System User's Guide for the EOSDIS Test System (ETS) MPS/Aura Simulator

Release 3.7
August 2003

Prepared Under Contract NAS9-98100
by Computer Sciences Corporation

System User's Guide
for the
EOSDIS Test System (ETS)
MPS/Aura Simulator

Release 3.7
August 2003
Prepared Under Contract NAS9-98100
by Computer Sciences Corporation

Prepared by:

_____
J. K. Carlson, CSC Date

Reviewed by:

_____
E. Quintin Date
ETS Technica1 Lead, CSC

_____
E. Noone Date
Simulations Technical Area Manager, CSC

Quality assured by:

_____
QA Officer, CSC Date

Approved by:

_____
W. Fuller Date
ETS Manager, Code 581

About This Document

This document is the System User's Guide for the Scalable, Integrated, Multimission Support System (SIMSS) Aura Simulator developed for the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC). This simulator is commonly referred to as the MPS/Aura simulator. The simulator is one of the deliverables of the EOSDIS Test System (ETS) Development Task under the CSOC contract. The System User's Guide provides an overview of operational concepts and procedures for this simulator. This document is approved by GSFC Code 581, and will be maintained by Computer Sciences Corporation (CSC) under the CSOC contract.

Who Should Use This Document

This document is for anyone who wishes to use the MPS/Aura Simulator. To use this simulator effectively you should already be familiar with the basics of the Windows NT and Windows 2000 Operating Systems as well as be familiar with the EOS Aura Spacecraft.

Document Overview

The MPS/Aura simulator is built with SIMSS components. The ETSAura, ETSGS, ETSXtract and ETS1553Bus modules were customised for the EOS Aura mission. All other modules are from the standard SIMSS toolkit. This document has been assembled with SIMSS and mission specific chapters.

The first chapter provides a description of the SIMSS client and server project user interface. It describes the core operations of installing, building, configuring, saving, restoring and running a project. All users should read this chapter to become familiar with SIMSS terminology and the common project user interface that will be referenced in later chapters.

Next are chapters that describe each of the modules that have been delivered as components of MPS/Aura. For each module, there is information on how to link it to other SIMSS modules, how to configure it, and how to control its run-time operations.

The last chapter provides a list of related Acronyms.

TABLE OF CONTENTS

How to Run a SIMSS-Based Simulator 12
SIMSS-1.0 TERMINOLOGY 12
SIMSS-2.0 INSTALLING A NEW RELEASE 13
SIMSS-2.1 INSTALLING THE CLIENT 13
SIMSS-2.2 INSTALLING THE SERVER 13
SIMSS-3.0 CONFIGURATION FILES 14
SIMSS-3.1 CLIENT/PROPERTIES/SERVERS.TXT 14
SIMSS-3.2 SERVER/PROPERTIES/PROPERTY.TXT 14
SIMSS-4.0 RUNNING A SIMSS APPLICATION 14
SIMSS-4.1 APPLICATION STARTUP SEQUENCE 15
SIMSS-5.0 SIMSS APPLICATION DISPLAYS 17
SIMSS-5.1 SIMSS CLIENT MAIN WINDOW 17
SIMSS-5.1.1 File Menu 17
SIMSS-5.1.2 Edit Menu 18
SIMSS-5.1.3 Project Menu 18
SIMSS-5.1.3.1 Add Project 18
SIMSS-5.1.3.2 Restore Project From 18
SIMSS-5.1.4 Help Menu 19
SIMSS-5.2 PROJECT WINDOW 19
SIMSS-5.2.1 Project System Menu 19
SIMSS-5.2.1.1 Connect to Server 20
SIMSS-5.2.1.2 Disconnect from Server 21
SIMSS-5.2.1.3 Save Project 21
SIMSS-5.2.1.4 Event Message Filtering 21
SIMSS-5.2.1.5 Show Event Log 22
SIMSS-5.2.1.6 Rename Project 22
SIMSS-5.2.2 Project Module Menu 23
SIMSS-5.2.2.1 Add Module 23
SIMSS-5.2.2.2 Create Links 23
SIMSS-5.2.2.3 Design Mode 24
SIMSS-5.2.2.4 Module Pop-Up Menu 25
SIMSS-5.2.2.4.1 Configure Module 25
SIMSS-5.2.2.4.2 Run-time Module 25
SIMSS-5.2.2.4.3 Remove Module 25
SIMSS-5.2.2.4.4 Rename Module 25
SIMSS-5.2.2.4.5 About Module 26
SIMSS-5.2.3 Project Run Menu 26
SIMSS-5.2.3.1 Locking a Project 26
SIMSS-5.2.3.2 Unlocking a Project 26
SIMSS-5.2.3.3 Running a Project 26
SIMSS-5.2.3.4 Stopping a Project 26
SIMSS-5.2.4 Project Directive Region 26
SIMSS-5.2.4.1 Directives/Scenario Files 27
SIMSS-5.2.4.2 Set Directive 27
SIMSS-5.2.4.2.1 Set directive using expressions 28
SIMSS-5.2.4.2.2 Set directive using arithmetic expressions 28
SIMSS-5.2.4.2.3 Set directive using arithmetic and Boolean functions 28
SIMSS-5.2.4.2.4 Set directive using mnemonics in expressions 29
SIMSS-5.2.4.3 Setbuffer Directive 29
SIMSS-5.2.4.4 Get Directive 30

SIMSS-5.2.4.5 Getbuffer Directive 30
SIMSS-5.2.4.6 Create Directive 31
SIMSS-5.2.4.7 Delete Directive 31
SIMSS-5.3 PROJECT EVENT MESSAGE WINDOW 31
Spacecraft Simulation Module for Aura (ETSAura) 32
ETSAURA-1.0 OVERVIEW 32
ETSAURA-2.0 INPUTS 32
ETSAURA-3.0 OUTPUTS 33
ETSAURA-4.0 CONTAINER ITEMS 33
ETSAURA-4.1 TELEMETRY CONTAINER ITEMS 33
ETSAura-4.1.1 Telemetry Status and Control Container Items 33
ETSAura-4.1.2 Telemetry Point Container Items 34
ETSAura-4.1.3 Telemetry Packet Container Items 34
ETSAura-4.1.4 Segmented Telemetry Container Items 35
ETSAura-4.1.5 Telemetry Dump Packet Container Items 36
ETSAura-4.1.6 Telemetry VCDU Container Items 36
ETSAura-4.1.7 External Telemetry Container Items 37
ETSAura-4.1.7.1 External APID Telemetry Container Items 37
ETSAura-4.1.7.2 External CLCW Telemetry Container Items 38
ETSAura-4.1.7.3 External Telemetry Status Container Items 39
ETSAURA-4.2 COMMAND CONTAINER ITEMS 40
ETSAura-4.2.1 Mission Specific Container Items 40
ETSAura-4.2.2 Command Validation/Debug Flags 40
ETSAura-4.2.3 Command Link Control Word Fields 41
ETSAura-4.2.4 Command Container Buffers 42
ETSAura-4.2.5 Command Triggering Scenarios Container Items 42
ETSAura-4.2.6 Command Submnemonic Container Items 42
ETSAura-4.2.7 Command Counter Container Items 42
ETSAURA-4.3 DATABASE CONTAINER ITEMS 44
ETSAURA-5.0 DISPLAYS 44
ETSAURA-5.1 CONFIGURATION MENU 44
ETSAura-5.1.1 Load Database 45
ETSAura-5.1.1.1 Maintaining Database Versions 45
ETSAura-5.1.1.2 Reloading Database Versions 46
ETSAura-5.1.2 Select Simulation Mode 46
ETSAura-5.1.3 Select Command Mode 47
ETSAURA-5.2 RUN-TIME MENU 47
ETSAura-5.2.1 Main Display 47
ETSAura-5.2.1.1 Telemetry Menus 48
ETSAura-5.2.1.1.1 Modify Packet Display 48
ETSAura-5.2.1.1.2 Telemetry Packet Display 49
ETSAura-5.2.1.1.3 Telemetry Status Display 50
ETSAura-5.2.1.1.4 Control Packet Display 50
ETSAura-5.2.1.1.5 Initiate Dump Packets 51
ETSAura-5.2.1.1.6 Display/Set Container Items 51
ETSAura-5.2.1.1.7 Display Container Buffer 52
ETSAura-5.2.1.1.8 Modify CCSDS Unsegmented Time Code 54
ETSAura-5.2.1.1.9 APID Status Display 54
ETSAura-5.2.1.1.10 External Packet Summary Display 55
ETSAura-5.2.1.2 Command Menu 55
ETSAura-5.2.1.2.1 Display Command Status 56
ETSAura-5.2.1.2.2 Modify Validation Criteria 56
ETSAura-5.2.1.2.3 Display Spacecraft Packet 58
ETSAura-5.2.1.2.4 Display Instrument Packet 59
ETSAura-5.2.1.2.5 Override CLCWs 59

ETSAura-5.2.1.2.6 Display Spacecraft Load 60
ETSAura-5.2.1.3 Time 61
ETSAura-5.2.1.3.1 Modify Times Display 61
ETSAURA-5.3 REMOVE 62
ETSAURA-5.4 ABOUT 62
ETSAURA-6.0 SPECIAL OPERATING INSTRUCTIONS 62
ETSAURA-6.1 SERIAL MODE OPERATION 62
ETSAURA-6.2 TRIGGERING SCENARIOS WITH COMMANDS 62
ETSAura-6.2.1 Using the Default Scenario Directory *62*
ETSAura-6.2.2 Using an Alternate Scenario Directory *63*
ETSAura-6.2.3 Update/Replace Command Trigger Definitions *63*
ETSAura-6.2.4 Using Full Path Scenario Filenames *64*
ETSAura-6.2.5 Using Relative Path Scenario Filenames *64*
ETSAURA-6.3 1553 BUS INTERFACE OPERATION 64
ETSAURA-6.4 SYNCHRONIZATION WITH EXTERNAL PACKET TIME 65
ETS Ground Station Module (ETSGS) 66
ETSGS-1.0 OVERVIEW 66
ETSGS-2.0 INPUTS 66
ETSGS-3.0 OUTPUTS 66
ETSGS-4.0 CONTAINER ITEMS 66
ETSGS-4.1 COUNTERS 66
ETSGS-4.2 BUFFERS 67
ETSGS-4.3 GROUND MESSAGE HEADER (GMH) FIELDS 67
ETSGS-4.4 EDOS DATA UNIT (EDU) HEADER FIELDS 67
ETSGS-5.0 DISPLAYS 68
ETSGS-5.1 CONFIGURATION 68
ETSGS-5.2 RUN-TIME 68
ETSGS-5.2.1 Show Status *69*
ETSGS-5.2.2 Show Command Packet *69*
ETSGS-5.2.3 Show Telemetry Channel 1 Packet *70*
ETSGS-5.2.4 Modify Telemetry Header *71*
ETSGS-5.2.5 Modify CLCW Header *72*
ETSGS-5.2.6 Set/Display GMT *73*
ETSGS-5.3 ABOUT 73
ETSGS-6.0 SPECIAL OPERATING INSTRUCTIONS 73
EOS Packet Extractor (EOSXtract) Module 74
EOSXTRACT-1.0 OVERVIEW 74
EOSXTRACT-2.0 INPUTS 74
EOSXTRACT-3.0 OUTPUTS 74
EOSXTRACT-4.0 CONTAINER ITEMS 74
EOSXTRACT-4.1 CONFIGURABLE PARAMETERS 75
EOSXTRACT-4.2 PROCESSING BUFFERS 75
EOSXTRACT-4.3 PROCESSING COUNTERS 75
EOSXTRACT-4.4 DEBUG FLAGS 75
EOSXTRACT-5.0 DISPLAYS 75
EOSXTRACT-5.1 CONFIGURATION DISPLAY 76
EOSXTRACT-5.2 RUN-TIME MENU 76
EOSXtract-5.2.1 Show Status Display *76*
EOSXtract-5.2.2 Dump CADU Buffer Display *77*
EOSXtract-5.2.3 Dump APID Buffer Display *77*
EOSXtract-5.2.4 Dump CLCW Buffer Display *77*
EOSXtract-5.2.5 Resume *77*
EOSXtract-5.2.6 Pause *78*
EOSXTRACT-5.3 REMOVE 78
EOSXTRACT-5.4 ABOUT 78

