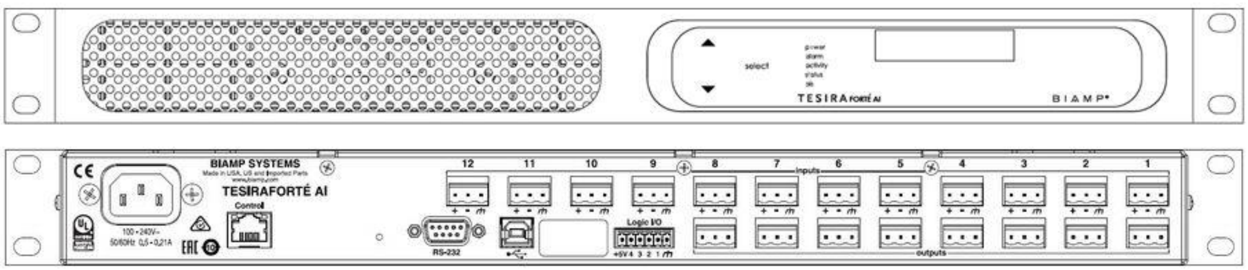
Features

- 128 x 128 channels of AVB (AVB model only)
- 12 mic/line level inputs, 8 mic/line level outputs
- Gigabit Ethernet port
- Up to 8 channels of configurable USB audio
- RS-232 serial port
- 4-pin GPIO
- 2-line LED display with capacitive-touch navigation
- Rack mountable (1RU)
- System configuration and control via Ethernet
- Internal universal power supply
- Fully compatible with Tesira servers, expanders, and controllers (AVB model)
- Signal processing via intuitive software allows configuration and control for signal routing, mixing, equalization, filtering, delay and much more
- CE marked, UL listed, and RoHS compliant
- Covered by Biamp Systems' 5-year warranty

TesiraFORTÉ AI AVB Front and Back Panels



TesiraFORTÉ AI Front and Back Panels



FORTÉ CI

The TesiraFORTÉ CI is a digital audio server with 12 analog inputs and 8 analog outputs and includes Sona™ Acoustic Echo Cancellation (AEC) technology on all 12 inputs.It also includes up to 8 channels of configurable USB audio. USB audio allows TesiraFORTÉ to interface directly with USB audio hosts, as well as to take full advantage of today's most sophisticated conferencing solutions.
TesiraFORTÉ AVB CI adds Audio Video Bridging (AVB) digital audio networking. The AVB model can be used as a standalone device or can be combined with other TesiraFORTÉ devices and Tesira servers, expanders, and controllers. TesiraFORTÉ CI also provides extensive audio processing, including but not limited
to: Sona™ AEC technology, signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools; all configured through the Tesira configuration software. TesiraFORTÉ CI is best suited for small- to medium-sized rooms that require high-quality audio solutions using AEC, voice lift, and mix-minus, such as conference rooms or distance learning environments.

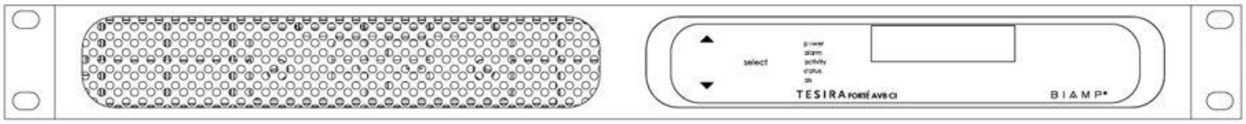
Benefits

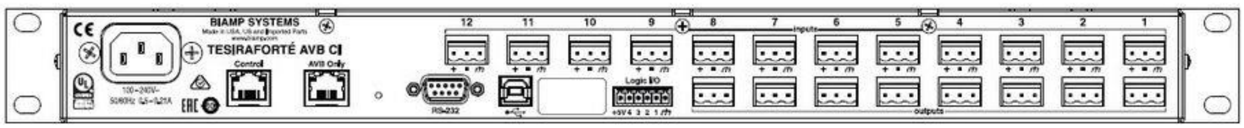
- Allows integrators to choose which model works best for the installation environment.
- Application-specific models make system design, configuration, and installation easier and faster.
- Included default configuration file allows for plug-and-play usage.
- Highly scalable and cost-effective solution that can grow over time with the needs of the customer.
- Sona™ AEC and SpeechSense™ technologies to enhance speech processing.
- Integrates directly with soft codecs and other USB audio hosts.

Features

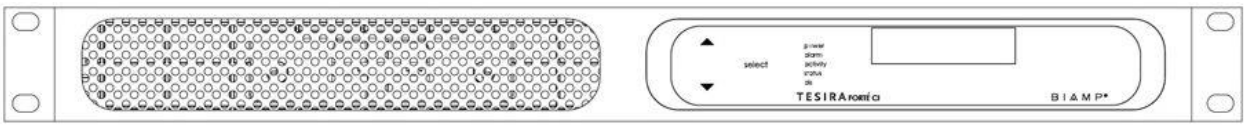
- 128 x 128 channels of AVB (AVB model only)
- 12 mic/line level inputs with AEC, 8 mic/line level outputs
- Gigabit Ethernet port
- Up to 8 channels of configurable USB audio
- RS-232 serial port
- 4-pin GPIO
- 2-line LED display with capacitive-touch navigation
- Rack mountable (1RU)
- System configuration and control via Ethernet
- Internal universal power supply
- Fully compatible with Tesira servers, expanders, and controllers (AVB model)
- Signal processing via intuitive software allows configuration and control for signal routing, mixing, equalization, filtering, delay and much more
- CE marked, UL listed, and RoHS compliant
- Covered by Biamp Systems 5-year warranty

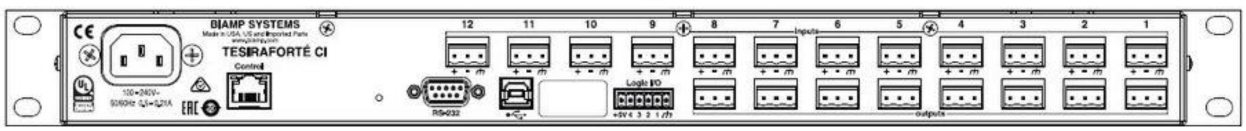
TesiraFORTÉ CI AVB Front and Back Panels





TesiraFORTÉ CI Front and Back Panels





FORTÉ TI

The TesiraFORTÉ TI is a digital audio server with 12 analog inputs and 8 analog outputs and includes Sona™ Acoustic Echo Cancellation (AEC) technology on all 12 inputs. It also includes up to 8 channels of configurable USB audio, and a standard telephone interface via a RJ-11 connector. USB audio allows TesiraFORTÉ to interface directly with USB audio hosts, as well as to take full advantage of today's most sophisticated conferencing solutions.
TesiraFORTÉ AVB TI adds Audio Video Bridging (AVB) digital audio networking. The AVB model can be used as a standalone device or can be combined with other TesiraFORTÉ devices and Tesira servers, expanders, and controllers.
TesiraFORTÉ TI also provides extensive audio processing, including but not limited to: Sona™ AEC technology, signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools; all configured through the Tesira configuration software. TesiraFORTÉ TI is best-suited for small- to medium-sized rooms that require high-quality audio solutions using voice lift, mix-minus, and AEC such as conference rooms or training facilities that require a standard telephone interface.

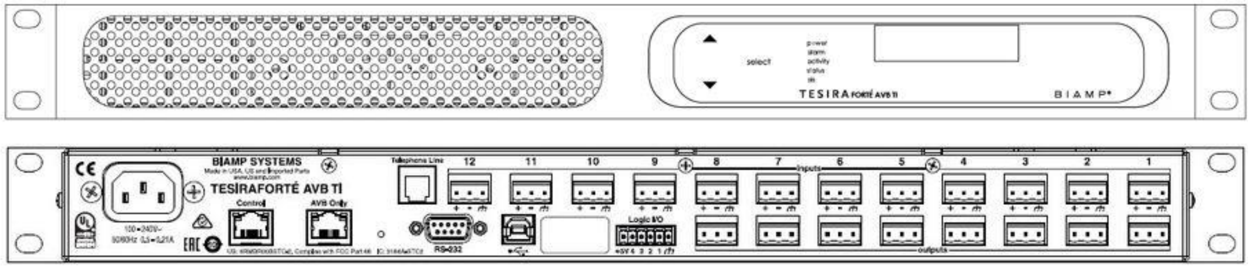
Benefits

- Allows integrators to choose which model works best for the installation environment.
- Application-specific models make system design, configuration, and installation easier and faster.
- Included default configuration file allows for plug-and-play usage.
- Highly scalable and cost-effective solution that can grow over time with the needs of the customer.
- Sona™ AEC and SpeechSense™ technologies to enhance speech processing.
- Integrates directly with soft codecs and other USB audio hosts.

