

# Telestra Training Manual

Revision H Version 1 March 2023 Document DOC-TEL-NA-TM-H-1 Advanced Simulation Technology inc. 500A Huntmar Park Drive • Herndon, Virginia 20170 USA (703) 471-2104 • asti-usa.com

Product Name: Telestra

Telestra Training Manual

© Copyright ASTi 2023

Restricted rights: copy and use of this document are subject to terms provided in ASTi's Software License Agreement (www.asti-usa.com/license.html).

ASTi 500A Huntmar Park Drive Herndon, Virginia 20170 USA

#### **Red Hat Enterprise Linux (RHEL) Subscriptions**

ASTi is an official Red Hat Embedded Partner. ASTi-provided products based on RHEL include Red Hat software integrated with ASTi's installation. ASTi includes a Red Hat subscription with every purchase of our Software and Information Assurance (SW/IA) maintenance products. Systems with active maintenance receive Red Hat software updates and support directly from ASTi.

#### **Export Restriction**

Countries other than the United States may restrict the import, use, or export of software that contains encryption technology. By installing this software, you agree that you shall be solely responsible for compliance with any such import, use, or export restrictions. For full details on Red Hat export restrictions, go to the following:

www.redhat.com/en/about/export-control-product-matrix

# **Revision history**

Date	Revision	Version	Comments
10/26/2018	G	0	(7.15.0) Converted Telestra Training Manual to XML. Updated screenshots, and edited content for gram- mar and style. Updated licensing content for USB License Keys.
1/9/2019	G	1	Updated the <b>Licensing Management</b> screenshot in "Install a USB License Key."
7/26/2019	Н	0	Formatted entire document and edited it for grammar and style. Clarified several unclear sections.
3/8/2023	Н	1	Added Red Hat Enterprise Linux subscription and export statement to the front matter.

## Contents

1.0 Introduction	1
2.0 Hardware overview	2
2.1 ACE Studio	3
2.2 Audio distribution devices	3
2.2.1 ACU	3
2.2.2 ACU2	4
2.2.3 ACE-RIU	5
2.2.4 Ashly Power Amplifier	6
2.2.5 Peripherals	8
3.0 Protocols, services, and networks	9
3.1 Abstraction of protocols	10
3.2 ACENet	10
3.3 ASTiNet	12
3.3.1 Voice over IP (VoIP)	12
3.4 DIS	13
3.5 HLA	14
3.6 Future protocols	14
4.0 Telestra concepts	15
4.1 Data flow	16
4.2 ACE Studio concepts	17
4.3 System default login credentials	17
4.4 Cold starts	18
4.5 System configuration	18
4.6 System Backup/Restore	19
4.6.1 Backup System Configuration	20
4.6.2 Restore System Configuration	23

5.0 Software	26
5.1 Project Manager	26
5.1.1 Project elements	28
5.1.2 ACE STUDIO Project Manager	29
5.1.3 Layout	30
5.2 Load Viewer	30
5.2.1 Models	31
5.2.2 Servers	32
6.0 Model services	35
6.1 Intercoms	35
6.2 Sound repositories	36
6.3 Math Plan	37
6.4 Radios	41
6.5 Commplan Editor	44
6.6 Radio Monitor	44
6.6.1 Radio filters	47
6.6.2 Statistics	48
6.7 Domain Editor	48
6.8 Helpers	49
6.8.1 Channels	50
7.0 Host Interface	52
7.1 Host control at the project level	53
7.2 Host control at the load and model level	54
7.3 Host interface exercise	57
8.0 Remote Management System	58
8.1 Getting started	58
8.2 Add a user account	59

8.3 System	60
8.3.1 Status	60
8.3.2 Health	61
8.3.3 Logs	62
8.4 Licensing	62
8.4.1 View licensing information	63
8.4.2 Install a USB License Key	64
8.4.3 Update a USB License Key	65
8.5 Audio devices in the RMS	67
9.0 ACE Studio model building	68
9.1 Sine wave	68
9.1.1 Build a model	69
9.1.2 Connect an ACU2 to the model	71
9.2 Mixer	73
9.3 Vox folder organization	74
9.4 Math Plan	77
9.5 Playsound	78
9.5.1 Upload Sound Files	79
9.5.2 Add a sound library	79
9.6 Intercoms	84
9.6.1 Intercom exercise	85
9.7 Radios	91
9.7.1 Local radios	93
9.7.2 Comm Plan	94
9.7.3 Add a model	99
9.8 Add a communication model with helpers1	112
10.0 Advanced topics and examples1	20

10.1 ACE Scope	
10.1.1 Main display	
10.1.2 1/3 Octave display	122
10.1.3 FFT display	
10.1.4 Comparison display	
10.2 ACE GLOBAL LINK EDITOR	
10.2.1 Example model	

## **1.0 Introduction**

ASTi's Telestra suite of products provides comprehensive sound and communication simulation software and equipment. With a wide range of capabilities and scalable solutions, Telestra products meet complex, high-fidelity, network distributed applications in today's training environments. This training course familiarizes you with Target and Studio hardware, the Remote Management System (RMS), and ACE Studio software.

This document discusses how to:

- Hardware overview
- Protocols, services, and networks
- Telestra concepts
- Software
- Model services
- Host Interface
- Remote Management System
- ACE Studio model building
- Advanced topics and examples

## 2.0 Hardware overview

Figure 1, "Hardware layout" below displays a typical system hardware setup. This system varies in complexity from program to program.



Figure 1: Hardware layout

This chapter discusses the following topics:

- ACE Studio
- Audio distribution devices

## 2.1 ACE Studio

ACE Studio is available on an ASTi Telestra platform with a removable hard drive or as a software-only application that runs on a virtual machine on a customer-furnished computer. ACE Studio is a suite of software tools that incorporates the following:

- Sound and communication model development
- Project management
- Communications monitoring and fault analysis
- Equipment status and configuration

ACE Studio software provides remote access to all networked simulation models and equipment from a single development workstation. For more information about ACE Studio, go to the *ACE Studio User Guide* (support.asti-usa.com/media/pdf/t4/ace\_studio\_ug.pdf).

## 2.2 Audio distribution devices

This section discusses the following topics:

- ACU
- ACU2
- ACE-RIU
- Ashly Power Amplifier
- Peripherals

#### 2.2.1 ACU

The ACU is an audio and input/output (I/O) remote interface for the Telestra suite of products. The ACU provides the Analog-to-Digital/Digital-to-Analog (AD/DA) conversion. All audio and I/O is digitally distributed between ACUs and Targets for maximum noise rejection and reliability. This unit may be connected directly to the Target or more typically through an ASTi-approved ACENet switch.

The Remote Management System (RMS) performs ACU firmware software updates and gain configuration. The hardware is available in a 1U (19-inch) two, four, and six-channel rack-mount configuration. Multiple Targets can share ACU channels when using a four or six-channel ACU; however, the channels are grouped A/B, C/D, and E/F. Different Targets cannot share two channels in a grouping.

The platform components consist of the following:

- Independent, software-configurable audio inputs and outputs (one per channel)
- Control inputs (three per channel)
- Digital outputs (one per channel)
- RS-422 serial ports (one per channel)
- 48 kHz digital audio distribution
- Two, four, or six DB-15 connectors

Figure 2, "ACU front panel" below and Figure 3, "ACU rear panel" below shows ACU rear and front panels:



Figure 2: ACU front panel



Figure 3: ACU rear panel

For ACU pinout diagrams, go to the *ASTi ACU Technical User Guide* (support.asti-usa.-com/media/pdf/acu\_ug.pdf).

#### 2.2.2 ACU2

The ACU2 audio and I/O distribution device features stereo operation for independent left and right output support on a single connector, a reduced footprint that can fit on a desktop. Two units can also fit in a 1U 19" rack space, or it accommodates a convenient power daisy chain connection for two units. The ACU2 has a sample rate of 48 kHz, ensuring high-fidelity audio processing with adjustable amplifier or preamplifier gains and microphone power. The ACU2 includes the following features:

- Four stereo audio outputs
- Four mono audio inputs
- Independent, software-configurable audio inputs and outputs
- Control inputs (three per channel)
- Digital outputs (one per channel)
- RS-422 serial ports (two per channel)

Figure 4, "ACU2 front panel" below and Figure 5, "ACU2 rear panel" below show the ACU2 front and rear panel:



Figure 4: ACU2 front panel



Figure 5: ACU2 rear panel

#### 2.2.3 ACE-RIU

The ACE-RIU is a compact interface module based on digital signal processing (DSP). It connects remote operator headsets, speakers, and control panels to a central Target via ACENet architecture. The ACE-RIU provides low-noise analog-digital conversion and low-latency distribution. The ACE-RIU has a sample rate of 48 kHz, which ensures high-fidelity audio processing. The hardware is available with a 19", 1U high rackmount kit, and each kit holds three ACE-RIUs.

The platform components consist of the following:

- Digital inputs (one per channel, four channels total)
- Digital outputs (one per channel, four channels total)
- RS-422 serial ports (two per channel)

Figure 6, "ACE-RIU front panel" below and Figure 7, "ACE-RIU rear panel" below show the ACE-RIU's front and rear panels:



Figure 6: ACE-RIU front panel



Figure 7: ACE-RIU rear panel

For more information about the ACE-RIU, go to the *ACE-RIU Technical User Guide* (<u>support.asti-usa.com/media/pdf/acenet\_ug.pdf</u>).

#### 2.2.4 Ashly Power Amplifier

The Ashly Power Amplifier provides power levels and features that meet the audio requirements for aural cue simulation. Each amplifier is a network component that integrates with ASTi's ACENet architecture. ASTi offers a four-channel amplifier and an eight-channel amplifier, both in a 2U chassis. This platform is for audio output only; customers generally use it for aural cue programs. The platform components consist of the following:

- Two Ethernet ports to the 1000 Megabits per second (Mbps) network (one used)
- One data port to the 1000 Mbps network
- Four or eight output connectors, depending on the platform
- Mode switches for every two channels
- Channel level controls providing gain control
- Four or eight input connectors, depending on the platform purchased



*Note*: *The input connectors are not used in the ASTi system setup.* 

Figure 8, "Ashly Power Amplifier front panel" below and Figure 9, "Ashly Power Amplifier rear panel" below show Ashly Power Amplifier rear and front panels:



Figure 8: Ashly Power Amplifier front panel



Figure 9: Ashly Power Amplifier rear panel

#### 2.2.5 Peripherals

In addition to Telestra audio hardware, ASTi also offers audio peripherals and user interfaces that connect to the equipment:

- Headsets, microphones, and speakers
- Press-to-talk (PTT) devices
- *Touchscreen tablets:* the generic solution for a radio control panel. ASTi provides custom software models.
- *Handheld terminals (HHTs):* a highly flexible solution to multioperator simulation requirements.





Figure 10: Telestra peripherals

For details about options, pricing, and ordering, go to ASTi Telestra Support (<u>support.asti-usa.com/telestra/index.html</u>).

## 3.0 Protocols, services, and networks

Models can connect to the protocols in ACE Studio using helpers. Simulated networked radios use standards such as ASTiNet, DIS, or HLA:

- DIS is a simulation standard that uses defined protocol data units (PDUs) to pass data between two sites.
- Simulated radio communications use DIS protocols for transmitter, receiver, and signal PDUs.
- HLA is a flexible simulation architecture managed by a run-time infrastructure.
- ASTiNet is an ASTi-proprietary protocol that provides communication networking for Target-to-Target operation and other ASTi-approved products.

In ACE Studio, **Domain Editor** provides the ability to set parameters for the standards.

Figure 11, "Network overview" below shows a network overview of Target protocols, services, and networks:



Figure 11: Network overview

This chapter discusses the following topics:

- Abstraction of protocols
- ACENet
- ASTiNet
- DIS
- HLA
- Future protocols

## 3.1 Abstraction of protocols

ACE Studio develops models independently of network protocols. Set up the network outside of the model. The domain acts as a gateway that maps protocols to the model, making it available to the outside world.



Figure 12: Layers of abstraction

## **3.2 ACENet**

The Audio Communications Environment Network (ACENet) is part of ASTi's latest-generation Telestra product family. It provides a low-latency, network-based audio and I/O distribution architecture for ASTi's ACE communications and sound modeling equipment and software. This flexible architecture provides a highly scalable distribution network of model processing systems and remote audio and I/O interface devices to add multiuser sound and communication applications.

ACENet has a wide array of features:

- *Remote distribution:* network-based, spoke, and hub architecture provides digital audio and I/O distribution across a wide area, hundreds of feet from Target platforms.
- *Ethernet-based:* uses commercial off-the-shelf (COTS) network cabling and ASTi-qualified equipment for easy connection and wide distribution. ACENet always operates on Eth1.

- *Highly scalable:* plugs multiple Target platforms, ACUs, and other ACENet-compatible equipment into a single ACENet network, providing a flexible modeling and distribution capability for applications ranging from a single operator to large, multioperator installations.
- *Flexible audio and I/O:* ACUs provide configurable audio, serial, analog, and discrete I/O interfaces to accommodate a wide range of peripherals, such as military and commercial headsets, audio amplifiers, speakers, microphones, recording equipment, press-to-talk (PTT) devices, simulated communication panels, handheld terminals (HHTs), and other peripheral devices.
- *High-fidelity:* ACENet supports synchronized, 48 kHz digital audio distribution for realistic sound and communication simulation.
- *Low-latency:* closed-network architecture and customized, real-time distribution software means extremely low transport latency, which is essential for realistic simulation and elimination of delay-related audio issues.

Figure 13, "ACENet audio distribution" below shows an example of an ACENet audio distribution:



Figure 13: ACENet audio distribution

For a list of ASTi-approved switches and FAQs, go to the *ACENet Technical User Guide* (support.asti-usa.com/media/pdf/acenet\_ug.pdf).

## 3.3 ASTiNet

ASTiNet is an ASTi-proprietary protocol that provides simple and flexible communication networking among Targets and other ASTiNet-enabled products. Some of ASTiNet's features include the following:

- *IPv6-based:* places it in position for use well into the future.
- *Auto configuration:* the IP broadcast and/or multicast addresses do not have to be set, providing a simple, plug-and-play setup.
- *Peer-to-peer:* eliminates the requirement and bottleneck associated with a central server.
- *Voice-over-IP capability:* provides easy setup and use for many-to-many communication mechanisms.
- *Radio simulation:* simple setup for when ease of use is more important than DIS.
- *Flexible message format:* provides extensibility for as-yet unforeseen applications.

ASTiNet is a fundamental networking protocol incorporated in Telestra with device domain configuration, providing support to other protocols, such as DIS and HLA.

#### 3.3.1 Voice over IP (VoIP)

ASTiNet VoIP was designed around the idea of a plug-and-play communication architecture that removes the need for a detailed understanding of the underlying principals of communication. At the heart of ASTi VoIP architecture are the core characteristics that ASTi considered during initial design:

- Easy setup and use
- Point-to-point and conference bridge support
- Based on IPv6, with features such as Quality of Service (QoS) and security
- Minimal configuration requirements for wide area networks (WANs) and firewalls
- Automatic setup when feasible
- Peer-to-peer paradigm (i.e., no single point of failure)
- Features geared toward DoD and the gaming world

## 3.4 DIS

When the Target is set for Distributed Interactive Simulation (DIS) operation, it connects directly to the DIS network. DIS is a simulation protocol standard developed jointly by industry and the military to enable interoperation of simulation and training devices over local area networks (LANs) and wide area networks (WANs).

One of the more difficult and often underestimated aspects of simulation over LANs and WANs is achieving a realistic radio communication environment. When DIS is active, the Target's local radio and intercom modeling extends over LANs and WANs. Communication simulation between multiple DIS-compatible network devices is invisible with full radio modeling across systems. ASTi supports recent versions of the DIS standard.

During DIS operation, the Target transmits and receives DIS-standard protocol data units (PDUs). Since the Target is involved strictly with communication simulation, it is only concerned with Transmitter, Signal, and Receiver PDUs. The exception to this is Entity State PDUs, which accommodate entity attach features, whereby a radio modeled on the Target is attached to an entity on the network.

## 3.5 HLA

Unlike many other high-level architecture (HLA) solutions, ASTi implemented HLA from the ground up to fully exploit the flexibility and interoperability of Defense Modeling and Simulation Office's (DMSO's) HLA 1.3 standard. Multiple run-time infrastructure (RTI) support, established and published radio simulation object model (SOM), agile federation object model (FOM) capabilities, back-channel communication options, and debug tools offer a well-supported HLA environment. In addition, ASTi's Target platform takes advantage of high-performance, industrial, off-the-shelf technology to provide increased HLA performance and reliability at a reasonable cost.



Figure 14: HLA network configuration

## **3.6 Future protocols**

ASTi based its core communication protocol around ASTiNet because it can easily translate to other protocols. Currently, this includes Distributed Interactive Simulation (DIS) and high-level architecture (HLA); however, we are always looking to add new protocols to our product suite based on market demands. So if Session Initiation Protocol (SIP) Voice over IP (VoIP), Test and Training Enabling Architecture (TENA), or others are required for your communication application, contact ASTi.

## 4.0 Telestra concepts

This chapter discusses the following topics:

- Data flow
- ACE Studio concepts
- System default login credentials
- Cold starts
- System configuration
- System Backup/Restore

### 4.1 Data flow

Telestra's concepts are very fundamental in understanding how the applications work together. Simply put, the project containing all model data and system configuration resides on the Target, and you can manipulate it using ACE Studio software. The complicated part is understanding the breakdown of information flowing between real-time and non-real-time. Figure 15, "Data flow" below displays the general flow of information from the Studio to the Target over the network.



-Repository is where Projects are stored -Load in Real-time -Project Manager only stores local copy until saved

Figure 15: Data flow

#### 4.2 ACE Studio concepts

In ACE Studio, a project consists of several layers of audio system hardware, software models, and network configuration parameters. ASTi created these layers of information to extract all networking configuration and hardware specifics from the model, which lets you change the model without reconfiguring parameters.

ACE Studio projects contain several layers. The first layer in a project is the layout, which contains the project's configuration. Each layout assigns the resources to the load. These resources include domains, comm plans, and sound repositories. The load consists of sets of models created in **ACE Model Builder**. The model layers are similar to past ASTi simulation models with components and primitives to drive the components.



Figure 16: Project layers

## 4.3 System default login credentials

Every system is set with the following default login username and password. Change system passwords as necessary to meet system administration requirements.

Telestra root login:

Username	Password
root	abcd1234

Studio login:

Username	Password
aceuser	aceuser

Target login:

Username	Password
admin	admin

Remote Management System (RMS) login:

Username	Password
admin	astirules

## 4.4 Cold starts

The cold-start procedure allows you to build the systems from scratch. You may wish to cold start your system for the following reasons:

- 1. Install the latest software version
- 2. Rebuild a damaged hard disk
- 3. Create spare hard disks

To cold start your system, go to the *ACE Target and Studio Cold Start Guide* (support.asti-usa.com/media/pdf/t4/ace target studio cs.pdf).

### 4.5 System configuration

The Remote Management System (RMS) provides network configuration for you to specify the network interface for the system, including an Internet Protocol (IP) address, a card mode, and a subnet mask for the Target's three Ethernet interface cards.

telestra-	de-17-56 Netwo	ork Devices
System War	nings	
This system is	s not licensed for this softwa	re release. Please upload the proper file for this system on the Licer
Interface et	th0	Interface eth3
Mac Address:	00:07:b8:de:17:56	Mac Address: 00:1b:21:61:8a:f1
IP 4 Address:	10.2.100.245	Status: Off
IP 6 Address:	fe80::207:b8ff:fede:1756	Edit eth3 Config.
Subnet Mask:	255.255.0.0	
Mode:	dhcp	
DHCP Client ID	:	
Ignore DNS:	off	
Ed	<u>lit eth0 Config.</u>	
Interface et	th1	
Mac Address:	00:07:b8:de:17:57	
IP 4 Address:	172.31.23.87	
IP 6 Address:	fe80::207:b8ff:fede:1757	
Subnet Mask:	255.255.0.0	
Mode:	fixed	
Ed	<u>lit eth1 Config.</u>	
Interface et	th2	
Mac Address:	00:1b:21:61:8a:f0	
IP 4 Address:	20.1.1.1	
IP 6 Address:		
Subnet Mask:	255.0.0.0	
Mode:	fixed	
Ed	lit eth2 Config.	

Figure 17: Network Devices

## 4.6 System Backup/Restore

You can easily manage archiving the system configuration files through the backup and restore options below:

- *Back up:* use **Backup System Configuration** to back up any or all of the individual sections of the overall system configuration or back up the entire system. A **Backup System Configuration as Text Only** option also exists. This setting does not back up any binary information and is useful when exporting a project from a classified area.
- *Restore:* use **Restore System Configuration** to install or restore system configuration files from a previously created backup file. This file can be uploaded to the system if it is not already resident on the Target.
- *Manage:* use **Manage Backup Archives** to inspect, download, and/or delete **System Configuration** backup archives.

Figure 18, "System Backup/Restore" below shows System Backup/Restore:



Figure 18: System Backup/Restore

This section discusses the following topics:

- Backup System Configuration
- Restore System Configuration

#### 4.6.1 Backup System Configuration

**Backup System Configuration** archives projects, sound libraries, and system configuration information on a local workstation.



Figure 19: Backup System Configuration

Figure 19, "Backup System Configuration" above shows a variety of check boxes that back up projects, sound files, or system configuration sections:

- *CrownAmpMapping:* when renaming a Crown Power Amplifier or Ashly Power Amplifier in the RMS, the amplifier does not allow you to store a name internally. The new name is mapped to the amplifier's media access control (MAC) address(es) in a file in the Telestra. This setting saves that mapping information.
- *DefaultLoadandLayout:* saves the default load and layout information on **System Status**, which allows the Telestra run the required layout after perhaps a power interruption.
- *ProjectsArchive*: saves any projects stored on the Telestra and lists all projects.



*Note*: *This check box does not select the sound files associated with the project; you must select them under SoundLibrary*.

- *RMSUsersandConfig:* saves information for all RMS users and their passwords on the Telestra as well as any user lockout settings.
- *RecordReplay*: saves any audio files recorded with **RecordReplay**.

- *SoundLibrary:* saves any **Sound Libraries** stored on the Telestra. Alternatively, select **SoundLibrary** to list all libraries on the Telestra.
- *SpeakerEQConfig:* backs up **SpeakerEQ** configurations, which automatically equalize a speaker by comparing referenced and measured sound responses.
- SpeechRecognitionConfig: backs up speech recognition license files and configuration.
- *TelestraConfig:* backs up the Telestra configuration, which includes the host name and network settings.
- *TextToSpeechConfig:* backs up basic text-to-speech (TTS) configuration and file locations. It does not back up the TTS license or the TTS installed voices.
- *Backup filename prefix:* by default, the backup file name is **telestraConfig**, with the date and time appended to the file name. This box replaces **telestraConfig** with something more related to the information or project you are backing up.