EOSXTRACT-6.0 SPECIAL OPERATING INSTRUCTIONS 78
1553 Bus Interface (ETS1553Bus) Module 79
ETS1553BUS-1.0 OVERVIEW 79
ETS1553BUS-2.0 INPUTS 79
ETS1553BUS-3.0 OUTPUTS 79
ETS1553BUS-4.0 CONTAINER ITEMS 79
ETS1553BUS-4.1 CONFIGURABLE PARAMETERS 79
ETS1553BUS-4.2 PROCESSING COUNTERS 80
ETS1553BUS-5.0 DISPLAYS 80
ETS1553BUS-5.1 CONFIGURATION DISPLAYS 81
ETS1553BUS-5.2 RUN-TIME MENU 81
ETS1553Bus-5.2.1 Status Display *82*
ETS1553Bus-5.2.2 Resume *82*
ETS1553Bus-5.2.3 Pause *82*
ETS1553BUS-5.3 REMOVE 83
ETS1553BUS-5.4 ABOUT 83
ETS1553BUS-6.0 SPECIAL OPERATING INSTRUCTIONS 83
Frame Monitor Module 84
FRAMEMONITOR-1.0 OVERVIEW 84
FRAMEMONITOR-2.0 INPUT 84
FRAMEMONITOR-3.0 OUTPUT 84
FRAMEMONITOR-4.0 CONTAINER ITEMS 84
FRAMEMONITOR-5.0 DISPLAYS 85
FRAMEMONITOR-5.1 CONFIGURATION 85
FRAMEMONITOR-5.2 RUN-TIME 87
FrameMonitor-5.2.1 Display Status *87*
FrameMonitor-5.2.2 Display Dump Menu *88*
FrameMonitor-5.2.3 Resume *90*
FrameMonitor-5.2.4 Pause *90*
FRAMEMONITOR-5.3 REMOVE 90
FRAMEMONITOR-5.4 ABOUT 90
FRAMEMONITOR-6.0 SPECIAL OPERATING INSTRUCTIONS 90
Internet Protocol (InputIP and OutputIP) Modules 91
INTERNETPROTOCOL-1.0 OVERVIEW 91
INTERNETPROTOCOL-2.0 INPUTS 91
INTERNETPROTOCOL-3.0 OUTPUTS 91
INTERNETPROTOCOL-4.0 CONTAINER ITEMS 91
INTERNETPROTOCOL-5.0 DISPLAYS 91
INTERNETPROTOCOL-5.1 CONFIGURATION MENU 92
InternetProtocol-5.1.1 IP Type Field *92*
InternetProtocol-5.1.2 Receiver (Sender) Address Box *92*
InternetProtocol-5.1.3 Source (Destination) Address Box *93*
InternetProtocol-5.1.4 Variable Length Input (Output) Check Box *94*
InternetProtocol-5.1.5 Frame Size Field *94*
INTERNETPROTOCOL-5.2 RUN-TIME MENU 94
InternetProtocol-5.2.1 Show Status *94*
InternetProtocol-5.2.2 Show Raw Packet *95*
InternetProtocol-5.2.3 Pause *97*
InternetProtocol-5.2.4 Resume *97*
InternetProtocol-5.2.5 Stop *97*
InternetProtocol-5.2.6 Restart *97*
INTERNETPROTOCOL-5.3 ABOUT 97
INTERNETPROTOCOL-6.0 SPECIAL OPERATING INSTRUCTIONS 97
Log Module 98
LOG-1.0 OVERVIEW 98

LOG-2.0 INPUTS 98
LOG-3.0 OUTPUTS 98
LOG-4.0 CONTAINER ITEMS 98
LOG-5.0 DISPLAYS 99
LOG-5.1 CONFIGURATION 100
Log-5.1.1 Logging Enabled Flag 101
Log-5.1.2 Log Directory 102
Log-5.1.3 Generated Log File Name Fields 102
Log-5.1.3.1 Project Identification Field 102
Log-5.1.3.2 Module Identification Field 103
Log-5.1.3.3 Description Fields 103
Log-5.1.3.4 Timestamp Field 103
Log-5.1.3.5 Log Format Specification Field 103
Log 5.1.3.6 File Type Extension Fields 104
Log 5.1.4 Log With Header Format 104
Log-5.1.5 Variable Length Format 104
Log-5.1.6 Size in bytes 104
Log 5.1.7 Maximum File Size (bytes) 104
Log 5.1.8 Maximum Full File Count 104
LOG-5.2 RUN-TIME 105
Log-5.2.1 Control and Status Display 105
Log 5.2.1.1 Reset Button 106
Log 5.2.1.2 Logging Enabled/Disabled Button 106
Log 5.2.1.3 Open New Log Button 106
Log 5.2.1.4 Format Button 106
Log-5.2.2 Pause/Resume 107
Log-5.2.3 Stop/Restart 107
LOG-5.3 ABOUT 108
LOG-6.0 SPECIAL OPERATING INSTRUCTIONS 108
LOG-6.1 OPTIONAL PROJECT DEFAULT LOG DIRECTORY 108
PACKET PROCESSOR MODULE 109
PACKET PROCESSOR-1.0 OVERVIEW 109
PACKET PROCESSOR-2.0 INPUTS 109
PACKET PROCESSOR-3.0 OUTPUTS 109
PACKET PROCESSOR-4.0 CONTAINER ITEMS 109
PACKET PROCESSOR-5.0 DISPLAYS 109
PACKET PROCESSOR -5.1 CONFIGURE 109
Packet Processor-5.1.1 VC Configuration Window 110
Packet Processor-5.1.2 Packet Configuration Window 110
Packet Processor-5.1.3 Select APID 111
PACKET PROCESSOR-5.2 RUN 111
Packet Processor-5.2.1 Show APID status 112
Packet Processor-5.2.2 Packet Status Window 113
Packet Processor-5.2.3 Configure 113
Scenario Module 114
SCENARIO-1.0 OVERVIEW 114
SCENARIO-2.0 INPUTS 114
SCENARIO-2.1 SERVER/PROPERTIES/PROPERTY.TXT 114
SCENARIO-2.2 MODULE TRIGGERED SCENARIOS 114
SCENARIO-2.3 SCENARIO FILE INPUT 114
Scenario-2.3.1 Scenario File Contents 114
Scenario-2.3.1.1 Specifying Modules to Receive Scenario Directives 116
Scenario-2.3.1.2 Specifying Conditional Expressions 116
Scenario-2.3.1.3 Simple Scenario Example 116
Scenario-2.3.1.4 IF Scenario Example 117

Scenario-2.3.1.5 While Scenario Example 117
SCENARIO-3.0 OUTPUTS 117
SCENARIO-4.0 CONTAINER ITEMS 117
SCENARIO-4.1 SCENARIO SCRIPT IDENTIFICATION NUMBERS 117
SCENARIO-4.2 REQUEST SCRIPT INFORMATION (SCRIPTSTATUS) 118
SCENARIO-4.3 TERMINATE SCENARIOS (KILLTHREAD) 118
SCENARIO-4.4 DEBUGGING AIDS 118
Scenario-4.4.1 Request Echoing of Directive Lines (EchoThread) *118*
Scenario-4.4.2 Request Additional Debug Messages (DebugThread) *118*
SCENARIO-4.5 DEFAULT SCENARIO DIRECTORY (SCENARIOPATH) 119
SCENARIO-5.0 DISPLAYS 119
SCENARIO-5.1 CONFIGURE 119
SCENARIO-5.2 RUN-TIME 119
Scenario-5.2.1 Scenario Control Display *119*
Scenario-5.2.2 Scenario File Browser *121*
SCENARIO-5.3 ABOUT 122
SCENARIO-6.0 SPECIAL OPERATING INSTRUCTIONS 122
SCENARIO-6.1 ADDITION OF DELAYS TO SCENARIO FILES 122
Serial Input Module 123
SERIALINPUT-1.0 OVERVIEW 123
SERIALINPUT-2.0 INPUT 123
SERIALINPUT-3.0 OUTPUT 123
SERIALINPUT-4.0 CONTAINER ITEMS 123
SERIALINPUT-5.0 DISPLAYS 123
SERIALINPUT-5.1 CONFIGURATION 123
SERIALINPUT-5.2 RUN-TIME 126
SerialInput-5.2.1 Display Status *126*
SerialInput-5.2.2 Display Dump *126*
SerialInput-5.2.3 Resume *127*
SerialInput-5.2.4 Pause *127*
SERIALINPUT-5.3 REMOVE 128
SERIALINPUT-5.4 ABOUT 128
SERIALINPUT-6.0 SPECIAL OPERATING INSTRUCTIONS 128
SERIALINPUT-6.1 "MAX CMD BLOCK" AND "CMD SIZE" CONFIGURATION 128
Serial Output Module 129
SERIALOUTPUT-1.0 OVERVIEW 129
SERIALOUTPUT-2.0 INPUT 129
SERIALOUTPUT-3.0 OUTPUT 129
SERIALOUTPUT-4.0 CONTAINER ITEMS 129
SERIALOUTPUT-5.0 DISPLAYS 129
SERIALOUTPUT-5.1 CONFIGURATION 130
SERIALOUTPUT-5.2 RUN-TIME 131
SerialOutput-5.2.1 Display Status *131*
SerialOutput-5.2.2 Display Dump *131*
SerialOutput-5.2.3 Resume *132*
SerialOutput-5.2.4 Pause *132*
SERIALOUTPUT-5.3 REMOVE 132
SERIALOUTPUT-5.4 ABOUT 133
SERIALOUTPUT-6.0 SPECIAL OPERATING INSTRUCTIONS 133
SERIALOUTPUT-6.1 OPERATION WITH INPUT IP MODULE 133
SERIALOUTPUT-6.2 SETTING FREQUENCY ON THE FLY 133
AVTEC Serial Input Module 135
AVTECSERIALINPUT-1.0 OVERVIEW 135
AVTECSERIALINPUT-2.0 INPUT 135
AVTECSERIALINPUT-3.0 OUTPUT 135

AVTECSERIALINPUT-4.0 CONTAINER ITEMS 135
AVTECSERIALINPUT-5.0 DISPLAYS 135
AVTECSERIALINPUT-5.1 CONFIGURATION 135
AVTECSERIALINPUT-5.2 RUN-TIME 137
AvtecSerialInput-5.2.1 Display Status 137
AvtecSerialInput-5.2.2 Display Dump Menu 137
AvtecSerialInput-5.2.2.1 Display Buffer Current Configuration 138
AvtecSerialInput-5.2.3 Resume 138
AvtecSerialInput-5.2.4 Pause 139
AVTECSERIALINPUT-5.3 REMOVE 139
AVTECSERIALINPUT-5.4 ABOUT 139
AVTECSERIALINPUT-6.0 SPECIAL OPERATING INSTRUCTIONS 139
AVTEC Serial Output Module 140
AVTECSERIALOUTPUT-1.0 OVERVIEW 140
AVTECSERIALOUTPUT-2.0 INPUT 140
AVTECSERIALOUTPUT-3.0 OUTPUT 140
AVTECSERIALOUTPUT-4.0 CONTAINER ITEMS 140
AVTECSERIALOUTPUT-5.0 DISPLAYS 140
AVTECSERIALOUTPUT-5.1 CONFIGURATION 140
AVTECSERIALOUTPUT-5.2 RUN-TIME 142
AvtecSerialOutput-5.2.1 Display Status 142
AvtecSerialOutput-5.2.2 Display Dump Menu 142
AvtecSerialOutput-5.2.2.1 Display Buffer Current Configuration 142
AvtecSerialOutput-5.2.3 Resume 143
AvtecSerialOutput-5.2.4 Pause 143
AVTECSERIALOUTPUT-5.3 REMOVE 143
AVTECSERIALOUTPUT-5.4 ABOUT 144
AVTECSERIALOUTPUT-6.0 SPECIAL OPERATING INSTRUCTIONS 144
Transmit File (TxFile) Module 145
TXFILE-1.0 OVERVIEW 145
TXFILE-2.0 INPUTS 145
TXFILE-3.0 OUTPUTS 145
TXFILE-4.0 CONTAINER ITEMS 145
TXFILE-5.0 DISPLAYS 146
TXFILE-5.1 CONFIGURATION DISPLAYS 146
TXFILE-5.2 RUN-TIME DISPLAYS 147
TxFile-5.2.1 Status Display 147
TxFile-5.2.2 Transmit Buffer Display 147
TxFile-5.2.3 Send From File 148
TxFile-5.2.3.1 File Browsing 150
TxFile-5.2.3.2 File Types 151
TxFile-5.2.3.3 Example 1 Manual Mode Transmission 152
TxFile-5.2.3.4 Example 2 Auto-Complete File Transmission 153
TxFile-5.2.3.5 Example 3 Auto-Blocks Transmission (Raw Fixed Data Files Only) 154
TxFile-5.2.3.6 Example 4 Use of Log File Timing (Log Files Only) 155
TxFile-5.2.3.7 Example 5 Serial Transmission Mode 156
TXFILE-5.3 ABOUT DISPLAY 157
TXFILE-6.0 SPECIAL OPERATING INSTRUCTIONS 157
VC Processor Module 158
VC PROCESSOR-1.0 OVERVIEW 158
VC PROCESSOR-2.0 INPUTS 158
VC PROCESSOR-3.0 OUTPUTS 158
VC PROCESSOR-4.0 CONTAINER ITEMS 158
VC PROCESSOR-5.0 DISPLAYS 159
VC PROCESSOR -5.1 CONFIGURE 159

VC Processor-5.1.1 Main Configuration Window *159*
VC Processor-5.1.2 Set Outputs *161*
VC Processor-5.1.3 Set Offsets *161*
VC PROCESSOR-5.2 RUN 162
VC Processor-5.2.1 Show Status *162*
VC Processor-5.2.2 Display VC *163*
VC Processor-5.2.3 ReConfigure *163*
ACRONYMS 164

How to Run a SIMSS-Based Simulator

The Scalable, Integrated, Multimission Support System (SIMSS) is a distributed, component-based, plug-and-play, client-server system useful for performing real-time operations, simulations and testing in support of NASA projects.

SIMSS-1.0 Terminology

SIMSS runs on one or more Windows NT or Windows 2000 workstations. It is designed to be user-configurable or to use predefined configurations for routine operations.

Client: The workstation on which the user interface runs. It also refers to the applications making up the user interface.

Server: The workstation on which the actual data processing is performed. It also refers to the applications making up the data processing component.

Module: A self-contained SIMSS component that receives, processes, or transmits data, or any combination of the three.

Channel: An interface or port through which a module receives or transmits data. A module may have zero to many channels.

Link: A directional connection between module channels. A link connects an output channel of one module with the input channel of another module.

Project: A collection of modules and links intended to perform a specific function such as spacecraft simulation, data quality monitoring, or data conversion.

Event Message: A time-tagged text message generated by any module to inform the operator. Event messages may report warning or error conditions as well as successful activities.

Directive: A text-based command line entry that operators submit to individual modules. The directive entry line window is below the configuration window in the main SIMSS Project display.

Container: A repository internal to each module that contains all of the vital data for that module. The container is used to exchange data between the client and server. Built-in functions of the container support save and restore operations.

Scenario: A file that contains directives. A scenario module may execute this file in order to send a timed repeatable sequence of directives to a linked module.