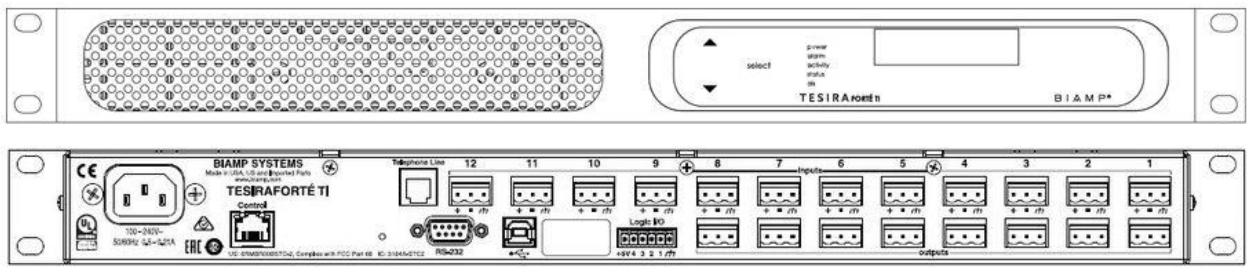
Features

- 128 x 128 channels of AVB (AVB model only)
- 12 mic/line level inputs with AEC, 8 mic/line level outputs
- Gigabit Ethernet port
- Up to 8 channels of configurable USB audio
- RS-232 serial port
- 4-pin GPIO
- 2-line LED display with capacitive-touch navigation
- Rack mountable (1RU)
- System configuration and control via Ethernet
- Internal universal power supply
- Standard telephone interface via a RJ-11 connector
- Fully compatible with Tesira servers, expanders, and controllers (AVB model)
- Signal processing via intuitive software allows configuration and control for signal routing, mixing, equalization, filtering, delay and much more
- CE marked, UL listed, and RoHS compliant
- Covered by Biamp Systems' 5-year warranty

Tesira Forte TI AVB Front and Back Panels



Tesira Forte TI Front and Back Panels



FORTÉ VI

The TesiraFORTÉ VI is a digital audio server with 12 analog inputs and 8 analog outputs and includes Sona™ Acoustic Echo Cancellation (AEC) technology on all 12 inputs. It also includes up to 8 channels of configurable USB audio, and a 2-channel VoIP interface via a RJ-45 connector. USB audio allows TesiraFORTÉ to interface directly with USB audio hosts, as well as to take full advantage of today's most sophisticated conferencing solutions. TesiraFORTÉ AVB AI adds Audio Video Bridging (AVB) digital audio networking. The AVB model can be used as a standalone device or can be combined with other TesiraFORTÉ devices and Tesira servers, expanders, and controllers.
TesiraFORTÉ VI also provides extensive audio processing, including but not limited to: Sona™ AEC technology, signal routing and mixing, equalization, filtering, dynamics, and delay, as well as control, monitoring, and diagnostic tools; all configured through the Tesira configuration software. TesiraFORTÉ VI is best-suited for small- to medium sized rooms that require high-quality audio solutions using VoIP, voice lift, mix-minus, and AEC such as board rooms or distance training facilities.

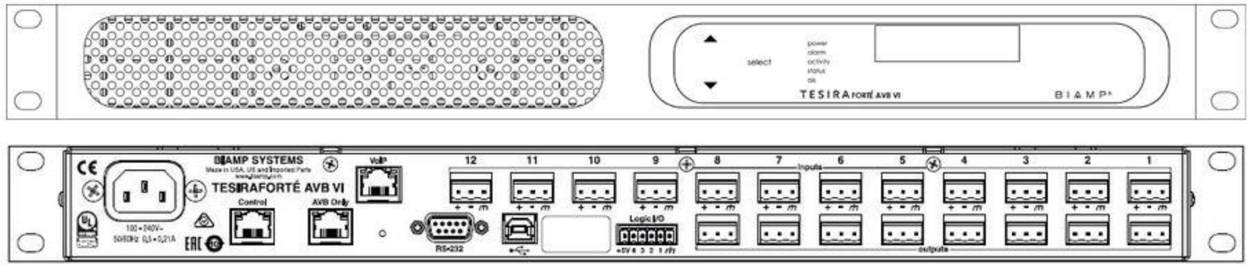
Benefits

- Allows integrators to choose which model works best for the installation environment.
- Application-specific models make system design, configuration, and installation easier and faster.
- Included default configuration file allows for plug-and-play usage.
- Highly scalable and cost-effective solution that can grow over time with the needs of the customer.
- Sona™ AEC and SpeechSense™ technologies to enhance speech processing.
- Integrates directly with soft codecs and other USB audio hosts.

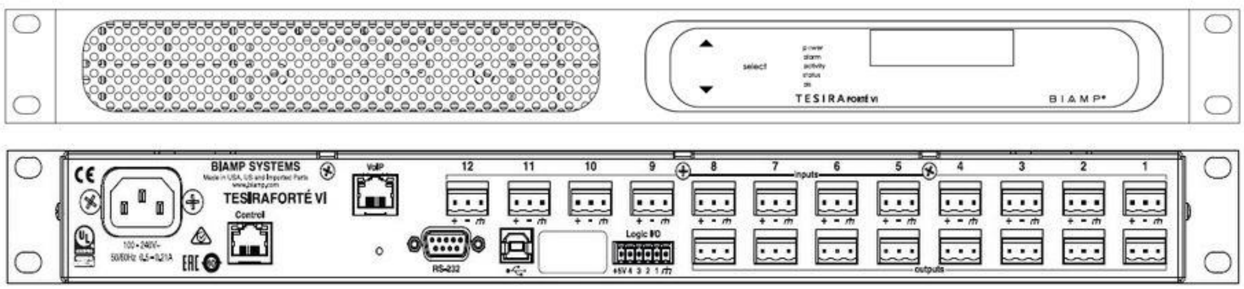
Features

- 128 x 128 channels of AVB (AVB model only)
- 12 mic/line level inputs with AEC, 8 mic/line level outputs
- Gigabit Ethernet port
- Up to 8 channels of configurable USB audio
- RS-232 serial port
- 4-pin GPIO
- 2-line LED display with capacitive-touch navigation
- Rack mountable (1RU)
- System configuration and control via Ethernet
- Internal universal power supply
- SIP VoIP interface via a RJ-45 connector
- Fully compatible with Tesira servers, expanders, and controllers (AVB model)
- Signal processing via intuitive software allows configuration and control for signal routing, mixing, equalization, filtering, delay and much more
- CE marked, UL listed, and RoHS compliant
- Covered by Biamp Systems' 5-year warranty

Tesira Forte VI AVB Front and Back Panels



Tesira Forte VI Front and Back Panels



Ex-Mod

EX-MOD

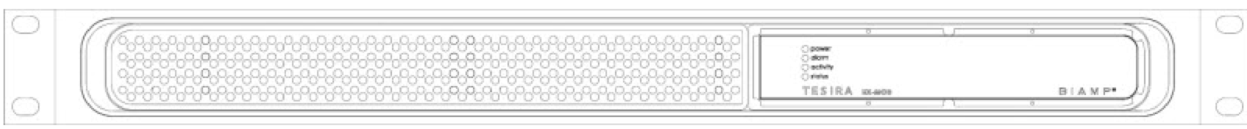
The Tesira EX-MOD is a modular expander device designed for use with Tesira SERVER, SERVER-IO and TesiraFORTÉ AVB devices. The Tesira EX-MOD allows remote placement of Tesira inputs and outputs for shorter analog cable runs. The Tesira EX-MOD is a single rack space, rack-mountable device, powered by 100-240VAC.