Figure 20, "Backup System Configuration options" below shows **Backup System Con-***figuration* options:

deathstar Backup System Configuration	
Configuration Sections	
Please select the desired type(s) of system configuration info to back up.	
Check All Clear All	
CrownAmpMapping	
CustomSRModels	
DefaultLoadAndLayout	
HLAConfig	
<u>ProjectsArchive</u>	
RMSUsersAndConfig	
RecordReplay	
SoftwareLicenses	
SoundLibrary	
SpeakerEQConfig	
✓ TelestraConfig	
TextToSpeechLicenseAndConfig	
Backup filename prefix: telestraConfig	
Cancel Start Backup	

Figure 20: Backup System Configuration options

Select Start Backup, and a status page shows a message asking you to wait.



Figure 21: Start Backup

This page shows the categories selected for backup. At this time, the archive file has generated on the hard drive in the Telestra but has not transferred to the local computer.

To transfer the archive to your local computer, select the archive name.

Some browsers may ask where you would like to store the archive. The browser may also store the archive in a preset location.

The archive is now stored safely on your local computer.

#### 4.6.2 Restore System Configuration

**Restore System Backup** restores a model archive on the Telestra server or uploads and restores a backup file from your local computer. Figure 22, "Restore System Backup" below shows **Restore System Configuration**:

telestra-de-17-56 Restor	e System Backup	
Select which backup to restore, or upload a b	backup archive using the form below.	
Existing Backup Files		
Filename	Restore	
SERA_AudioBridge_RevA20181003-2051.tgz	restore now	
SpkrEq20181003-2049.tgz	restore now	
hail_test20181003-2050.tgz	restore now	
s97_Test20181003-2050.tgz	restore now	
telestraConfig20181003-2033.tgz	restore now	
Upload Previous Backup File		
Choose File No file chosen		
Upload Backup File		

Figure 22: Restore System Backup

To restore an archive, follow these steps:

You can restore a file (e.g., **ASTi-20151214-1603.tgz**) from the Telestra itself. Alternatively, select **Choose File** and **Upload Backup File** to upload an archive on your local computer.

Select Choose File, and find the backup archive on your local computer.

To transfer the archive to the Telestra, select **Upload Backup File**, and it displays under **Existing Backup Files**.

Under **Configuration Sections**, choose configuration(s) to restore, and select **Start Res-toration**.

Each check box allows you to restore a specific project, sound file, or section of system configuration:

- *CrownAmpMapping:* when renaming a Crown Power Amplifier or Ashly Power Amplifier in the RMS, the amplifier does not allow you to store a name internally. The new name maps to the amplifier's MAC address(es) in a Telestra file. Selecting this box restores that mapping information.
- DefaultLoadandLayout: restores the default load and layout on System Status.
- *ProjectsArchive*: restores any projects in the archive. Alternatively, selecting **ProjectArchive** lists projects in the archive so you can individually select them.
- *RMSUsersandConfig:* restores all information for the RMS users and their passwords in the archive as well as any user lockout settings.
- *RecordReplay:* restores any archived audio files recorded using **RecordReplay**.

- *SoundLibrary:* restores archived sound libraries. Alternatively, selecting **SoundLibrary** lists all sound libraries in the archive so you can individually restore them.
- *SpeechRecognitionConfig:* restores the speech recognition license file and configuration from the archive.
- *TelestraConfig:* restores the Telestra configuration, which includes the host name and network settings.



*Caution*: If you created a backup on a different Telestra, the configuration settings might not be applicable to this Telestra.

- *TextToSpeechConfig:* restores basic text-to-speech (TTS) configuration and file locations. It does not back up the TTS license or the TTS installed voices.
- Start Restoration: brings up a window showing that the restoration is in progress.



*Note*: A warning message may indicate if a project already exists on the Target.

Figure 23, "Select Sections to Restore" below shows Select Sections to Restore:



Figure 23: Select Sections to Restore

Once the restoration is complete, a page similar to Figure 24, "Restoration Complete" below appears:



Figure 24: Restoration Complete

**TelestraConfig** replaced the Telestra's settings with the archive's network configuration. To use the newly restored files, select **restart your Telestra system**. In the confirmation message, select **Reboot Telestra System Now**.



Figure 25: Reboot Telestra System Now

## 5.0 Software

Three main system software areas exist:

- Project Manager:
  - ° Manages the entire program or system
  - Acts as a configuration tool
  - ° Builds and installs layout
- Load Viewer:
  - ° Loads the configuration
  - <sup>°</sup> Develops, builds, and debugs models
- RMS:
  - Provides hardware configuration
  - ° Configures the network
  - Manages USB License Key(s)
  - ° Manages ACENet device (e.g., gain settings and firmware updates)
  - ° Manages system health and debug settings

This chapter discusses the following topics:

- Project Manager
- Load Viewer

## 5.1 Project Manager

Many of today's simulation and training applications have transitioned from beyond simple, stand-alone training devices to multiplatform, complex, networked simulation applications. **ACE STUDIO Project Manager** provides the ability to develop, set up, and manage sound and communications models, simulation applications, and other related elements across a set of platforms and applications. Projects can manage greater simulation complexities and allow successful interoperation.

A project is a sound and communications simulation scenario that consists of hardware (e.g., modeling platforms, audio and I/O distribution, simulation servers), simulation software (e.g., sound and communication models, satellite communications (SATCOM), terrain, data link), and configuration elements (e.g., communication plans, entity assignments, exercise parameters).

A project can represent the sound and communication hardware, software, and models for a simple, standalone desktop simulator. On the opposite end of the spectrum, a project can encompass many training devices and applications participating in a WAN-based simulation architecture or exercise.

This section discusses the following topics:

- Project elements
- ACE STUDIO Project Manager
- Layout

Figure 26, "ACE STUDIO Project Manager" below shows the ACE STUDIO Project Manager:



Figure 26: ACE STUDIO Project Manager
### **5.1.1 Project elements**

Projects contain elements that develop, set up, and manage a complex sound and communication simulation scenario across network-based ASTi hardware and simulation applications. Projects elements include the following:

- *Targets:* embedded modeling platforms that run sound and communication models and other ASTi simulation applications that you can develop and set up.
- *Audio and I/O distribution devices:* include ASTi's ACUs, ACU2s, ACE-RIUs, and ACENet-compatible audio amplifiers.
  - *ACUs and ACU2s:* provide remote digital audio and I/O distribution between Targets and audio peripherals (e.g., military and commercial headsets, powered speakers, tape units, DVRs, real-world communication equipment). Distribution is via ASTi's ACENet protocol over dedicated Ethernet-based networks.
  - *ACE-RIUs:* remote audio interface devices that connect remote operator headsets, speakers, and control panels to a central Target via ACENet.
  - *ACENet:* compatible audio amplifiers that reinforce sounds in environmental cue applications. These amplifiers connect to ACENet, eliminating the need for an individual ACU and audio amplifier in environmental cue applications.
- *Simulation servers:* Telestra simulation server software runs server-based simulation applications and services, such as satellite communications (SATCOM), terrain, high-fidelity (HF) propagation environments, high-level architecture (HLA), data link, and Network Time Protocol (NTP).
- *Communication and sound models:* distribute, link, and manage communication and sound model elements as part of the project. Develop models with ACE Studio's model generation tools.
- *Sound repositories:* develop and manage recorded sound libraries from sound and communication models.
- *Host interface configuration:* set up host interfaces to sound and communication models as part of project development. Develop model elements that are reusable across platforms and are agnostic to any particular host simulation software's structure.
- *Comm plans:* with ACE Studio's communications planning tool, radio, intercom, and other communication assets, set up and manage across a set of models and applications to help ensure interoperability. Comm plans change, store, and reuse communication parameters for different exercises.
- *Domain:* the project manages domain-related parameters, such as entity assignments, DIS, and HLA parameters.

• *Loads:* the project configures models, simulation applications, host interfaces, and other elements for each Target.

### 5.1.2 ACE STUDIO Project Manager

ACE STUDIO Project Manager develops and manages projects. This tool is part of ASTi's ACE Studio software suite. When launched, the ACE STUDIO Project Manager searches the network for available Targets. It then queries found Targets and generates a list of existing projects on them. You can then choose to work with any project from the available list. Alternatively, build a new project, and develop on any available Target. Install, develop, and manage all projects on that Target.

In ACE STUDIO Project Manager, perform typical file operations on a project, such as Save, Save As, Open, Close, and New. To push model changes back to the Target, save in ACE STUDIO Project Manager.

A built-in control management system also tracks changes to the Target. As a result, you can manage projects much like software source code. Features such as change tracking, change descriptions, release management, and the ability to return work with earlier release instances provide powerful configuration management capabilities.

Some project management features are also available through the Remote Management System (RMS). Pointing a network browser at a Target from any convenient computer connected to the same network accesses the RMS application on that Target.

Within the RMS, select ACE STUDIO Project Management. From the pages under this tab, view local and global projects.

- Local projects (i.e., projects on the Target):
  - ° Display list of projects on the current Target
  - ° Back up projects from the Target
  - ° Delete projects from the Target
  - View Change Logs of each project
- Global projects (projects on other machines visible to Targets over the network):
  - ° Display list of projects on other Targets
  - View change logs of each project on other Targets

### 5.1.3 Layout

A layout is a graphical and textual representation of your project with configuration parameters. The layout collects user and tool-generated elements:

- Hardware
- Models
- Interfaces
- Communications assets
- Exercise and communication-planning parameters

In ACE STUDIO Project Manager, select links and set these elements from the current project libraries to create an executable layout. You can also add and generate new project elements.

Links between icons show dependencies and associations of the individual project elements. For example, a link from a load element to a Target element indicates the load will be installed and run on that particular Target. Link a communication plan element to several Targets, indicating that each Target will use the communication plan after executing the layout.

# 5.2 Load Viewer

A load is a collection of models linked to form a communication and sound model to run on a selected Target. A load is a model set for a specific Target, whereas a project is a complete configuration of loads, comm plans, and servers across one or more Targets, servers, and simulation applications. Generate models via ACE Studio's model development tools, through the various helpers, or a combination of both.

To generate an empty load in ACE STUDIO Project Manager, right-click Loads, and select Add.

The ACE Studio model builder generates the models in the load. Create a load from scratch. To add the current load to the project, select **Save**.

To apply a load to a Target, double-click the Target in **Layout**. The Target configuration window appears with a load list. Choose a load to run on a Target.

Figure 27, "Text view of a load" below shows the text view of a load in ACE STUDIO Project Manager:



Figure 27: Text view of a load

This section discusses the following topics:

- Models
- Servers

### 5.2.1 Models

Models are the individual modeling elements generated by either helpers and/or ACE Studio's modeling environment. A model can be small and simple (e.g., a set of components that model an engine sound, a shipboard binaural operator, or an F-16 Caution Warning system). A model can also be large and complex, such as the entire communication system for an F-18 platform.

Models are self-contained and can link together; therefore, you can create a library of reusable model components to build larger, more complex models.

Add models that the project helpers or that ACE Studio's model development tools build to a project. The model folder shows all models that Targets in the project use. Additionally, add models to the project to create a library of reusable components, regardless of whether the current layout uses them.





Figure 28: Models view

### 5.2.2 Servers

In addition to Targets, you can add server platforms to a project. From a hardware standpoint, server platforms are additional Target platforms connected to a set of Target systems over a network. The servers run ASTi's server-based simulation applications:

- High-level architecture
- Satellite communications (SATCOM)
- High frequency (HF)
- Automatic link establishment (ALE)
- Terrain

Server platforms can also provide traditional, server-based services, such as Network Time Protocol (NTP).



Figure 29: Servers

As the name implies, these simulation applications provide simulation capabilities and features to a collection of Targets. To add servers to the project, right-click the **Layout** canvas and add **Server**. Double-click **Server** to open up the configuration tool, and choose the simulation services that support the application.

The most commonly used feature is **DIS Gateway**, which provides the interface to the DIS network:

- a. Set DIS version to 4, 5, or 6.
- b. Set the **DIS interface** to eth0, 2, 3, or 4.
- c. Set the DIS RX/TX port (e.g., 53000).

d. In **main**, set the outgoing destination address for DIS packets (e.g., 255.255.255.255). Choose **broadcast** or **multicast**.

	DIS Gateway	×
File Info:	(* indicates required field)	
Name* :	DIS	
General		
version	6	
Interface	s	
DIS inter	face eth0 🔻 port 53000	
main	10.2.255.255	$\odot$ bcast $\bigcirc$ mcast
signal		$\odot$ bcast $\bigcirc$ mcast
receiver		$\odot$ bcast $\bigcirc$ mcast
entity		$\odot$ bcast $\bigcirc$ mcast
TDL link	interface 🕶 port	
TDL		ucast
path inte	rface • port	
path		$\odot$ bcast $\bigcirc$ mcast
hf interfa	ace v port	
hf		● bcast ⊖ mcast
		Cancel OK

Figure 30: DIS Gateway

# 6.0 Model services

This chapter discusses the following topics:

- Intercoms
- Sound repositories
- Math Plan
- Radios
- Commplan Editor
- Radio Monitor
- Domain Editor
- Helpers

### 6.1 Intercoms

**Intercom** components relate to internal communication paths within the model. This group includes the communication panel and local **Intercom** buses. Audio on **Intercom** buses never transmits to the voice network. These buses internally pass audio around.

**Intercoms** provide an **Intercom** audio bus structure that other components can use to distribute audio throughout a model and to simulate **Intercom** bus structures in simulation applications. A network version allows an extension of **Intercom** buses between systems using simulation industry standard Distributed Interactive Simulation (DIS) or high-level architecture (HLA) protocols.

Figure 31, "Intercom example" below demonstrates how Intercom works:



Figure 31: Intercom example

# 6.2 Sound repositories

Sound repositories are the sound file libraries that support sound and communication models. For example, an AH-64D Apache model may use a library of recorded sounds (e.g., weapon launches, engine ignitions, touch-down thump, caution/ warning voice alerts, threat alerts, crypto tones, FH equipment beeps and squawks).



Figure 32: ACE STUDIO Project Manager

**Sound Library Editor** creates, reuses, and manages these libraries. The editor also structures these sounds, allowing you to build sound groups in an individual sound library. For example, you might create an F18 communications sound library with around 20 sound files. You can then group these sounds inside the library by function (e.g., ARC-182, caution, weapons). Reuse the sound repository apply it to multiple F18 platforms.

A waveset is necessary to uploading sound files, because it creates a folder location where all sound files are stored. To upload sound files, go to the *Remote Management System User Guide* (support.asti-usa.com/media/pdf/t4/rms4\_ug.pdf)

All files must be 48 kHz, mono, 16-bit windows pulse-code modulation (PCM). All sound files are uploaded to the following file location on the Target: **var/local/asti/soundfiles**/*wave-set*, where *waveset* is the waveset's name. To load sound files at model load, set the buffer to **True**.



Figure 33: Sound Library Editor

# 6.3 Math Plan

**Math Plan** provides access to various mathematical functions that you may apply to the layout. These functions permit local data manipulation in the models. This tutorial demonstrates how to use the **Math Plan** in **MathFunction**.

When you create a new project, ACE Studio automatically creates a default **Math Plan** called "mathplan." To rename the **Math Plan**, follow these steps:

- 1. On the canvas, right-click Math Plan (5).
- 2. Select Clone and Replace.
- 3. Enter a name for the Math Plan.
- 4. Right-click the server icon (\Box), and select edit. On core, you can now view the Math Plan's new name in select.

To set up a Math Plan, follow these steps:

- 1. Add a new project, or open an existing project.
- 2. In the layout, double-click mathplan.



Figure 34: Icon for mathplan

3. Select Add Group, and name it New\_Group.

· ·	Mathplar	n Editor - mathp	olan		-		×
Add Group Add Function	one Function	🗵 Delete	Notify Tare	jet			
Function Groups	Functions						
Group	Handle	Function Type	Description				
basic	Add	ADD	None	F(x,y) = x	+ y		
new_group	Subtract	SUBTRACT	None	F(x,y) = x	- y		
	Multiply	MULTIPLY	None	F(x,y) = x	* у		
	Linear	LINEAR	None	F(x,y,z) =	x * y	+ z	
	Logical-and	LOGICAL-AND	None	F(x,y) = x	& у		
	Logical-or	LOGICAL-OR	None	F(x, y) = x	(   y		
	Logical-xor	LOGICAL-XOR	None	F(x, y) = x	( î y		
			Cancel	Apply		ОК	

Figure 35: Mathplan Editor

4. Select Add Function, and name it Table\_Function.

Add Group Add Function Cl	one Function	😢 Delete	Notify Target	
Function Groups	Functions			
Group	Handle	Function Type	Description	
basic	Add	ADD	None	F(x,y) = x + y
new_group	Subtract	SUBTRACT	None	F(x,y) = x - y
	Multiply	MULTIPLY	None	F(x,y) = x * y
	Linear	LINEAR	None	F(x,y,z) = x * y + z
	Logical-and	LOGICAL-AND	None	F(x,y) = x & y
	Logical-or	LOGICAL-OR	None	F(x, y) = x   y
	Logical-xor	LOGICAL-XOR	None	$F(x, y) = x \uparrow y$
	Table_Function	None	None	none
		Cane	cel App	ок

Figure 36: Math Plan function

5. Select **Function Type**, and then select **TABLE**.

· · ·	Mathplan Editor - mathplan								
Add Group Add Function Cl	one Function	🛞 Delete	Notify Target						
Function Groups	Functions								
Group	Handle	Function Type	Description						
basic	Add	ADD	None	F(x,y) = x + y					
new_group	Subtract	SUBTRACT	None	F(x,y) = x - y					
	Multiply	MULTIPLY	None	F(x,y) = x * y					
	Linear	LINEAR	None	F(x,y,z) = x * y + z					
	Logical-and	LOGICAL-AND	None	F(x,y) = x & y					
	Logical-or	LOGICAL-OR	None	$F(x, y) = x \mid y$					
	Logical-xor	LOGICAL-XOR	None	$F(x, y) = x^{y}$					
	Table_Function	TABLE -	None	none					
		Can	cel App	ок ОК					

Figure 37: New Table Function

6. To the right of **Description**, double-click **edit**.

7. Fill in the table values:

r			Table I	uncti	on		×
Han	dle: Table_	Functio	'n		✓ Interpol	ate	
× =	0.0	f(x) =	0.0	х =		f(x) =	
× =	0.1	f(x) =	0.01	x =		f(x) =	
× =	0.2	f(x) =	0.04	х =		f(x) =	
× =	0.3	f(x) =	0.09	х =		f(x) =	
× =	0.4	f(x) =	0.16	х =		f(x) =	
× =	0.5	f(x) =	0.25	× =		f(x) =	
× =	0.6	f(x) =	0.36	× =		f(x) =	
× =	0.7	f(x) =	0.49	х =		f(x) =	
× =	0.8	f(x) =	0.64	х =		f(x) =	
× =	0.9	f(x) =	0.8	× =		f(x) =	
× =	1.0	f(x) =	1.0	× =		f(x) =	
× =		f(x) =		х =		f(x) =	
× =		f(x) =		х =		f(x) =	
× =		f(x) =		× =		f(x) =	
× =		f(x) =		х =		f(x) =	
× =		f(x) =		х =		f(x) =	
					💥 Cance	el	∉ОК

Figure 38: Table Function

- 8. To close **Table Function**, select Apply
  - select Apply and Cok
- 9. Select ok to close Math Plan Editor.
- 10. Save and install the project.
- 11. Double-click the server icon ( $\blacksquare$ ) to open **Model Viewer**.
- 12. To add a simulation model, right-click the canvas.
- 13. To open the model, double-click the **Model** icon.
- 14. Add a MathFunction component.

Add	to Model: 'Fo	lders_Model' _		×					
Items P	alette								
Int	FlexTable			Ш					
IntTable									
Latch									
LogicTable									
Ma	MathFunction								
Nu	NumToString								
Pa	ssThrough								
Dynar	nics								
<ul> <li>EnvCu</li> </ul>	ie								
Destination	n Folder :	Maths	<u>cha</u>	inge					
Item Name	e : Lookup_T	able							
		Cancel	Add						

Figure 39: Add Table Function

15. To open the Ace Data Viewer, double-click MathFunction.

- 16. Double-click select in Value for the Function variable.
- 17. In the new group, select the **Table\_Function** that you created in Step 5.
- 18. Select 🖉ок
- 19. In the Ace Data Viewer, double-click Modifier for *Input X*, and enter 0.6.

				Folde	rs_Model:Lookup_Tabl	e (/Maths	:/) - A	ce Da	ita Viewer	-	×
Lo	oku	р_Та	able [Cont	rol/MathF	unction]						
Da	ata	Link	s Schema	atic Info	View/Edit Description						
							J	Filte	r View	Full View	
	Fr	om	Variable	Туре	Value	Ор	Mod	ifier	Result	То	
			Function	function	new_group.Table_Func	tion					
			Input_X	float32	1.0	*	0.60	0000	0.600000		
			Input_Y	float32	1.0	*	1.0		1.0		
			Input_Z	float32	1.0	*	1.0		1.0		- 1
			Gain	float32	1.0		1.0 1.0			- 1	
			Result	float32	0.360000		0.360000			- 1	
											- 1
											- 1
											- 1
											- 1
											- 1
											- 1

Figure 40: MathFunction

The *Result* is now 0.36, which matches the table's output for an input of 0.6.

### 6.4 Radios

**Radios** are the largest, most complex, and most used components in ACE Studio. The following is a summary list of radio features for all simulated radios:

- World Position: defines the X, Y, Z coordinates of the radio's location.
- *Frequency:* defines the center of the radio tune frequency for transmit and receive. Optionally, it can define separate transmit and receive frequencies.
- *Antenna Gain:* simulates the size and radiative efficiency of the antenna. All modeled antennae are isotropic.
- *Squelch*: a noise gate that only allows signals exceeding a specified strength to filter through and play.
- *Background Noise:* the general noise created when using radios.

- *Fill:* allows you to choose one of a set of *N* predefined radio fills, as defined in a global comm plan.
- *Multiple Net Support*: provides the radio with the ability to support multiple nets per a specific fill. Nets define the following core radio characteristics:
  - *Multiple Modulation Type:* describes the modulation parameters of the radio, such as AM, FM, SATCOM, HQ, Intercom, etc.
  - *Amplitude Modulation (AM) and Frequency Modulation (FM):* two primary modulations for radio operation.
  - *Modulation Discrimination*: occurs when radios can only receive signals from radios with the same modulation type.
  - *AM Mixing:* when multiple AM signals broadcast on the same channel frequency, the AM receiver will receive all signals and mix them according to signal strength.
  - *FM Capture Effect:* when several FM radios are transmitting on the same frequency, an FM receiver will only be able to receive the strongest signal.
  - *Sensitivity:* receiver sensitivity in dB.
  - *Bandwidth and Bandwidth Overlap*: determines the amount of audio noise mixed into the received audio, based on the simulated bandwidth of the radio band. This parameter does not affect the in-tune calculation.
  - *Encoding Type and Rate:* defines the audio encoding type, including muLaw, continuously variable slope delta (CVSD), pulse-code modulation (PCM), and the sample rate.
  - ° Transmit Power: indicates the transmission power of the radio in Watts/dBm
  - *Automatic Gain Control (AGC)*: adjusts the gain to appropriate levels for a range of input signal levels.
  - SATCOM Parameters: defines satellite mode parameters, if applicable.
  - *Frequency Hopping*: a method of rapidly switching frequencies while a receiver and transmitter communicate. The receiver and transmitter have to jump between the same frequencies, at the same speed, and at the same time.
  - *Crypto Parameters:* radios that scramble the signals before they are transmitted so that only receivers who know the special key will have the ability to decode them, producing a secure voice transmission across any frequency.
- *Half Duplex and Full Duplex:* half-duplex mode is when the radio is able to transmit and receive signals but cannot do both at the same time. Full-duplex mode allows radios to transmit and receive signals at the same time. Typically, full-duplex mode is only used for intercom systems but never for real radios.