SIMSS-2.0 Installing A New Release

SIMSS-2.1 Installing the Client

The Java Runtime Engine (JRE) must be installed on the same PC as the SIMSS client. To determine whether the correct JRE version is already installed, open a DOS window and type ***java –version***. If the response indicates a version different from the one called for in the current delivery package, follow the delivery package instructions for installing the JRE. For the convenience of the user, the correct version of the JRE should be included in the root folder of the delivery CD.

Once the JRE is installed, the next step is to run **setup.exe** in the *\client* folder on the CD and follow the prompts. Upon completion of the client installation, a SIMSS Client icon will be installed on the desktop. The SIMSS Client icon will point to the executable directory for the client software. Below this directory are three additional directories: images, properties, and save.

Directory	Description
Images	Contains bitmapped image files needed by the client software.
Properties	Contains client configuration files.
Save	This directory is where client and server information is stored when projects are saved. The user may want to delete old files from this directory.

SIMSS-2.2 Installing the Server

To install the SIMSS server, run **setup.exe** in the *\server* folder on the CD and follow the prompts. Upon completion of the server installation, a SIMSS Server icon will be installed on the desktop. The SIMSS Server icon will point to the executable directory for the server software. Below this directory are five additional directories: elog, log, properties, save and scenario.

Directory	Description
Elog	Contains event message log files. The user may want to delete old event message log files from this directory to reclaim disk space.
Log	This directory is the default location for data log files generated by the generic Log module.
Properties	Contains the server property.txt configuration file.
Save	Unused.
Scenario	Contains delivered scenario files. The user may store additional scenario files here.

SIMSS-3.0 Configuration Files

SIMSS-3.1 Client/Properties/Servers.txt

Before running the SIMSS Client, please check that the correct IP address for running the SIMSS Server is in the *properties*/*servers.txt* file. This file must be present and must contain the *localhost* entry. If the SIMSS Client is to be operated remotely from the SIMSS Server then additional entries, giving the IP addresses of each SIMSS Server desired, must be added to the file.

The format for entries in this file is:

Servername, IP address

Example:
localhost,127.0.0.1

SIMSS-3.2 Server/Properties/Property.txt

There are several parameters that may be configured in the server. Most of them are for debugging purposes. These parameters are stored in the *properties*/*property.txt* file. This file is read in each time that the server is started. The values used from the property.txt file are written into the server text window. The property.txt file contains descriptions of the parameters and their normal settings. Use caution in making modifications to this file. Some parameters that may be of interest to users are listed below.

Property	**Description**
EventMsgLogSize	This is a maximum event message log size. When this many event messages have been logged, the oldest messages will be overwritten. When set to zero, there is no limit on the number of event messages that are saved to the log.
DelOldLog	This flag is for deleting the old event log file on startup. (0=no, 1=yes). This is only effective if the event log file name is the same for each execution. Use with DefaultEventLogSave.
DefaultEventLogSave	This flag specifies use of the same event log file name for each execution (0=use different names, 1=use same name).

SIMSS-4.0 Running a SIMSS Application

The SIMSS modular architecture can support a variety of applications by connecting generic and mission-specific modules in different combinations. The general steps for starting up any SIMSS project are summarized in the next section. Detailed examples of each of these steps are provided in SIMSS-5.0, which describes the graphical user interface (GUI).

Every SIMSS application must run within a project. Each project consists of interconnected executable software modules. Every software module may have configuration and run-time displays. Every module that accepts operator directives can execute the directives described later in this chapter. Every module that executes directives may also be configured to accept directives from a scenario file. Every module that generates event messages for the operator sends them to the event log region of its project window.

Other chapters in this system user's guide are devoted to the various generic and mission-specific SIMSS modules. Please refer to these chapters for detailed information on module-specific configuration and run-time operations.

SIMSS-4.1 Application Startup Sequence

This paragraph provides an overview of the steps necessary to create or restore a project for operations. The subsections under paragraph SIMSS-5.0 provide more details about the steps where options are available.

Startup Steps	**Description**
1) Start the Server	Double-click the Server icon on the PC desktop. When the Server is started, a text window is created.
2) Start the Client	Double-click the SIMSS Client icon on the desktop. When the Client is started, a Client text window is created and then the graphical user interface is started with the SIMSS Client main window.
3) Add (or Restore) a Project	Click on the SIMSS Client main window's Project menu and select "Add Project". Alternatively, a "Restore From" selection may be used to load a previously saved project. A project window is created within the main window and the project event message window is created. If a previously saved project is restored, skip to step 9.
4) Connect Project to Server	In the project window, click the System menu and select the "Connect" option. A list of available servers will be shown. Select a server for the connection. When you connect to the server, the message in the lower left-hand corner of the project changes from "Status" to "Project Loaded."
5) Add Modules	In the project window, click the Module menu and select "Add Module". From the Module Selection window, click on the module to be added and the OK button. Click again in the project window for placement of the module's icon. Repeat this step until all desired modules have been added.

6) Create Links Note: Refer to the module specific chapters for input and output channel data descriptions in order to link the output channels to the appropriate input channels!	In the project window, click the Module menu and select "Create Links". Click on the sending module border then move the cursor to the receiving module border and click again. If either module has multiple links, a popup window will appear for specification of the exact link numbers. Repeat this step to create all desired links. To cancel link creation mode, click on the Module menu and select "Design".
7) Configure Modules	Click in the center of the module and select "Configure" from the popup menu. Perform module-specific configurations. Repeat for as many modules as needed.
8) Save the Project (optional)	Click the project window's System menu and select the "Save Project" option. Specify a name for the project save file and apply it from the save or browse window.
9) Lock the Project	In the project window, click the Run menu and select "Lock".
10) Run the Project	In the project window, click the Run menu and select "Run". All of the modules in the project will be started. Click in the center of any module to access its run-time options and displays.

SIMSS-5.0 SIMSS Application Displays

This section describes the graphical user interface common to all SIMSS applications. Every application has graphical client and project windows. Every module that accepts operator directives can execute the directives described later in this SIMSS chapter. Every module that generates event messages sends them to the project event message window.

SIMSS-5.1 SIMSS Client Main Window

When the SIMSS Client is started, the main window appears.

SIMSS-5.1.1 File Menu

The File menu contains an "Exit" option.

SIMSS-5.1.2 Edit Menu

The Edit menu contains a "Look and Feel" option. When this option is chosen, the operator may select from "System", "Metal", "Motif", or "Windows". The default setting is "Metal".

SIMSS-5.1.3 Project Menu

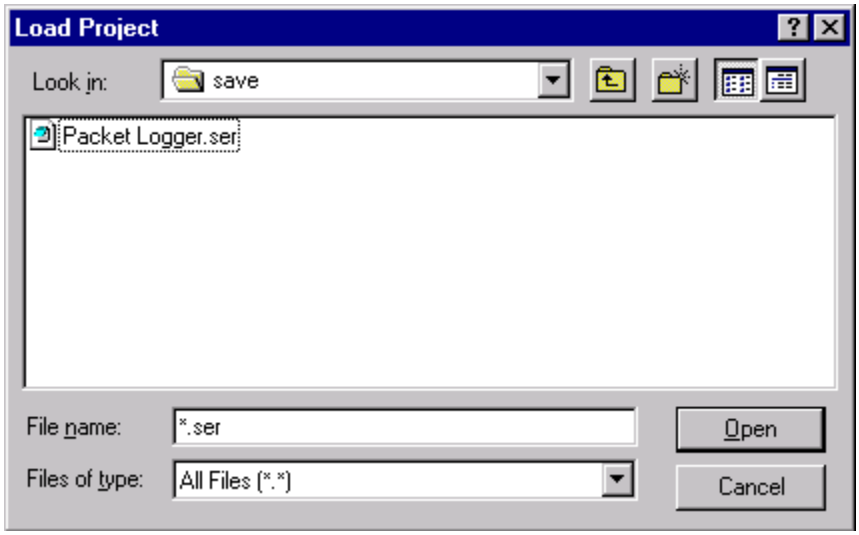
The Project menu contains two selections; "Add Project" and "Restore Project From .

SIMSS-5.1.3.1 Add Project

To create a new project, select the "Add Project" item from the Project menu. This will add a project window within the SIMSS client window and open the separate project event message window.

SIMSS-5.1.3.2 Restore Project From

To restore an existing project, select the "Restore Project From" item from the Project menu. A file selection screen like the following will appear.



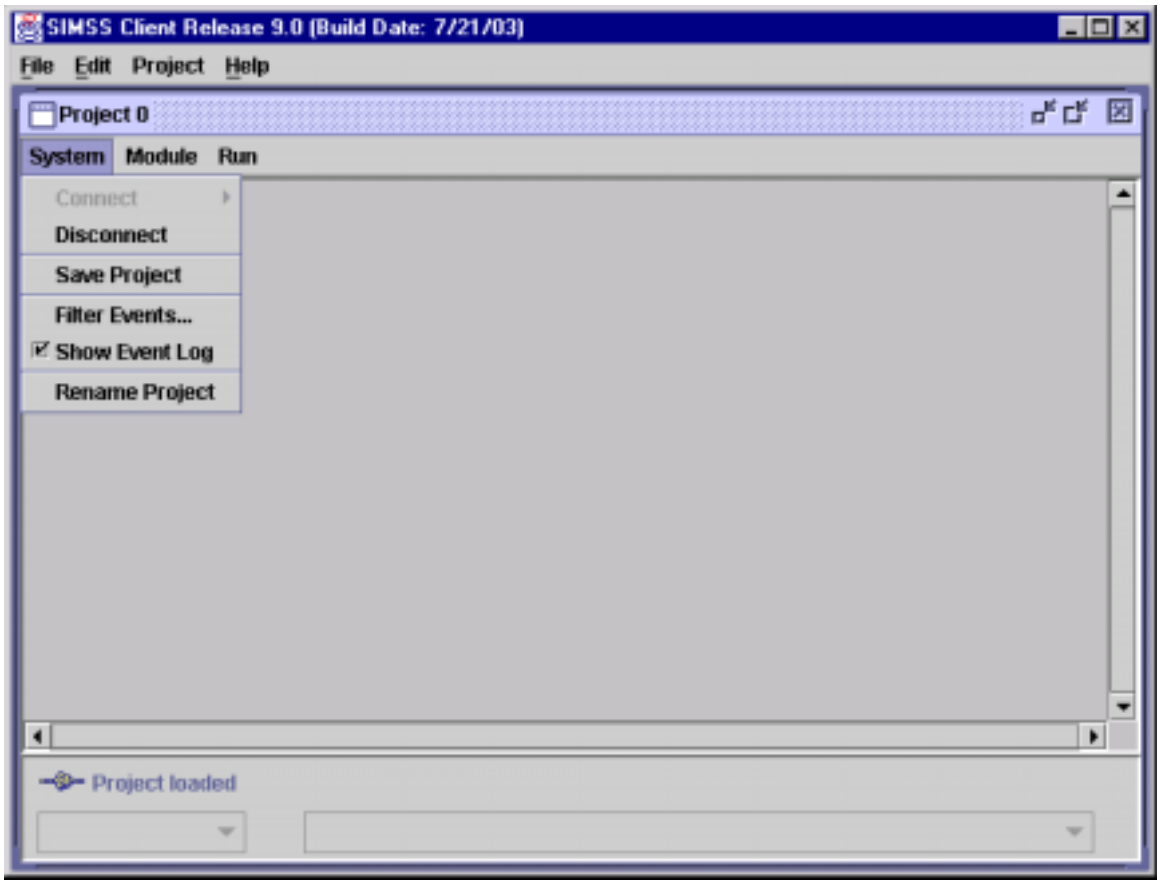
The system will search the save directory by default. Another directory may be selected by clicking the folder selection down-arrow and navigating to the desired directory. Select a file and click the **Open** button. Click on the **Cancel** button to dismiss the display without restoring a project.

SIMSS-5.1.4 Help Menu

The Help menu contains an "About" option that provides client version information.

SIMSS-5.2 Project Window

When a project is added (or restored) its project window is displayed and its Project Event Message window (see SIMSS-5.3) is displayed.



The project window contains a System menu, a Module menu and a Run menu. At the bottom left of the project window is a status field and a drop-down module selection menu . At the bottom right is the directive region of the screen.

SIMSS-5.2.1 Project System Menu

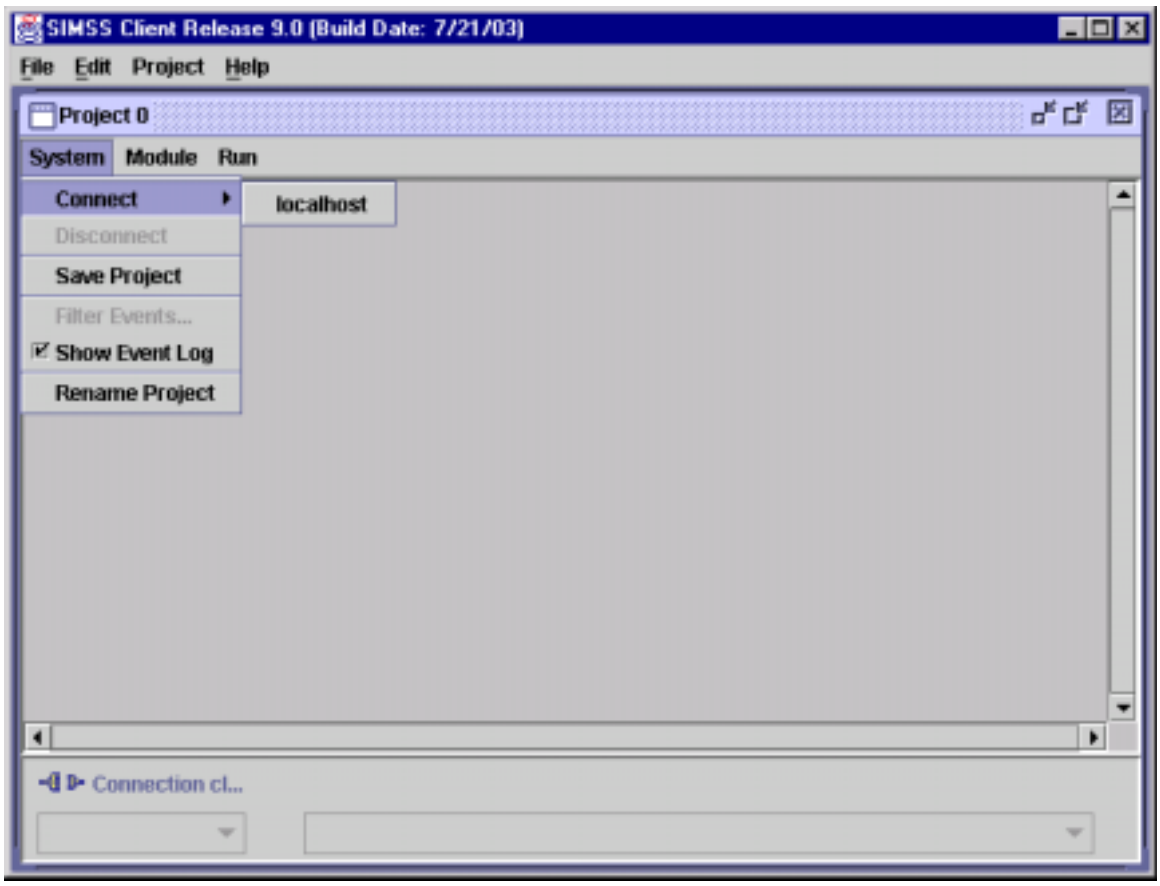
The System menu on the Project window contains the following choices.