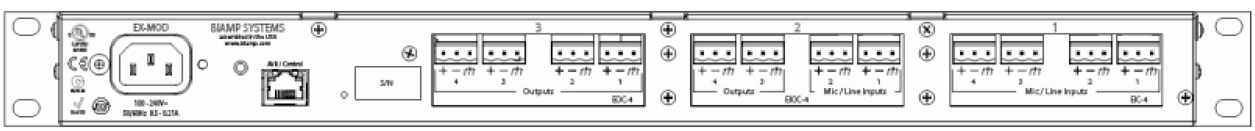
Exact I/O configuration can be customized specifically for the local zone. The modular design allows for flexibility to change or expand the local system. The EX-MOD can be configured with up to three 4-channel input and/or output expander cards for a maximum of 12 channels.

Tesira expanders must be associated (proxied) via the equipment table to an AVB enabled SERVER, SERVER-IO, or TesiraFORTÉ device. Expanders have one Ethernet port that accommodates both control functions and AVB audio traffic.
A Ethernet link between the control and AVB audio networks must be present for correct operation. Please refer to the Supported Network Topologies section for more details

Supported I/O Cards Include:

- EEC-4 -4 channels of mic/line level audio input with AEC (acoustic echo cancellation)
- EIC-4 - 4 Channel Mic/ Line Input card
- EOC-4 - 4 channel Mic/ Line Output card
- EIOC-4 - 2 channel Mic/Line Ipputs and 2 channel Mic/ Line Outputs per card





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

		the system		

EEC-4

The Tesira EEC-4 is an expander card with 4 analog input channels that includes AEC (Acoustic Echo Cancellation) for use with the Tesira EX-MOD device. The EEC-4 is one of several optional cards that can be installed in Tesira EX-MOD.
Each EEC-4 provides 4 channels of mic/line level audio input with AEC. The inputs are electrically balanced and are provided with plug-in barrier strip connectors. Software control of each input includes input gain (0-66dB) with clip indicator, +48V phantom power, mute, level, signal invert, and Biamp's Sona™ AEC active/ bypass.

EIC-4

The Tesira EIC-4 is an expander card with 4 analog input channels for use with the Tesira EX-MOD device. The EIC-4 is one of three optional cards that can be mounted in Tesira EX-MOD. Each EIC-4 provides 4 channels of mic/line level audio input. The inputs are electrically balanced and provided on plug-in barrier strip connectors. Software control of each input includes gain with clip indicator, +48V phantom power, mute, level and signal invert.

EOC-4

The Tesira EOC-4 is an expander card with 4 analog output channels for use with the Tesira EX-MOD device. The EOC-4 is one of three optional cards that can be mounted in Tesira EX-MOD. Each EOC-4 provides 4 channels of line level audio output. The outputs are electrically balanced and provided on plug-in barrier strip connectors. Software control of each output includes mute, level, signal invert and full-scale output reference.

EIOC-4

The Tesira EIOC-4 is an expander card with 2 input and 2 output channels for use with the Tesira EX-MOD device. The EIOC-4 is one of three optional cards that can be mounted in Tesira EX-MOD. Each EIOC-4 provides 2 channels of mic/line audio input and 2 channels of line level audio output. The inputs and outputs are electrically balanced and provided on plug-in barrier strip connectors. Software control of each input includes gain with clip indicator, +48V phantom power, mute, level and signal invert whereas for output includes mute, level, signal invert and full-scale output reference.

Expanders

Expander Hardware

Expander Hardware devices are designed for use with Tesira SERVER, SERVER-IO and TesiraFORTÉ AVB devices.
Expander Audio Hardware must be used with AVB enabled hardware. Server or Server I/O devices must be fitted with the AVB-1 Server card and TesiraFORTÉ devices must be the AVB capable devices.

- EX-AEC
- EX-IN
- EX-OUT
- EX-IO

Expander control hardware can be used with all Tesira SERVER, SERVER-IO and TesiraFORTÉ devices. As the Control network is used, AVB capable devices are not required.

- EX-Logic

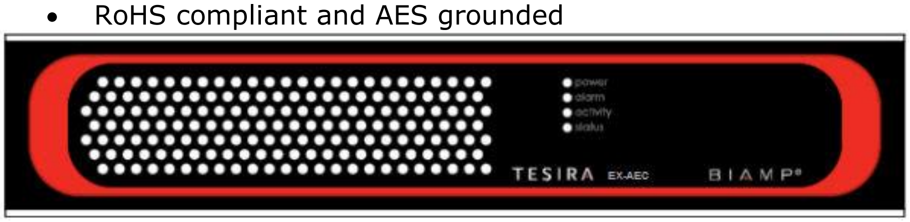
EX-AEC

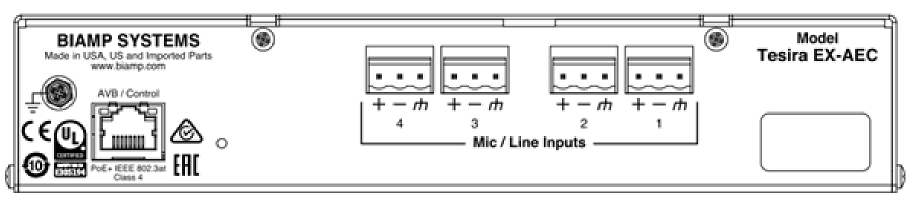
The Tesira EX-AEC is a modular expander device designed for use with Tesira SERVER, SERVER-IO, and TesiraFORTÉ AVB devices. The Tesira EX-AEC offers remote placement of Tesira inputs. The Tesira EX-AEC is a half-rack space device powered by PoE+.

The Tesira EX-AEC is a 4-channel mic/line input expander which include Biamp's Sona™ AEC (Acoustic Echo Cancellation).

Tesira expanders must be associated (proxied) via the equipment table to an AVB enabled SERVER, SERVER-IO, or TesiraFORTÉ device. Expanders have one Ethernet port that accommodates both control functions and AVB audio traffic.
A Ethernet link between the control and AVB audio networks must be present for correct operation. Please refer to the Supported Network Topologies section for more details

- 4 channels of balanced mic or line level input
- Sona AEC algorithm
- 0–66dBgain, adjustable in 6dB increments
- +48V phantom power
- -100 to +12dB fader range for level
- Audio and control networking over AVB
- Powered by PoE+
- Plug-in barrier strip connectors
- Front panel LEDs for device status indications
- Half-rack chassis
- RoHS compliant and AES grounded





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system

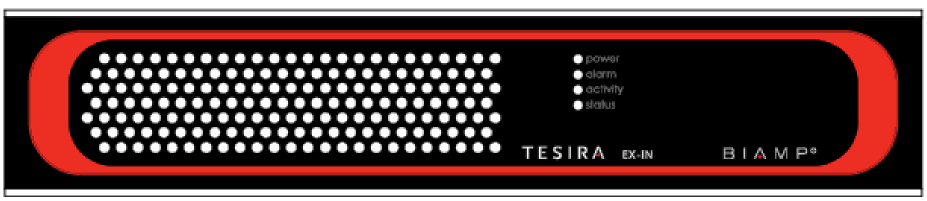
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

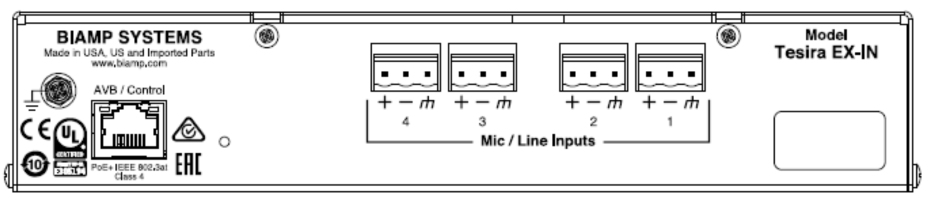
EX-In

The Tesira EX-IN is a half-rack expander box for use with Tesira SERVER and SERVER-IO devices. The EX-IN is a 4-channel mic/line input expander. The expander communicates with the Tesira AVB network for audio networking, configuration and control and is powered by PoE+.