- *Propagation:* the movement of the radio waves as they move away from the transmitting antenna.
  - *Ranging:* an effect that occurs as a result of the distance between two radios. The greater the distance between the radios, the weaker the signal due to the dissipating power of the signal as it traverses a large area.
  - *Occulting*: the loss of radio signal due to the curvature of the earth's horizon.
  - *Ionosphere Effects:* the loss of signal due to the changes in the earth's atmosphere such as time of day or different seasons. The ionosphere effects only occur with high-frequency (HF) radios.
  - *Line of Sight (LOS)*: when radio waves traveling in a straight line are dispersed due to obstacles or obstructions.
  - *Fresnel Diffraction*: loss of signal due to the reflection of off obstacles in the path of the radio waves from transmitter to receiver.
  - *Terrain Effects*: the loss of signal due to land obstruction, such as a mountain.

This section discusses the following topics:

- Local radios
- Comm Plan
- Add a model

# 6.5 Commplan Editor

**Commplan Editor** creates a library of radio "fills" consisting of crypto, frequency hop, waveform types, nets, and other necessary parameters for the simulated radios in the project. You can create multiple comm plans and store them as part of the project.

In this way, you can apply and install different plans with relative ease to support changing operational or exercise requirements. For example, the day-to-day operations of an F-16 simulator may use one plan that provides the trainer communication simulation as tested and signed off with the device. However, other plans may be applied when the F-16 device is used in network-wide exercises.

Commplan Editor - commplan									
	+								Expand All
commplan	AM FM INTERCOM	Mode AM FM INTERCOM	Encoding MULAW MULAW MULAW	Rate 8000 8000 8000	Bandwidth(Hz) 25000 25000 25000	Tx Power(Watts) 1.0 1.0	RECEIVERGAIN rxgain.rxgain rxgain.rxgain Off	Voice ON OFF	e Effects
Opened Commplan 'commplan'	[						Cancel App	oly	ОК

Figure 41: Commplan Editor

# 6.6 Radio Monitor

**Radio Monitor** is a network-debugging tool that examines radios in the local radio environment and other servers. Updating in real time, **Radio Monitor** views the radios on the network. Radio details include the following:

- Source or domain
- Ether
- Frequency
- Mode
- Target IP address
- Name
- Protocol ID

Under Name, view the DIS ID. The DIS ID is defined in the following order:

- Exercise ID
- Site and application IDs from the IP address of the local Target
- Entity
- Radio ID

The Radio Monitor view tabs:

- *Radios:* shows receivers and transmitters on the network.
- *DIS Maps:* displays the DIS gateway mapping to DIS identifiers to Universally Unique Identifiers (UUIDs). ACE radios are identified via UUIDs.
- *Statistics*: provides diagnostics information.

			-		×					
File Help	File Help									•
Radios DIS Maps Statistics										
ASTiNet 8	ASTiNet & DIS  All Ethers  Center-of-Earth DIS Port 53							00		<b></b>
Transmitters   DIS:24			Expired 5 sec updates     Active			Multicast Addr				
Domain 🔻	Ether	Frequency	Mode	Target	Name			Protocol Id		
DIS:24	Generic	200.000000MHz	FM	IP:10.2.132.107	DIS:132.	107.35395.62	2908	DIS:		
DIS:24	Generic	200.000000MHz	FM	IP:10.2.132.107	DIS:132.	107.35396.62	2907	DIS:		
DIS:24	Intercom	Channel 48	Intercom	IP:10.2.132.107	DIS:132.	107.35407.62	2896	DIS:		

Figure 42: Radio Monitor

In the radio list, all transmitting or receiving radios are green. Double-click any radio available on the network to view the details. The radios time out after 12.5 seconds and turn white in the list. The radio disappears from the list after 25 seconds if it receives no further updates.

**Online** displays all Targets available on the network and allows selection of a specific Target to view traffic visible only to that Target.

To view the general radio details, open a radio in the radio list. The radio general information includes the following:

- Radio Name
- Model Name
- Target Name
- IP Address
- Protocol ID
- Universally unique identifier (UUID)
- Domain

At the bottom of the window, view in-tune radios, and select each name to view their details. **Out-Of-Tune** is blank if all radios are in tune. Factors include occulting, ranging, terrain, and squelch.

	Transmi	tter DIS:132	.107.35396.62	907 on IP:10.2.13	2.107 ×
General	Properties				
Name: Model Na	ame:	DIS:132.1	07.35396.6290	)7	
Target Na	ame:	IP:10.2.13	2.107		
IP Addres	is:	10.2.132.J	107		
UUID:	iu.	e198f50b9	9d50e103500	4157c42471c6	
Domain:		DIS:24			
NetName	2:				
Mode:		FM		Power:	30dbm
Frequenc	iy:	200.00000	0MHz	Bandwidth:	25khz
On or Off	r/ //tom:	On		Active?	INO
SatCom (	hannel <sup>.</sup>			SatCom Mode:	
Ha Sync	Time Off:			Hg Time of Day:	
Hq Sec K	ley:			Hq Word of Day:	
Hq Syste	m:			Hq Start of Msg?	•
Sg Sync	Time Off:			Sg Hop Set:	
Sg Sec K	ey:			Sg Lockout Set:	
Sg Start o	of Msg?	ν.	0	Latin de l	
Location		X: Y-	0	Latitude:	
		Z:	0	Elevation:	
Updates:		67329			
Off Coun	tdown:	12482		Expire Countdow	n: 24982
Audio Cor	unt:	0		Bytes:	0
Last Enco	ode:	Invalid		Last Rate:	0
Last Sam	ples:	0		Last Bytes:	0
Local Inti	ine Rv:			Audio Plaving?	
TDL Cour	nt:	0		Total Bits:	0
Local Intu	ine Rx:			Total Messages:	0
Last Rate	:	0		Last Bit Length:	0
Last Type	2:	0		Last Class:	0
Last Msg	Count:	0			
Target	Receiver	Frequency	Out-Of-Tune	e Reason Range	Path Factor
					Close

Figure 43: Radio Monitor detail

### 6.6.1 Radio filters

**Radio Monitor** displays radios available on the ASTiNet, Distributed Interactive Simulation (DIS), or both networks. Filter out a network, or view both simultaneously. Filter the radios as transmitters or receivers or by assigned domain name. Alternatively, you can view all domains.

Ethers identify groups of radio types that interoperate with each other. For example, AM and FM radios operate in the same generic ether, so when an AM and FM radio have the same frequency, they interfere with each other. On the other hand, Intercom radios and Voice-over-IP (VoIP) radios do not interact with AM or FM radios or with each other. Intercom radios and VoIP radios exist in separate ether groups.

Filter radios by the following:

- All Ethers: view all radio types.
- *Generic:* view AM, FM, continuous waveform (CW), upper sideband (USB), lower sideband (LSB), single sideband filter (SSBF), jammer, pulse, and satellite communications (SATCOM). Generic radios tune via frequency.
- *Intercom*: view only intercoms. Intercom radios tune via channel number.
- VoIP: view only VoIP radios, which tune via net name.
- *HaveQuick*: view only HaveQuick radios, which tune via spread spectrum net ID.
- *SINCGARS*: view only SINCGARS radios.

Other filters include the following:

- *Center of the Earth:* radios located at the center of the earth (i.e., 0,0,0).
- *Expired*: radios that are timed out.
- *Active*: active radios on the network.

### 6.6.2 Statistics

		ASTINet/DIS Monitor				_ 0
File Help					Online	
Radios DIS	Maps Statistics					
Domain 🔹	<ul> <li>Channel</li> </ul>	Address	Domain:	All Don	nains	
*Monitor	linux-4	FF18:4571:FEED:030c:29a5:e35c:f7eb:8406	Channel:	All Cha	nnels	
*Monitor	linux-6	FF18:4571:FEED:0307:b8dc:a5a6:b31a:45b2	Address:			
*Monitor	linux-6.local	FF18:4571:FEED:030c:29a5:e35c:4515:b1f5				
*Monitor	localhost.localdomain	FF18:4571:FEED:0315:175e:441a:ea87:90d9	Other:		Net:	
*Monitor	Mordor-RH6-Work	FF18:4571:FEED:0307:b8de:f604:1f28:eede	Transmitte	er: 1506	IParm:	
*Monitor	mordor-rh7	FF18:4571:FEED:0307:b8de:f604:48f2:bfe2	Receiver:	9034	CR Publication:	
*Monitor	mwy167	FF18:4571:FEED:03d0:c9ad:f774:29c7:2064	TDL Link:		Msngr Cmd:	
*Monitor	mwy49	FF18:4571:FEED:0316:7638:c6be:96a0:fc4e	Audio:		Msnar Resp:	
*Monitor	mwy67	FF18:4571:FEED:0307:b8de:1d68:58:2161	TDL		Publication:	
*Monitor	mwydev	FF18:4571:FEED:0316:7638:c6be:f5d0:2575	Entitu		Dath	
*Monitor	NAK-Target2	FF18:4571:FEED:03d0:c99e:120b:e741:d76	Entity.		Faul.	
*Monitor	OLD-ONE	FF18:4571:FEED:0307:b8de:f604:8964:1cc	Local Entit	y: 24082	Path Info:	
*Monitor	p8-oft-target2	FF18:4571:FEED:0307:b8dc:63a1:38e1:cc4	Bid:		DIS Map Report:	
*Monitor	p8-wtt-target1	FF18:4571:FEED:0307:b8dc:65e4:c4f7:fa64	Declare:	6092	NM Query:	450
*Monitor	p8-wtt-target3	FF18:4571:FEED:031a:d401:5806:7a13:3360	Solicit:		NM Report:	570
*Monitor	PHETelestra	FF18:4571:FEED:03d0:c99e:33b0:d648:fe92	Listen:	32806	ANZAC:	
*Monitor	SDMT4	FF18:4571:FEED:0307:b8dc:60a0:72ad:fcbe	Announce		Fire:	
*Monitor	speechrec7	FF18:4571:FEED:0307:b8dc:9ccb:b729:94a0	Offer		Detonation:	
*Monitor	telestra-00-10-38	FF18:4571:FEED:0307:b8dd:ddd6:84dd:1b6a	Connecti		Environmental Dro	
*Monitor	telestra-01-ac-68	FF18:4571:FEED:03bf:101:ac68:2dcb:ee76	Connect.		Environmental Pro	cess:
*Monitor	telestra-01-AC-68	FF18:4571:FEED:03bf:101:ac68:a780:5346	Control:		lotal:	648

In the Radio Monitor, Statistics displays the radio details for diagnostic purposes:

Figure 44: Radio Monitor Statistics

### 6.7 Domain Editor

**Domain Editor** sets standard DIS or HLA parameters to a specific exercise in the project. In a DIS exercise, **Exercise ID**, **Site ID**, **App ID**, **Entity ID**, and **Radio ID** must uniquely identify each simulated radio. Simulated radios must operate on the same exercise ID and frequency to communicate with each other.

First, add a domain, and enter an exercise ID. Select the **Site ID** and **App ID**, which default to the last two IP octets:

			Domain Editor - domain	-		×
Names	HLA					
-Add Do	omains		Domain Info:			
	-	-	HLA:		]	
			Cancel Apply		ОК	

Figure 45: Domain Editor

# 6.8 Helpers

Helpers are specialized tools in **Project Manager** that build various types of project elements. Helpers include the following:

- Channel Helper
- Communication Planner
- Math Plan
- Domain Editor
- Sound Library Editor
- Radio Helper
- Host Interface Helper
- Loads
- Models
- Server Configuration Helper
- Test Plan
- HIT Plan

#### • Speech Recognition (SR) Plan

#### • Cell Plan

Each helper produces project elements for the project tree. Elements live in their own respective folders for easy visibility and access. In this fashion, libraries of reusable elements are created for future use.

Helpers allow you to quickly build and manage complex simulation models by creating reusable elements, which ensures consistency and interoperability within the simulation application.

### 6.8.1 Channels

At their basic level, channels are audio connection points in the sound and communication model. A simple example is a mix of environmental cue sounds that must route to a particular speaker location. Another, more complex example is an operator with a microphone input and headset output and a communication panel structure and communication asset links (e.g., radios, intercoms).

While you can generate these structures using ACE Studio, **Channel Helper** is a useful alternative, because it can automatically generate some of these more complex, audio-related modeling structures. **Channel Helper** creates modeling elements that you can reuse in a sound or communication model.

	Channel Helper	×
Channel Group Info (* indicate Group Name* : channels	s required field):	
Add Channels:	OP SINCGARS VOISUS	
Channel Name ACU2:Left_Front ACU2:Right_Front ACU2:Left_Side	Name: ACU2:Left_Side Comment: Left Side Speaker (Assignment:	
ACU2:Right_Side ACU2:Subwoofer	Bus: channels Bus Group: Left_Side	
	Highway: Out     Left_Side     Left     Right       Speaker:     Left_Side       Speaker Position: X     0.0     Y     0.7     Z     0.2	
	Cancel	date

Figure 46: Channel Helper

For example, it is much easier and more consistent to use **Channel Helper** to generate three generic IOS operator positions than to manually generate them in the modeling environment. To avoid losing work, do not edit objects that you created with **Channel Helper** in **Model Viewer** or helper-generated submodels that you generated in ACE Studio.

**Channel Helper** has the ability to create Audio I/O channels for ACUs, ACU2s, ACE-RIUs, and amplifiers. It can also create radio operator stations. Set up the operator stations with up to 16 radios for each operator. Set up SINCGARS for use with ASTi's simulated SINCGARS panels.

# 7.0 Host Interface

Hosts are representations of packet interfaces that link state data from your host application software to user-developed sound and communication models in a project. Each packet interface structure built into a model has a corresponding host at the project level. You can input or output hosts, since state information can flow between models and host applications in either direction.

The ACE Studio host interface consists of two parts:

- Host containers and their corresponding sockets at the project level
- Host I/O packets at the model level

Each host container icon on the canvas sets the packet interface with the appropriate User Datagram Protocol (UDP) port and physical Ethernet port (i.e., Eth0, Eth2) on a Target.

In the host model, each host I/O packet defines the information in the host UDP packet. The host I/O packet is commonly called the Interface Control Document (ICD). The ICD defines and controls input offsets, data types, and the UDP port number.

This approach removes the reference link to the models because they carry no specific network configuration information, making them reusable across platforms without configuration changes.

*Socket* refers to a network endpoint. This is where the IP address, Ethernet port number, and UDP port number are specified for input packets. Additionally, the destination IP address and UDP port number, payload size in bytes, and iteration rate for output packets are specified. **Host Packet Editor** points to these sockets.

This chapter discusses the following:

- Host control at the project level
- Host control at the load and model level
- Host interface exercise

# 7.1 Host control at the project level

In **Project** > **Host**, the host containers and sockets are created in each host container. Each socket is defined as *HostIn* or *HostOut*. Figure 47, "Host In" below shows *HostIn* in the **Host Interface Helper**:

	Host Interfa	ce Helper	×
Host Group Info (* indicates rec Group Name* : hosts	quired field):		
Add Hosts: Host Name HOSTIN:AURAL_DATA HOSTOUT:STATUS_DATA HOSTOUT:STATUS_DATA	Host Info: H Comment: Configure:- Interface: Port: Multicast IP* * Indicates of	OSTIN:COMMS_DATA Communications Interface eth2	
+ -		Cancel	date

Figure 47: Host In

For each *HostIn* socket, define the interface and port number. The port number selects the default network receive port for the packet data if it is an input packet or the transmit network port if it is an output packet.

For each *HostOut* socket, define the following:

- Destination IP address
- Port
- Packet length in bytes
- Send rate in Hertz (Hz)



Note: The send rate limit is approximately 20 Hz.

Figure 48, "Host Out" below shows the **Host Out** in the interface helper:

	Host Interface I	Helper >
Host Group Info (* indicates n Group Name* : hosts	equired field):	
Add Hosts: Host Name HOSTIN:AURAL_DATA HOSTIN:COMMS_DATA HOSTOUT:STATUS_DATA	Host Info: Name: HOS Comment: No Configure: Destination IP: Port: Payload Size: Send Rate: Send from port: Interface:	TOUT:STATUS_DATA Comment 10.2.132.106 10015 47 bytes 20 Hz • Random Same as Destination eth2
+ -		
		Cancel

Figure 48: Host Out

## 7.2 Host control at the load and model level

In the load, add a host model. Create the host I/O packets in a readily locatable block. Inside the host model, add either a *HostIn* or *HostOut* variable when creating a host I/O packet component:

	Add to Load: 'load' _		×			
Items						
▶ Har	rdwareModel					
▼ Hos	stModel					
	Aural					
	CommsIn					
1	HostDataIn					
	HostModel					
	Host_SystemAlerts		ш			
Rac	RadioModel					
▶ SimModel						
ltem Na	ame : Host_Hostdatain					
	Cancel	Add				

Figure 49: Add a host simulation model

Before getting into the details of *HostIn* and *HostOut*, there are a few general things to note about creating packets. You must assign each packet to a socket, and only one host I/O packet can link to a single socket at a time. Select **change** to assign the packet to a socket.

	Host_Aural-MaralHostin (Root) - ACE IO Packet Editor _ C X												
AuralH	turalHostin [hosts.Aural_Cue_In](change) 🚯 Controller Import ICD Live Capture 🗋 Enable Testing Little Endian 👻												
Packe	t Notes												
	Align Offset	Clear Testmode	Add to Co	nnector	Use Init Value:	Load/Sour	te Fail 🔻	Timeout	(s)	3.00			\$
Id.	Name	Offse 🔻	MsgLer Typ	e Init.	Value Function	Y	z	Rscale	Test	Test Value	Used by	Other	Description
1	Total_Sound_Freeze	0	0 bool	ean 0	Select	0.000000	0.000000	0.000000	off	0	multiple	<ramp></ramp>	Total Sound Freeze flag 0=Run 1=Freeze
2	Sound_Level	1	0 float	32 0.0	Select	0.000000	0.000000	0.000000	off	100.0	multiple	<ramp></ramp>	Aural cue system overall sound level (0-100%)
3	True_Airspeed	5	0 float	32 0.0	Select	0.000000	0.000000	0.000000	off	0.0	Aural_Data/True_Airspeed	<ramp></ramp>	Aircraft true airspeed (knots)
4	Groundspeed	9	0 float	32 0.0	Select	0.000000	0.000000	0.000000	off	0.0	Aural_Data/Groundspeed	<ramp></ramp>	Aircraft ground speed (knots)
5	Calibrated_Airspeed	13	0 float	32 0.0	Select.	0.000000	0.000000	0.000000	off	0.0	Aural_Data/Calibrated_Airspeed	<ramp></ramp>	Aircraft calibrated airspeed (knots)
6	Mach_Number	17	0 float	32 0.0	Select	0.000000	0.000000	0.000000	off	0.0	Aural_Data/Mach_Number	<ramp></ramp>	Aircraft Mach number (m)
7	Altitude_AGL	21	0 float	32 0.0	Select	0.000000	0.000000	0.000000	off	0.0	Aural_Data/Altitude_AGL	<ramp></ramp>	Aircraft altitude above ground level (ft)
8	Left_Engine_N1_RPM	25	0 float	32 0.0	Engines.LEM	1.000000	0.000000	1.000000	off	100.400001526	Aural_Data/Left_Engine_N1_RPM	<ramp></ramp>	Left engine N1 RPM
9	Left_Engine_N2_RPM	29	0 float	32 0.0	Select	0.000000	0.000000	0.000000	off	101.0	Aural_Data/Left_Engine_N2_RPM	<ramp></ramp>	Left engine N2 RPM
10	Left_Engine_Lit	33	0 uint8	0	Select	0.000000	0.000000	0.000000	off	0	Aural_Data/Left_Engine_Lit	<ramp></ramp>	Left engine ignition
11	Left_Engine_Thrust	34	0 float	32 0.0	Select.	0.000000	0.000000	0.000000	off	0.0	Aural_Data/Left_Engine_Thrust	<ramp></ramp>	Left engine thrust (lbs)
12	Left_Engine_Surge	38	0 uint8	0	Select	0.000000	0.000000	0.000000	off	0	Aural_Data/Left_Engine_Surge	<ramp></ramp>	Left engine surge bang trigger
13	Left_Engine_Thrust_Reverse	39	0 uint8	0	Select	0.000000	0.000000	0.000000	off	0	Aural_Data/Left_Engine_Thrust_Reverse	<ramp></ramp>	Thrust Reverse (deselected = 0 selected = 1)
14	Left Engine Thrust Rev Pct	40	0 float	32 0.0	basic.Multin	ŀ.0.010000	0.000000	1.000000	off	0.0	Aural Data/Left Engine Thrust Rev Pct	<ramn></ramn>	Left Engine Thrust Reverse Percentage

Figure 50: Add a host simulation model

Select **Controller** to view the socket and packet statistics. For *HostIn*, a column exists for the **Fail Threshold** time. If a packet has not arrived within the set amount of time, then ACE Studio assumes a communication failure. For *HostOut* variables, a column exists for **Rate (Hz)**; this is the rate for variables coming in.

hosts.A	ural_Cue_In (	load) - Ace Hos	tln Viewer	•	×
hosts.Aural_Cue_In					
Configure					
Interface eth0 change					
Port 10000 change					
Statistics					
Total Pkts Recd 0					
Hostin	Pkts Recd.	Pkts Thrown	Fail Threshold	Fail Co	bun
Host_Aural:AuralHostIn	0	0	0	69799	

Figure 51: Controller

Host Packet Editor includes the following settings:

- *Enable Testing*: allows operation of the **Test Mode**, **Test Values**, and other columns to be used for local testing. When deselected, these columns are disregarded.
- *Live Capture:* opens a window showing the current values for each entry in the ICD in real time if packets are being received. The display may also be captured and saved as a comma-separated value (.csv) file.
- *Endianness:* set to big or little endianness to match the endianness of the UDP data packet's source.

• *Align Offset:* correctly byte-aligns the entries in the ICD. Alignment commences at the location of the highlighted entry in the ICD.

1	<b>-</b> ا ٦	 -1-
		_
	12	-

Note: Align Offset does not work for packed bytes.

- *Clear Testmode:* turns off entities in Test Mode.
- *Add to Connector...*: links entries that you selected in the ICD to a specified connector.
- Use Init Value:
  - *Never*: if the UDP data stops, **HostIn** component outputs equal zero.
  - *Load:* when the project loads, **HostIn** outputs **Init Values** until it receives UDP packets and needs to use **Init Values**.
  - *Load/Source Fail*: when this setting does not receive UDP packets for a period exceeding the value in the Timeout (s), HostIn outputs Init Value until Load/Source Fail receives UDP packets again.
- *Timeout (s)*: sets the threshold for assumed communication failure when a packet has not arrived within the set amount of time in seconds.