System Menu Item	Description
Connect	Connects the project to a server
Disconnect	Disconnects the project from a server

Save Project	Saves project design to file on local PC
Filter Events…	Change event message filtering
Show Event Log	When this option is enabled, the project event message log window is displayed.
Rename Project	Allows user to rename project.

SIMSS-5.2.1.1 Connect to Server

The client must be connected to a server in order to function. To make a connection to a server, select the "Connect" option from the System menu. A menu of available servers will be displayed. Click to select a server for the connection. In the following picture, only one server was available.



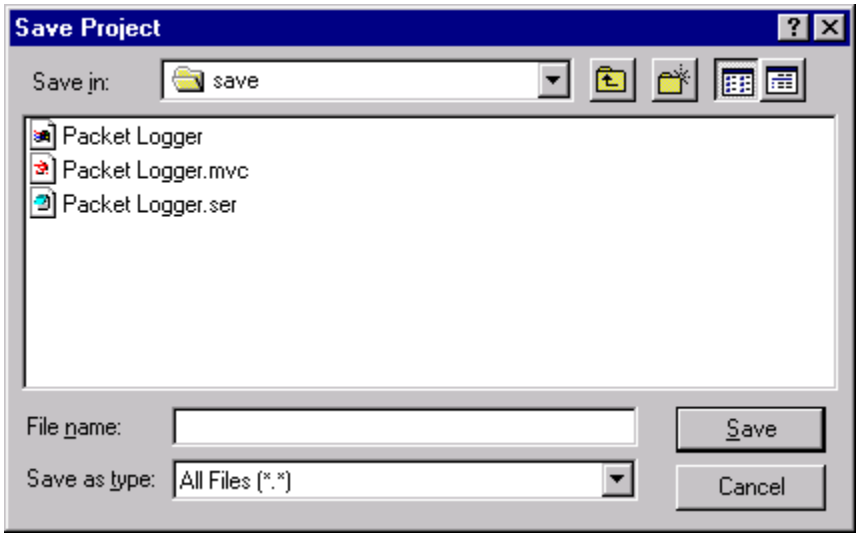
After the client has been connected to a server, the "Disconnect" option will be enabled on the System menu and the "Connect" option will be disabled.

SIMSS-5.2.1.2 Disconnect from Server

To disconnect the client from the server, select the "Disconnect" option from the System menu. After the client has disconnected from the server, the "Disconnect" option will be disabled on the System menu and the "Connect" option will be enabled.

SIMSS-5.2.1.3 Save Project

This option allows the user to store a created and configured project to a file for later restoration and use. The following screen will appear.



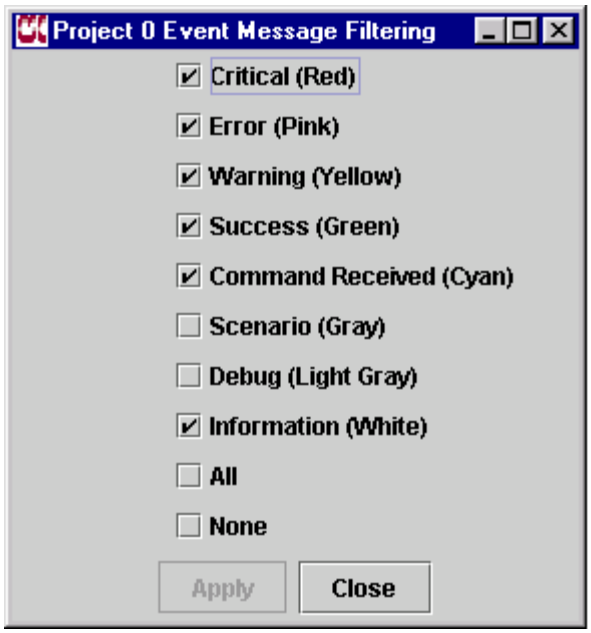
Enter the filename for storage of the current project and click the **Save** button to have project information written. Two files containing Client project information and one file containing Server project information will be created in the Client \save folder.

The Client \save folder is the default storage location. However, the project may be saved elsewhere by clicking on the folder selection down-arrow and navigating to the desired directory.

Clicking **Cancel** will close this window without performing a save.

SIMSS-5.2.1.4 Event Message Filtering

When "Filter Events…" is selected, the Event Message Filtering screen appears. This image shows the default filtering scheme. Only event messages with the selected colors will be displayed in the event message window.



Filtering only affects the display of event messages in the Project Event Message Window. All event messages will continue to be generated and written into the event message log file.

To disable the display of any of these event message types, clear its checkbox.

To enable the display of any of these event message types, check its checkbox.

Use the **Apply** button for changes to take effect.

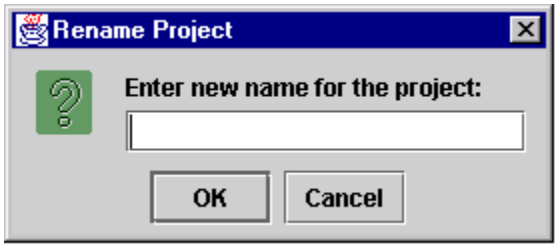
Use the **Close** button to dismiss this screen.

SIMSS-5.2.1.5 Show Event Log

Each project has an event message log that contains color-coded, time-tagged event messages received from its modules. When the Show Event Log box is checked the event log is displayed as a separate scrolling window. When this box is cleared, the event window is closed. Closing the event window does not affect the writing of event messages to the Event Message Log file.

SIMSS-5.2.1.6 Rename Project

When the user selects the "Rename Project" option from the Project menu, the following popup appears. If a new project name is entered, the new name appears on the title line for all project windows. The new name is saved and restored with the project.



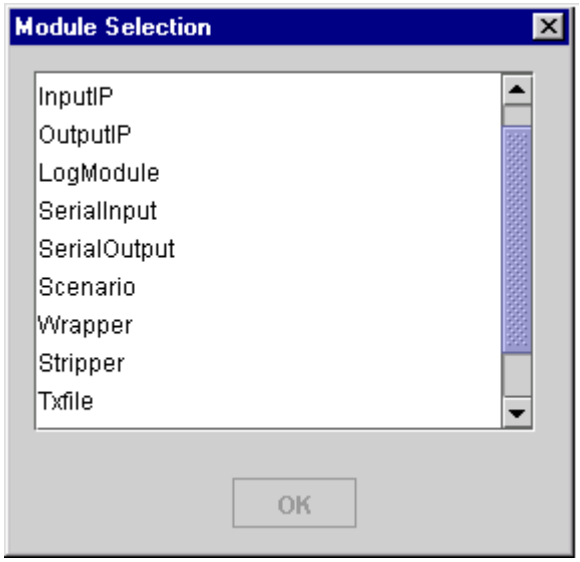
SIMSS-5.2.2 Project Module Menu

The Module menu on the Project window contains the following choices.

Module Menu Item	Description
Add Module	Add a module to the project's design diagram
Create Links	Create a link between modules
Design	Allows editing of the project design diagram

SIMSS-5.2.2.1 Add Module

To add a module to the project, select the "Add Module" option from the Module menu.



A list of the modules available from the server will be displayed. Click on the module's name to select a module and then click the **OK** button, or double-click the module name. The module selection box will disappear.

Position the cursor within the project window where the top left corner of the module's symbol should be drawn and click. A rectangle representing the module will be drawn at that location. The rectangle will have a wide cyan border while in Design mode.

If a module is added that accepts operator directives, the directive region of the project window becomes accessible.

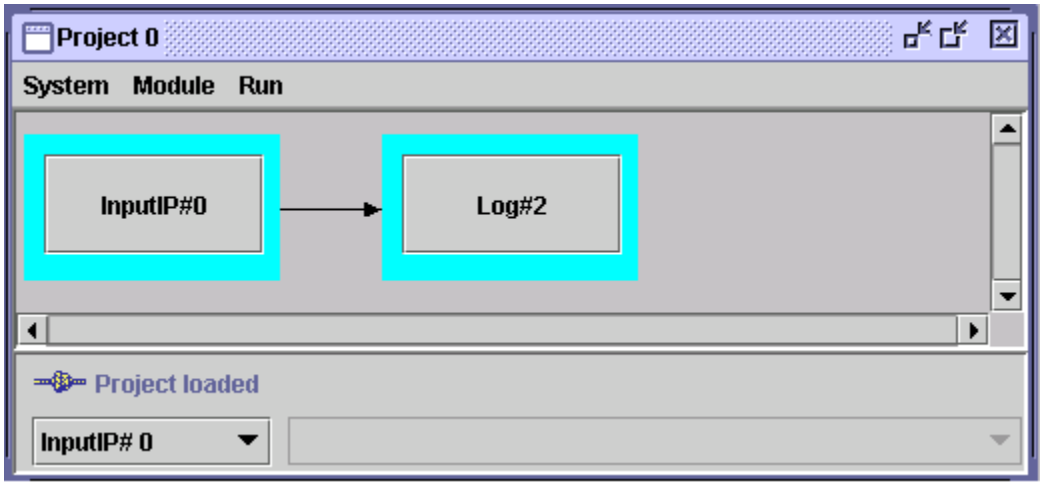
Repeat the entire sequence given in this paragraph for every module to be added to the project. If a module is added in error, click on the interior of the module and select the "Remove" option from the pop-up menu.

SIMSS-5.2.2.2 Create Links

Links are used to connect the input and output channels between modules for a given project. In the example shown below, the output channel of an Input IP module is linked to the input channel of a Log module.

To create a link,

- Select the "Create Links" option from the Module menu,
- Position the crosshair cursor on the highlighted edge of the source module and click,
- Move the cursor to the destination module's edge and click again.



When multiple links are defined for a module, the user will be prompted to choose a channel number for the source and destination links as appropriate. Refer to module-specific information for the number of input and output channels a module has and how they should be configured.

The cursor remains in Create Links mode to allow for the creation of additional links. The cursor appears as a crosshair symbol until a different menu option is selected.

If multiple links are created between two modules, additional clicks may be done on the project background to anchor the link lines apart for visibility. To avoid having the link line go through intervening modules or overlay another link, click on the sending module, move the cursor to a convenient point on the Project background and click once, then move the cursor to the receiving module and click again.

If a mistake is made when creating a link, or to remove an undesired link, click on the Module menu and select "Design". Select the desired link by clicking on it. Click "OK" in the pop-up box, then press the Delete key to remove the link. Finally, click the Module menu and select "Create Links" to return to link creation mode.

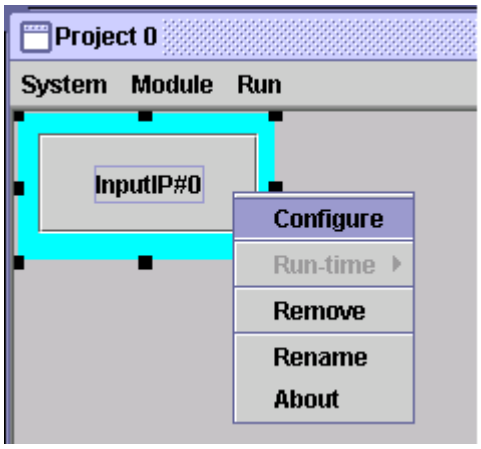
SIMSS-5.2.2.3 Design Mode

After the "Create Links" option has been selected, the cursor remains as a crosshair symbol and the system remains in link creation mode. To cancel this mode, click the "Design" option from the Module menu. The cursor is changed back into a pointer symbol and may be used to select modules or links for modification. While in this mode the operator may click on a link line to review the channel numbers of the modules it connects. A popup window provides the source and destination channel numbers. While a link line is selected, any of its background anchors may be moved by clicking and dragging.

SIMSS-5.2.2.4 Module Pop-Up Menu

Clicking in the center of a module activates a pop-up menu with the following choices. Not all choices will be available for all modules.

Module Pop-Up Menu	Description
Configure	Provides access to module-specific configuration options.
Run-time	Provides access to module-specific run-time options.
Remove	Removes the module from the project.
Rename	Renames the module
About	Provides information about the module.



SIMSS-5.2.2.4.1 Configure Module

The actions initiated by the "Configure" option are specific to each module's functionality and are fully described in the module chapters of this document.

SIMSS-5.2.2.4.2 Run-time Module

The actions initiated by the "Run-time" option are specific to each module's functionality and are fully described in the module chapters of this document.