Tesira expanders must be associated (proxied) via the equipment table to an AVB enabled SERVER, SERVER-IO, or TesiraFORTÉ device. Expanders have one Ethernet port that accommodates both control functions and AVB audio traffic.
A Ethernet link between the control and AVB audio networks must be present for correct operation. Please refer to the Supported Network Topologies section for more details

- 4 channels of balanced mic or line level input
- 0–66dBgain, adjustable in 6dB increments
- +48V phantom power
- -100 to +12dB fader range for level
- Audio and control networking over AVB
- Powered by PoE+
- Plug-in barrier strip connectors
- Front panel LEDs for device status indications
- Half-rack chassis
- RoHS compliant and AES grounded





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an inactive part of an active	Not applicable	The Host device is part of an

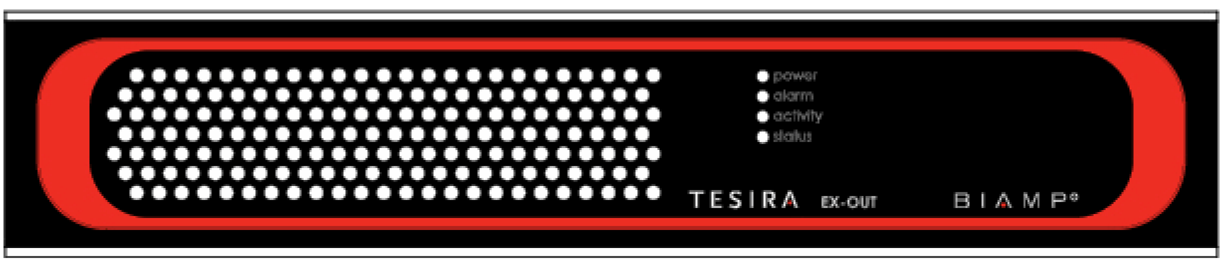
		system		inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

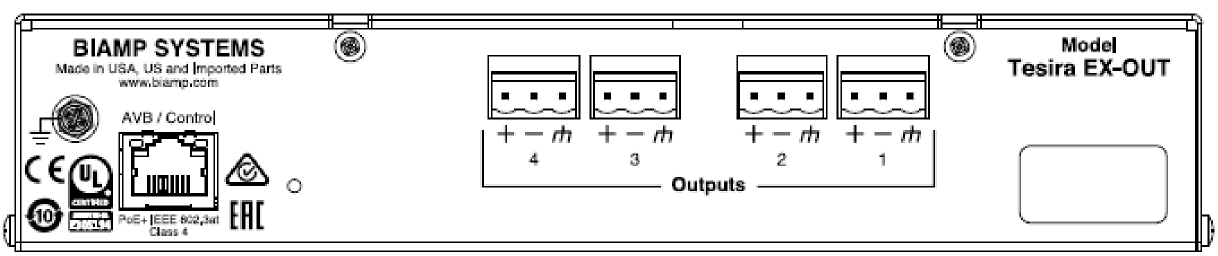
EX-Out

The Tesira EX-OUT is a half-rack expander box for use with Tesira SERVER and SERVER-IO devices. The EX-OUT is a 4-channel line level output expander. The expander communicates with the Tesira AVB network for audio networking, configuration and control and is powered by PoE+.

Tesira expanders must be associated (proxied) via the equipment table to an AVB enabled SERVER, SERVER-IO, or TesiraFORTÉ device. Expanders have one Ethernet port that accommodates both control functions and AVB audio traffic.
A Ethernet link between the control and AVB audio networks must be present for correct operation. Please refer to the Supported Network Topologies section for more details

- 4 channels of balanced mic line level output
- -100 to +12dB fader range for level
- Signal invert for reverse polarity
- Selectable full-scale output reference level (24dBu, 18dBu, 12dBu, 6dBu, 0dBu, -31dBu) for best interface/performance
- Audio and control networking over AVB
- Powered by PoE+
- Plug-in barrier strip connectors
- Front panel LEDs for device status indications
- Half-rack chassis
- RoHS compliant and AES grounded





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the	Not applicable	Minor fault is active in the device	Major fault is active in the

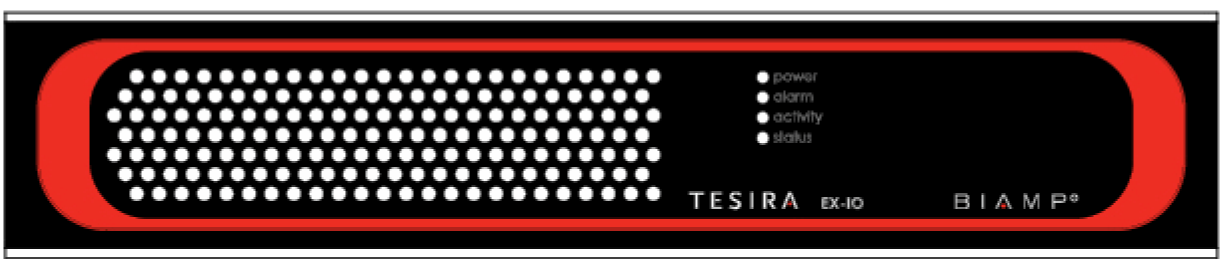
	device			device
Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

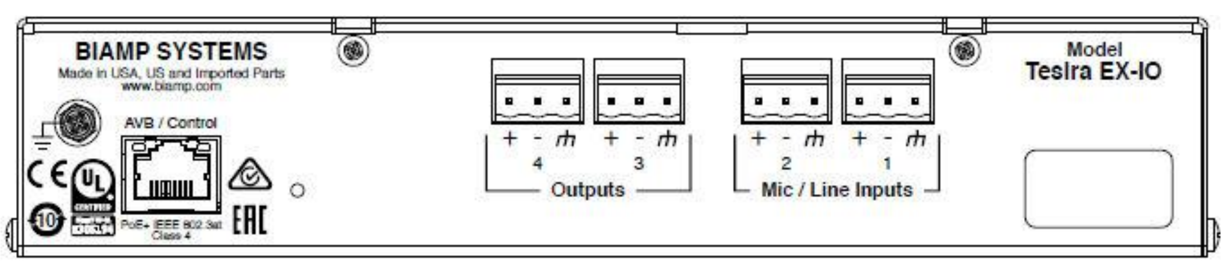
EX-IO

The Tesira EX-IO is a half-rack expander box for use with Tesira SERVER and SERVER-IO devices. The EX-IO is a 4-channel input and output expander. The expander features two channels of mic/line level input and 2 channels of mic level output. The expander communicates with the Tesira AVB network for audio networking, configuration and control and is powered by PoE+.

Tesira expanders must be associated (proxied) via the equipment table to an AVB enabled SERVER, SERVER-IO, or TesiraFORTÉ device. Expanders have one Ethernet port that accommodates both control functions and AVB audio traffic.
A Ethernet link between the control and AVB audio networks must be present for correct operation. Please refer to the Supported Network Topologies section for more details

- 2 channels of balanced mic or line level input
- 2 channels of balanced line level output
- 0–66 dB gain, adjustable in 6dB increments on input
- +48V phantom power on inputs
- Selectable full-scale output reference level (24dBu, 18dBu, 12dBu, 6dBu, 0dBu, -31dBu) for best interface/performance
- Powered by PoE+
- Half-rack chassis
- RoHS compliant and AES grounded





Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device

Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

EX-Logic

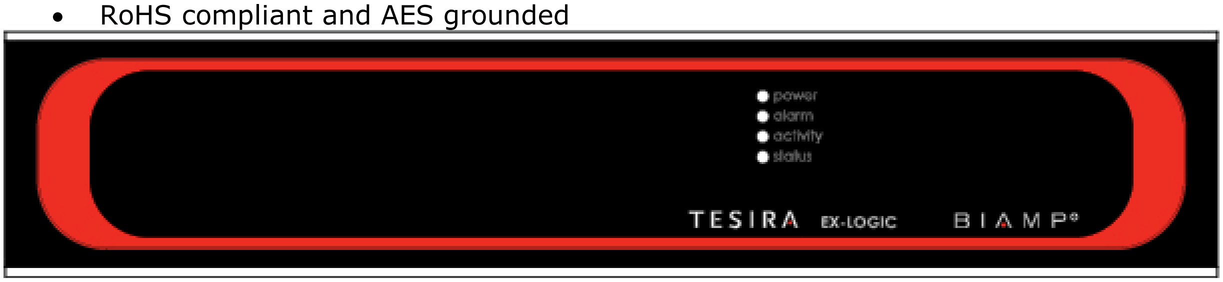
The Tesira EX-LOGIC is a half-rack logic box for use with Tesira SERVER, SERVER-IO and TesiraFORTÉ devices.