For *HostIn*, the host I/O packet defines values for information entering the port:

- *Name:* defines the *HostIn* variable name.
- *Offset:* sets the offset location in the Ethernet packet in bytes for the variable's data.
- *Type:* sets the data type for the variable.
- *Init. Value:* sets the initial value for the variable. Options include **never**, **load**, and **load/source fail**.
- *Function*: adds a **MathFunction** to apply to the variable.
- *Scaler*: applies a scale factor to the **MathFunction** output as a gain entry.
- *Test Mode:* toggles between using the host value or the value set in **Test Value**.
- *Test Value*: sets the value used for overriding the host value when in **Test Mode**.
- *Used by:* sets the variable's destination (e.g., a connector).
- *Other:* uses a linear ramp as a test value that cycles between minimum and maximum; toggles Boolean values on and off at a set rate.
- *Description*: adds variable details (i.e., maximum and minimum value range).

For *HostOut*, the host I/O packet defines the values for the information going out on the port. The I/O packet values include the following:

- *Name:* defines the *HostOut* name.
- *Offset:* sets the offset location in the Ethernet packet in bytes for the data associated with the variable.
- *Type:* sets the data type for the variable.
- Used by: sets where the variable is coming from (e.g., a connector).
- *Description:* adds details about the variable.

### 7.3 Host interface exercise

This exercise assumes you are familiar with the ACE modeling environment. Figure 52, "Host interface" below shows how to set up the host interface:



Figure 52: Host interface

# 8.0 Remote Management System

The Remote Management System (RMS) is a specialized web server that provides complete sight and control of all ASTi devices on the simulation network, ranging from standalone to multisite, exercise-wide network configurations. The RMS provides an easy-to-navigate, user-friendly interface. Using a standard web browser from anywhere on the network, you can accomplish the following tasks:

- View system status and health
- Edit network configurations and upload licenses
- Perform project management
- Set up audio distribution devices and hosts

This chapter discusses the following topics:

- Getting started
- Add a user account
- System
- Licensing
- Audio devices in the RMS

For additional information, go to the *Remote Management System User Guide* (<u>support.asti-usa.com/media/pdf/t4/rms4\_ug.pdf</u>).

# 8.1 Getting started

To log into the Remote Management System (RMS), follow these steps:

- 1. Open a web browser on a computer sharing a network with the Telestra server.
- 2. To access the RMS, in the address bar, enter the Telestra server's IP address.
- 3. In the top-right corner, select Login.
- 4. Log into the system using the following default credentials:

Username	Password
admin	astirules

To set up the Target's network, go to the *Telestra Quick Start Guide* (<u>support.asti-usa.-</u> <u>com/media/pdf/t4/ace\_t4\_qsg.pdf</u>).

### 8.2 Add a user account

Before adding a new user account, log in using the default username and password:

Username	Password
admin	astirules

Figure 53, "RMS login screen" below shows the RMS login screen:



Figure 53: RMS login screen

After logging into the RMS, select **Manage Users** in the top right. On **RMS User Management**, add or edit new user accounts with administrator privileges.

Current System: eist	1 2 3 167 168 0.11 05 3 15 168 0.11 05 3 15 15 4 5 6 055		ASTi Remot	te Management System
System Status Health Logi Reset, Power Configuration Metworko Metworko Backup Restore Description Terrian Terrian Terrian Terrian Projects Project Management Network Targets ACENet HLA DIS Audio Upload Sound Files Spectral Analysis Archive Recordings Play Sound Files Licenses	ejst4-rh7 System Status CPU Load 7 Non-realtime 14.9% Non-realtime 1.0% Non-realtime 0.0% Realtime 31.0% 43.0% Memory Used 34% Swap Used 0%	System Info OS Version: ACE Version: ACE Build: ACE Build: ACE Build Date: ACE Build Date: ACE Security Version: Current Layout: Default Project: Default Layout: eth0: eth1: eth2: eth3: Credits:	RedHatEnterpriseWorkstation 7.6 v7.2.0 4CSe1adb 2019/04/03 06:25pm EST none SIMC1_05_CltMustang main [Running] none none [Changg]   [Remove] 10.2.135.70 172.31.99.111 20.1.1.1 100000 [33250 used] [report]	Contact Settings
	ASTi ·	support@asti-usa.com	• <u>www.asti-usa.com</u>	18:02:46 up 1 day

Figure 54: RMS User Management

## 8.3 System

This section discusses the following topics:

- Status
- Health
- Logs

### 8.3.1 Status

Go to **System** > **Status** to view the system and installation information. Select **Contact Settings** to enter installation and contact information for the system. This page also displays the Target software version:

telestra-de-17-56	5 Sy	ster	n Status		
CPU Load 🛛			System Info		Contact Settings
	Avg.	Peak	OS Version:	RedHatEnterpriseWorkstation 7.4	
Non-realtime	8.1%		ACE Version:	v7.1.0	
Non-realtime	0.0%		ACE Build:	8332563b	
Non-realtime	0.0%		ACE Build Date:	2018/10/03 11:02am EST	
Bealtime	0.0%	1.0%	ACE Security Version:	none	
			Current Project:	none	
Memory Used			Current Layout:	none [Stopped]	
6%			Default Project:	none	
Swan Used			Default Layout:	none [ <u>Change</u> ]   [ <u>Remove</u> ]	
Swap Used			eth0:	10.2.100.245	
0%			eth1:	172.31.23.87	
			eth2:	20.1.1.1	
			eth3:		
			Credits:	[0 used] [ <u>report</u> ]	

Figure 55: System Status

### 8.3.2 Health

**System** > **Health** verifies that the software is running properly. **ACE System Health** allows system debugging by providing low-level, raw information. Most of this information provides ASTi with informative, accurate debugging details. The health system is made of a tree-like structure. Each section includes subsections, and those subsections include additional subsections.

telestra-de-17-56 ACE Sy	stem Health
Тор	[1279 registered]
	Show Detail Stop Refresh
Name	
? Overall	
? ASTi Realtime	
✓ HostInterface	
✓ <u>Licensing</u>	
✓ <u>Platform Info</u>	
🖌 Project	
V Publisher	
✓ <u>RT-XPoint</u>	
Radio Networking	
✓ <u>Sound</u>	
Speech Recognition	
🖌 Text To Speech	
✓ <u>XPoint</u>	
🗸 AshlyAmplifiers	
🖌 Configuration Daemon	

Figure 56: ACE System Health

### 8.3.3 Logs

**System** > **Log** displays 100 of the most recent log entries. Download the log files to the local system to view the log details. Filter capabilities provide quick search capabilities for debug, information, warnings, and errors.

telestra-de-17-56 System Log							
Color Key: Debug Info Warning Error Critical							
Displaying 100 most recent entries.	Filter Settings						
<pre>Oct 3 18:10:01 systemd Starting Session 25 of user root. Oct 3 18:10:01 systemd Started Session 25 of user root. Oct 3 18:01:01 systemd Starting Session 24 of user root. Oct 3 18:01:01 systemd Started Session 24 of user root. Oct 3 18:00:03 systemd Started Session 24 of user root. Oct 3 18:00:45 server.rms IDError while sending response ignored: [Errno 32] Broken pipe Oct 3 18:00:45 server.rms LOGIN: User admin logged in from 10.2.100.100. Oct 3 18:00:01 systemd Starting Session 23 of user root. Oct 3 18:00:01 systemd Started Session 23 of user root. Oct 3 17:50:18 systemd Started Session 22 of user root. Oct 3 17:50:01 systemd Started Session 22 of user root. Oct 3 17:50:01 systemd Started Session 22 of user root. Oct 3 17:50:01 systemd Started Session 22 of user root. Oct 3 17:50:01 systemd Started Session 22 of user root. Oct 3 17:50:01 systemd Started Session 22 of user root. Oct 3 17:50:01 system Started Session 22 of user root.</pre>	Group: Level: model radio system Start: Oct V 3 V 0:00 V End: Oct V 3 V 23:00 V List oldest entries first? Show all entries? Filter Log File View						
Oct 3 17:46:38 server.rms IOError while sending response ignored: [Errno 32] Broken pipe							
Oct 3 17:40:01 systemd Starting Session 21 of user root.	Upload Log File						
Oct 3 1/:40:01 systemd Started Session 21 of user root.	Select log file to upload:						
Oct 3 17:35:42 Kernel nrtimer: interrupt took 3576 ns	Choose File No file chosen						
Oct 3 17:30:01 systemd Started Session 20 of user root.	Upload Log File						
Oct 3 17:20:01 systemd Starting Session 19 of user root.	opious cog r ito						
Oct 3 17:20:01 systemd Started Session 19 of user root.							
Oct 3 17:10:01 systemd Starting Session 18 of user root.							

Figure 57: System Log

# 8.4 Licensing

ASTi's licenses are tied to USB License Keys, which are DoD-approved devices covered under ASTi's Authority to Operate (ATO) and Risk Management Framework (RMF) accreditation. Upon delivery, each USB License Key activates a predefined set of software functionality for any system running Telestra software. This includes ASTi-provided hardware, customer-furnished equipment (CFE), government-furnished equipment (GFE), and virtual machines (VMs). USB License Keys also give you the ability to transfer functionality among systems and receive loaner and trial licenses. To learn more about USB License Key benefits and FAQs, go to <u>USB License Keys and your ASTi System (#123)</u>.

This section discusses how to:

- View licensing information
- Install a USB License Key
- Update a USB License Key

### 8.4.1 View licensing information

On Licensing Management, under Available Options, view the total credits and options available to your system:

Available Options		
Credits	100000	
Data Link Interface	<b>i</b> 1	
HLA	<b>11</b>	
IA	<b>11</b>	
Level D Spectral Analysis	<b>i</b> 1	
Propagation Loss Interface	68 - C	
Speech Recognition	68 - C	2 streams
Telestra Runtime	<b>i</b> 1	
Terrain Database Server	<b>i</b> 1	
Text-to-Speech	68 - C	2 streams
Vibration Analysis	<b>ii</b>	

Figure 58: License Available Options

Licenses shows a list of installed USB License Keys:

- License ID
- License type (i.e., hardware or software)
- Model
- Version
- The date the license was generated
- Error message, if applicable

Green licenses are active with no errors, while yellow licenses will expire in the next 90 days or sooner. Red licenses have already expired. If a system is unlicensed, a warning displays at the top of the page. Minimum functionality may still be available on unlicensed systems. However, if you would like to use the application's full feature set, contact ASTi to update your USB License Key.

Figure 59, "Installed licenses" below shows an example of installed licenses:

Licenses											
ID	Туре	Model	Version	Date	Error	Details					
88275080	Hardware	Max	4.27	09-Oct-2018		Show	Download				
102768155	Hardware	Max	4.27	29-Sep-2018		Show	Download				
1830087180	Hardware	Max	4.27				Download				

Figure 59: Installed licenses
Under **Details**, select **Show** to view the license's individual credits, options, and expiration dates:

L	icenses.									
I	D	Туре	Мо	del	Version	Date		Error	Details	
	88275080	Hardware	M	ax	4.27	09-Oct-	2018		Hide	Download
	Name			Qty.	.	Expired?	Exp. [	Date		
	Credits			250	0000	No				
	Data Link Interface					No				
	HLA			1		No				
	IA			1		No				
	Level D Spec	ctral Analysis		1		No				
	Propagation	Loss Interfac	e	1		No				
	Speech Reco	gnition		4 s	treams	No				
	Telestra Runtime			1		No				
	Terrain Database Server Text-to-Speech					No				
					treams	No				
	Vibration An	alysis		1		No				

Figure 60: Individual license details

### 8.4.2 Install a USB License Key

When you first receive one or more ASTi USB License Key(s) with a shipment, you must insert your licenses into the applicable Telestra server(s).

To install your USB License Key(s), follow these steps:

- 1. Insert the USB License Key(s) into the applicable servers.
- 2. Open a web browser on a computer sharing a network with the Telestra server.
- 3. To access the RMS, in the address bar, enter the Telestra server's IP address.
- 4. In the top-right corner, select Login.
- 5. Log into the system using the following default credentials:

Username	Password
admin	astirules

6. On the left, go to Licenses.

System Status	telestra	-dc-a5-	a6 L	icens	e Manage	ement						
Health	Dage does	not auto-refr	och Mor	e info hel	200							
Logs	• rage does	not auto ren-	con. <u>Pior</u>	e into ber								
Reset / Power	wer											
Configuration	Software I	Maintenan	ICE 7									
Networking												
Network Devices												
Backup Restore	The softw	ware and /	or TA	mainter	ance for thi	e eveten	n is a	ctive u	til April 1	and the second second		
Description												
Terrain												
Text-to-speech	Available	Ontions										
Projects	Artanabie .	options										
Project Management	Credits		15000									
Targoto	Data Link Inte	erface	1									
ACENet	HLA		1									
HLA	IA		ii -									
DIS	Level D Spect	ral Analysis	61 - C									
Audio	Propagation Loss Interface											
Upload Sound Files	Speech Recoo	nition	61	4 streams								
Archive Recordings	Telestra Runti	ime	11									
Play Sound Files	Terrain Datab	ase Server	61									
Licensing	Text-to-Speed	ch	61	4 streams								
Licenses	Vibration Ana	lveie										
	VIDIATION ANA	19313	- C									
	Licenses											
	ID	Туре	Model	Version	Date	Revision	Error	Details				
	102768155	Hardware	Time	4.51	14-Dec-2018	13		Show	Download			

Figure 61: Licenses navigation

7. Under Licenses, find your license's ID to confirm that it is active. This ID is printed on the tag attached to your USB License Key.

	Licenses	icenses											
	(D	Туре	Model	Version	Date	Error	Details						
•	102768155	Hardware	Time	4.51	15-0ct-		Show	Download					

Figure 62: Active license ID

### 8.4.3 Update a USB License Key

You may need to update ASTi USB License Key(s) to expand or alter system functionality (e.g., receive a software trial, add more clients). ASTi provides a License Key Update (.lku) file that you can upload to the Telestra server. Updating a USB License Key or changing its license terms does not require returning it to ASTi. A single file can also update multiple keys.

To update USB License Key(s), follow these steps:

- 1. Insert the USB License Key(s) into a Telestra server.
- 2. Open a web browser on a computer sharing a network with the Telestra server.
- 3. To access the RMS, in the address bar, enter the Telestra server's IP address.
- 4. In the top-right corner, select Login.
- 5. Log into the system using the following default credentials:

Username	Password
admin	astirules

6. On Licensing Management, under Update/Install a License, select Choose File, and find the License Key Update (.lku) file on your local system.

Update/Install a License
Apply changes by uploading a License Key Update (.lku) or License File (.lf) here.
Choose File No file chosen
Update / Install

Figure 63: Update/Install a License

The new license displays under Licenses:

Licenses							
ID	Туре	Model	Version	Date	Error	Details	
 102768155	Hardware	Time	4.51	15-0ct-		Show	Download

Figure 64: Active license

# 8.5 Audio devices in the RMS

**Network** > **ACENet** displays all the devices available on the ACENet network. Select the device to rename it, or set the input and output gains for each device channel.

Г	- - - -  <b>-</b>  7
Т	=
Т	=

*Note*: You can only set or change the gains if the device is part of the model currently running on the system.

SLi Current System: pshr System	hel7	hrhel7 Devi	s s or	r F			906	¢////		Legit			
Health Logs	Up Up	ACENet Lost Beat Packets Count: 0											
Reset / Power Configuration	Res	Reset ACENet Counters											
Network Devices	0	Device Name	Layout	Network	Firmware	Device Number	Latency Mode	Status	мас	Message			
Description	8 ACU2(s)												
SR & TTS Terrain		ACU01	- 11		2.17	4051	Normal	1	00:1a:18:00:0f:d3	ок			
Projects		ACU02	O	ii 👘	2.17	3846	Normal	<ul> <li>Image: A second s</li></ul>	00:1a:18:00:0f:06	ок			
Project Management		ACU03	O	il 👘	2.17	3849	Normal	×	00:1a:18:00:0f:09	ОК			
Targets		ACU04	O	il 👘	2.17	3863	Normal	<ul> <li>Image: A second s</li></ul>	00:1a:18:00:0f:17	ок			
ACENet		ACU07	O	- 11	2.17	3843	Normal	<ul> <li>Image: A set of the set of the</li></ul>	00:1a:18:00:0f:03	ок			
HLA		ACU08	0	il 👘	2.17	3856	Normal	<ul> <li>Image: A set of the set of the</li></ul>	00:1a:18:00:0f:10	ок			
DIS		ACU09	O	il 👘	2.17	3839	Normal	×	00:1a:18:00:0e:ff	ОК			
Lipload Sound Files		ACU10	O	il 👘	2.17	3861	Normal	<ul> <li>Image: A second s</li></ul>	00:1a:18:00:0f:15	ок			
Spectral Analysis Archive Recordings Play Sound Files Licensing Licenses	* Sets master clock for audio timing across the system. Used for debug purposes.												
	T II	This information is accurate as of 2018-10-05 at 15:26:03. This page will not automatically refresh; doing so may temporarily increase the load on system CPU resources by as much as 10-15%. This has the potential to adversely affect audio quality. If errors or unexpected conditions are displayed on this page, first try to update it by clicking your browser's reload button.											
										15:26:03 up 1			

Figure 65: Device List

The Remote Management System provides a wide variety of functions beyond the scope of this document. For additional information, go to the *Remote Management System User Guide* (support.asti-usa.com/media/pdf/t4/rms4\_ug.pdf).

# 9.0 ACE Studio model building

This section discusses ACE Studio model building from simple sine waves to creating radio models. The hands-on exercises build on each other because it assumes a certain level of knowledge as you work your way through the exercises.

This document does not take the place of an ASTi training course. ASTi recommends a threeday training course, which includes intensive hardware and software familiarization and model building assistance oriented to your application.

This chapter discusses the following topics:

- Sine wave
- Mixer
- Vox folder organization
- Math Plan
- Playsound
- Intercoms
- Radios
- Add a communication model with helpers

### 9.1 Sine wave

Figure 66, "Sine wave exercise" below shows an example of a sine wave exercise:



Figure 66: Sine wave exercise

This section discusses how to do the following:

- Build a model
- Connect an ACU2 to the model

### 9.1.1 Build a model

To build a model, follow these steps:

- 1. From the desktop, open ACE Studio.
- 2. To open a project, select **Project**. A screen opens, showing all the Targets on the net-work.
- 3. To view the projects located on each Target, expand each Target in the list.
- 4. Choose an existing project, or create a new one. To create a new project, choose a Target, and select the plus sign ().
- 5. Enter a name for the new project.
- 6. To view the layout, under **Projects**, select **main**. Main is the default layout icon view of the project. You can also add a new layout, starting with a blank canvas. Add each item one by one.
- 7. To install the contents of the project on the designated Target, select **Install Layout**. The project continues to run as you build models.
- 8. In the layout, right-click the server icon ( $\square$ ).
- 9. To open **Model Viewer**, double-click the server icon (
- 10. Right-click the Load canvas, and select Add.
- 11. To create the model canvas, select Sim Model, and name it Audio Out Example.



*Note*: *When naming files, use an underscore* (\_) *instead of a space.* 

- 12. To open the model, double-click the Audio Out Example model icon.
- 13. To add an item, right-click the canvas. Expand the Audio tab, and select Wave.
- 14. Name the component **Sine\_Wave1**.

15. Select Add.

Add to Model: 'Operators' 🗕 🗖 🗙											
Items	Items Palette										
	RecordReplay										
RecordReplayPath											
	Sequencer										
	SimpleMixer										
	VolumeControl										
	Vox					11					
	Wave										
Auc	diolO										
Cor	nmPanel										
Destinat	ion Folde	r:	Operators:/		<u>cha</u>	ange					
ltem Na	me : Sir	e_Wav	e_1								
			Cancel		Add						

Figure 67: Add sine wave

- 16. To open ACE Data Viewer, double-click Sine\_Wave1.
- 17. Set Frequency to 440.
- 18. Set Gain to 0.20.
- 19. Expand **OutSignal**, and select **view scope**. This action creates a sine wave generator with a frequency of 440 Hz and an amplitude of 0.2.

	1111	Operat	ors:Sine_W	ave_1 (Root) -	Ace	Data Viewe	r .	×	/Sine_Wave_1 - Signal(Value).audio _	o x
Sine_	Wave_	1 [Audio/Wave]	0						File Common	
Data	Links	Schematic Info	View/Edit [	Description					Period: 25 (msec) Trigger Threshold: 0.0 • ascending descendir	g Freeze
				•	🖉 [F	ilter View	Full Vie	w	Signal 1/3 Octave FFT	
	From	Variable	Туре	Value	Ор	Modifier	Result	То		
•		FreqModSignal	audio						V Offset:	
		Frequency	float32	1.0	*	440.0	440.0			
		Gain	float32	1.0	*	0.200000	0.200000			$\land \land$
		FreqModDepth	float32	1.0	*	0.0	0.0			$\Lambda = \Lambda$
		MarkSpaceInput	float32	1.0	*	0.5	0.5			
		MarkSpaceUnits	width_units	DUTY_CYCLE						
		WaveType	waveshape	sine						$ \setminus $
-		OutSignal	audio							V
		active	boolean	TRUE			TRUE			
		audio	audio	View Scope			View Scope			

Figure 68: Sine wave

#### 9.1.2 Connect an ACU2 to the model

To connect an ACU2 to the model, follow these steps:

- 1. To add an item, right-click Model on the canvas.
- 2. Under IOInterfaces, select ACU2channel, and name it ACU2\_1.
- 3. Select Add.
- 4. Double-click ACU2\_1 to open ACE Data Viewer.
- 5. In **Identifier**, under **Value**, double-click and enter the ACU2's name. If an ACU2 is connected to ACENet, select it from the drop-down box.



*Note*: By default, each ACU2 comes with a unique name. You can change this name in the Remote Management System (RMS). To view the names of all ACU2s on the network, go to **RMS Network** > ACUNet.

- 6. In Channel, double-click None.
- 7. In the drop-down box, select Channel A.

	Add to Mo	odel: 'Operators'	-	٥	×							
Items	Palette											
- IOI	nterfaces											
ACE_RIU_SerialByteOut												
ACE_RIU_channel												
	ACU2_SerialByteOut											
	ACU2channel											
	ACUchannel				Ш							
	AmpOut											
	SerialPort_IO				Ш							
▶ Inte	ercom											
Destinat	tion Folder :	Operators:/		<u>cha</u>	ange							
ltem Na	me : ACU2_	1										
					_							
		Cancel		Add								

Figure 69: Add ACU2

Figure 70, "Channel A ACU2" below shows acu2\_channel A:

					Operato	ors:ACU2_	1 (Ro	ot) - Ace D	ata Viev	ver		-	×
AC	U2_:	<b>1</b> [IC	Interfaces/AC	U2cha	annel]	ð							
Da	ta I	Links	Schematic	Info	View/Edit	Descriptio	n						
								9	🥖 🛛 Filte	er View	Full View		
	Fro	m	Variable	Ту	pe	Value	Ор	Modifier	Result	То			
			ldentifier	dev	/ice_id	ACU2_1							
			Channel	acu	12_channel	А							J
			Volume	floa	at32	1.0	*	1.0	1.0				
		,	AudioOutGain	floa	at32	1.0	*	1.0	1.0				
►		,	AudioOuts	auc	lio[32]								
۲			AudioOut	auc	lio								
		,	AudioOutGainL	. floa	at32	1.0	*	1.0	1.0				

Figure 70: Channel A ACU2

- 8. To route the generated audio out, link Sine\_Wave1 to ACU2\_1.
- 9. To connect Sine\_Wave1 to ACU2\_1, select Sine\_Wave1, which opens ACE MODEL LINK EDITOR.
- 10. To link the signal, select **Sine\_Wave1** with the *OutSignal* to ACU2\_1 with the *Audi-oOuts* signal, as shown in Figure 71, "ACE MODEL LINK EDITOR" below. The signal options appear after selecting each component.