SIMSS-5.2.2.4.3 Remove Module

The "Remove" option deletes the module from the project.

SIMSS-5.2.2.4.4 Rename Module

The "Rename" option allows the user to provide a different name for a module. The name must be unique within the project. The new name is saved and restored with the project.

SIMSS-5.2.2.4.5 About Module

The "About" option provides generic information on a module. Of interest during project configuration are the number of input and output channels.

SIMSS-5.2.3 Project Run Menu

The Run menu on the Project window contains the following choices.

Run Menu Item	Description
Lock/Unlock	Locks/Unlocks the project's design diagram
Run/Stop	Starts/Stops the project's execution

SIMSS-5.2.3.1 Locking a Project

Prior to running a project, the user must lock its design by clicking the "Lock" option from the Run menu. After the design is locked, the "Lock" option is replaced by the "Unlock" option and the "Run" option becomes available on the Run menu.

SIMSS-5.2.3.2 Unlocking a Project

Click the "Unlock" option from the Run menu to allow modification to the project's design diagram. The "Unlock" option is then replaced on the Run menu with a "Lock" option.

SIMSS-5.2.3.3 Running a Project

Click the "Run" option from the Run menu to start running all of the modules of the project. The borders of the modules will change to green to indicate run mode. The "Run" option is then replaced with a "Stop" option.

Click on specific modules to get their pop-up menus. See paragraph SIMSS-5.2.2.4 for an example. Click on the Run-time option for module-specific displays or actions. Please refer to the module-specific chapters in this system user's guide for information on each module's configuration and run-time displays.

SIMSS-5.2.3.4 Stopping a Project

Click the "Stop" option from the Run menu to stop execution of the modules of the project. The borders of the modules will change to red to indicate the stopped condition. The "Stop" option is then replaced with a "Run" option on the Run menu.

SIMSS-5.2.4 Project Directive Region

If any module in a project accepts operator directives, the directive region in the lower right side of the project window frame is activated. The directive region has a module indicator button and a data entry field. When the module indicator button is clicked, a list

of the modules accepting directives is shown. Click to select the module to execute a directive. The button is labeled with the currently selected module. In the example below, the Input IP module is selected.

Click in the data entry field to the right of the module button to start entering a directive. Alternatively, click on the down arrow to the right of the field to view up to ten previously entered directives. A previous entry may be selected, optionally edited, and submitted. Use the keyboard's Enter key to send the directive to the module. Directives are available to any module that accepts directives and has container variables.

SIMSS-5.2.4.1 Directives/Scenario Files

This section describes all of the directives currently defined for all SIMSS modules that accept operator directives. These directives may be placed in text files to be used as scenario files. Any module that processes directives can also receive directive lines read from a scenario file by the Scenario module. In addition to the directives described in this section, there are directives that are executed only by the Scenario module. These scenario directives control the timing and conditional execution of scenario directive lines. Please see the chapter on the Scenario module for more information on creating and executing scenario files. Refer to the container descriptions in each module's chapter for the names (mnemonics) of container items for each module.

SIMSS-5.2.4.2 Set Directive

The **set** directive may be used to change the value of any modifiable variable in the container for the indicated module. The format of the **set** directive is

set name value

where *name* is a container variable name and *value* is either a decimal, octal, or hexadecimal number. Octal numbers are identified by a leading zero. Hex numbers are identified by a leading 0x. Negative numbers may be entered as signed decimal integers in the usual manner, provided that the container variable chosen accepts negative numbers. Example: Telemetry points that are 2's complement integers.

The names of variables are provided in the container description for each module. If the variable is successfully set, an event message of the form

"module: *name* set to *value*"

informs the operator of the change. If the variable name is not recognized or the variable cannot be modified, an event message of the type

"module: *name* invalid, read-only or not found"

will inform the operator. If unsuccessful, verify that the module indicator is correct, that the variable name spelling is correct and that the variable is not defined as read-only.

SIMSS-5.2.4.2.1 Set directive using expressions

More complicated assignments of values to container items are possible. Telemetry mnemonics (or any other type of container item) may be set to values derived from any combination of arithmetic expressions, arithmetic functions, Boolean functions, and other container items. These directives may be used to create sophisticated scenario files. The length of an individual set directive is limited to 100 characters. Intermediate variables may be created to allow writing of expressions that would otherwise be too long. In order to use any of the following expressions, specification of an equal sign in one of the following forms is required (=, +=, -=, *=, /=, ^=) after the mnemonic name and before the expression. The following sections provide syntax information and examples of set directives using expressions.

SIMSS-5.2.4.2.2 Set directive using arithmetic expressions

A container item may be set to an arithmetic expression using the symbols for addition (+), subtraction (-), division (/), multiplication (*) and raise to a power (^ or **). Parentheses are recommended to specify the precedence order.

set mnemonic = 2 * (7^3)

SIMSS-5.2.4.2.3 Set directive using arithmetic and Boolean functions

A container item may also be set to an arithmetic expression using functions from the following table. The following rules apply:

- The function names must be in lower case.
- The parentheses around the function argument are required syntax.
- Constant values must be in decimal.
- The Boolean functions perform a bit-wise AND, OR, etc.
- A Boolean function argument that is a mnemonic must be in parenthesis.

Function Description	**Set directive example**
Arcsine (x)	Set mnemonic = asin (x)
Arccosine (x)	Set mnemonic = acos (x)
Arctangent (x)	Set mnemonic = atan (x)
Degrees to radians (x)	Set mnemonic = dtor (x)
Radians to degrees (x)	Set mnemonic = rtod (x)
Square root (x)	Set mnemonic = sqrt (x)
Log (x)	Set mnemonic = ln (x)
Exponent (x)	Set mnemonic = exp (x)
Tangent (x)	Set mnemonic = tan (x)
Sine (x)	Set mnemonic = sin (x)
Cosine (x)	Set mnemonic = cos (x)

Function Description	**Set directive example**
Sine in radians (x)	Set mnemonic = rsin (x)
Cosine in radians (x)	Set mnemonic = rcos (x)
Arcsine in radians (x)	Set mnemonic = rasin (x)
Arccosine in radians (x)	Set mnemonic = racos (x)
Arctangent in radians (x)	Set mnemonic = ratan (x)
Add x to mnemonic's current value.	Set mnemonic += x
Subtract x from mnemonic's current value	Set mnemonic -= x
Multiply mnemonic's current value by x	Set mnemonic *= x
Divide mnemonic's current value by x	Set mnemonic /= x
Raise mnemonic's value to the power of x	Set mnemonic ^= x **or** Set mnemonic **= x
AND two values together	Set mnemonic = mnemonic and 1
OR two values	Set mnemonic = mnemonic or 7
OR two values and Negate the result	Set mnemonic = mnemonic nor 1
Exclusive OR two values	Set mnemonic = mnemonic xor 7
NAND two values	Set mnemonic = mnemonic nand 15

SIMSS-5.2.4.2.4 Set directive using mnemonics in expressions

Container names may be used instead of numbers in an expression. In the simplest case, a container item may be set to the current value of another container item.

set mnemonicA = mnemonicB

Container items may also be combined together to create expressions.

set mnemonicA = (mnemonicB + mnemonicC)/2

SIMSS-5.2.4.3 Setbuffer Directive

The **setbuffer** directive may be used to change the value of any modifiable buffer in the container for the indicated module. The format of the **setbuffer** directive is

[setbuffer | setbufferle | setbufferandlength] *buffername* <*offset*> [byte | word | dword] <*value*> <*value*> <*value*> …

buffer. If "setbufferandlength" is used, then the size of the buffer is set to the total size of the data values set. The **setbuffer** directive entry line is limited to 100 characters. The names of alterable buffers are provided in the container description for each module. An example of setbuffer usage is:

setbuffer TlmPacket0140 20 byte 10 012 0xA

In this example, three bytes of the buffer TlmPacket0140 beginning at offset 20 are set to 10. If the buffer is successfully set, an event message of the form

"module: values set in buffer *buffername*"

informs the operator of the change. If the buffer name is not recognized or the buffer cannot be modified, an event message of the form

"module: *buffername* not found"

will inform the operator. If unsuccessful, verify that the module indicator is correct, that the buffer name spelling is correct, and that the buffer is not defined as read-only.

SIMSS-5.2.4.4 Get Directive

The **get** directive may be used to display the value of any variable in the container for the indicated module. The format of the **get** directive is

get *name*

If the name matches a variable in the module's container, an event message of the form

"module: *name* = value"

is written to the event message log. If the name doesn't match a container variable, an error event message of the form

"module: *name* not found"

will be produced. If unsuccessful, check the setting of the module indicator and the spelling of the container variable name.

SIMSS-5.2.4.5 Getbuffer Directive

The **getbuffer** directive may be used to observe the value of any modifiable buffer in the indicated module. The format of the **getbuffer** directive is

getbuffer *buffername* <*offset*> [byte | word | dword] <*count*>

where *buffername* is a container buffer name, *offset* is the decimal byte offset to the first item to be retrieved, the keywords "byte", "word", and "dword" indicate the size of data to retrieve at one time, and *count* is the number of items of data to retrieve. If the suffix "le" is added to the typein, the data is retrieved in little-endian (Intel) byte order before being displayed. The **getbuffer** directive entry line is limited to 100 characters. The names of retrievable buffers are provided in the container description for each module. An example of **getbuffer** usage is:

getbuffer TlmPacket0140 20 byte 10

In this example, ten bytes of the buffer TlmPacket0140 beginning at offset 20 are retrieved. If there are no errors, an event message of the form

"module: *buffername* = 14 15 16 17 18 19 1a 1b 1c 1d"

informs the operator of the data with values in hexadecimal. If the buffer name is not recognized, an event message of the form

"module: *buffername* not found"

will inform the operator. If unsuccessful, verify that the module indicator is correct, and that the buffer name spelling is correct.

SIMSS-5.2.4.6 Create Directive

The **create** directive may be used to create additional mnemonics in the indicated module. The format of the **create** directive is

create *mnemonic*

where *mnemonic* is the name for a new variable in the module's container. The new name must not already exist and cannot include spaces. There are no other restrictions on the name but if the mnemonic is to be used in calculations, no arithmetic symbols or function names should be used. The new variable is a real number. It exists until removed by a delete directive or its associated module is deleted.

SIMSS-5.2.4.7 Delete Directive

The **delete** directive may be used to delete user created mnemonics in the indicated module. The format of the **delete** directive is

delete *mnemonic*

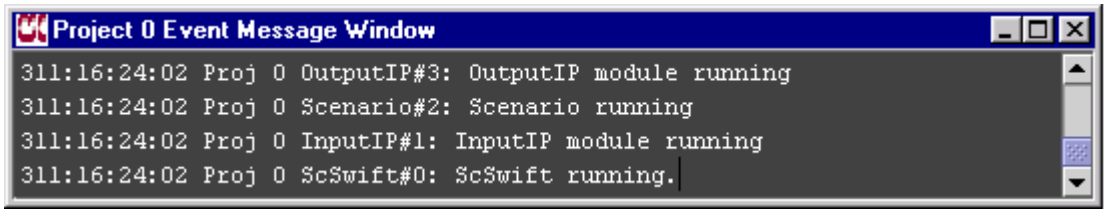
where *mnemonic* is the name of the container variable to be deleted. Only user created container items may be deleted.

SIMSS-5.3 Project Event Message Window

The Project Event Message window is automatically displayed when a project is added or restored. This window may be resized, minimized or closed. This window defaults to an automatic scrolling mode where the most recent event messages are always displayed. The scroll bar may be used to manually scroll back to display older event messages. The window then stays at the manually scrolled position. To return to automatic scrolling mode, return the scroll bar to the bottom edge of the scroll region.

The project event log window may be opened and closed by checking and unchecking the "Show Event Log" option from the System menu.

The contents of this window may be filtered by event message type. This window defaults to displaying only information, warning and error level messages. See the section on Event Message Filtering under the project system menu for more information.



Spacecraft Simulation Module for Aura (ETSAura)

ETSAura-1.0 Overview

The functions of the ETS Aura Spacecraft module (ETSAura) include ingest, generation and transmission of formatted telemetry packets, receipt and validation of commands in command link transmission units (CLTUs) or 1553 bus packet formats, and maintenance of GMT and spacecraft time. Spacecraft time is maintained as an offset from the Aura epoch and is inserted into telemetry packets prior to transmission. This module is capable of updating telemetry and starting the execution of scenario script files in response to commands. This module is also capable of displaying telemetry packets and received commands under operator direction.

ETSAura-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Scenario directives	Same as for operator directives	Directives from a scenario module are processed as if they had come from the user interface directive line.
2	Command stream (IP/serial CLTU, or 1553 command packets)	CCSDS and Aura specific	Commands parsed and validated per CCSDS standard and Aura extensions. Event messages generated on valid or invalid commands.
3	APID and EOS Enhanced CLCW Telemetry packets	APID, VCID validated with database and external load processing flags	If external telemetry receipt is enabled, the received packet is copied to the appropriate internal buffer and its telemetry points are updated with the received values. These packets are then merged into the outgoing telemetry stream.

The ETSAura module is capable of receiving two types of command input on channel 2. The default is used for the receipt of IP or serial commands in CCSDS CLTU format. If ETSAura is being used to simulate instruments on the 1553 bus, the container item named "CommandInputType" should be set to a non-zero value. When this flag is non-zero, the expected input is CCSDS command packets that are not contained within transfer frames and not codeblock encoded.

ETSAura is capable of ingesting telemetry and CLCW packets on input channel 3. Several sets of flags control which APIDs and which CLCWs may be overwritten with this external data. By default, processing of all external telemetry packets is disabled. Refer to section ETSAura-4.1.7 External Telemetry Container Items for further information.

ETSAura-3.0 Outputs

The ETSAura module has 2 types of output channels. When the module is using IP mode, only channels 1 and 2 are used. When the module is using Serial mode only channel 3 is used.