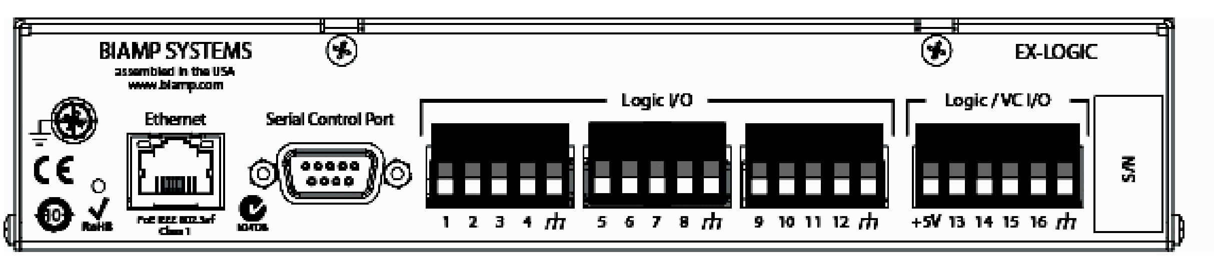
The EX-LOGIC provides both logic inputs and outputs and through software can be configured as a control interface. There are 16 total connections that can be used as inputs or outputs. Pins 1 – 12 are designated as Logic connections only and will accept contact closure or 5V TTL for input or provide contact closure or LED power (5V/10mA) for output. The remaining 4 connections can also be used for Logic connection or as variable voltage control input (e.g. interface to a potentiometer). Please review the Logic I/O Wiring section for more details.

The EX-LOGIC also provides a serial port for the output of command strings that can be used to send action commands to other equipment in the system.

Tesira EX-LOGIC expanders must be associated (proxied) via the equipment table to a SERVER, SERVER-IO, or TesiraFORTÉ device. EX-LOGIC Expanders have one Ethernet port and connect to the Tesira control network. Please refer to the Supported Network Topologies section for more details

- 16 total logic connections can be used as inputs or outputs
- 4 connections can also be used as voltage control inputs
- Inputs can control actions within the software including: presets, mutes, ducking, room combining, paging functions, and much more
- Outputs can trigger status relays, indicators or provide logic input to other controllable equipment
- Serial port for the output of command strings
- Connects to Tesira SERVER or SERVER-IO over AVB
- Powered by PoE
- Plug-in barrier strip connections
- Front panel LEDs for device status indications
- Half-rack chassis
- RoHS compliant and AES grounded





Grounding screw. This is for grounding the chassis of the expander

Ethernet data connection. Standard RJ-45 connector for connection with minimum CAT-5e cabling. The expander must receive PoE (IEEE 802.3af) power on this connector in order for proper operation. This connection is for sending and receiving control data with the Tesira server. The expander will not operate if it is not on a network that includes a Server-Class Tesira device.

Serial port. This is a command string output connection. The serial port can be configured by a Command String block in the software to send a string to another device. The serial port Baud rate is configurable in Device Maintenance using the Remote Devices selection The Baud can be set to 110, 300,1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200. Default is 115200.

Connections 1-12. These GPIO connections can be used as either inputs as outputs. They can be assigned to actions within the software using the Logic Input and Logic Output blocks.

Inputs
Each of the 12 GPIO pins can be configured individually by a logic input block as a logic input. By default the voltage on these pins is high causing a 1 on the input. A contact closure grounds the voltage causing a 0 on the input.

Outputs
Each of the 12 GPIO pins can be configured individually by a logic output block as a logic output. These GPIO pins can be driven high or low depending on the state of the logic output block. Due to hardware constraints, the internal pull-up resistor is always enabled.

Current Source.
When configured as logic output, each of the 16 GPIO pins can be configured to enable a current source capable of driving an LED. The current source is enabled depending on the state of the logic output block.

Connections 13 – 16. These connections can function in the same way as connections 1-12 for Digitial GPIO. But they can also be assigned as variable voltage input controls to allow analog control within the Tesira System by connection to a potentiometer.

GPIO pins 13 – 16 can be configured individually by the Control Voltage block in the software. If any one of these is configured for voltage control, then the logic expander will turn on the 5V potentiometer power.

Voltage Control Calibration
The 4 analog GPIO pins support voltage control calibration because a potentiometer may not be able to achieve the full range of voltage expected by the internal analog/digital convertors. When calibrated, the logic expander records both the minimum and maximum voltage levels caused by the potentiometer to achieve the full range of voltage.

Front Panel LED indicators

LED	Off	Green	Yellow	Red
Power	Unit is not powered	Unit is powered	Not applicable	Not applicable
Alarm	No Fault is active in the	Not applicable	Minor fault is active in the device	Major fault is active in the

	device			device
Activity	Not applicable	The Host device is an inactive part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration

Lab.gruppen Amplifier

Lab.gruppen Amplifier

All 3 power models in the D Series platform are available as Tesira variants for seamless integration into a comprehensive networked environment based on AVB protocols. All amplifiers function as a discrete high-power output device in the Tesira environment, with multi-channel audio inputs, control and monitoring all carried over a single Cat-5e/6 network cable. Full amplifier specifications and manuals are available at Lab.Gruppen.com

- D 200:4T
- D 120:4T
- D 80:4T

An analog input option is available for analog failover, which may be required in critical PA/VA applications. These are specified as separate model variant -

- D 200:4Ta
- D 120:4Ta
- D 80:4Ta

Software features

- **D Series Tesira system integration** - Tesira expander class device with full integration of amplifier configuration and amplifier surveillance
- **Device swap -** Host name hot swap with full configuration transfer
- **Auto power down -** Configurable auto power down to ultra low consumption SLEEP state
- **Channel processing** - Mute, Level, Signal invert

AVB Audio Network

- AVB I/O 4x4
- Sample rates / transport 48 kHz / Unicast
- Network latency Configurable 1 or 2 milliseconds

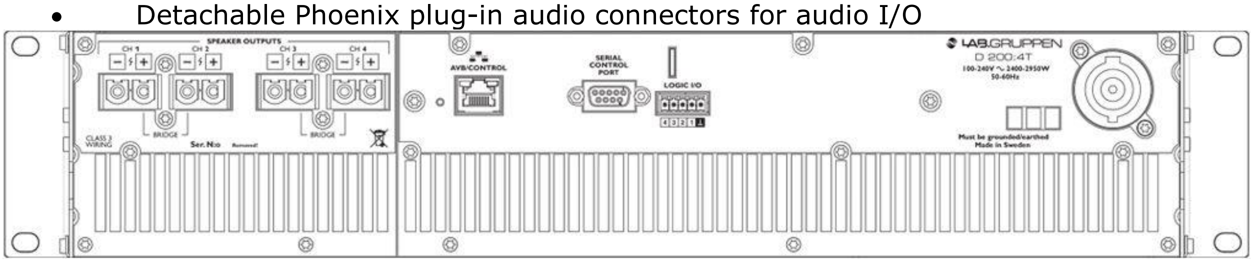
Optional 4 channels analog inputs

- Usage Configurable as Tesira system inputs or analog failover
- Balanced mic or line level input +48V phantom power and digitally controlled input gain
- Maximum input and digital reference +24 dBu
- Sampling rate / resolution 48 kHz / 24 bit
- Input impedance 8 k ohm
- Frequency response (20 Hz - 20 kHz @ 4 dBu) +/- 0.25 dB
- THD+N (20 Hz - 20 kHz @ 0 dB gain +4 dBu in) 0.006%
- Dynamic range (20 Hz - 20 kHz, 0 dB) > 108 dB
- Cross talk (20 Hz - 20 kHz @ 0 dB gain + 4 dBu in) > 85 dB

Amplifier Features

- Rational Power Management™ (RPM) – True flexibility in allocating total available power output across channel count for most efficient use of system inventory
- Up to 5000 W per channel for driving high power elements (e.g. subwoofers) on all models
  - all 4 channels on D 200:4T
  - up to two channels on D 120:4T
  - up to one channel on D 80:4T
- Integrates seamlessly with Tesira DSP platform as a high-power output object
- Control D Series from within Tesira software
- Platform indicators for Alarm, Activity, Status and Power and per-channel indicators for Load, Amp, Signal and Mute