ACE MODEL LINK EDITOR _										
Operators Swap										
Component 🛖 😑	Signal 🌵 🚥		Signal 🍁 📟		Component 🛖 😑					
<ul> <li>Operators</li> </ul>	르 OutSignal audio		AudioInGain	float32	<ul> <li>Operators</li> </ul>					
ACU2_1		+->>	AudioOutGain	float32	ACU2_1					
Sine_Wave_1			AudioOutGainL	float32	Sine_Wave_1					
Operator_1			AudioOutGainR	float32	Operator_1					
Operator_2		Added	📁 AudioOuts	audio	Operator_2					
			AudioOutsLeft	audio						
Links										
Sine_Wave_1 OutSignal ->	ACU2_1 AudioOuts									

Figure 71: ACE MODEL LINK EDITOR

- 11. To create the link, select 4
- 12. Apply the changes.
- 13. Connect a headset to Channel A of the ACU2, and listen to the sine wave.

## 9.2 Mixer

**Mixer** mixes up to eight signals into a single, composite signal. The **Mixer** variables determine which of the eight signals should mix with individual and overall gain control. A ninth signal always mixes into the output signal and cascades multiple **Mixers**.



Figure 72: Mixer project

To add a **Mixer**, follow these steps:

- 1. Add a new project, and install it.
- 2. In the load, add a simulation model, and name it.
- 3. Open the new model.
- 4. Add two Wave components, and name them Sine\_Wave and Square\_Wave.
- 5. Add a Mixer and ACU2 Channel component.
- 6. Name the components **Mixer** and **ACU2channel\_A**.
- 7. Open **Sine\_Wave**, and set the frequency (e.g., 400).
- 8. Route Sine\_Wave OutSignal to Mixer signal1.
- 9. Open **Square\_Wave**, and set the frequency (e.g., 300). Route the **Square\_Wave** out signal to the **Mixer** input **signal2**.
- 10. Open the ACU2 object, and select an ACU2 and channel A.

11. Route the Mixer *OutSignal* to the ACU2 channel *AudioOut*.



Figure 73: Mixer links

- 12. Apply the changes.
- 13. Listen to the mixed sound waves from the output device connected to the proper ACU2 channel.

				(	Operat	tors:Audio_Mixe	er (Ro	ot) - Ac	e Data Vie	ewer			-	×
Audio	udio_Mixer [Audio/Mixer]													
Data	Links	Schema	atic	Info	View/	/Edit Description	ı							
							Out							
Source variable Destination						Destination va	ariable							
OutSignal <u>/ACU2_1</u>					<u>AudioOuts</u>									
							In							
Sour	rce		Sou	irce v	ariable	e Destinatio	on var	iable						
<u>/S</u>	/Sine_Wave_1 OutSignal					Signal1								
<u>/S</u>	quare_	Wave_1	Outs	Signa	L	Signal2								

Figure 74: Mixer component links

# 9.3 Vox folder organization

**Vox** allows voice-activated or press-to-talk (PTT) control over an audio input signal. The model also demonstrates using model folders for organization.



*Note:* First connect a speaker to the ACU2 on Channel A and connect a headset with the microphone and PTT to the ACU2 on Channel B.

To use the **Vox** component, follow these steps:

- 1. Add a new project, and install it.
- 2. Open Model Viewer, add a new simulation model, and name it.
- 3. Open the model, and add two folders. Name the folders ACU2\_Folder and Audio\_ Folder.

- 4. Open ACU2\_Folder, add two ACU2 Channels, and name them Channel\_A and Channel\_B.
- 5. Open Audio\_Folder, and add Vox, Mixer, and Wave.
- 6. Open **Wave**, and set *Frequency* and *Gain*.
- 7. Using the link editor, route Wave AudioOut to Mixer.
- 8. Open Vox, and route AudioOut to Mixer.
- 9. Open Mixer, and route AudioOut to ACU Channel B.
- 10. Open **Channel\_A**, and route the audio in to **Vox**. In **Model View**, the folders show connections between them:

System Load	
▼ 🖧 load	
▼ S Folders_Model	Folders_Model
ACU2_Folder	
🔄 Audio_Folder	
Views : /Folders_Model/	/ / / /
1	
	→ Ē
Audio_Fold	ACU2_Folde
er	r

Figure 75: Folders\_Model

Opening Audio\_Folder shows the objects in other folders that are connected to the Audio\_Folder objects.



Figure 76: Folders\_Model links

When opened, ACU2\_Folder shows the objects in other folders that are connected to ACU2\_Folder:



Figure 77: ACU2\_Folder

To add PTT capability to the headset and microphone using **Vox**, add a link from **Channel A** *PTT* output to the **Vox** *PTT* input.

# 9.4 Math Plan

The **Math Plan** in ACE Studio provides access to various mathematical functions, which may be applied to the layout. The functions permit local manipulation of data within the models. The **Math Plan** objects include the following:

- Add
- Multiply
- Logical-AND
- Logical-XOR
- TableDB
- Scale and Limit
- Random Number
- Max Min
- Polynomial
- Antilog

Figure 78, "Mathplan Editor" below shows Mathplan Editor:

	Mathplan	Editor - mathp	lan	-		×					
Add Group Add Function	Add Group Add Function Clone Function Delete										
Function Groups Functions											
Group	Handle	Function Type	Description								
basic	Add	ADD	None	F(x,y) = x +	У						
Panel	Subtract	SUBTRACT	None	F(x,y) = x - y	/						
Radio	Multiply	MULTIPLY	None	F(x,y) = x *	ý						
	Linear	LINEAR	None	$F(x,y,z) = x^{2}$	* y + z	4					
	Logical-and	LOGICAL-AND	None	F(x,y) = x &	У						
	Logical-or	LOGICAL-OR	None	$F(x,y)=x\mid$	У						
	Logical-xor	LOGICAL-XOR	None	F(x, y) = x	У						
			Cancel	Apply	OK						

Figure 78: Mathplan Editor

- SubtractDivide
- Logical-OR
- Table
- TableXY
- Lag Filter
- Comparator
- Switch
- Log

# 9.5 Playsound

Figure 79, "Playsound process" below demonstrates the use of a **Playsound** and the sound library by creating a simple **Playsound** model. **Playsound** plays digitally encoded sound files. Sounds that have no dynamically varying elements except for overall volume level are best handled as fixed, offline, recorded sound files (e.g., missile launch).



Tigure 77. Timysouniu proc

This section discusses how to do the following:

- Upload Sound Files
- Add a sound library

.

### 9.5.1 Upload Sound Files

**Upload Sound Files** provides a two-step process to uploading sound files on the system. You must first select a waveset or create a new one. The **waveset** folder contains the sound files in the sound repository. Upload the selected sound files to **waveset**.



*Important*: All sound files must be in the following format: 16-bit, PCM, MONO, WAV files with a 48 kHz sample rate. The RMS yields you an error if the file is not in this format.

telestra-de-17-56 Unload Sound Files
telestra-de-17-50 opioad Sound Thes
Use this form to upload individual sound files (in <b>.wav</b> format), or to upload one <b>.tgz</b> archive containing multiple sound files.
Important! All sound files must have the following attributes:
<ul> <li>.wav format</li> <li>48 kHz sample rate</li> <li>16-bit PCM</li> <li>mono audio (1 channel)</li> </ul>
At this time, only sound files with the above traits can be used by the system. RMS will inspect the uploaded file(s) after the transfer, and the subsequent page will display valid files, as well as details for any sound file deemed invalid.
1. Where do the files belong?
Select an existing Waveset: WST_Comms ▼
OF: Specify the name of a new Waveset: Creating a new Waveset will ignore the selected Waveset above (if any). Valid characters are a-z, A-Z, 0-9, underscores(_) and dashes (-); spaces will be automatically converted to underscores.
2. Select file to upload:
Choose File propeller_plengine.wav
If you are uploading a new WAV file with the same name as a file inside the selected Waveset, the new file will overwrite the existing one.
Upload File

Figure 80: Upload Sound Files

### 9.5.2 Add a sound library

By default, when a new project is created, ACE Studio automatically creates a default sound repository called **sounds**. *(Optional)* To rename the sound repository, follow these steps:

- 1. Right-click sounds on the canvas.
- 2. Select Clone and Replace.
- 3. Enter a new name for the repository.
- 4. Right-click **Telestra**, and select **edit**. Under **core**, you can view the sound repository's new name.

To upload sound files to the Target, open the Remote Management System (RMS), and go to the **Audio** section. Open **Upload Sound Files**, and follow the instructions to add individual .wav files, or upload one .tgz archive containing multiple sound files. For more information about uploading sound files, go to the *Remote Management System User Guide* (support.asti-usa.com/media/pdf/t4/rms4\_ug.pdf).



*Important*: All sound files must be in the following format: 16-bit pulse-code modulation (PCM) MONO .wav files with a 48 kHz sample rate. The RMS gives you an error if the file is not in this format.

To locate sound files on the Target, log into the Target via Secure Socket Shell (SSH) protocol. From the **Project Manager**, right-click **Telestra** in the layout, and select **ssh**. In the terminal, enter the following: **cd /var/local/asti/soundfiles**/*waveset*, where *waveset* represents the waveset's name.

To add a sound library, follow these steps:

- 1. Create a new project, and install it.
- 2. In the layout, open **Telestra Editor**.
- 3. Add a folder name in **Waveset**. The **waveset** folder contains the sound files when you upload them to the Target.

		Tele	stra Editor		×			
Configu	re (* require	d fields)						
Name:	* Mordor-r	h7						
Target:	mordor-r	h7						
CORE	SIM SERVE	R SM	TESTING	OTHER				
Select	t							
🖧 Lo	ad:	load			•			
🚺 So	ound Repo:	sounds	5		•			
	Waveset	Trng_S	ounds					
Co	ommplan:	commplan 💌						
Ma Ma	athplan:	mathplan 🔹						
🐥 Do	omain:	domain 💌						
T Fil	terplan:	filterplan 💌						
Add H	lelpers							
👿 He	elper: Sele	ct		-	+ -			
chan	nels/channe	ls						
radio	s/radios							
host	s/hosts							
			×	Cancel	💻 Update			

Figure 81: Set a waveset

- 4. Select Update.
- 5. Open Model Viewer, add a new simulation model, and name it.
- 6. Go to to the model canvas, and add **Playsound**.
- 7. Add two ACU2channel components, and name them ACU2Channel\_A\_Speaker and ACU2Channel\_B\_PTT.
- 8. Save and close Model Viewer.
- 9. On a web browser, go to **RMS Audio** > **Upload Sound File**. Follow the directions on the page to upload two sound files. The files must be previously located the work-station to do this.



*Note*: All sound files must be in the following format: 16-bit PCM MONO wave files with a 48 kHz sample rate. If needed, convert the audio files to this format using Audacity, which is included on the ACE Studio virtual machine.

- 10. Go to the project, and open the sound repository by double-clicking Sounds.
- 11. Add a new sound library, and name it.
- 12. Add a new sound, and enter the name of the sound. Double-click **Path**, where it says **Select...**, and choose the sound file to associate with that name.

Select File	×							
▼								
🎢 Marker_1.wav								
∬d Marker_2,wav								
Cancel								

Figure 82: Select a playsound

13. Repeat Step 2 for the second sound file.

						Sou	nd Librar	y Editor - so	unds				-		×
Image: Second Coup     Image: Second Coup     Image: Second Coup     Image: Second Coup       Add Library     Add Group     Add Coup     Add Sound     Copy     Paste     Delete															
Sound Libra	aries		Sound File	es											
Library Nar	me Index		Name	Index	Path	Playall	Start	Stop	Num Samples	Buffer	Loop	Post-delay (sec)	Audio-level		
Training			Sound1	1	Marker_1.wa	av true	0	24937	24937	bysize	true	0	1.0		
			Sound2		Marker_2.wa	av true		36344	36344						
		1										💥 Cancel 🔍	🖉 Apply	4	ж

Figure 83: Sound Library Editor

- 14. To close the window, select Apply and OK
- 15. Save the project, and reinstall the layout.
- 16. Open **Model Viewer**, go to the **Model** canvas, and open the **Playsound** component. In the **Value** of *LibraryId*, enter the sound library's name.
- 17. Open ACU2channel\_B\_PTT, and set the ACU2 *Identifier* and *Channel*.
- 18. Open ACE MODEL LINK EDITOR, and route the ACU2channel\_B\_PTT *PTTselect* to the Playsound *SoundIndex*.

	ACE	MODEL LINK E	DITOR	_ = ×
Folders_Model Swap				
Component 🐈 💻	Signal 🛖 📟		Signal 🛖 📟	Component 🐈 🚍
<ul> <li>Folders_Model</li> </ul>	AnalogIn1 uint8		BeginOffset float32	▼ Folders_Model
ACU2_Folder	AnalogIn2 uint8		EndOffset float32	ACU2_Folder
Audio_Folder	AnalogIn3 uint8		GroupId playsound_g	rc • Audio_Folder
Maths	Audioln audio	+->>	OutGain float32	Maths
✓ Playsound_1	DigitalIn1 boolean		Pause boolean	✓ Playsound_1
ACU2channel_A_Speaker	DigitalIn2 boolean		Randomize boolean	ACU2channel_A_Speaker
ACU2channel_B_PTT	DigitalIn3 boolean	Added	🛏 SoundIndex playsound_s	ACU2channel_B_PTT
PlaySound	PTT boolean		Trigger boolean	PlaySound
	■ PTTselect uint8			
Links				
ACU2channel_B_PTT PTTselect -	> PlaySound SoundIndex			

Figure 84: PTTselect link

19. To trigger the **Playsounds**, route the **ACU2Channel\_B\_PTT** component's *PTT* output to the **Playsound** component's *Trigger* input.

	ACE M	10DEL LINK E	DITOR		_ = ×
Folders_Model Swap					
Component 🖶 🚍	Signal 🍨 📟		Signal 🛖 📟		Component 🛖 🚍
<ul> <li>Folders_Model</li> </ul>	AnalogIn1 uint8		BeginOffset floa	at32	<ul> <li>Folders_Model</li> </ul>
ACU2_Folder	AnalogIn2 uint8		EndOffset floa	at32	ACU2_Folder
Audio_Folder	AnalogIn3 uint8		GroupId pla	aysound_gro	Audio_Folder
Maths	AudioIn audio	+->>	OutGain flo	at32	Maths
✓ Playsound_1	DigitalIn1 boolean		Pause boo	olean	▼ Playsound_1
ACU2channel_A_Speaker	DigitalIn2 boolean		Randomize boo	olean	ACU2channel_A_Speaker
ACU2channel_B_PTT	DigitalIn3 boolean	Added	📼 SoundIndex pla	aysound_so	ACU2channel_B_PTT
PlaySound	🛋 PTT boolean		🛏 Trigger 🛛 boo	olean	PlaySound
	➡ PTTselect uint8				
Links					
ACU2channel_B_PTT PTT ->	> PlaySound Trigger				
ACU2channel_B_PTT PTTselect ->	> PlaySound SoundIndex				

Figure 85: PTT link

20. Route the **Playsound** audio output to the ACU2 audio output to play through the speaker.

		ACE	MODEL LINK E	DITOR		_ = ×
Folders_Model Swap						
Component 🛖 😑	Signal 🍨 🖷			Signal 🍁 📟		Component 🛖 💻
<ul> <li>Folders_Model</li> </ul>	\Rightarrow Out			AudioInGain	float32	<ul> <li>Folders_Model</li> </ul>
ACU2_Folder	Playing	boolean		AudioOutGain	float32	ACU2_Folder
Audio_Folder				AudioOutGainL	float32	Audio_Folder
Maths			+->>	AudioOutGainR	float32	Maths
✓ Playsound_1				🛏 AudioOuts	audio	✓ Playsound_1
ACU2channel_A_Speaker				AudioOutsLeft	audio	ACU2channel_A_Speaker
ACU2channel_B_PTT			Added	AudioOutsRight	audio	ACU2channel_B_PTT
PlaySound				DigitalOut	boolean	PlaySound
				Volume	float32	
Links						
PlaySound Out -:	> ACU2chann	el_A_Speaker A	AudioOuts			

Figure 86: Audio output link

- 21. Open ACU2channel\_A\_Speaker, and set the *Identifier* and *Channel A*.
- 22. Save and exit Model Viewer.

23. In **Project Manager**, save and reload the model. Each sound should play from the speaker by switching between PTT Channels 1 and 2.



Figure 87: Playsound model

24. Connect a speaker to the ACU2 in Channel D, and connect a headset with a microphone and four-channel PTT to the ACU2 in Channel B. Use the PTT channel select to choose a **Playsound** to review.

# 9.6 Intercoms

This section demonstrates the use of Intercoms by creating a model with two operators operating on four buses using a PTT switch with the **CommPanel** component. The basic concept of the **CommPanel** is the connection of an operator input (usually a microphone, often via a **Vox** component) through a PTT gate and gain stage with an optional control input and scaling factor via a control selector switch to an intercom channel (bi-directional) provided by the **Intercom** service. The intercom channel may be connected to various other component types to provide connectivity to other audio components, or may simply be used as a basic intercom to provide standard intercom voice communications. A simple example using the intercom bus includes two operators each with a four-channel PTT switch. When both operators are on the same channel (intercom bus) they can communicate and listen to the same audio. Each operator needs their own **CommPanel** component.



Figure 88: Intercom system diagram

#### 9.6.1 Intercom exercise

To complete the Intercom exercise, follow these steps:

- 1. In ACE Studio, create a new project, and install it. In **Model Viewer**, add a simulation model to create the canvas for building the model.
- 2. In the model canvas, add a new folder, and name it **Op1**.
- 3. In the folder, add a CommPanel4 component, and name it Op\_Panel.
- 4. Add an I/OInterface > ACU2channel component, and name it Op\_ACU2channel.

5. Add a link to route the audio in from the headset microphone to the **Op\_Panel** (**Com-mPanel**), select the **Op\_ACU2channel** *AudioIn*, and link it to the **Op\_Panel** *InSignal*.

	ACI	E MODEL LINK EI	DITOR		×
Intercom_Exercise Swap					
Component 🐈 😑	Signal 🍨 📟		Signal 🛖 📟		Component 🐈 🚍
<ul> <li>Intercom_Exercise</li> </ul>	AnalogIn1 uint8		InControl	byte	<ul> <li>Intercom_Exercise</li> </ul>
▼ Op1	AnalogIn2 uint8		InGain	float32	▼ Op1
Op_ACU2channel	AnalogIn3 uint8		📼 InSignal		Op_ACU2channel
Op_Panel	🖿 AudioIn 🛛 audio		OutControl	byte	Op_Panel
	DigitalIn1 boolean		OutGain	float32	
	DigitalIn2 boolean		PTT	boolean	
	DigitalIn3 boolean	Added	PTTselect	byte	
	PTT boolean		Power	boolean	
	PTTselect uint8		SideControl	byte	
			SideGainContro	ol byte	
Links					
Op_ACU2channel AudioIn -> O	p_Panel InSignal				

Figure 89: Audio input connection

6. Add a link to route the PTT from the ACU2 to the **Op\_Panel** (**CommPanel**) PTT, select the **Op\_ACU2channel** *PTT* signal, and link it to the **Op\_Panel** *PTT* signal.

· · · · · · · · · · · · · · · · · · ·	ACE M	10DEL LINK E	DITOR		-	×
Intercom_Exercise Swap						
Component 🛖 🚍	Signal 🍨 📟		Signal 🛖 📟		Component 🐈 🚍	
<ul> <li>Intercom_Exercise</li> </ul>	AnalogIn1 uint8		InControl	byte	<ul> <li>Intercom_Exercise</li> </ul>	
▼ Op1	AnalogIn2 uint8		InGain	float32	▼ Op1	
Op_ACU2channel	AnalogIn3 uint8		📁 InSignal	audio	Op_ACU2channel	
Op_Panel	🕩 Audioln 🛛 audio	+->	OutControl	byte	Op_Panel	
	DigitalIn1 boolean		OutGain	float32		
	DigitalIn2 boolean	- 2	🛏 PTT	boolean		
	DigitalIn3 boolean	Added	PTTselect	byte		
	➡ PTT boolean		Power	boolean		
	PTTselect uint8		SideControl	byte		
Links						
Op_ACU2channel PTT -> Op_I	Panel PTT					
Op_ACU2channel AudioIn -> Op_F	Panel InSignal					

Figure 90: PTT connection

7. Create a link to route the audio out to the headset. Select the **Op\_Panel** *OutSignal*, and link it to the **Op\_ACU2channel** *AudioOuts* signal.

	ACE MODEL LINK EDITOR _								
Intercom_Exercise Swap									
Component 🐈 🚍	Signal 🍨 📟		Signal 🍨 📟		Component 🐈 💳				
<ul> <li>Intercom_Exercise</li> </ul>	🕩 OutSignal audio		AudioInGain	float32	<ul> <li>Intercom_Exercise</li> </ul>				
▼ Op1	SideSignal audio		AudioOutGain	float32	▼ Op1				
Op_ACU2channel			AudioOutGainL	float32	Op_ACU2channel				
Op_Panel		+->>	AudioOutGainR	float32	Op_Panel				
			🛏 AudioOuts	audio					
		= ->>	AudioOutsLeft	audio					
		Added	AudioOutsRight	audio					
			DigitalOut	boolean					
			Volume	float32					
Links									
Op_Panel OutSignal -> Op	_ACU2channel AudioOuts								

Figure 91: OutSignal connection

8. Add a link to route the sidetone of the headset back into the headset. This allows you to hear yourself through the headset. Select the **Op\_Panel** *SideSignal*, and link it to the **Op\_ACU2channel** *AudioOuts*.

	ACEN	10DEL LINK E	DITOR	_ = ×
Intercom_Exercise Swap				
Component 🐈 🚍	Signal 🌪 📟		Signal 🌪 📟	Component 🐈 🚍
<ul> <li>Intercom_Exercise</li> </ul>	💷 OutSignal audio		AudioInGain float32	<ul> <li>Intercom_Exercise</li> </ul>
▼ Op1	➡ SideSignal audio		AudioOutGain float32	✓ Op1
Op_ACU2channel			AudioOutGainL float32	Op_ACU2channel
Op_Panel		+->>	AudioOutGainR float32	Op_Panel
			🛏 AudioOuts 🛛 audio	
		- 2	AudioOutsLeft audio	
		Added	AudioOutsRight audio	
			DigitalOut boolean	
			Volume float32	
Links				
Op_Panel SideSignal -> C	p_ACU2channel AudioOuts			
Op_Panel OutSignal -> O	p_ACU2channel AudioOuts			
				7

Figure 92: Side signal connection

9. Add three links for the PTT knob selection control. Select the **Op\_ACU2channel** PTT-select, and link it to the **Op\_Panel** *OutControl*, *InControl*, and *SideControl*.