Ch.	Description
1	IP mode Housekeeping telemetry packets
2	IP mode Command Link Control Words (CLCWs)
3	Serial mode telemetry transmission

ETSAura-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the **Set** and **Get** directives to access items with a fixed or string type. Use the **SetBuffer** and **GetBuffer** directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

ETSAura-4.1 Telemetry Container Items

The container items for telemetry processing are organized into several sections: control and status of telemetry streams, telemetry points, packet buffers, segmented packets, dump packets, VCDU buffers, and external telemetry.

ETSAura-4.1.1 Telemetry Status and Control Container Items

Name	Type	Description
CheckpointTelemetry	Fixed	Set this variable to create a file of checkpointed telemetry values. The file created is actually a scenario script that may be executed to restore the telemetry to a checkpoint state. The file is written into the scenario directory defined by the ScenarioPath variable.
CLCWFrameCount	Fixed	CLCW channel transmitted count
CLCWPkt	Buffer	CLCW packet buffer (12 bytes)
TlmChannel1Enabled	Fixed	Channel 1 enabled flag (0=disabled, 1=enabled) (default is disabled)
TlmChannel1FrameCount	Fixed	Channel 1 frames transmitted count
TlmCUCpfield	Fixed	P-field for CUC time in packet header
TlmCUCpfieldextension	Fixed	P-field extension for CUC time in packet header
TlmEnabledCH1	Fixed	Flag to start/stop telemetry channel 1
TlmEnabledCH2	Fixed	Flag to start/stop telemetry channel 2
TlmInterval3	Fixed	Interval for automatic sending of VCDUs on physical channel 3
TlmOutputFormat	Fixed	Flag to change output format (0=IP, 1=serial)

TlmPktDropCnt	Fixed	Used to check if any VCDU packets were dropped
TransmitCLCW	Fixed	CLCW enabled flag (0=disabled, 1=enabled) (default is disabled)
CLCWInterval	Fixed	Interval between CLCW packet transmissions in seconds
CLCWTransmitLinked	Fixed	CLCW transmission linked to H/K transmission (0=no, 1=yes) (default is linked)

ETSAura-4.1.2 Telemetry Point Container Items

Each telemetry point read from the database is defined in the container four ways. The raw value is identified by the telemetry mnemonic name and by "TLM#" followed by the parameter identification number in decimal digits. The value in engineering units is identified by one of the raw names with "__EU" (two underscores) added to the end. For example, the raw value for telemetry point COM_BR_TXB (parm_id=3525) may be accessed with the name "COM_BR_TXB" or "TLM#3525". Its engineering value is identified by "COM_BR_TXB__EU" or "TLM#3525__EU". The value of any telemetry point can be requested using a **get** directive. The value of any telemetry point can be changed using the **set** directive. Whenever a telemetry point's value is changed, its raw and engineering values are updated using the conversion equations from the database. Set directives may also be used within scenario files.

Note: The raw values of telemetry points are stored as integers and the engineering unit values are stored as floating point values. When floating point values are assigned to raw values, the fractional portion is dropped.

Note: For telemetry points identified as 1750A floating point in the database, conversions of engineering unit values into the raw 1750A floating point format is supported. Remember to assign the floating point value to one of the "__EU" versions of the point.

ETSAura-4.1.3 Telemetry Packet Container Items

Container items are created for each valid telemetry packet defined in the database. Where "<APID>" appears in the following table, substitute the packet's APID number in 4 decimal digits. For example, the container buffer for APID 397 is named "TlmPacket0397". For packets that have secondary keys, substitute the APID in 4 decimal digits plus the secondary key in 4 decimal digits for the "<APID>". For example, the container buffer for APID 114 with secondary key of 55 is named "TlmPacket01140055".

Name	Type	Description
TlmPacket<APID>	Buffer	Buffer for specified APID's telemetry packet.
TlmPacket<APID>SequenceCount	Fixed	Sequence count for APID's packet header
TlmPacket<APID>SequenceFlag	Fixed	Sequence flag for APID's packet header
TlmPacket<APID>APID	Fixed	APID for packet header
TlmPacket<APID>SecondaryHeaderFlag	Fixed	Secondary header flag for APID's packet header
TlmPacket<APID>DataLength	Fixed	Data length for APID's packet header
TlmPacket<APID>Version	Fixed	Version for APID's packet header
TlmPacket<APID>Type	Fixed	Type for APID's packet header
TlmPacket<APID>Interval	Fixed	Time between transmissions in seconds (floating point number)
TlmPacket<APID>Enabled	Fixed	Enabled flag for internal packet formatting and transmission (0=no, 1=yes) (default is not enabled)
TlmPacket<APID>ExternalEnabled	Fixed	Flag to enable overwriting of packet with external packet (0=no, 1=yes) (default is not enabled)

ETSAura-4.1.4 Segmented Telemetry Container Items

In order to implement segmented telemetry packets, a file named **AuraSegmentedApids.txt**, containing APID numbers, one per line, must be in the Server\properties folder. This causes a segment data buffer and some associated variables to be created in addition to the telemetry items shown in the previous section. In the table below, substitute the appropriate segmented APID in 4 decimal digits for the <APID> in the Name. Scenario files should be used to store data into these segmented buffer items and control their transmission. A sample scenario file to initialize and transmit APID 1715 is **TES Boilerplate 3 segments.txt**, which should be in the default scenario directory.

Name	Type	Description
FPCResponseData<APID>	Buffer 5000 bytes	Segmented telemetry buffer that is sized to hold all segment data. Contents should be configured using setBuffer directives.
FPCResponseLength<APID>	Fixed	Variable that specifies the number of data bytes in the above segment data buffer.
FPC_ID<APID>	Fixed	Variable used to set the FPC cooler Id, a 2-byte field that follows the secondary header. Can be 1 or 2 for TES APIDs.

ETSAura-4.1.5 Telemetry Dump Packet Container Items

The dump packets are not generated from the database. In order to simulate a dump, the following dump parameters must be set. **As soon as the dump word count is given**, the dump packets will be formatted and transmitted until all of the specified words have been dumped. If dump parameters are being set via a scenario script, the dump word count must be set last. The dump word buffer may be set by a series of setBuffer directives, if not copied from a command load. The dump header information may be set using directives or the "Initiate Dump Packets…" selection from the Telemetry Menu.

Command load processing optionally copies the first segment of data words in a command load to the telemetry dump data buffer. As long as successive command loads contain consecutive memory locations, data words will be copied. Event messages produced during command load processing will provide information for starting a telemetry dump of the loaded memory locations.

Name	Type	Description
CmdLoadCopyToDump	Fixed	Automatic copying of command load buffer to telemetry dump buffer enabled flag (0=disabled, 1=enabled) (default is enabled)
CmdLoadValidChecksum	Fixed	Flag for validating command load buffer's checksum (0=disabled, 1=enabled) (default is enabled)
DumpMemoryAddress	Fixed	Memory address expected for next consecutive command load
DumpMemoryPosition	Fixed	Current byte position for automatic copy into the telemetry dump data buffer
CmdSpaceLoadBuffer	Buffer	Spacecraft memory load buffer (1024 bytes)
TlmDumpAPID	Fixed	Dump packet APID
TlmDumpInterval	Fixed	Interval in milliseconds between dump packets: 10000 = 10 seconds, defaults to 1000 (1 second)
TlmDumpPacketSize	Fixed	Size of dump packet (default is 256)
TlmDumpSecondaryHeader	Fixed	0 = no secondary header (default value) 1 = spacecraft secondary header with CUC
TlmDumpStartAddress	Fixed	Starting memory dump address
TlmDumpWordCount	Fixed	Number of 16 bit words to dump
SCMemory	Buffer	6000 byte telemetry dump data buffer

ETSAura-4.1.6 Telemetry VCDU Container Items

For the transmission of serial mode telemetry, certain VCDU Primary Header field values are defined in container items so that they may be easily changed. These are listed in the following table. All Aura telemetry is normally transmitted on virtual channel 2. Modification of the Fill VCDU (Virtual Channel 63) is not possible. To set the replay flag, for example, enter the directive 'set Vcdu2replay 1'.

Name	Type	Size	Description
Vcdu2version	Fixed	2 bits	Version Number
Vcdu2scid	Fixed	8 bits	Spacecraft ID
Vcdu2vcid	Fixed	6 bits	Virtual Channel ID
Vcdu2replay	Fixed	1 bit	Replay Flag
Vcdu2reserved	Fixed	7 bits	Reserved Flags

It is also possible to inject data values into VCDU headers that will appear as intermittent errors in the data stream. There are 50 internal buffers where VCDUs are constructed. To inject an error that will appear once in every 50 VCDUs, enter a set directive using one of the following container names as the argument. Where "<buffer>" appears in the following table, substitute a buffer number between zero and 49. For example, the container item for buffer 11's Primary Header Version field is named "VCDUVERSION11".

Name	Type	Size	Description
VCDUVERSION<buffer>	Fixed	2 bits	Version Number
VCDUSCID<buffer>	Fixed	8 bits	Spacecraft ID
VCDUVCID<buffer>	Fixed	6 bits	Virtual Channel ID
VCDUREPLAY<buffer>	Fixed	1 bit	Replay Flag
VCDURESERVED<buffer>	Fixed	7 bits	Reserved Flags

ETSAura-4.1.7 External Telemetry Container Items

The default configuration of this module does not allow any of its packet buffers to be overwritten with external data. Several sets of flags have been created to allow a flexible scheme of mixing received external data with internally generated data. Telemetry packets may be enabled for external input on an APID basis. CLCW packets may be enabled for external input on a virtual channel basis. When an internal APID or CLCW telemetry buffer is overwritten with an external packet, data from the external packet is then used to update the telemetry point variables. If these telemetry points are also reported in internally generated packets, the external values then appear in those packets.

Whether internally or externally generated, the operator has the ultimate control over all values in APID and CLCW packets. Any operator modification of telemetry points; from a GUI screen, from the directive line, or through the execution of a scenario file will "permanently" override new external values. The operator may later choose to allow the external data to take precedence again.

Due to the number of flags and complexity of these interactions, scenario files are recommended for configuring and controlling external data input.

ETSAura-4.1.7.1 External APID Telemetry Container Items

If a telemetry packet is enabled for internal generation (TlmPacket<APID>Enabled), it cannot directly receive external data. To allow any APID packet to be overwritten with

external packets, its associated external flag must be enabled and its internal flag must be disabled. Only APIDs loaded from the project database may be enabled for external input. External packets must not exceed the database-defined size of internal packet buffers. Operator changes to telemetry points are considered "sticky" and will override subsequent external values for those fields in the packet. To remove all "sticky" changes for a packet, briefly enable the packet for internal generation (for at least one transmission) and then enable the packet for external loading again. Two other ways to remove "sticky" updates using the RemoveUpdates variable are described below.

External APID Flags	**Type**	**Description**
TlmPacket<APID>ExternalEnabled	Fixed	External load enable flag for the <APID> packet (0=disabled, 1=enabled). Default is disabled.
UseExternalPacketTime	Fixed	Flag for synchronizing with external time. (0=disabled, 1=enabled). Default is enabled. If synchronization with external timing is not desired, this variable should be set to zero.
ExternalCUCTime	Fixed	Holds the most recent external CUC time received.
RemoveUpdates	String	This flag allows the user to remove "sticky" updates to telemetry values. To remove all updates from all packets, set this string to ALL. To remove updates for a specific telemetry item, set this string to the name of the specific telemetry item. Either version of the raw telemetry mnemonic name may be used. Do not enter quotes with the string value.

ETSAura-4.1.7.2 External CLCW Telemetry Container Items

The first two flags in the table control loading of the CLCW buffers with external values. Even when a CLCW is accepting external values, the operator may override any field's value. Operator changes to individual fields are considered "sticky" and will override subsequent external values for the field. To enable the field for external values again, set the field's "Updated" flag back to 0.

External CLCW Flags	**Type**	**Description**
SpaceClcwExternalEnabled	Fixed	External load enable flag for the Spacecraft CLCW buffer (0=disabled, 1=enabled). Default is disabled.
InstrClcwExternalEnabled	Fixed	External load enable flag for the Instrument CLCW buffer (0=disabled, 1=enabled). Default is disabled.