- CAFÉ (Configuring Amplifiers For the Environment) Software incorporating ESP (Equipment Specification Predictor) for design, system planning and equipment planning, installation and commissioning.
- Regulated Switch-Mode Power Supply (R.SMPS™)
- Breaker Emulation Limiter (BEL™) to prevent power interruption
- Under-voltage Limiting (UVL™) for continued operation despite severe voltage drops
- Best-in-class Power Factor Correction (PFC) with Current Draw Modelling
- Proven and reliable Class TD® Output Stage with patented Intercooler® high-efficiency cooling system
- Dedicated on-board surveillance & load monitoring system
- Optional analog inputs for local system inputs or analog failover (D 200:4Ta, D 120:4Ta, D 80:4Ta)
- Detachable Phoenix plug-in audio connectors for audio I/O



Chassis Front Panel LED indicators

LED	Off	Green	Yellow	Red
Alarm	No Fault is active in the device	Not applicable	Minor fault is active in the device	Major fault is active in the device
Activity	Not applicable	The Host device is an active part of an active system	Not applicable	The Host device is part of an inactive system
Status	Not applicable	Device has received its configuration and is ready to participate in the system	Device is ready and waiting to receive a configuration	Device is not ready to receive its configuration
Power	Device is powered Off	Power State is ON	Power State is STANDBY	Power State is SLEEP

Amplifier channel Front Panel LED indicators

LED	Off	Green	Yellow	Red
Load	Amplified Output Load Status UNKNOWN	Amplified Output Load Status OK	Amplified Output Load Status WARNING	Amplified Output Load Status ERROR
Amp	Amplified Output Amp Status UNKNOWN	Amplified Output Amp Status OK	Amplified Output Amp Status WARNING	Amplified Output Amp Status ERROR
Signal	Amplified Output Signal Status UNKNOWN	Amplified Output Signal Status OK	Amplified Output Signal Status WARNING	Amplified Output Signal Status ERROR

Mute	Power state is standby	Audio is not muted	Audio is muted	Not applicable

Rear Panel Connections:

Speaker Outputs

Connect your speaker cabling here.

AVB / Control

The amplifier is configured and connected to in the same way as any other Tesira AVB Expander. Please review the AVB Network Considerations section for more details.

Serial Connector

The RS232 interface is used to control the power state of the amplifier externally. It operates at 9600 baud with 8 data bits, 1 stop bit, and no parity bits. These settings cannot be modified.
The following are the list of text commands. Each command and response ends with a carriage return <CR>, which is hexadecimal 0x0D. Each command and response is case-sensitive.

Text Command	Text Response	Description
wake	OK	When in the SLEEP state, transitions to the ON state. Because this wakes up the Tesira host card from SLEEP, the Tesira host card will respond with the text response once it is powered up and ready to receive new commands. Will also reply If already in the ON or STANDBY states
Wake	ERR	While in the ON or STANDBY state, an error will be returned as a text response of ERR<CR> if an unrecognized command or an invalid power value is received. While in the SLEEP state no error is returned. When powering up or transitioning from, or to, SLEEP, a text response of ERR<CR> will be returned. It is necessary to wait for the OK<CR> before issuing valid commands.
status	0- Sleep 1- On 2- Standby	Returns the status of the power state of the Tesira host card.
set power <value>	OK	Transitions to the power state specified by the value. This text command is not available in the SLEEP state. Use “wake” command to turn on the amplifier and then use this command to change the power state.

Logic Connector

The Logic connector is used for power control and status monitoring. The connector use is fixed and is defined as follows:

Pin	Function	Description
1	Ground	
2	Input	A transition from logic 0 to logic 1 transitions the power state to ON. A transition from logic 1 to logic 0 transitions the power state to SLEEP. This is edge triggered.
3	Input	Not used
4	Output	A logic 0 when the amplifier is in the ON state and logic 1 otherwise (STANDBY and SLEEP).
5	Output	A logic 0 when either the amplifier or an audio channel has at least one fault. A logic 1 when there are no faults. Faults detected by the Tesira application that runs on the Tesira host card do not apply to this output.

Note: A logic 1 is defined as TTL high level and a logic 0 is defined as TTL low level

Audio-Technica Dante Mic

ATND971 Dante Enabled Microphone

The ATND971 is a Dante-enabled, wide-range condenser microphone with a cardioid polar pattern. It is designed for surface-mount applications such as high-quality sound reinforcement, conferencing, distance learning and other demanding sound pickup applications. The microphone can be configured using the dedicated Dante Mic Component processing object in Tesira software.

The microphone should be placed on a flat, unobstructed mounting surface, with the front of the microphone facing the sound source. The sound source should not be below, or higher than 60° above, the plane of the mounting surface. Full microphone specifications and manuals are available at audio-technica.com

When this microphone is used with Tesira software it will override local control, causing the audio to remain on and the red/green LED status indicator, low-cut filter and input gain level to be controlled remotely. In Remote mode the microphone's user switch can be programmed to trigger functions on compatible Dante-enabled devices. When in remote mode the Remote LED is illuminated. For full routing capabilities the use of Audinate Dante Controller software is required. Please review the Dante Network Considerations section for more details.

Features

- Connects directly to network via Ethernet cable—no soldering or additional cable required
- Integrated user switch controls talk/mute in Local mode and triggers Dante-enabled devices in Remote mode
- Local or remote control of gain, low-cut UniSteep® filter and red/green LED status indicator
- Powered by network PoE -802.3af (802.3at Type 1)
- Scalable across Dante's 512 bidirectional audio channels
- Touch-sensitive capacitive-type user switch
- UniGuard® RFI-shielding technology offers outstanding rejection of radio frequency interference (RFI)
- UniSteep® filter provides a steep low-frequency attenuation to improve sound pickup without affecting voice quality
- Available interchangeable elements permit angle of acceptance from 100° to 360°
- Heavy die-cast case and non-slip silicone foam bottom pads minimize coupling of surface vibration to the microphone

ATND8677 Dante Enabled Microphone

The ATND8677 is a Dante-enabled microphone desk stand for use with any gooseneck microphone with a three-pin XLRM-type output connector. The desk stand is designed for surface-mount applications such as high-quality sound reinforcement, conferencing, distance learning and other demanding sound pickup applications. The microphone can be configured using the dedicated Dante Mic Component processing object in Tesira software.

Full specifications and manuals are available at audio-technica.com

When this microphone is used with Tesira software it will override local control, causing the audio to remain on and the red/green LED status indicator, low-cut filter and input gain level to be controlled remotely. In Remote mode the microphone's user switch can be programmed to trigger functions on compatible Dante-enabled devices. When in remote mode the Remote LED is illuminated. For full routing capabilities the use of Audinate Dante Controller software is required. Please review the Dante Network Considerations section for more details.

Features

- Connects directly to network via Ethernet cable—no soldering or additional cable required
- Integrated user switch controls talk/mute in Local mode and triggers Dante-enabled devices in Remote mode
- Local or remote control of gain, low-cut UniSteep® filter, red/green LED status indicator and phantom power
- Powered by network PoE - 802.3af (802.3at Type 1)
- Scalable across Dante's 512 bidirectional audio channels
- Touch-sensitive capacitive-type user switch
- UniSteep® filter provides a steep low-frequency attenuation to improve sound pickup without affecting voice quality
- 3-pin XLRF-type input for quick mounting of any gooseneck microphone with an XLRM-type output
- Heavy die-cast case and non-slip silicone foam bottom pads minimize coupling of surface vibration to the microphone

Control

TEC-1

TEC-1s Ethernet Controller is a remote control for Tesira® systems. It offers a simple, intuitive interface for end users and can be installed and configured to fit the unique needs of a particular application. The device connects via standard CAT-5/6/7 cabling and is powered over Ethernet, eliminating the need for custom cabling and local power sources. Multiple remote control panels can be connected over large distances using standard network technology.