		ACE	MODEL LINK E	DITOR		-	×
Intercom_Exercise Swap							
Component 🖶 💻	Signal 🍨 📟			Signal 🌸 📟		Component 🛖 😑	
<ul> <li>Intercom_Exercise</li> </ul>	AnalogIn1	uint8	1	🝋 InControl	byte	<ul> <li>Intercom_Exercise</li> </ul>	
✓ Op1	AnalogIn2	uint8		InGain	float32	▼ Op1	
Op_ACU2channel	AnalogIn3	uint8		📁 InSignal	audio	Op_ACU2channel	
Op_Panel	르 Audioln	audio		🝋 OutControl	byte	Op_Panel	
	DigitalIn1	boolean	∞≥ -	OutGain	float32		
	DigitalIn2	boolean		PTT	boolean		
	DigitalIn3	boolean		PTTselect	byte		
	PTT	boolean	1	Power	boolean		
Links							
Op_ACU2channel PTTselect -> O	p_Panel	SideControl					
Op_ACU2channel PTTselect -> O	p_Panel	InControl					
Op_ACU2channel PTTselect -> O	p_Panel	OutControl					
	- ·						

Figure 93: PTTselect connection

10. To open the Intercom Bus Service, in Op\_Panel, select IntercomBusService.



Figure 94: IntercomBusService link

11. In IntercomBusService, add four new buses, and name them.



Figure 95: Add intercom buses

12. Go to **Op\_Panel**, and assign **Signals 1–4** to the buses by double-clicking in **Value**. In **Intercom Bus Service**, select a bus, and select **Set Value**.

	Intercom_Exercise:Op_Panel (/Op1/) - Ace Data Viewer _ 🛛 🗙											
Op_Pa	anel [C	ommPanel/C	ommP	Panel4] 👩 Pow	/erService	<u>IntercomBus</u>	sServic	<u>e</u>				
Data	Links	Schematic	Info	View/Edit Descrip	ew/Edit Description							
						4	Filter	View	Full View			
Fre	om			Variable SidetoneLocal	Type byte	Value 255	Ор &	Modifier 0	Result 0	То		
				SideGainControl	byte	255	&	255	255			
				Sig1	id	IC_Bus1						
				Sig2	id	IC_Bus2						
				Sig3	id	IC_Bus3						
				Sig4								1
				Sig1_RxGain	float32	1.0	*	1.0	1.0			Т
				Sig2_RxGain	float32	1.0	٠	1.0	1.0			
				Sig3_RxGain	float32	1.0	*	1.0	1.0			1
				Sig4_RxGain	float32	1.0	*	1.0	1.0			

Figure 96: Operator panel intercom bus assignments

13. Open the **Op\_ACU2channel**, and assign the *Identifier* and *Channel*.

	Intercom_Exercise:Op_ACU2channel (/Op1/) - Ace Data Viewer >									×	
Op_ACU2channel [IOInterfaces/ACU2channel]											
Da	ita Lin	ks Schematic	Info	View/Edit D	escription						
							🚽 🛛 Filte	er View	Full View		
	From	Variable	Т	ype	Value	Ор	Modifier	Result	То		
		Identifier	d	evice_id	ACU2_1						
		Channel	a	:u2_channel							
		Volume	fl	oat32	1.0	*	1.0	1.0			
		AudioOutGa	in fl	oat32	1.0	*	1.0	1.0			
۲	multip	le AudioOuts	au	udio[32]							
											-

Figure 97: ACU2 channel assignment

14. Copy and paste the **Op1 folder**, and name it **Op2 folder**.



Figure 98: ACU2 channel assignment

15. In **Op2 folder**, change the ACU2 channel to a different channel that has a connected PTT and headset.

		In	terco	m_Exercise	:Op_ACU	2chai	nnel (/Op2/	) - Ace D	ata Vie	wer	-	×
Op.	ACU2cl	<b>hannel</b> [IOInte	erface	s/ACU2char	nel] 👩	]						
Dat	a Links	Schematic	Info	View/Edit [	Descriptior	ı						
							8	Filter	View	Full View		
	From	Variable	Ту	ype	Value	Ор	Modifier	Result	То			
		Identifier	de	vice_id	ACU2_1							
		Channel		u2_channel								
		Volume	flo	pat32	1.0	*	1.0	1.0				
		AudioOutGain	flo	pat32	1.0	*	1.0	1.0				
	multiple	AudioOuts	au	idio[32]								
		AudioOut	au	ıdio								

Figure 99: Op2\_ACU2 channel assignment

16. Verify the ACU2 channel gains set in the RMS. These gains will vary depending on the ACU firmware.

ACU2_1 - Channel A							
This device is running firmware version: 2.17. Acceptable values for <b>input gain</b> are: -8 or +2 through +58 Acceptable values for <b>output gain</b> are: -25 through +10							
Standard Gain S	Settings						
Mic Power Enable:	Enable						
Input Gain:	40	dB					
Output Gain: 10 dB							

Figure 100: ACU2 channel gains

17. In **Model Viewer**, select **Reload**. Apply the changes. Two users should be able to talk to each other through all four-channel selections on the PTTs when the selector switches are in matching positions.

# 9.7 Radios

**Radios** are the largest, most complex, and most used components in ACE Studio. The following is a summary list of radio features for all simulated radios:

- *World Position:* defines the X, Y, Z coordinates of the radio's location.
- *Frequency*: defines the center of the radio tune frequency for transmit and receive. Optionally, it can define separate transmit and receive frequencies.
- *Antenna Gain:* simulates the size and radiative efficiency of the antenna. All modeled antennae are isotropic.
- *Squelch*: a noise gate that only allows signals exceeding a specified strength to filter through and play.
- Background Noise: the general noise created when using radios.
- *Fill:* allows you to choose one of a set of *N* predefined radio fills, as defined in a global comm plan.
- *Multiple Net Support*: provides the radio with the ability to support multiple nets per a specific fill. Nets define the following core radio characteristics:
  - *Multiple Modulation Type:* describes the modulation parameters of the radio, such as AM, FM, SATCOM, HQ, Intercom, etc.
  - *Amplitude Modulation (AM) and Frequency Modulation (FM)*: two primary modulations for radio operation.

- *Modulation Discrimination*: occurs when radios can only receive signals from radios with the same modulation type.
- *AM Mixing:* when multiple AM signals broadcast on the same channel frequency, the AM receiver will receive all signals and mix them according to signal strength.
- *FM Capture Effect:* when several FM radios are transmitting on the same frequency, an FM receiver will only be able to receive the strongest signal.
- *Sensitivity:* receiver sensitivity in dB.
- *Bandwidth and Bandwidth Overlap*: determines the amount of audio noise mixed into the received audio, based on the simulated bandwidth of the radio band. This parameter does not affect the in-tune calculation.
- *Encoding Type and Rate:* defines the audio encoding type, including muLaw, continuously variable slope delta (CVSD), pulse-code modulation (PCM), and the sample rate.
- *Transmit Power*: indicates the transmission power of the radio in Watts/dBm
- *Automatic Gain Control (AGC)*: adjusts the gain to appropriate levels for a range of input signal levels.
- SATCOM Parameters: defines satellite mode parameters, if applicable.
- *Frequency Hopping:* a method of rapidly switching frequencies while a receiver and transmitter communicate. The receiver and transmitter have to jump between the same frequencies, at the same speed, and at the same time.
- *Crypto Parameters:* radios that scramble the signals before they are transmitted so that only receivers who know the special key will have the ability to decode them, producing a secure voice transmission across any frequency.
- *Half Duplex and Full Duplex:* half-duplex mode is when the radio is able to transmit and receive signals but cannot do both at the same time. Full-duplex mode allows radios to transmit and receive signals at the same time. Typically, full-duplex mode is only used for intercom systems but never for real radios.
- *Propagation*: the movement of the radio waves as they move away from the transmitting antenna.
  - *Ranging:* an effect that occurs as a result of the distance between two radios. The greater the distance between the radios, the weaker the signal due to the dissipating power of the signal as it traverses a large area.
  - *Occulting*: the loss of radio signal due to the curvature of the earth's horizon.
  - *Ionosphere Effects:* the loss of signal due to the changes in the earth's atmosphere such as time of day or different seasons. The ionosphere effects only occur with high-frequency (HF) radios.

- *Line of Sight (LOS):* when radio waves traveling in a straight line are dispersed due to obstacles or obstructions.
- *Fresnel Diffraction*: loss of signal due to the reflection of off obstacles in the path of the radio waves from transmitter to receiver.
- *Terrain Effects*: the loss of signal due to land obstruction, such as a mountain.

This section discusses the following topics:

- Local radios
- Comm Plan
- Add a model

### 9.7.1 Local radios

This section focuses on the radio in its simplest form: the local radio. Local radios operate between two operators on one Target. In this exercise, the following services will be used:

- IntercomBusService
- RadioControlService

IntercomBusService passes audio signals between the radio transceivers and the associated CommPanels. RadioControlService links the RCUbasic components to their associated Transceiver components. The components for the first operator and radio are located in a folder named Operator1; the second operator and radio is located in a folder named Operator2.

Each folder contains the following components and names:

Component	Name
ACU2channel	Headset_PTT
CommPanel4	Panel
RCUbasic	Control_Head
Transceiver	RX_TX
GeodeticWorldPosition	Position

#### Table 1: Local radio components and names

As in the previous exercise, create a new project. This time, call it Local\_Radios. Do not install it at this time.

### 9.7.2 Comm Plan

On **Project Manager**, there is an icon that looks like a hard hat and is named Comm Plan. Double-clicking it opens **Commplan Editor**, which has two panes:

- Left pane: a tree-type view of the major sections of the Comm Plan.
- Right pane: contains details of each of the major section.

The Comm Plan is a road map for the radios and allows or disallows the radios to operate in certain manners. Initially, the Comm Plan is created with a default plan installed. To see the default plan, in the left pane, select **Fill** twice. In the right pane, select the arrowhead next to **Fill**.

The default Comm Plan contains functions most commonly used in typical radio models. Use the default Comm Plan or modify it to suit your requirements. For this exercise, you will remove the majority of the functionality and add what you need.

In the right pane, highlight the default fill. To delete, select the minus sign ().

	Commplan Editor - commplan	- 0	×
	4 - E	Expand	I All
commplan	Group Name		
<ul> <li>▶ <sup>™</sup> <u>Net</u></li> <li>▶ <sup>™</sup> Waveform</li> </ul>			
Crypto			
<ul> <li>Treqhop</li> <li>Receivergain</li> </ul>			
▶ 🔄 <u>Satcom</u>			
Opened Commplan 'commplan'	Cancel Apply	Oł	<

Figure 101: Comm Plan fill cleared

In the left pane, select **Net**. In the right pane, select **Net**. To delete all of the default nets, at the top of the window, select the minus sign (...).

	Commplan Editor - commplan			_ 0	×
commplan Fill Vaveform Crypto Freqhop Receivergain	Group Name			Expand A	All
Opened Commplan 'commplan'		Cancel	Apply	ОК	

Figure 102: Comm Plan net cleared

In the left pane, select **Waveform**. To delete all of the default waveforms, at the top of the window, select the minus sign (...).

	Commplan Editor - commplan		- 0	×
commplan Fill Vet Vet Crypto Freqhop Receivergain Satcom	Group Name		Expand	All
Opened Commplan 'commplan'		Cancel Apply	OK	¢

Figure 103: Comm Plan waveform cleared

The Comm Plan is structured in a bottom-up format. You cannot build it using a top-down method.

In the left pane, highlight **Waveform**. To create a highlighted line that includes the name **file1**, at the top of the window, select the plus sign ( ). Enter the various modulations here. Select **file1**, and replace it with **Modulations**.

	Commplan Editor - commplan		_ = ×
commplan Fill Net Waveform	Group Name       Modulations		Expand All
<u>Modulations</u> <u>Grypto</u> <u>Freqhop</u> <u>Receivergain</u> <u>Satcom</u>			
Opened Commplan 'commplan'		Cancel Apply	ОК

Figure 104: Add modulations to waveform

In the left pane, highlight **Modulations**. Radio functionality headings now display at the top of the right pane.

At the top of the window, select the plus sign  $(\textcircled{\bullet})$  to add a new line of data.

In Name, the entry name is wavef1. Change this to a more meaningful name: INTERCOM.

			Commpla	n Editor	- commplan			-	o x
	+ <b>-</b>							[	Expand All
commplan	Name	Mode	Encoding	Rate	Bandwidth(Hz)	Tx Power(Watts)	RECEIVERGAIN	Voice	Effects
Sint Sint	INTERCOM	INTERCOM	MULAW	8000	25000	1.0	Off	OFF	
✓ <u>Net</u> ✓ <u>Waveform</u>									
Modulations									
Opened Commplan 'commplan'							Cancel App	ly	ОК

Figure 105: Add intercom to modulations

At the top of the window, select the plus sign  $(\textcircled{\bullet})$  to add a new line of data.

In Name, the entry is called wavef1. Change this to a more meaningful name: AM.



*Note*: The pane automatically saves entries in alphabetical order, so check that your new entry hasn't moved before making changes to other columns.

In Mode, select INTERCOM. When the list opens, press the Down arrow to select AM.



*Important*: Select the window's background before applying changes. This action ensures the drop-down arrow is not still in view. If it is, when you select *Apply* and close the window, ACE Studio won't save your changes.

The *Encoding* and *Rate* are typical for this type of modulation, as is the bandwidth. You can leave *TxPower* as is and enter the default value directly into **RCUbasic**, or enter the *TxPower* value here.

The radio receiver must have a gain and AGC profile. By default, it is off. Select this entry, and press the Down arrow to access the drop-down list. Select **rxgain.rxgain**, a default entry that's typical for AM and FM radios.

**Voice Effects** adds an effect that makes the radios sound more realistic. Several parameters derive the effect. Turn on **Voice Effects**.

Create another entry named **FM** for FM radios. All other columns must be the same as the **AM Mode**.

			Comm	plan Edito	r - commplan			_
	+							Expand All
commplan	Name	Mode	Encoding	Rate	Bandwidth(Hz)	Tx Power(Watts)	RECEIVERGAIN	Voice Effects
S Not	AM	AM	MULAW	8000	25000	1.0	rxgain.rxgain	ON
inet inet	FM	FM	MULAW	8000	25000	1.0	rxgain.rxgain	ON
<u>Vaverorm</u> <u>Modulations</u> <u>Crypto</u>	INTERCOM	INTERCOM	MULAW	8000	25000	1.0	Off	OFF
Opened Commplan 'commplan'							Cancel	Арріу ОК

Figure 106: Completed waveform section

Ensure that no drop-down arrows are visible.

In the left pane, highlight Net. To add a new line named file1, at the top of the window,

select the plus sign ( ). You can enter details about the net here. Select **file1**, and rename it **Nets**.

	Commplan Editor - commplan			- 0	×
	<b>+</b> = ₽			Expand	All
commptan	Group Name				
▼ <sup>™</sup> <u>Net</u>	Nets				
Nets					
▼					- 11
Modulations					
Opened Commplan 'commplan'		Cancel	Apply	OK	:

Figure 107: Add nets to net

In the left pane, select **Nets**. Along the top of the right pane, there are now headings related to some radio specifics.

At the top of the window, select the plus sign ( ) three times. Three blank nets have been created. In Name, rename net1, net2 and net3 to AM, FM, and INTERCOM.

In AM, select Off in Waveform. From the drop-down box, select Modulations.AM. This action links the values in Waveform, which are relevant to the AM modulation to the AM net. Link the FM and INTERCOM lines also.

			Commplan Editor - co	mmplan			_ = ×
	<b>+ - -</b>						Expand All
commplan	Name	Frequency(Hz)	TxFrequency(Hz)	WAVEFORM	CRYPTO	FREQHOP	SATCOM
<u>□</u> <u>Fill</u>	AM	0		Modulations.AM	Off	Off	Off
▼ <u>Net</u>	FM	0		Modulations.FM	Off	Off	Off
Nets	INTERCOM			Modulations.INTERCOM	Off	Off	Off
Modulations							
Opened Commplan 'commplan'					Ca	ncel Appl	у ок

Figure 108: Completed nets

If **Crypto**, **Frequency Hopping**, or **SATCOM** modes are required, link them in a similar manner in their respective columns. Make default value selections here for default values for the radio to function in those modes. For example, if you don't make cipher selections, and cipher modes correctly populate in **RCUbasic** and **Transceiver**, they do not function in cipher mode.

You can also set a default frequency or receive and transmit frequency. In this case, if no input is set in the **RCUbasic** for frequency, ACE Studio uses this value.



*Note*: If the input frequency to *RCUbasic* is set back to 0, *RCUbasic* uses the frequency of 0, not the default in the Comm Plan.

In the left pane, select **Fill**. At the top of the window, select the plus sign ( ) to create a highlighted line called **file1**. You can add the fills here. Rename **file1** to **Plan**.

In the left pane, select **Plan**, and select the plus sign ( ) twice. This action adds two entries to the right pane. Rename fill1 and fill2 to **RADIO** and **INTERCOM**. RCUbasic references **Plan**, **INTERCOM**, and **RADIO** as the fill.

		Commplan Editor - commplan		•	- • ×
commplan	<b>+</b> = F				Expand All
✓ <sup>(C)</sup> <u>Fill</u> Plan	Name	NE	T		
✓ <sup>™</sup> Net	RADIO				
▼ <sup>™</sup> Waveform					
Opened Commplan 'commplan'				Cancel Apply	ОК

Figure 109: Added fills

Select INTERCOM, and then select the plus sign (+).

A subentry appears with the number 1. **RCUbasic** references this number as the net number. **None** is highlighted in red because **none** is not a valid net name. Select **none**, and from the drop-down list, select **Nets.INTERCOM**. This action links the appropriate net and waveform to this net number.

Select **RADIO**, and then select the plus sign (<sup>+</sup>). Complete this action again.

Two subentries appear. None highlights red, as **none** is not a valid net name. In line 1, select **none**, and select **Nets.AM**. To link the appropriate net and waveform to this net number. For line 2, reference to **Nets.FM**.

r		Commplan Editor - commplan	_ = ×
	<b>+</b> - <b>E</b>		Expand All
commplan	÷ -		
▼ 🖾 <u>Fill</u>	Name	NET	
Plan Plan	▼ INTERCOM		
▼ 🖾 <u>Net</u>	1	Nets.INTERCOM	
Nets	▼ RADIO		
▼ <sup>™</sup> Waveform	1	Nets.AM	
Modulations	2	Nets.FM	
<ul> <li>Crypto</li> </ul>			
🕨 🔄 <u>Freqhop</u>			
Opened Commplan 'commplan'		Cancel	Apply OK

Figure 110: Completed fills

Nothing should be highlighted red. The Comm Plan is now complete. Select Apply, then select OK.

In the toolbar, select **Save Project**. When **Saving Changes** appears, make a relevant entry (e.g., "Completed Commplan"). Select OK.

Saving changes	×				
Enter commit message					
Completed Commplan					
Cancel OK					

Figure 111: Save Comm Plan

#### 9.7.3 Add a model

To add a model, follow these steps:

- 1. To install the layout, in Project Manager, select Install Layout.
- 2. Wait for installation to complete. To open Model Viewer, double-click Telestra.
- 3. Right-click the main canvas, and select Add.
- 4. In Add to Load, select SimModel.
- 5. In Item Name:, enter Radios, and select Add.
- 6. In Item Name:, enter Operators, and select Add.
- 7. Two simulation models now exist on the canvas. Close Add to Load.
- 8. You can build the operators, panels, audio inputs, and audio outputs in the **Operators Sim Model**, and you can build the radios in the **Radios Sim Model**. Open the **Operators Sim Model**.
- 9. Right-click the canvas, and select Add.
- 10. In Add to Model, add a folder, and name it Operator\_1.
- 11. Open Operator\_1.
- 12. Add the following components and names:

Components	Names
CommPanel4	Panel
ACU2channel	Operator
Filter	Mic_Filter
MathFunction	Rad_Int_Select

Table 2: Components and names

Figure 112, "Add components to Operator\_1" below shows adding a component to **Operator\_1**:



Figure 112: Add components to Operator\_1

 The setup includes two intercom buses: one for intercom communication, and one for radio communication. Open Panel, assign Sig1 to Intercom, and assign Sig2 to Radio1. Close Panel.

Γ	 OutControl	byte	255	&	255	255		
	 SideControl	byte	255	&	255	255		
	 SidetoneLocal	byte	255	&	0	0		
	 SideGainControl	byte	255	&	255	255		
	Sig1	id	Intercom					
	Sig2	id	Radio1					
	Sig3	id	UNASSIGNED					II.
	Sig4	id	UNASSIGNED					II.
	 Sig1_RxGain	float32	1.0	*	1.0	1.0		
	 Sig2_RxGain	float32	1.0	*	1.0	1.0	k	

Figure 113: Assign intercom buses

14. Open **Operator**. For **Identifier**, double-click **Select**, and select the ACU2. For the channel, choose a channel connected to the first headset and four-way PTT device.

	Operators:Operator (/Operator_1/) - Ace Data Viewer _													
Ор	erato	or [10	OInterfaces/A	CU2cł	nannel]	0								
Da	ta L	.inks	Schematic	Info	View/Edi	t Descript	tion							
										Full View				
	Fror	n \	/ariable	Ту	pe	Value	Ор	Modifier	Result To					
						ACU2								
		C	hannel	acu	12_channe	ΙA								
		~ \	/olume	floa	at32	1.0	*	1.0	1.0				L	

Figure 114: Assigning Operator 1 ACU2

- 15. Link the following components and variables:
  - a. **Operator's** AudioIn to the **Mic\_Filter's** InSignal
  - b. **Operator's** *PTT* to the **Panel's** *PTT*
  - c. **Operator's** *PTTselect* to the **Rad\_Int\_Select's** *Input\_X*
  - d. Mic\_Filter's OutSignal to the Panel's InSignal
  - e. Rad\_Int\_Select's *Result* to the Panel's *InControl*
  - f. Rad\_Int\_Select's *Result* to the Panel's *OutControl*
  - g. Rad\_Int\_Select's Result to the Panel's SideControl
  - h. Panel's OutSignal to the Operator's AudioOuts
  - i. Panel's SideSignal to the Operator's AudioOuts

16. Close ACE MODEL LINK EDITOR.



Figure 115: Operator\_1 component links

17. Open **Mic\_Filter**. By default, the filter is a low-pass filter with a roll-off frequency of 2000 Hz and a Q Factor of 0.7071. This filter often acts as a microphone filter. These default values help remove audio feedback, especially when the audio input and output are closely coupled.