UpdatedSpaceClcwCWT UpdatedInstrClcwCWT	Fixed	Sticky update flag for spacecraft/instrument CLCW Control Word Type field (0=false, 1=true)
UpdatedSpaceClcwVersion UpdatedInstrClcwVersion	Fixed	Sticky update flag for spacecraft/instrument CLCW Version field (0=false, 1=true)
UpdatedSpaceClcwStatus UpdatedInstrClcwStatus	Fixed	Sticky update flag for spacecraft/instrument CLCW Status field (0=false, 1=true)
UpdatedSpaceClcwCOP UpdatedInstrClcwCOP	Fixed	Sticky update flag for spacecraft/instrument CLCW COP In Effect field (0=false, 1=true)
UpdatedSpaceClcwVCID UpdatedInstrClcwVCID	Fixed	Sticky update flag for spacecraft/instrument CLCW VCID field (0=false, 1=true)
UpdatedSpaceClcwSpare1 UpdatedInstrClcwSpare1	Fixed	Sticky update flag for spacecraft/instrument CLCW spare 1 field (0=false, 1=true)
UpdatedSpaceClcwNoRfAvail UpdatedInstrClcwNoRfAvail	Fixed	Sticky update flag for spacecraft/instrument CLCW No RF Avail field (0=false, 1=true)
UpdatedSpaceClcwNoBitLock UpdatedInstrClcwNoBitLock	Fixed	Sticky update flag for spacecraft/instrument CLCW No Bit Lock field (0=false, 1=true)
UpdatedSpaceClcwLockout UpdatedInstrClcwLockout	Fixed	Sticky update flag for spacecraft/instrument CLCW Lockout field (0=false, 1=true)
UpdatedSpaceClcwWait UpdatedInstrClcwWait	Fixed	Sticky update flag for spacecraft/instrument CLCW Wait field (0=false, 1=true)
UpdatedSpaceClcwRetransmit UpdatedInstrClcwRetransmit	Fixed	Sticky update flag for spacecraft/instrument CLCW Retransmit field (0=false, 1=true)
UpdatedSpaceClcwFarmCount UpdatedSpaceClcwFarmCount	Fixed	Sticky update flag for spacecraft/instrument CLCW FARM-B Count field (0=false, 1=true)
UpdatedSpaceClcwSpare2 UpdatedInstrClcwSpare2	Fixed	Sticky update flag for spacecraft/instrument CLCW spare 2 field (0=false, 1=true)
UpdatedSpaceClcwReport UpdatedInstrClcwReport	Fixed	Sticky update flag for spacecraft/instrument CLCW Report field (0=false, 1=true)

ETSAura-4.1.7.3 External Telemetry Status Container Items

External Telemetry Status	Type	Description
ExternalPacketHeaderBytes	Fixed	Size of optional header on all external packets. If external packets have an EDOS Service Header, this field should be set to 20. Default header size is 0 bytes.
ExternalCLCWCount	Fixed	Count of all external CLCW packets processed
ExternalAPIDCount	Fixed	Count of all external APID packets processed
ExternalIgnoredCLCWCount	Fixed	Count of all CLCW packets ignored because external loading was disabled
ExternalIgnoredAPIDCount	Fixed	Count of all APID packets ignored because external loading was disabled
ExternalPacketErrorCount	Fixed	Count of all external packets in error

ETSAura-4.2 Command Container Items

The container items for command processing fall into several groups: mission specific items, validation/debug flags, CLCW fields, processing buffers, command triggering scenarios, command sub-mnemonics and counts. Initial values for some items are shown in parentheses following the description.

ETSAura-4.2.1 Mission Specific Container Items

The initial values for these items are shown in parentheses. Values may be changed using **set** or **setBuffer** directives from the directive line or from a scenario file.

Name	Type	Description
CmdCltuCodeblockSize	Fixed	Codeblock size in bytes (8)
CmdCltuExpectedStartSequence	Buffer	Expected CLTU start sequence buffer (2 bytes) ($EB90_{16}$)
CmdCltuExpectedTailSequence	Buffer	Expected CLTU tail sequence buffer (8 bytes) ($C5C5\ C5C5\ C5C5\ C579_{16}$)
CmdCriticalTieAVCID	Fixed	Virtual Channel ID (VCID) for TIE-A critical commands (16)
CmdCriticalTieBVCID	Fixed	Virtual Channel ID for TIE-B critical commands (17)
CmdSCID	Fixed	Spacecraft Identifier (CC_{16})
CmdSlidingWindowSize	Fixed	FARM-1 Sliding Window Size (180)

ETSAura-4.2.2 Command Validation/Debug Flags

Name	Type	Description
CmdEnabled	Fixed	Command processing enabled flag. (0=disabled, 1=enabled) (Default is enabled)
CommandInputType	Fixed	Command input type flag. (0 = CLTU, other = 1553 command packets) (Default is CLTU)
CodeblockValidation	Fixed	Command codeblock parity validation enabled flag. (0=disabled, 1=enabled) (Default is enabled)
FrameValidation	Fixed	Command transfer frame header validation enabled flag. (0=disabled, 1=enabled) (Default is enabled)
FARMValidation	Fixed	Command Frame Acceptance and Reporting Mechanism (FARM) validation enabled flag. (0=disabled, 1=enabled) (Default is enabled)
PacketValidation	Fixed	Command packet validation enabled flag. (0=disabled, 1=enabled) (Default is enabled)
CmdDebugEnabled	Fixed	Master command debug messages flag (0=disabled, 1=enabled). When enabled, all of the command debug event messages are generated. (Default is disabled)

CmdDebugCLTU	Fixed	Debug CLTU messages flag (0=disabled, 1=enabled). When enabled, CLTU debug event messages are produced. (Default is disabled)
CmdDebugCodeblock	Fixed	Debug codeblock flag (0=disabled, 1=enabled). When enabled, codeblock debug event messages are produced. (Default is disabled)
CmdDebugFrame	Fixed	Debug frame flag (0=disabled, 1=enabled). When enabled, frame debug event messages are produced. (Default is disabled)
CmdDebugSpacePacket	Fixed	Debug spacecraft packet flag (0=disabled, 1=enabled). When enabled, debug event messages are produced for each spacecraft channel packet. (Default is disabled)
CmdDebugInstrPacket	Fixed	Debug instrument packet flag (0=disabled, 1=enabled). When enabled, debug event messages are produced for each spacecraft channel packet. (Default is disabled)

ETSAura-4.2.3 Command Link Control Word Fields

Name	Type	Description
CmdSpaceClcwCWT	Fixed	Spacecraft CLCW Control Word Type
CmdSpaceClcwVersion	Fixed	Spacecraft CLCW Version
CmdSpaceClcwStatus	Fixed	Spacecraft CLCW Status
CmdSpaceClcwCOP	Fixed	Spacecraft CLCW COP In Effect (1)
CmdSpaceClcwVCID	Fixed	Spacecraft CLCW VCID (0)
CmdSpaceClcwSpare1	Fixed	Spacecraft CLCW Spare field 1
CmdSpaceClcwNoRFAvail	Fixed	Spacecraft CLCW No RF Avail Flag
CmdSpaceClcwNoBitLock	Fixed	Spacecraft CLCW No Bit Lock Flag
CmdSpaceClcwLockout	Fixed	Spacecraft CLCW Lockout Flag
CmdSpaceClcwWait	Fixed	Spacecraft CLCW Wait Flag
CmdSpaceClcwRetransmit	Fixed	Spacecraft CLCW Retransmit Flag
CmdSpaceClcwFarmCount	Fixed	Spacecraft CLCW Farm-B Counter
CmdSpaceClcwSpare2	Fixed	Spacecraft CLCW Spare field 2
CmdSpaceClcwReport	Fixed	Spacecraft CLCW Report Value
CmdInstrClcwCWT	Fixed	Instrument CLCW Control Word Type
CmdInstrClcwVersion	Fixed	Instrument CLCW Version
CmdInstrClcwStatus	Fixed	Instrument CLCW Status
CmdInstrClcwCOP	Fixed	Instrument CLCW COP In Effect (1)
CmdInstrClcwVCID	Fixed	Instrument CLCW VCID (1)
CmdInstrClcwSpare1	Fixed	Instrument CLCW Spare field 1
CmdInstrClcwNoRFAvail	Fixed	Instrument CLCW No RF Avail Flag
CmdInstrClcwNoBitLock	Fixed	Instrument CLCW No Bit Lock Flag
CmdInstrClcwLockout	Fixed	Instrument CLCW Lockout Flag
CmdInstrClcwWait	Fixed	Instrument CLCW Wait Flag

CmdInstrClcwRetransmit	Fixed	Instrument CLCW Retransmit Flag
CmdInstrClcwFarmCount	Fixed	Instrument CLCW Farm-B Counter
CmdInstrClcwSpare2	Fixed	Instrument CLCW Spare field 2
CmdInstrClcwReport	Fixed	Instrument CLCW Report Value

ETSAura-4.2.4 Command Container Buffers

Name	Type	Description
CmdPolyRemainderTbl	Buffer	Polynomial remainder table for parity calculation. (256 bytes)
CmdSpaceCLCW	Buffer	Spacecraft virtual channel CLCW buffer (4 bytes)
CmdInstrCLCW	Buffer	Instrument virtual channel CLCW buffer (4 bytes)
CmdSpacePkt	Buffer	Spacecraft command packet buffer (128 bytes)
CmdInstrPkt	Buffer	Instrument command packet buffer (128 bytes)
CmdFrameBuffer	Buffer	Command transfer frame buffer (256 bytes)
CmdCLTU	Buffer	Command link transmission unit (6000 bytes)
CmdCodeblock	Buffer	Compressed codeblock buffer (holds codeblock bytes without parity bytes) (6000 bytes)

ETSAura-4.2.5 Command Triggering Scenarios Container Items

Name	Type	Description
ScenarioPath	String	Full pathname of directory where the command triggering scenario definition file can be found. Path should end with slash character.
ReadScenFile	Fixed	Whenever a set directive accesses this flag, the command triggering scenario definition file is re-read, replacing the previous definitions.

ETSAura-4.2.6 Command Submnemonic Container Items

For each variable command in the database, container items are created for each sub-mnemonic field. The format for the names of these items is subname#parm_id. For example, the value received in the ANGLE sub-mnemonic for the command with PARM_ID 32081 will be stored in "ANGLE#32081". During run-time as variable commands are identified in the database, event messages display each command's mnemonic name, its parameter ID in decimal digits, and any sub-mnemonic names and values. At the same time that the event message is being created, the received sub-mnemonic values are stored into their associated container items.

ETSAura-4.2.7 Command Counter Container Items

Name	Type	Description
TIE_A_Active	Fixed	Flag indicates whether TIE A or TIE B is online for the purpose of counting codeblocks (0=TIE B, 1=TIE A) (default is TIE_A)

CmdCounter0	String	Name of telemetry point counter for TIE A valid codeblocks
CmdCounter1	String	Name of telemetry point counter for TIE B valid codeblocks
CmdCounter2	String	Name of telemetry point counter for VCID 16 valid critical commands
CmdCounter3	String	Name of telemetry point counter for VCID 17 valid critical command
CmdCounter4	String	Name of telemetry point counter for valid ISC commands
CmdCounter5	String	Name of telemetry point counter for valid PC commands
CmdCounter6	String	Name of telemetry point counter for valid GNCC commands
CmdCounter7	String	Name of telemetry point counter for valid CTC online commands
CmdCounter8	String	Name of telemetry point counter for valid CTC offline commands
CmdCounter9	String	Name of telemetry point counter for invalid ISC commands
CmdCounter10	String	Name of telemetry point counter for invalid PC commands
CmdCounter11	String	Name of telemetry point counter for invalid GNCC commands
CmdCounter12	String	Name of telemetry point counter for invalid CTC online commands
CmdCounter13	String	Name of telemetry point counter for invalid CTC offline commands
CmdTotalCLTUs	Fixed	Count of all CLTUs received
CmdValidCLTUs	Fixed	Count of valid CLTUs
CmdRejectCLTUs	Fixed	Count of invalid CLTUs
CmdTotalCodeblocks	Fixed	Count of all Codeblocks
CmdValidCodeblocks	Fixed	Count of valid Codeblocks
CmdRejectCodeblocks	Fixed	Count of invalid Codeblocks
CmdTotalTransferFrames	Fixed	Count of all Transfer Frames
CmdValidTransferFrames	Fixed	Count of valid Transfer Frames
CmdErrorTransferFrames	Fixed	Count of invalid Transfer Frames
CmdADFrames	Fixed	Count of Type AD Transfer Frames
CmdACFrames	Fixed	Count of Type AC Transfer Frames
CmdBCFrames	Fixed	Count of Type BC Transfer Frames
CmdBDFrames	Fixed	Count of Type BD Transfer Frames
CmdTotalSpacePkts	Fixed	Count of all Spacecraft Packets
CmdValidSpacePkts	Fixed	Count of valid Spacecraft Packets
CmdErrorSpacePkts	Fixed	Count of invalid Spacecraft Packets

CmdTotalInstrPkts	Fixed	Count of all Instrument Packets
CmdValidInstrPkts	Fixed	Count of valid Instrument Packets
CmdErrorInstrPkts	Fixed	Count of invalid Instrument Packets
CmdIgnoredCLTUs	Fixed	Number of CLTUs ignored while command processing is disabled.

ETSAura-4.3 Database Container Items

Name	Type	Description
DatabaseCmdId	Fixed	Command identification flag (0=off, 1=on). When this flag is on, command identification using the database occurs with each received command. When this flag is off, all received commands are processed as if they were not found in the database.
CmdFixDataFile	String	Loaded Command Fixed Data Word Specification File
CmdParmFile	String	Loaded Command Parameter Specification File
CmdVarDataFile	String	Loaded Command Variable Data Word Specification File
CmdVerifyFile	String	Loaded Command Execution Verification (CEV) File
TlmCalcurveFile	String	Loaded Telemetry EU Conversion Specification File
TlmDescFile	String	Loaded Telemetry Description File
TlmDStateFile	String	Loaded Discrete States Specification File
TlmPacketFile	String	Loaded Telemetry Packet Specification File
TlmParmFile	String	Loaded Telemetry parameter specification File
TlmPolyConvFile	String	Loaded Polynomial Coefficients Specification File
TlmRYLimitsFile	String	Loaded Red/Yellow Limit Specification File

ETSAura-5.0 Displays

To access the displays for this module, first click on the center of the ETSAura module in the project window. The following items will appear in a pop-up menu.

Module Pop-Up Menu Item	Description
Configure	Access the configuration menu for the module
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

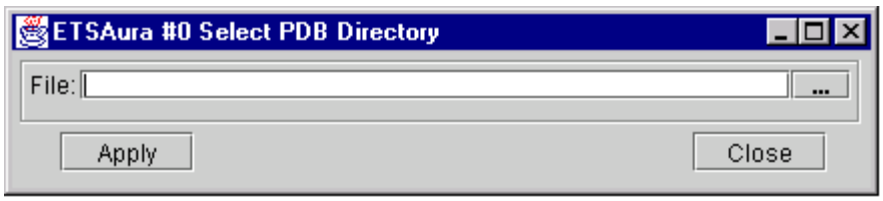
ETSAura-5.1 Configuration Menu

The configuration menu for the ETSAura module contains two items.

Configuration Menu Item	Description
Load Database	Load project database information
Select Simulation Mode	Select IP or Serial Mode
Select Command Mode	Select Command input format

ETSAura-5.1.1 Load Database

This display specifies the directory location of the project database flat files for initialization of command and telemetry information in the simulator.