Can be included in software programming by adding the Device from the TEC-1 item on the **Component Objects > Controls toolbar**

Tesira TEC-1 devices are associated with a SERVER, SERVER-IO, or TesiraFORTÉ device via the TEC-1 Device ID and Device Allocation (configured in the TEC-1 Property Sheet). TEC-1 devices have one Ethernet port and connect to the Tesira control network. Please refer to the Supported Network Topologies section for more details

FEATURES

- Adjustment and/or initiation of 32 selectable system volumes and actions
- Volumes are any individual or grouped system levels, including inputs, outputs, matrix cross-points, etc.
- Actions are any individual or grouped system operations, including presets, source selection, mutes, ducking, combining, etc.
- Control functions are programmed in the Tesira system design software
- High contrast OLED display with a wide viewing angle
- The display brightness can be adjusted to fit the ambient light present in the application and automatically dims when not in use
- Capacitive touch technology eliminates protruding and moving parts to increase product reliability and longevity while simplifying cleaning
- Surface-mounts to any wall; multiple mounting options accommodate international back boxes
- 330' (100m) Ethernet cable length can be extended with standard PoE 802.11af network technology (routers, switches, hubs, media converters)
- Connects with standard RJ-45 or IDC connector
- Covered by Biamp Systems' five-year warranty

System Control

Software User Interface

Once a system design is compiled and downloaded into Tesira Server devices, the system can be controlled in real-time via the Tesira software. The extent of control can be limited with different password levels.

In addition, Control Dialog Boxes for controls can be minimized to create customized control surfaces (room combiners, meters, level controls, mute buttons, & preset buttons). These control surfaces can then be made accessible to the User, only with a specific Password.

These control surfaces remain functional, even if other component settings are made inaccessible to the User (via Password Level). The control surfaces can also remain visible, even if the components they represent are made invisible (via Layer View). Therefore, a custom User control surface can be created in the Layout, with User access allowed, but with all other system settings inaccessible (and hidden). The size & shape of the Layout may be changed, and Toolbars hidden, to customize the appearance of the User control screen.

Third Party Control

After initial programming & configuration,Tesira systems may be controlled by RS-232 communication from third-party control systems, such as AMX®or Crestron®, using a Serial Control Port or Ethernet Connection.

Tesira can be controlled via the control dialogs in the Tesira software, or via third-party controllers using RS-232, Telnet or Secure Shell Console (SSH) .

A Tesira Text Protocol is used to facilitate interfacing to third party controllers

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.

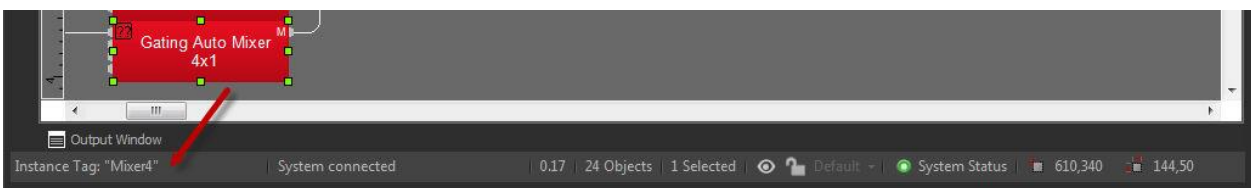
To facilitate external control of Tesira servers Biamp uses Tesira Text Protocol (TTP). This allows external control of a programmed Tesira system via ASCII characters.

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

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:

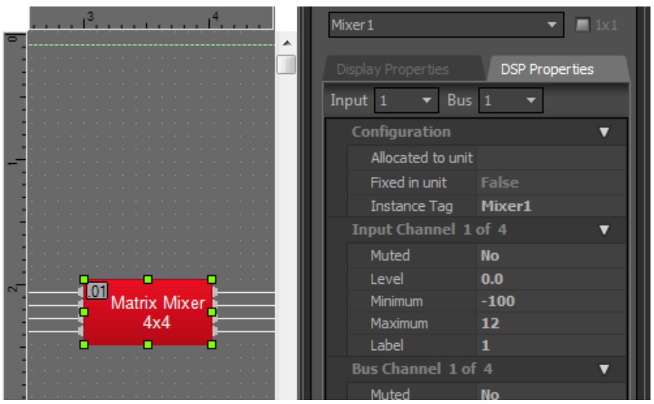
Instance_Tag Service [Index] [Value]

- **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.

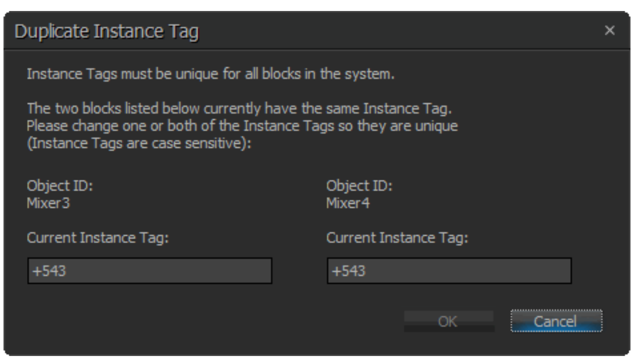
Instance Tag	Service Code	Value
DEVICE	recallPreset	1001

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	A attribute is to be read. The value will be returned in the response
set	A attribute is to be set to a specific value. String: **Instance_Tag Service [Index][Value]** Example: **Level1 set mute 1 true**
increment	A 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	A 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	A attribute is to be toggled. String: **Instance_Tag Service Attribute [Index]** Example: **Level1 toggle mute 1**
subscribe	A attribute is to be subscribed to.
unsubscribe	A 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

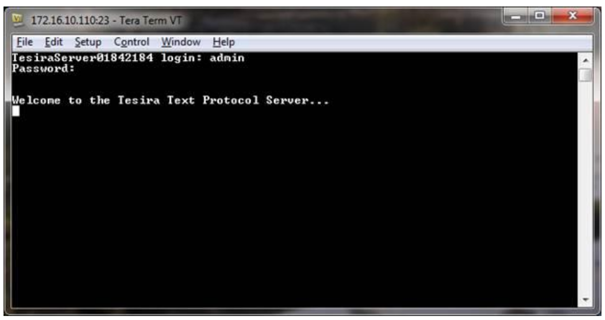
TTP Security

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.

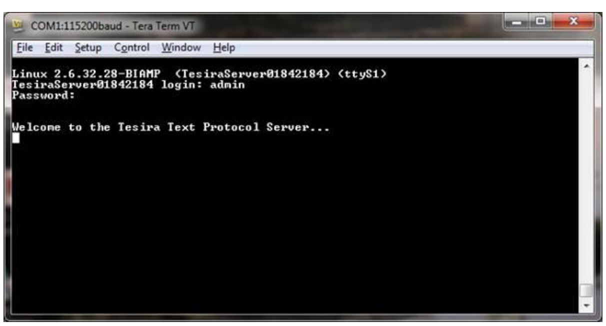
Opening a Telnet or SSH session to a Tesira Server results in a login prompt. Valid credentials must be provided to access the system in any way. One must be logged in as controller or higher level to make any changes to the system, while an observer can only query the system for levels and other current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



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.

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 Examples are shown using PuTTY

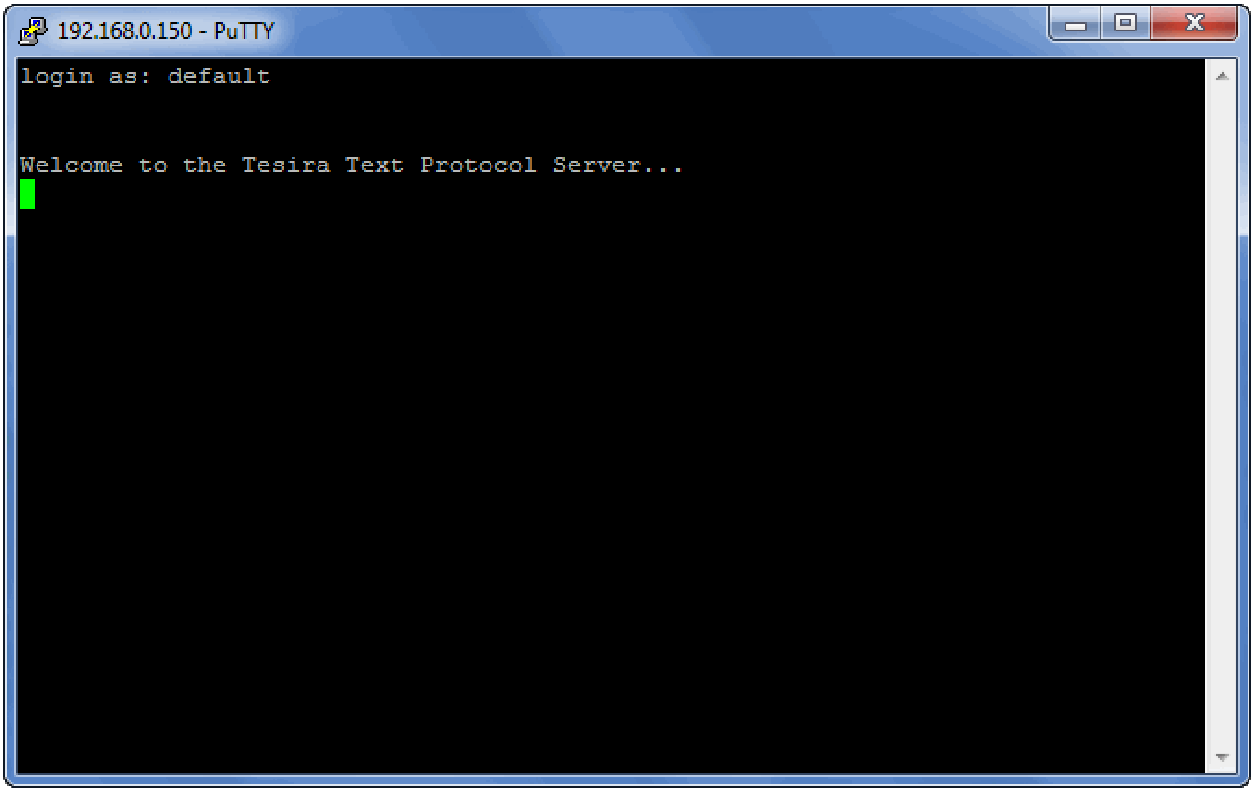
Configuring a PC to connect to Tesira using Telnet.