					Оре	erators:Mic_	Filter (/Op	erato	or_1/) - Ace	Data Viev	wer		-	×
Mic.	Filte	r [Aud	lio/Fil	ter]	Ø									
Dat	a Link	s S	chem	atic	Info	View/Edit D	escription							
									$\checkmark$	Filter Vie	w	Full View		
	From			Var	iable	Туре	Value	Ор	Modifier	Result	То			
					ble		FALSE		TRUE	TRUE				
				Filte	erType	filter_type	2 LowPass							
.				Free	quency	float32	1.0	*	2000.0	2000.0				
				QFa	ctor	float32	1.0	*	0.707100	0.707100				
				Out	Gain	float32	1.0	*	1.0	1.0				1
	Operat	or/Au	idioln	InSi	gnal	audio								
•				Out	Signal	audio					Panel	/InSignal		

Figure 116: Mic\_Filter

- 18. Close the filter.
- 19. Open Rad\_Int\_Select. This component connects to a MathFunction, which is defined in the Math Plan.
- 20. To use this feature, you must establish what you want the **MathFunction** to do, build it, and link it. *PTTSelect's* output is binary-weighted based on switch position 1, 2, 3, and 4 resulting in 1, 2, 4, and 8. The first three positions are relevant to this exercise:
  - Position 1: intercom communication
  - Position 2: AM
  - Position 3: FM

**Panel** includes two buses: *Intercom* and *Radio*. Choose **AM** or **FM** in the radio, not the panel.

21. Next, add a table with no interpolation:

Input	Output
0	0
1	1
2	2
3	2

- 22. Select ACE STUDIO Project Manager.
- 23. Open MathPlan Editor.
- 24. To create a new group, in the toolbar, select **Add Group**, and name it **Panel**. Store any panel-related **MathFunctions** in this group.
- 25. In the toolbar, select add function. Rename the new function Rad\_Int\_Sel.
- 26. In Function Type, select None on Rad\_Int\_Sel. From the drop-down box, select TABLE.
- 27. To open Table Function, in the right column, select edit.
- 28. To make the function switch between values when selected, clear Interpolate.
- 29. Under x=, enter 0.0, 1.0, 2.0, 4.0, and 8.0.



Note: PTTSelect yields a transient value of 0.0 between channels.

30. Under f(x)=, enter 0.0, 1.0, 2.0, and 2.0.

					-			Table	Function		×
	Mathplan	Editor - mathpla	in	_ = ×	Han	dle: Rad_Ir	nt_Sel		🗌 Interpo	Interpolate	
-		$\otimes$	Net C. Terry		x =	0.0	f(x) =	0.0	× =	f(x) =	
Add Group Add Function C	lone Function	Delete	Notiry Targe	et	x =	1.0	f(x) =	1.0	× =	f(x) =	
Function Groups	Functions				x =	2.0	f(x) =	2.0	× =	f(x) =	
Group	Handle	Function Type	Description		x =	4.0	f(x) =	2.0	× =	f(x) =	
basic	Add	ADD	None	F(x,y) = x + y	x =		f(x) =		× =	f(x) =	
Panel	Subtract	SUBTRACT	None	F(x,y) = x - y	x =		f(x) =		× =	f(x) =	
Radio	Multiply	MULTIPLY	None	F(x,y) = x * y	x =		f(x) =		х =	f(x) =	
8	Linear	LINEAR	None	F(x,y,z) = x * y + z	x =		f(x) =		× =	f(x) =	
	Logical-and	LOGICAL-AND	None	$F(x,y) = x \otimes y$	x =		f(x) =		× =	f(x) =	
	Logical-xor	LOGICAL-OR	None	F(x, y) = x   y F(x, y) = x   y	x =		f(x) =		x =	f(x) =	
	Rad_Int_Sel	TABLE	None	<edit></edit>	х =		f(x) =		× =	f(x) =	
					x =		f(x) =		× =	f(x) =	
					x =		f(x) =		× =	f(x) =	
					x =		f(x) =		× =	f(x) =	
		x =		f(x) =		× =	f(x) =				
Caral Araly OK							f(x) =		× =	f(x) =	
					2 Ca	ncel	≪∎ок				
Cancel Apply OK							f(x) =		x = x = x = x = x =	f(x) = f(x) = f(x) = f(x) = f(x) = f(x) =	СК

Figure 117: Rad\_Int\_Sel MathFunction

- 31. Go to **MathPlan Editor**. To implement changes immediately after application, select **Notify Target**. You do not need to save and reload the layout.
- 32. Apply your changes, and select Yes to confirm.
- 33. Go to Model Viewer, and open Rad\_Int\_Select.
- 34. On function, double-click Select, and Select File opens.
- 35. To link **Rad\_Int\_Select** to **Math Plan**, select **Panel** > **Rad\_Int\_Sel**, and select **OK**. The first operator is now complete.

Dod. In			opera	o ×									
Rau_II	nt_Sele	ect [Contro	ol/MathF	unction]	0							Select File	×
Data	Links	Schematio	: Info	View/Edit D	Description							Linear	
							Ľ	Filter \	/iew	Full View		Logical-and	
Fre	om		Variable	е Туре	Value	C	рM	lodifier	Result	То		Logical-or	
			Function	n function	<select></select>							Logical-xor	
Ор	erator/F	PTTselect	Input_X	float32	255.0	*	1.	0	255.0			Multiply	
			Input_Y	float32	1.0	*	1.	0	1.0			Rad_Int_Sel	
			Input_Z	float32	1.0	*	1.	0	1.0			Subtract	
			Gain	float32	1.0	•	1.	0	1.0			🕨 🗐 Radio	ш
			Result	float32	255.0				255.0	multiple		🕨 🔄 basic	
												Cancel OK	

Figure 118: MathFunction selected

- 36. To view the **Operator\_1** folder, in the top-left of **Model Viewer**, select **Operators**.
- 37. Copy and paste the folder, and rename it **Operator\_2**.
- 38. Open **Operator\_2** and the **Operator** component.
- 39. Change the *Identifier* and *Channel* to the ACU2 channel connected to the second headset and PTT device.
- 40. Open **Panel**, and change the *Radio1* value to **Radio2**.
- 41. Close the component window and **Model Viewer**. Apply the changes.



Figure 119: Apply changes

42. Close MathPlan Editor.

- 43. On the **Project Manager**, select **Save Project**. In the commit dialog, make a useful comment (e.g., "Operators completed"). The operators section of the model is now complete.
- 44. From the toolbar, reinstall the layout. When complete, open the Model Viewer.
- 45. Open the Radios Sim model.
- 46. Right-click on the canvas, and select **Add**. Add a folder, name it **Operator1\_Radio**, and open it.
- 47. Add the following components, and name them:

Components	Names
RCUbasic	Controller
Transceiver	RX_TX
MathFunction	AM_FM_Select





Figure 120: Adding components to Operator1\_Radio

- 48. Open Controller and RX\_TX.
- 49. To open RadioControlService Editor in Controller, double-click UNASSIGNED in *TransceiverId*. Select New Bus, and rename it RX\_TX1. Press Enter, and select Set Value.
- 50. To open RadioControlService Editor in RX\_TX, double-click UNASSIGNED in *TransceiverId*. Select New Bus, and rename it RX\_TX1. Press Enter, and select Set Value.
- 51. *TransceiverId* pairs **RCUbasic** and **Transceiver** together for control purposes. In the **Controller**, double-click **Select** in the **Value** column of the **Fill** line. Using the drop-down box, select **Plan.RADIO**.

52. In the **Modifier** column of the *Frequency* row, enter **200,000,000**. The **RX\_TX** values for *Frequency* and *Mode* fill in.

	Radios:Controller (/Operator1_Radio/) - Ace Data Viewer									Radios:RX_TX (/Operator1_Radio/) - Ace Data Viewer							
Controlle	r [Radio/RCUbasic]	Radic	ControlServi	<u>ce</u>				RX_T	<b>X</b> [Ra	dio/Transceiver]	PowerServ	ice RadioControlS	ervice	Intercomi	BusService	WorldPositionServi	<u>ce</u>
Data Link	ks Schematic Info	View/Edi	t Description					Data	Link	s Schematic Info	View/Edit De	escription					
					4	Filter View	Full View							- 🖌 🛛	ilter View	Full View	
From	Variable	Туре	Value	Ор	Modifier	Result To		F	rom	Variable	Туре	Value	Ор	Modifier	Result	То	
												TRUE					
	Fill	fill	Plan.RADIO							PowerBus	id	UNASSIGNED					
	Net	uint32	1	+	0	1				PowerState	boolean	TRUE					
	Frequency	uint64	0	+	200000000	20000000				RadioBus	id	Radio1					
	TxFrequency	uint64	0	+	0	0				ReceiveGain	float32	1.0	*	1.0	1.0		
	TxPower	float32	0.0	+	0.0	0.0				SquelchLevel	float32	0.200000	*	1.0	0.200000		
	CryptoKey	uint16	0	+	0	0				RxDataThreshold	float32	0.200000		1.0	0.200000		
	CryptoSys	uint16	0	+	0	0				TransceiverId	id	RX_TX1					
	CodecType	uint16	0	+	0	0				Mode	TunerMode	AM					
	FreqHopNetId	uint16	0	+	0	0				TxFrequency	uint64	20000000					
	FreqHopSyncTOD	uint32	0	+	0	0				RxFrequency	uint64	20000000					

Figure 121: Initial radio entries

- 53. For **RadioBus**, double-click **UNASSIGNED**, and **IntercomBusService** opens. **Panel** in **Operators** creates three entries.
- 54. Highlight **Radio1**, and select **Set Value**. Audio to and from the **RX\_TX** is present on **Radio1** of the **IntercomBusService**. **Panel** and **RX\_TX** can now communicate with each other.
- 55. For **RadioName**, double-click **Edit**, and enter a valid name for the radio (e.g., **Oper-ator1\_Radio**).
- 56. Fill **DomainName**. For radios to communicate, they must be on the same domain and have matching radio parameters. Even if the parameters match, if the domains do not match, the radios cannot see each other. Double-click **Edit**, and enter **local\_radios**.



Figure 122: Final radio entries

- 57. The radio needs to know to whether to choose AM or FM modes. For this, we will use the AM\_FM\_Select component. In Project Manager, open MathPlan Editor.
- 58. Add a new group called **Radio**. This group stores any **MathFunctions** relevant to the radios. Add a new function table called **AM\_FM\_Sel**.

59. *PTTSelect*, a binary-weighted value, drives the table. **AM** is 2, and **FM** is 4. For the table to work, an input of 2 produces an output of 1 and an input of 4 produces an output of 2. Add the table with interpolate off and the following values:

Input	Output
0	0
1	1
2	2
3	2

Figure 123, "Mathplan Editor" below shows the radio select table in Mathplan Editor:

	Mathplan	Editor - mathpla	n	_ = ×		4		Table I	unction	×
-		$\otimes$	Notify Targ	at	Han	dle: AM_F	M_Sel		🗌 Interpola	te
Add Group Add Function	Clone Function	Delete		x =	0.0	f(x) =	0.0	× =	f(x) =	
Function Groups	Functions		x =	1.0	f(x) =	0.0	x =	f(x) =		
Group	Handle	Function Type	Description		x =	2.0	f(x) =	1.0	× =	f(x) =
basic	Add	ADD	None	F(x,y) = x + y	x =	4.0	f(x) =	2.0	× =	f(x) =
Panel	Subtract	SUBTRACT	None	F(x,y) = x - y	x =		f(x) =		× =	f(x) =
Radio	Multiply	MULTIPLY	None	F(x,y) = x * y	x =		f(x) =		x =	f(x) =
	Linear	LINEAR	None	F(x,y,z) = x * y + z	<sub>x =</sub>		f(x) =		x =	f(x) =
	Logical-and	LOGICAL-AND	None	F(x,y) = x & y			64.A -			5(L) =
	Logical-or	LOGICAL-OR	None	F(x, y) = x   y	x =		r(x) =		x =	$T(\mathbf{x}) =$
	Logical-xor LOGICAL-XOR		None	$F(x, y) = x \uparrow y$	x =		f(x) =		× =	f(x) =
	AM_FM_Sel TABLE None			<edit></edit>	x =		f(x) =		x =	f(x) =

Figure 123: Mathplan Editor

- 60. In MathPlan Editor, select Notify Target, and apply your changes.
- 61. In Model Viewer, link the AM\_FM\_Select component to that MathFunction.
- 62. Link AM\_FM\_Select *Result* to Controller *Net*.
- 63. In ACE MODEL LINK EDITOR, use a bus to link the Radios and Operators simulation models.
- 64. Close ACE MODEL LINK EDITOR.
- 65. Go to the **Radios** simulation model, and copy the **Operator1\_Radio** folder to **Operator2\_Radio**. Open the **Operator2\_Radio** folder.
- 66. Open **Controller**, and change the *TransceiverId* to **RX\_TX2** by adding a new entry in the **Radio Control Service**.
- 67. Close Controller, and open RX\_TX.
- 68. Change *RadioBus* to **Radio2**.
- 69. Change the *TransceiverId* to **RX\_TX2**.

70. Change *RadioName* to **Operator2\_Radio**.

		Radios:Controller (/C	perator	ata Viewer		×	Ĩ		R	adios:RX_TX	/Operator2_Radio	/) - Ao	e Data Vi	ewer	_ = = :	×		
Cont	roller [Radio/RCUbas	ic] 👩 RadioControl	Service						RX_1	<b>FX</b> [Ra	dio/Transceiver]	PowerServ	ice RadioControlS	ervice	Intercom	usService	WorldPositionService	e
Data	Links Schematic	Info View/Edit Descri	ption						Data	Link	s Schematic Info	View/Edit De	scription					
				4	Fi	lter View	Full View		1					4	Filter \	/iew	Full View	
	rom	Variable	Туре	Value	Ор	Modifier	Result	То	F	rom	Variable	Туре	Value	Ор	Modifier	Result	То	T
		TransceiverId	id	RX_TX2							PowerIn	boolean	TRUE			TRUE		1
		Fill	fill	Plan.RADIO							PowerBus	id	UNASSIGNED					1
4	M_FM_Select/Result	Net	uint32	2	+	0	2				PowerState	boolean	TRUE					I
-		Frequency	uint64	0	+	200000000	200000000				RadioBus	id	Radio2					I
-		TxFrequency	uint64	0	+	0	0		-		ReceiveGain	float32	1.0	*	1.0	1.0		I
-		TxPower	float32	0.0	+	0.0	0.0				SquelchLevel	float32	0.200000	•	1.0	0.200000	1	I
-		CryptoKey	uint16	0	+	0	0		-		RxDataThreshold	float32	0.200000	*	1.0	0.200000	1	I
-		CryptoSys	uint16	0	+	0	0				TransceiverId	id	RX_TX2					I
-		CodecType	uint16	0	+	0	0				Mode	TunerMode	FM					L
-		FreqHopNetId	uint16	0	+	0	0				TxFrequency	uint64	20000000					I
-		FreqHopSyncTOD	uint32	0	+	0	0				RxFrequency	uint64	200000000					L
-		FreqHopTranSecKey	uint16	0	+	0	0				RadioName	string	Operator1_Radio					I
- 1		FreqHopHopSetWOD	uint16	0	+	0	0		-		RadioNameIn	string						1
-		FreqHopLockOutId	uint16	0	+	0	0				DomainName	string	local_radios					1
-		FreqHopSystem	uint16	0	+	0	0		-		DomainNameIn	string						1

Figure 124: Radio2 final values

- 71. Close **RX\_TX**.
- 72. In **Operator2\_Radio**, right-click in the canvas, and select **Add**. Add a connector, and name it **PTT\_data**. Open the connector.
- 73. At the top of the window, next to Group: Unassigned, select change. Set the group to Operator2.
- 74. To add a signal, select the plus sign (+). Name the new signal **PTT\_Select**. Close the connector.
- 75. Right-click the connector, and select Create Link...
- 76. Link **PTT\_Data**/*PTT\_Select* to **AM\_FM\_Select**/*Input\_X*. Close **ACE MODEL LINK EDITOR**.

	R	adios:PTT_Data (/Operator2_Radio/) - Ace Connector Viewer	-		×
PTT_Da	TT_Data [Connector]				
Connec	tor View				
÷	📿 Group:	Operator2 change			
From	Signal	То			
	PTT_Select	AM_FM_Select/Input_X			
	PTT_Select	AM_FM_Select/Input_X			

Figure 125: Connector group assign

77. Right-click the connector, and select **Assign to Bus**. Name the bus **Inter\_Model\_ Data**.



Figure 126: Operator2 radio completed

- 78. Right-click the connector, and select Copy.
- 79. Go to **Operator1\_Radio\_Folder**.
- 80. Right-click the canvas, and select **Paste**. Open the new connector. Change the **Group** to **Operator1**.
- 81. Right-click the connector, and select Create Link...
- 82. Link PTT\_Data/PTT\_Select to AM\_FM\_Select/Input\_X. Close the ACE MODEL LINK EDITOR.
- 83. Right-click the connector, and select **Assign to Bus**. Select the bus **Inter\_Model\_ Data**.
- 84. Go to **Operators** > **Operator\_2**.
- 85. Right-click the canvas, and select Paste.
- 86. Link **Operator**/*PTT\_Select* to **PTT\_Data**/*PTT\_Select*.

87. Assign the **PTT\_Data** connector to the **Inter\_Model\_Data** bus.



Figure 127: Operator2 panel completed

- 88. Go to **Operators** > **Operator\_1**.
- 89. Right-click the canvas, and select **Paste**. Open the connector, and change the group to **Operator1**.
- 90. Link **Operator**/*PTT\_Select* to **PTT\_Data**/*PTT\_Select*.
- 91. Assign the **PTT\_Data** connector to the **Inter\_Model\_Data** bus.



Figure 128: Completed layout

92. Open the Inter\_Model\_Data bus.

93. Two groups display in Ace Bus Viewer. If you expand the groups, you can see the signals in the groups and the sources and destinations. The signal names in the groups are the same, but because they are in different groups; they are not connected to each other.

	Inter_Model_Data (load) - Ace Bus Viewer						×
Inter.	nter_Model_Data [BUS]						
Bus \	View	Icon View					
Fron	n		Signal	То			
-			Operator1				
	Оре	rators:PTT_I	Data PTT_Select	Radios:PTT_Data			
-			Operator2				
	Оре	rators:PTT_I	Data PTT_Select	Radios:PTT_Data			

Figure 129: Inter Model Bus

An analogy would be a large cable containing two multicore cables.

94. **Icon View** gives a visual representation of the source and destination connectors and the groups within the bus.

	Inter_Model_Data (load) - Ace Bus Viewer			×
Inter_Model_Data [BUS]	0			
Bus View Icon View				
Operators: PTT_Data	→ 🚰 Operator1	Radios:PTT _Data		
Operators: PTT_Data	→ ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ	Radios:PTT _Data		

Figure 130: Inter Model Bus icon view

Exit **Model Viewer**, and apply the changes. In **Project Manager**, save the project, and reload the layout.

The project is complete. Two operators can now communicate over a local intercom or a radio in AM or FM mode. Radios and intercoms in the same Target can communicate with each other, but radios and intercoms in different Targets cannot.

## 9.8 Add a communication model with helpers

The following instructions explain how to create a communication model using the ACE Studio helpers. This example assumes a level of familiarity with ACE software. To create a communication model with ACE Studio helpers, follow these steps:

- 1. Add a new project called Helpers.
- 2. Under Servers, create a DIS\_Gateway.
- 3. Set the DIS version to 6.
- 4. Set the **DIS interface** to **eth0**.
- 5. Set the DIS RX/TX port to 53000.
- 6. Next to **main**, set the outgoing destination address for DIS packets to **192.168.255.255**. Select **bcast**.

DIS Gateway					
File Info:	(* indicates required field)				
Name* :	DIS				
General					
version	6				
Interfaces	5				
DIS inter	face eth0 v port 53000				
main	192168.255.255 • bcast 🔿 mca	ast			
signal	● bcast ○ mca	ast			

Figure 131: DIS Gateway

- 7. To create the domain, follow these steps:
  - a. In the tree view, select main.
  - b. To add a domain, in **Domain Editor**, select the plus sign (+).
  - c. In Set Domain Name, enter DIS.
  - d. Set the Exercise ID to 24.
  - e. Set the Site and App ID to the last two octets of the Telestra's IP octet.

f. Select Apply and OK.

	Domain Editor - domain	-	۰	×
Names HLA				
Add Domains:	Domain Info: Name: DIS Comment: No Comment DIS: Exercise ID: 24 © Set IDs to Last Two IP Octets © Set IDs Manually Site ID: App ID:			
+ -	File:			
	Cancel Apply		ОК	

Figure 132: Domain Editor

8. Create a Comm Plan with four FM nets (i.e., FM1–FM4).

- 9. Set the following default frequencies:
  - 20100000
  - 20200000
  - 20300000
  - 20400000

			Commp	lan Edit	or - commplan			_ = ×
	+	1						Expand All
commplan	Name	Mode	Encoding	Rate	Bandwidth(Hz)	Tx Power(Watts)	RECEIVERGAIN	Voice Effects
<sup>™</sup> <u>Fill</u>	FM	FM	MULAW	8000	25000	1.0	rxgain.rxgain	ON
wei								
Waveform								
Radios							A	

### Figure 133: Commplan waveform

		c	Commplan Editor - com	mplan				_
						Expand All		
commplan	Name	Frequency(Hz)	TxFrequency(Hz)	WAVEFORM	CRYPTO	FREQHOP	SAT	гсом
Fill	FM1	201,000,000		Radios.FM	Off	Off	Off	
▼ □ <u>Net</u>	FM2	202,000,000		Radios.FM	Off	Off	Off	
Nets	FM3	203,000,000		Radios.FM	Off	Off	Off	
▼ <u>vvaveform</u>	FM4	204,000,000		Radios.FM	Off	Off	Off	
Radios								

#### Figure 134: Commplan nets

·	Commplan Edit	tor - commplan	_
commplan	+ - 5		Expand All
	Name	NET	
▼ 🖾 <u>Net</u>	▼ FMRADIO 1	Nets.FM1	
Nets Waveform	2	Nets.FM2	
Radios	4	Nets.FM3 Nets.FM4	
Chypto			

Figure 135: Commplan fills

- 10. Install the layout.
- 11. On the main canvas, double-click **radios**.

12. In the bottom-left of the Radio Helper, select the plus sign (). Add a World Position, and name it WP1.

Enter Rad	×	
WORLD_POS	•	
WP1		
Cancel	OK	

Figure 136: Add a WORLD\_POSITION

13. Select the plus sign (+), add a radio, and name it **RADIO\_1**.

Enter Rad	×	
RADIO		•
RADIO_1		
Cancel	OK	

Figure 137: Add a radio

- 14. Select **Domain**, and then select DIS.
- 15. Set the Entity ID (e.g., 5) and the Radio Number (e.g., 1).
- 16. Select Fill, and then select Radio.FMRADIO.
- 17. From World Position, select WP1.