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 enable 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 and 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 has 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 110, 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

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 or TesiraFORTÉ devices to use Telnet can be disabled via a DEVICE TTP command or in the Device Network Settings in Device Maintenance.

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.

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.

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 and Password are ‘default’ and ‘default’ respectively. In a protected system, the credentials configured in the system must be provided.

Attribute tables

Interface tables

Service Addresses
Device Session
IO Blocks:
Audio Input Block Audio Output Block CobraNet Input Block CobraNet Output Block Dante Input Block Dante 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 Call State Commands Dtmf Decode 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
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.

Examples:

To reboot the device you are connected to:

Instance Tag	Service
DEVICE	reboot

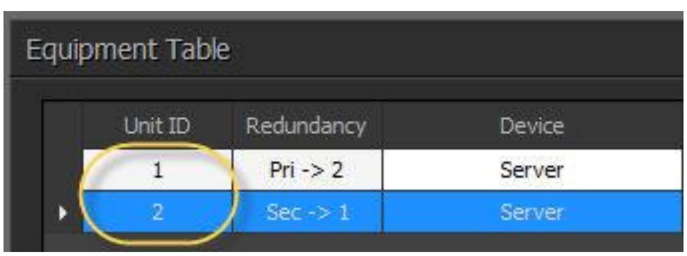
Result: DEVICE reboot

To start Audio on a device :

Instance Tag	Service
DEVICE	startAudio

Result: DEVICE startAudio

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

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	
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)

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 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 a single space. 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 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 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" }

Attribute Description	Attribute Code	Command	Indexes	Value Range
Active Faults	activeFaultList	get		
Discovered Servers	discoveredServers	get		
DNS Config	dnsConfig	get/set		
DNS Status	dnsStatus	get		
Host Name	hostname	get/set		
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
Firmware Version	version	get		

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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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.

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	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	streamActive	get		false, true

Connection Status				
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 a single space. 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 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.

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	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		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 a single space. 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 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.

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 subscribe unsubscribe	channel	false, true

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 a single space. 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 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.

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

Dante 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 a single space. 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 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.

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, 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	peak	get	channel	false, true

Occurring		subscribe unsubscribe		
All Peaks	peaks	get subscribe unsubscribe		
Phantom Power	phantomPower	get set toggle	channel	false, true

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 a single space. 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 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 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 a single space. 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 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 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	channel	minLevel - maxLevel dB

		unsubscribe		
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 a single space. 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 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.

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 a single space. 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 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.

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 a single space. 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 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 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	channel	false, true

		toggle subscribe unsubscribe		
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 a single space. 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 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 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
Reset AEC	aecReset	get / set toggle	channel	false, true
Bypass AGC	agcBypass	get / set toggle	channel	false, true
Conferencing Mode	confMode	get / set	channel	TEST, TELEPHONE, VOIP, VIDEO, CONF_MODE_CUSTOM
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
All Meter States	meters	get subscribe unsubscribe	channel	
Min SNR	minSnr	get / set increment decrement	channel	10.0 - 50.0 dB
Min Threshold	minThreshold	get / set increment decrement	channel	-30.0 - 10.0 dBu (Max Value equal to Target Level)
Mute	mute	get / set toggle subscribe unsubscribe	channel	false, true
All Mute States	mutes	get subscribe unsubscribe		
Noise Reduction	nrdMode	get / set	channel	OFF, LOW, MED, HIGH, NOISE_RED_MODE_CUSTOM
Channel Count	numChannels	get		1 - 24
Pre-Emphasis Slope	preEmphasisSlope	get / set	channel	Slope_0, Slope_1, Slope_2, Slope_3

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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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	-100.0 - 0.0 dB

		decrement	

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 a single space. 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 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 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 a single space. 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 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 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	dialToneLevel	get / set	-100.0

Level		increment decrement	-12.0 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 - 12.0 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	false, true

		toggle	
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.

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 a single space. 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 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 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 a single space. 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 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 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	

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 a single space. 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 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 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		
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)	
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
DTMF Off Time	dtmfOffTime	get / set increment decrement	line	40 - 1000 ms
DTMF On Time	dtmfOnTime	get / set increment decrement	line	40 - 1000 ms
Last Number Dialed	lastNum	get subscribe unsubscribe	line	
Line In Use	lineInUse	get subscribe	line (1 or 2), call	false, true

		unsubscribe	appearance index (1-6)	
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		
Line Count	numChannels	get		Always 2
Protocol Info	protocols	get subscribe unsubscribe		
Redial Enabled	redialEnable	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

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

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_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

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

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]]]

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 a single space. 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 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 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. 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
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 a single space. 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 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.

Attribute Description	Attribute Code	Command	Indexes	Value Range
Amplifier	ampName	get		

Name				
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	minLevel	get / set increment	channel	-100.0 - maxLevel dB

Min		decrement		
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

Mixer Blocks

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 a single space. 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 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 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	channel	-100.0 - inputMaxLevel

		increment decrement		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 a single space. 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 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 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 a single space. 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 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 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	input	-100.0 - inputMaxLevel

		decrement		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 a single space. 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 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 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	input, output	-100.0 - 0.0 dB

		decrement		
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 a single space. 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 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 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 a single space. 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 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 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	room	levelOutMin - levelOutMax dB

		subscribe unsubscribe		
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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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	band	-30.0 - 0.0 dB

		decrement		
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 a single space. 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 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 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	false, true

		toggle	
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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 &	filterTypeSlope	get / set	band,	[Type, slope]

Slope			filter	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 a single space. 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 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 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 a single space. 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 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.

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 a single space. 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 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.

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 a single space. 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 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 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	false, true

		toggle	
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 a single space. 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 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 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 a single space. 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 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 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	0.0 - 30.0 dB

		decrement	
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 a single space. 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 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 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	
Input Count	numInputs	get		1 - 256
Output Count	numOutputs	get		1 - 256
Output Label	outputLabel	get/set	output	

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 a single space. 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 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 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		false, true

		subscribe unsubscribe		
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

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 a single space. 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 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 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}

Control Blocks

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 a single space. 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 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 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	channel	minLevel - 12.0 dB

		decrement		
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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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		

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 half 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 a single space. 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 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 Code	Index
Dialer1	get	lastNum	1

Attribute Description	Attribute Code	Command	Indexes	Value Range
Auto Answer	autoAnswer	get / set toggle subscribe	line	false, true

		unsubscribe		
Call State	callState	get subscribe unsubscribe		
Display Name Label	displayNameLabel	get / set		
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 a single space. 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 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 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	channel	0 - 60000 ms

		increment decrement		
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 a single space. 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 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 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

Generator Blocks

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 a single space. 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 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 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	20.0 - 20000.0 Hz

		decrement	
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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 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 a single space. 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 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 Code
ControlVoltage1	get	numchannels

Example
ControlVoltage1 get numChannels +OK "value":1

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