18. Repeat this for the next three radios (i.e., RADIO2, RADIO3, and RADIO4), and use the radio numbers 2, 3, and 4.

	Radio Helper	×
Radio Group Info:		
Group Name: radios		
Add Radios:	General Advanced	
Radio Name	Info:	
WORLD_POSITION:WP1	Name: RADIO:RADIO_1	
RADIO:RADIO_1	Comment: No Comment	
	Settings:	
	Domain: DIS	
	Exercise ID: 24	
	Set IDs from Domain	
	Set IDs Manually	
	Site: App: Entity: 5 Radio: 1	
	Marking Field:	
	Fill: RADIO.FMRADIO	
	Crypto Library:	
	World Position: WP1	

#### Figure 138: Radio information

- 19. To close the window and update the layout, select Update.
- 20. On the main canvas, double-click channels.
- 21. In Channel Helper, select the plus sign (+), add a RADIO\_OPERATOR, and name it OPERATOR1.
- 22. For the bus, enter **Operators**.
- 23. For the Bus Group, enter Operator1.

### 24. For bus1, bus2, bus3, and bus4, select RADIO\_1-RADIO\_4.

Channel Helper						
Channel Group Info (* indicate	s required field):					
Group Name* : channels						
Add Channels:	OP SINCGARS VOISUS					
1_RADIO_OPERATOR:OPE	Channel Info: Name: 1_RADIO_OPERATOR:OPERATOR1					
1_RADIO_OPERATOR:OPE	Comment: No Comment					
	Assignment:					
	Bus:					
	Bus Group:					
	Highway:					
	Highway: Out Left Right					
	Intercom Service:	-11				
	bus1 RADIO_1 v bus9 Select Radio v					
	bus2 RADIO_2   bus10 Select Radio					
	bus3 RADIO_3   bus11 Select Radio					
	bus4 RADIO_4   bus12 Select Radio					
	bus5 Select Radio 🔹 bus13 Select Radio 👻					
* =	bus6 Select Radio 🗸 bus14 Select Radio 🗸					
	Cancel 👿 U	pdate				

Figure 139: Add an operator

25. Add Operator2.

26. For the **Bus Group**, enter **Operator2**.

r	Channel Helper	General Contraction of the second	×
Channel Group Info (* indicates Group Name* : channels	required field):		
Add Channels: Channel Name 1_RADIO_OPERATOR:OPE 1_RADIO_OPERATOR:OPE	OP SINCGARS VOISUS Channel Info: Name: 1_RADIO_OPERATOR:OP Comment: No Comment Assignment: Bus: Bus Group: Highway: Highway: Out Left Left	PERATOR2	
+ -	bus1 RADIO_1 bus2 RADIO_2 bus3 RADIO_3 bus4 RADIO_4 bus5 Select Radio bus6 Select Radio	bus9     Select Radio       bus10     Select Radio       bus11     Select Radio       bus12     Select Radio       bus13     Select Radio       bus14     Select Radio	

Figure 140: Add a second operator

27. Select Update.

28. Right-click the main canvas, and add an ACU2. Identify it, and set channel A to be channels > 1\_RADIO\_OPERATOR:OPERATOR1 and channel B to be channels > 1\_RADIO\_OPERATOR:OPERATOR2. Select Add.

				ACU2 Ed	itor	×
Cor	nfigure (	* required	l fields)			
Na	me: *	ACU2_1	L			
AC	U2: *	ACU2_1	L			
Lo	cation:					
Te	lestra: *	Mordor	-rh7	•		
Cha	annels	4 💌				
A	channe	els 🔹	-	1_RADIO	OPERATOR:OPER	ATOR1 🔻
в	channe	els 🔹	-	1_RADIO	OPERATOR:OPERA	ATOR2 🔻
с	Select	Group	-		Select Channel 🔻	
D	Select	Group	-		Select Channel 🔻	
					💥 Cancel	- Add

Figure 141: ACU2 Editor

- 29. Save and install the layout.
- 30. Open Model Viewer.
- 31. In **Model Viewer**, substantial chunks of the layout are now automatically built, thus saving time in model generation. This method is not for everyone due to helpers' limitations, but it may be worth considering.



Figure 142: Model layout

# **10.0 Advanced topics and examples**

This chapter discusses the following topics:

- ACE Scope
- ACE GLOBAL LINK EDITOR

# 10.1 ACE Scope

The scope function in ACE Studio is a graphical representation of an oscilloscope display. Various additional features are included in the scope function. This section discusses the following topics:

- Main display
- 1/3 Octave display
- FFT display
- Comparison display

## 10.1.1 Main display

Below is a typical scope display. It is displaying a sine wave, amplitude +1 to -1 peak-to-peak with a frequency of 500 Hz.



Figure 143: Scope main display

When the scope is first initiated, the default **Period** is 100 msec. To make the screen clearer, you can change it to 10 msec.

The peak-to-peak value is exactly 2, or +1 to -1. The main display shows five wave cycles, which equate to 500 Hz.

In **Threshold**, enter a trigger value for the time base. Select an **ascending** or **descending** signal, exactly like an oscilloscope.

To freeze the screen for detail inspection, select Freeze.

The Y axis has several adjustments. Figure 144, "Adjusted scope display" below shows the display after making several adjustments.



Figure 144: Adjusted scope display

In this example, **Y Range** is **4**. The vertical range goes from -3.0 to 1.0, which is 4 units.

**Y Offset** is **1.0**, so the vertical center of the scope is -1.0. The +1 refers to moving the 0 reference up by 1 unit.

**Y Resolution** changes to **0.5**. This value has no effect on the signal; however, it moves the grid lines closer together for more detailed examination of the plot. In this case, they are 0.5 units apart.

## 10.1.2 1/3 Octave display

To view an alternate display, select 1/3 Octave:



Figure 145: 1/3 Octave view

In Figure 145, "1/3 Octave view" above, the amplitude of the signal displays in 1/3 octave increments from 50-16,000 Hz. Since the signal is a 500 Hz sine wave, the peak is at 500 Hz, and the signal amplitude rapidly drops off on both sides of the peak.

To see the amplitude of the signal at any 1/3 octave step, move the mouse over the display, and the band's value shows under **Mouse Over Band**. Alternatively, select individual bands, and the amplitude values display under **Selected Bands**.



Figure 146: Highlighting bands

In Figure 146, "Highlighting bands" on the previous page, no indications exist in **Mouse Over Band**. However, four bands are highlighted:

- 250 Hz with an amplitude of -52.3 dB
- 500 Hz with an amplitude of -3.0 dB
- 630 Hz at an amplitude of -20.9 dB
- 2,500 Hz at an amplitude of -78.6 dB

These amplitudes update in real time.

Scale includes four selections:

- *dBFS or Full Scale:* scales the display vertically from -96 dB to 0 dB. This scale is provided with a 96 dB range, since the 16-bit audio used in the .wav files for use with the Telestra gives a dynamic range of 96 dB.
- *Normalized*: scales the display from 0 to 1. Unlike the dB scales, this is a linear scale and has the effect of compressing the parts of the signal with a lower level.
- *Cal94DB SPL*: useful for sound tuning.
- Call14dB SPL: useful for sound tuning.



Important: Do not use Cal 94 dB.

Figure 147, "Calibration tone" below shows a plot at 94 dB, 1000 Hz:



Figure 147: Calibration tone

When you are making Level D recordings on an aircraft, a 1000 Hz calibration tone records at the start of each recording session. The audio source is normally a 1000 Hz tone at either 94 dB SPL or 114 dB SPL. For most applications, 94 dB is more than adequate. For extremely loud aircraft, normally military transport or rotary wing, the reference is usually 114 dB SPL.

When tuning an aural cue system, use a calibrated microphone. Start by recording a reference level just like in the aircraft. We use the scope 1/3 Octave display set to 114 SPL mode (not 94 dB), and adjust the input channel gain until we see either 94 dB or 114 dB at the center frequency, depending upon the reference level calibrating the microphone. The input channel is now calibrated to the microphone. For comparison purposes, if we were to compare the aircraft calibration recording to the microphone level, they should match.

As a result, the signal amplitude could be the same for both 94 dB SPL and for 114 dB SPL. It all depends upon the sensitivity and scaling of the microphones being used. All we are doing is adjusting the scale of the display to make it easier to read and giving us a calibrated display of the actual signal amplitude.

Clearing **Moving** and **Time** when a dynamic signal displays causes the band amplitudes to change rapidly, making them difficult to read. Level D tuning requires you to compare sounds over a period of time (e.g., 15 seconds to three minutes, in some cases). You can use individual band averages.

**Moving** averages each individual band over time. The longer **Moving** is selected, the longer the averaging period. If **Time** is selected, the display shows the peak value of each band and a line showing the average values.

### 10.1.3 FFT display

Any time signal (e.g., noise, speech, music) is a function of fundamental frequencies, a combination of sinusoidal waves of varying frequency, amplitude, and phase. The FFT display shows the amplitudes and frequencies of these root sine waves, which build the resultant sound. This theory is useful while duplicating an engine sound, for example. You can transpose the frequencies and amplitudes in the display directly into the **Engine** component.



Figure 148, "FFT display" below is the FFT display for the sine wave:

Figure 148: FFT display

The peak is 1000 Hz, as expected.

To assist in the analysis, adjust the Y gain with **Y Factor**. The **X Max** limits the upper frequency for analysis, which stretches the lower frequency end of the display, improving clarity. The **X Resolution** adjusts the graphical scale's resolution.

**Tears** detects transients that may appear and disappear too quickly to analyze. The tears show the max value for a few seconds and return to their original value.

In Figure 149, "Filtered noise FFT display" below, the 1000 Hz sine wave was replaced with a noise source after passing through a bandpass filter set to 1000 Hz.



Figure 149: Filtered noise FFT display

Even though the filter is a bandpass type, it has a finite bandwidth. Below, the display sums three sine waves of equal amplitude with the following frequencies:

- 500 Hz
- 1750 Hz
- 6500 Hz

The scope display doesn't look like sine waves. The peaks in 1/3 Octave give some idea of what is going on, but the FFT display has isolated the three frequencies that created the wave.



Figure 150: Three sine waves scope

To recreate the display, view the normal scope display. To repeat the scope display three times, from **File**, set the frame number to three. From these displays, choose the individual tabs and scaling as desired.

### 10.1.4 Comparison display

When tuning an aural cue system, you can use the scope in **1/3 Octave** as a comparison tool. When creating a reference file during Level D recordings, a tightly controlled known set of conditions exists. Convert the reference file to a calibrated format, and store them on the Telestra server.

You may now use the known conditions to drive the aural cue system, which is set to a calibrated volume (i.e., normally 100 percent). The spectral analysis microphone can pick up the signal once you seal it for calibration. Listen to it with a **Playsound**, and compare it to the referenced sound file. To use the comparison display, follow these steps:

 Instigate a scope using View Scope from Pink Noise Reference. Open Pink Noise Filtered, and right-click View Scope. Figure 151, "Add to Scope" below shows how to compare a pink noise reference source to a 1000 Hz bandpass-filtered pink noise source.

		Sim	_Model:Aud	io_NoiseSo	urce2	(Root) - Ac	e Data View	er		-	×
Audio	_Noise	Source2 [Audio,	NoiseSource	]							
Data	Links	Schematic Info	View/Edit [	Description							
						$\checkmark$	Filter View		Full View		
	From	Variable	Туре	Value	Ор	Modifier	Result	Тс	)		
		NoiseColor	noise_color	White							
		FilterEnable	boolean	FALSE	^	TRUE	TRUE				
		FilterFrequency	float32	1.0	*	1000.0	1000.0				
		FilterQFactor	float32	1.0	*	0.707100	0.707100				
		FilterType	filter_type2	BandPass							
		OutGain	float32	1.0	*	1.0	1.0				
-		OutSignal	audio								
		active	boolean	TRUE			TRUE				
		audio	audio	Add t	to Sco	pe 🕨	View Scope				
							-				

Figure 151: Add to Scope

2. Select Add to Scope.

		Sim	_Model:Aud	io_NoiseSou	irce2	(Root) - Ac	e Data View:	/er _		×
Audio	_Noise	Source2 [Audio/	NoiseSource	] (0)						
Data	Links	Schematic Info	View/Edit [	Description						
						1	Filter View	Full View		
	From	Variable	Туре	Value	Ор	Modifier	Result	То		I
		NoiseColor	noise_color	White						
		FilterEnable	boolean	FALSE	^	TRUE	TRUE			
		FilterFrequency	float32	1.0	*	1000.0	1000.0			
		FilterQFactor	float32	1.0	*	0.707100	0.707100			
		FilterType	filter_type2	BandPass						
		OutGain	float32	1.0	*	1.0	1.0			
•		OutSignal	audio							
		active	boolean	TRUE			TRUE			
									( , , , , , , , , , , , , , , , , , , ,	

Figure 152: Select Add to Scope

3. Choose a scope, and add a signal to it.



Figure 153: Added to scope

Both scope displays display in a single window. They are different, but it would be nice to see the spectral difference so you can tune the pink noise to match the reference.

4. At the top of the window, select File.

	Ace Scope	-		×
File Common				
Change Layout	- OutSignal(Value).audio			
Quit	c) Trigger Threshold: 0.0	ending	Fre	eze

Figure 154: Change Layout

5. Select **Change Layout** to reset the display.

(	Change Layout	-		×
1 Frame				•
/Audio_NoiseSourc	el - OutSignal(Va	alue).	audio	•
	ОК			

Figure 155: Change Layout window

- 6. To use the comparison function, we need to show the display in one frame, so the top line is correct. The next line shows **Pink\_Noise-Reference**. Select **OK**, and the scope shows a scope display with a single window showing the **Pink\_Noise\_Reference** signal.
- 7. To select the comparison mode, select the second line, and then select Comparison.

(				-
	Change Layout	-		×
1 Frame				•
/Audio_NoiseSou	rce1 - OutSignal(V	/alue).a	audio	
/Audio_NoiseSou	irce2 - OutSignal(Va	lue).au	ıdio	
Comparison				

Figure 156: SelectComparison layout

8. When **Comparison** opens, choose which signals to compare.

Comparison _ 🗆	×
Compare	
/Audio_NoiseSource2 - OutSignal(Value).audio	•
to	
/Audio_NoiseSource1 - OutSignal(Value).audio	•
ОК	

Figure 157: Select comparison signals

- 9. Compare the filtered signal to the reference signal. The decision as to which signal is the master and which is the one to compare it to is important.
- 10. Select OK twice, and select Change Layout.



Figure 158: Comparison mode

- 11. In **Comparison**, choose which signals to place on Lines 1 and 2.
- 12. Compare the filtered source to the reference source. In the display, if the amplitude of a frequency band in the filtered signal is lower that the same band in the reference signal, the display shows the band dropping from the 0 dB reference level.

In the displayed comparison, the filtered signal passes through a bandpass filter of 1,000 Hz. At the 1000 Hz band, a positive gain of around 7 dB exists. On either side, the bands move farther away from the center frequency. The filtered signal's amplitude drops to 50 Hz and 16,000 Hz, approximately 30 dB and 10 dB below the reference signal.

13. To make a display match, remove the bandpass filter. In a complex simulated audio system, use selective dynamic filtering of the propellers or engines to increase or decrease gains at different frequencies to match the output from the model to the referenced audio.

# **10.2 ACE GLOBAL LINK EDITOR**

ACE GLOBAL LINK EDITOR is an alternative method to create buses and connectors, interlinking sim models. The tool allows a visualization of a bus connection from the source component to the destination component. You can use this feature to add signals to existing connectors and buses.

### 10.2.1 Example model

To create an example model and layout, follow these steps:

1. Install the layout, and in **Model Viewer**, create a new simulation model, and a new host model.

load	main
Sim_Model	
load-servi	
ces	
HOSL_MODEL	

Figure 159: Example model

2. Add a **HostIn** component, which provides the labels to connect to the simulation model.



Figure 160: Host interface

3. Populate Host Packet in Host Packet Editor.

					н	ost_Model:H	ost_Data (	Root) - AC	E IO Packe	t Edito	or				-	×
Host_D	ata [N/A](change	e) 👩 🖸	ontroller In	nport ICD	Live Capture	🕑 Enable T	esting Littl	e Endian	•							
Packet	Notes															
+ -	🖣 🖳 📿 Align	Offset	Clear Test	tmode	Add to Conne	ctor Use	Init Value:	Never		•	Timeout (s)	1.00				-
ld. ▼	Name	Offse	MsgLer	Туре	Init. Value	Function	Υ	Z	Rscale	Test	Test Value	Used by	Other	Description		
1	Test_Boolean1	0	0	boolean	0	Select	0.000000	0.000000	0.000000	off	0		<ramp></ramp>	None		
2	Test_Boolean2	1	0	boolean	0	Select	0.000000	0.000000	0.000000	off	0		<ramp></ramp>	None		
3	Test_Boolean3	2	0	boolean	0	Select	0.000000	0.000000	0.000000	off	0		<ramp></ramp>	None		
4	Test_Boolean4	3	0	boolean	0	Select	0.000000	0.000000	0.000000	off	0		<ramp></ramp>	None		
5	Test_Float1	4	0	float32	0.000	Select	0.000000	0.000000	0.000000	off	0.000		<ramp></ramp>	None		
6	Test_Float2	8	0	float32	0.000	Select.	0.000000	0.000000	0.000000	off	0.000		<ramp></ramp>	None		
7	Test_Float3	12	0	float32	0.000	Select	0.000000	0.000000	0.000000	off	0.000		<ramp></ramp>	None		
8	Test_Float4	16	0	float32	0.000	Select	0.000000	0.000000	0.000000	off	0.000		<ramp></ramp>	None		

Figure 161: Test host packet

4. In **Host Packet Editor**, highlight all of the signals, select **Add to Connector...**, and add these signals to a new connector.

					н	ost_Mode	l:Host_Data (F	loot) - AC	E IO Packe	t Edit	or				-	•	×
Host_D	ata [N/A] <u>(change</u>	1 @ C	ontroller In	nport ICE	Live Capture	🗹 Enabl	e Testing Little	Endian	•								
Packet	Notes																
+ -	e 📑 🕵 🛛 Align	Offset	Clear Test	mode	Add to Conne	ctor	Jse Init Value:	Never		•	Timeout (s)	1.00					
Id. 🕶	Name	Offse	MsgLer	Туре	Init. Value	Function	Y	Z	Rscale	Test	Test Value	Used by	Other	Description			
1																	
2							0.000000	0.000000	0.000000								
3						Select	Select/Creat	e Connect	or × 00								
4						Select	Data Conner	tor	- 00								
5						Select			00								
6						Select	Cancel	0	к 00								
7						Select	0.000000	0.000000	0.000000								
8	Test_Float4	16	0	float32	0.000	Select	0.000000	0.000000	0.000000	off	0.000		<ramp></ramp>	None			

Figure 162: Add signals to connector

- 5. Assign the connector to a bus named **Data\_Bus**, and copy the connector.
- 6. Go to the simulation model, and paste the connector to the canvas, keeping its name. Add ten **MathFunctions** (i.e., **Function1–10**).
- 7. Link the eight signals in the connector to **Functions1–8** (x) inputs, and assign the connector to the **Data\_Bus**.



Figure 163: simulation model example

8. A bus connects the host and simulation models:



Figure 164: Load overview

To refresh the load and ensure all changes are current in the temporary files, select **Reload** in **Model Viewer**.

9. Go to Host Packet Editor, and add two signals to the host packet.

9	Test_Float5	20	0	float32	0.0	Select	0.000000 0.000000 0.000000 off	0.0	 <ramp></ramp>	None
10	Test_Float6	24	0	float32	0.0	Select	0.000000 0.000000 0.000000 off	0.0	 <ramp></ramp>	None

#### Figure 165: Labels in host packet

- Next, use ACE GLOBAL LINK EDITOR to connect these labels to Function9 and Function10 in the simulation model. At the top of Model Viewer, select Tools > Global Link Editor.
- 11. ACE GLOBAL LINK EDITOR is arranged from top to bottom, from source to destination. From the top, select your source model (i.e., **host\_model**) in **Select Src Model**. This action populates the next pane down with the relevant components in the selected model.
- 12. In this case, select **Host\_Data**, which populates the right pane with signals that you can link.
- 13. Select Test\_Float5, which is the first signal.

- 14. In **Host Model**, select the connector, which populates the left **Signal** pane. To add a signal to the connector, select the plus sign (+), and enter a name. **Load** only displays one bus. As a result, the bus displays in the right **Bus** pane. Normally, **Load** would display all buses.
- 15. Select Data\_Bus.

r		ACE GLOBAL LINK EDITOR	×
✓ Select Src Model     Host_Model			Bus 🔮 💳 Data_Bus
Component 🛊 📟	Signal  Signal	Signal * Connector * Connector Test_Float2 Test_Float3 Test_Float5	
Links Host_Data Test_Float2 -> D Host_Data Test_Boolean1 -> D Host_Data Test_Boolean4 -> D	Data_Connector Test_Float2 Data_Connector Test_Boolean1 Data_Connector Test_Boolean4		*

Figure 166: Source selections

- 16. Follow the same procedure for destinations. From Select Dest Model, select Sim\_ Model.
- 17. To populate **Signal**, in the lower-right pane, select **Data\_Connector**.
- 18. Add the signal label to match the new signal label you added in the source connector. This is done in the exact same way we added the signal in the source connector.



*Important*: The signal names in the source and destination connectors must be an exact match. The system allows you to use differing names, and they both appear in the bus but are not connected.

19. In the lower left panel, select Function9. In the right pane, select input\_X.

▼ Select Dest Model				
Sim_Model 👻				
Component 🐈 😑	Signal 🌵 📟	Signal 🐈 💻	Connector 🐈 😑	
▼ Sim_Model	Gain float32	Test_Boolean4	▼ Sim_Model	12.
Function7	Input_X float32	📀 🕂 📼 Test_Float1	Data_Connector	
Function10	Input_Y float32	Test_Float2		
Function9	Input_Z float32	C Test_Float3		
Function8		Test_Float4		
Function3		Test_Float5		
Links	I			

Figure 167: Destination selections

20. The entire signal path now displays. If everything is correct, select Link All. This creates the link for that signal. Do the same for Test\_Float6.

		ACE GLOBAL LINK EDITOR	_
▼ Select Src Model			Bus 🛖 🚍
Host_Model 🗸			Data_Bus
Component 🐈 💻	Signal 💠 📟	Signal 💠 💻 Connector 🐈 💻	
✓ Host_Model Host_Data	<ul> <li>Test_Float1 float32</li> <li>Test_Float2 float32</li> <li>Test_Float3 float32</li> <li>Test_Float4 float32</li> <li>Test_Float5 float32</li> <li>Test_Float6 float32</li> </ul>	Test_Float1     Float_Model     Data_Connector     Test_Float3     Test_Float4     Test_Float4     Test_Float5     Test_Float5     Test_Float6	
Links Host_Data Test_Float1 -> Data_CC Host_Data Test_Float5 -> Data_CC Host_Data Test_Float4 -> Data_CC Host_Data Test_Boolean3 -> Data_CC • Select Dest Model	onnector Test_Float1 onnector Test_Float5 onnector Test_Float4 onnector Test_Boolean3		1
Component 🔮 💻	Signat  Gain float32 Input_X float32 Input_Y float32 Input_Z float32 Input_Z float32	Signal 🛊  Connector  Connector	I⊗⊕ I⊗= Exists
			Link All

Figure 168: Test\_Float6 links

21. Close ACE GLOBAL LINK EDITOR. In the model, the load contains two additional signals.