Display Field	Description
File:	Specifies the full path file name for the project database directory. This field may be entered directly or via the file-browsing button to the right of this field.
File Browse Button ...	Use this button to navigate to the appropriate directory on the local machine or anywhere within the machine's network neighborhood. (Sample file browsing screens are shown in the Scenario module chapter.)
Apply	Use this button to attempt database loading from the specified directory. Information about the success or failure of loading the project database files is reported to the event message log.
Close	Use this button to dismiss the display

ETSAura-5.1.1.1 Maintaining Database Versions

Users must keep different versions of the database in separate directories. The individual flat files do not usually require renaming. The simulator loads the "first" database file matching each wildcard pattern from the following table. If multiple files in a directory match a wildcard pattern, the database file that is loaded is unpredictable. If a file is surpassed by a newer version (with a slightly different version field), the original file should either be deleted or renamed so that it no longer matches the wildcard pattern. Event messages will list the selected database directory and the names of the loaded files.

Wildcard File Name	Description
cmd_fixdata_*.pdb	Command Fixed Data Word Specification File
cmd_parm_*.pdb	Command Parameter Specification File
cmd_vardata_*.pdb	Command Variable Data Word Specification File
cmd_verify_*.pdb	Command Execution Verification (CEV) File
tlm_calcurve_*.pdb	Telemetry EU Conversion Specification File
tlm_desc_*.pdb	Telemetry Description File
tlm_dstate_*.pdb	Discrete States Specification File
tlm_packet_*.pdb	Telemetry Packet Specification File

tlm_parm_*.pdb	Telemetry parameter specification File
tlm_polyconv_*.pdb	Polynomial Coefficients Specification File
tlm_rylim_*.pdb	Red/Yellow Limit Specification File

Note: The asterisk '*'character in each wildcard pattern enables the matching of zero to many version specification characters in each database file name.

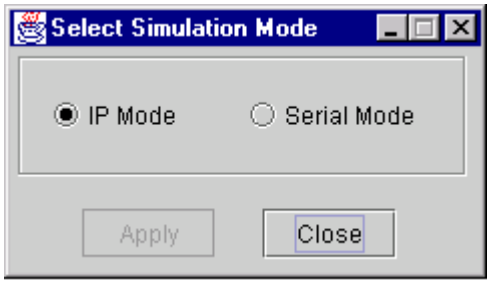
ETSAura-5.1.1.2 Reloading Database Versions

The user has the capability to change the loaded version of the database. If still in configuration mode, a different PDB directory may be selected and applied. Event messages will report the deletion of the current database and the loading of the newly selected database.

If the simulator is running, use the project's Run menu to "Stop" the project. Then use the module pop-up's Configure menu to access the "Load Database" option. After the current database information is deleted and the new database has finished loading, use the Run menu's "Run" option to restart the project. When ETSAura resumes, all the newly created telemetry packets are disabled and all the newly created telemetry points are set to their database defined initial values. Scenario scripts must be executed again to enable the appropriate packets for transmission and to make other desired modifications to telemetry points.

If the user has modified the database files within a directory, the same directory may be reloaded again only if a character in the directory name is changed from lowercase to uppercase or vice versa. Subsequent loads from the same case-sensitive directory name are inhibited so that inadvertent double clicks do not cause repeated loading, deletion and reloading of the same set of database files.

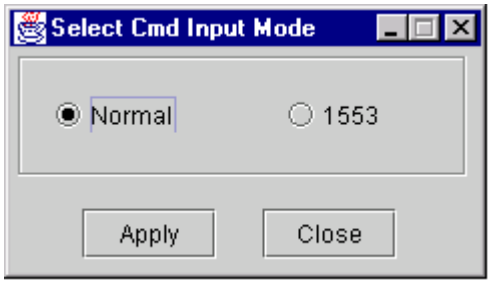
ETSAura-5.1.2 Select Simulation Mode



This display is used to set the Simulation Mode. Depending on the simulation mode, different output interface modules must be used in the project's design. Clicking the **Apply** button causes the action defined by the fields to take effect. Clicking the **Close** button closes the window with no additional action taken.

Mode Field	Description
IP Mode	The module will use IP protocol for the output channels. See the IP module chapter for configuration of the Output IP modules.
Serial Mode	The module will use a serial output channel. See the Serial module chapter for configuration of the Serial Output module.

ETSAura-5.1.3 Select Command Mode



This display is used to define the format (and validation) for expected command input.

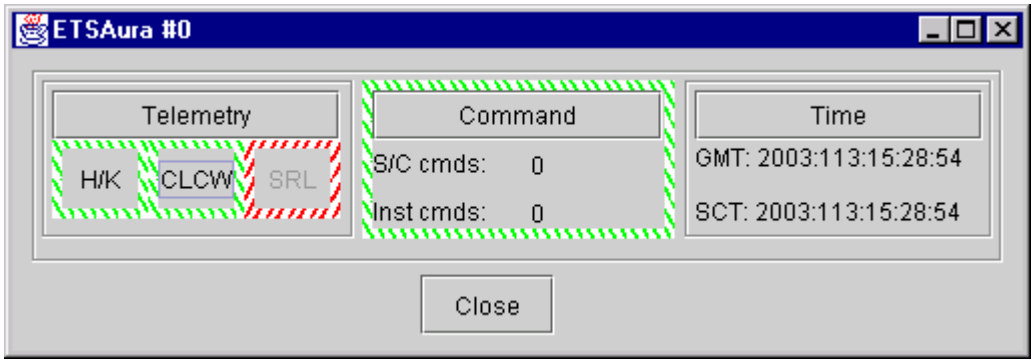
Clicking the **Apply** button causes the action defined by the fields to take effect. Clicking the **Close** button closes the window with no additional action taken.

Mode Field	Description
Normal	When this mode is selected, the simulator is expecting commands in CCSDS CLTU format. This is the default command mode.
1553	When this mode is selected, the simulator is expecting command packets as received from the 1553 bus interface.

ETSAura-5.2 Run-time Menu

Run-time Menu Item	Description
Control	Request the module's main display
Resume	Resume the module's execution
Pause	Pause the module's execution

ETSAura-5.2.1 Main Display



This is the main display for the ETSAura module, providing basic information about its current state, including the current GMT and spacecraft times, the number of valid spacecraft and instrument commands received, and the enabled/disabled status of command receipt and telemetry transmission. The latter are indicated by colored diagonal stripes around the appropriate box: the entire command box for command status and the telemetry channel boxes for each telemetry channel. Red stripes angling up and to the right indicate disabled, while green stripes angling down and to the right indicate enabled. When the module is using IP mode, the telemetry region buttons **H/K** and **CLCW** are available and the **SRL** button is unavailable. When the module is using serial mode, only the **SRL** button is available. Clicking on the **Telemetry**, **Command**, and **Time** buttons in the main window provides access to subordinate displays.

ETSAura-5.2.1.1 Telemetry Menus

There are four buttons in the telemetry area of the main display: **Telemetry**, **H/K**, **CLCW** and **SRL**. Clicking the **Telemetry** button brings up a menu with the following items.

Telemetry Menu Item	Description
Modify Packet…	Modify packet data area on byte basis
Display Packet…	Display packet header and contents
Display Status	Display status of the telemetry channels
Control Packet…	Change the frequency of the packet transmission
Initiate Dump…	Set parameters to initiate dump packets
Display/Set Container Items…	Display and optionally modify container items
Display Container Buffer…	Display any container buffer
Modify CUC…	Modify the CCSDS Unsegmented Time Code
APID Status	Display status of database defined packets
External Packet Summary	Display status of externally generated packets

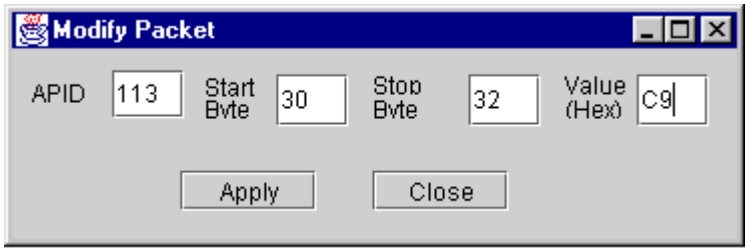
Clicking the **H/K**, **CLCW** or **SRL** channel buttons brings up a menu with the following choices.

Channel Menu Item	Description
Start	Enable telemetry transmission on the channel
Stop	Disable telemetry transmission on the channel

Note: The **H/K** and **CLCW** buttons have been linked together to simplify operations. Whenever the **H/K** button is used to start or stop housekeeping telemetry transmission, the same action will be applied to the CLCW channel. The **CLCW** button may be used to start or stop the CLCW channel independently as well. If decoupling of these two buttons is desired, enter the following directive on the directive line or in a scenario file.

Set ClcwTransmitLinked 0

ETSAura-5.2.1.1.1 Modify Packet Display

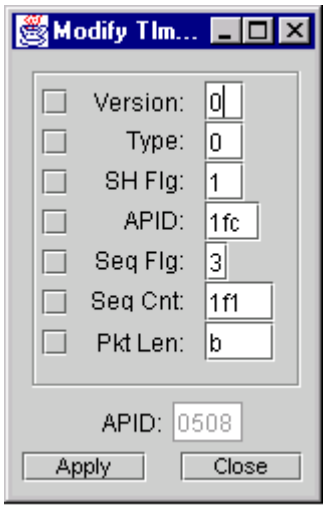


This display allows the user to modify the data area of the packet on a byte or repeating byte basis. All of the display fields must be entered.

Data Entry Field	Description
APID	Application ID of the packet to be modified
Start Byte	First byte to change
Stop Byte	Last byte to change
Value (Hex)	Value to change byte(s) to, in hexadecimal

ETSAura-5.2.1.1.2 Telemetry Packet Display

The telemetry packet display shows the header and, optionally (by pressing the **Dump** button), the contents of the most recent telemetry packet sent with the given APID. The user must enter a valid value in the APID field and then click the **Apply** button before any other values will be reported. If there are multiple formats for an APID, click the "Use Secondary Key" box and enter the secondary key value in the space next to the APID. Clicking the **Dump** button will bring up a standard dump display with the contents of the packet. Clicking the **Edit** button will bring up the Modify Tlm Packet Header Display shown below. Clicking the **Close** button closes the window with no additional action taken.

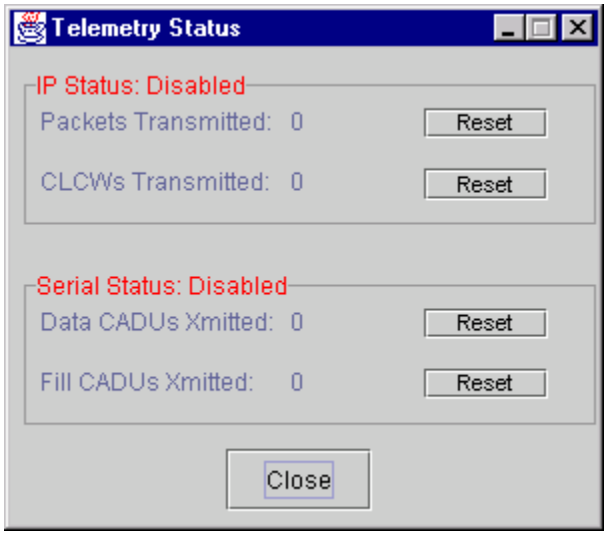


The Modify Tlm Packet Header display allows the user to modify fields in the packet header. Click on an enable box to the left of the field to enable modification of a specific header field. Clicking the **Apply** button puts the settings into effect. Clicking the **Close** button closes the display with no additional action taken.

The purpose of the Modify Tlm Packet Header display is to inject data values into transmitted packets that could be perceived as erroneous by the receiving system. Changing the APID and Pkt Len fields does not change the name or length of the buffer, only the data in the header fields.

Data Entry Field	Description
Version	CCSDS packet version number
Type	CCSDS packet type
SH Flg	Secondary header flag
APID	Application identifier (packet number)
Seq Flg	Sequence flag
Seq Cnt	Packet sequence counter
Pkt Len	Packet length for data zone (seven less than actual length)

ETSAura-5.2.1.1.3 Telemetry Status Display

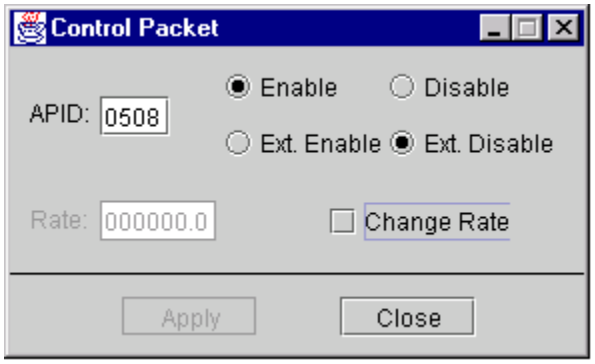


The telemetry status display shows the enabled or disabled status and the number of packets or CADUs transmitted for each of the telemetry channels.

Clicking the **Reset** button will reset the associated counter to zero.

Clicking the **Close** button dismisses this window.

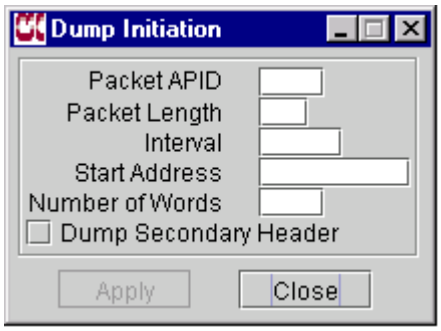
ETSAura-5.2.1.1.4 Control Packet Display



This display allows the user to control whether or not a telemetry packet is internally generated or overwritten with external data and if internally generated, optionally changes the rate at which it is transmitted. Clicking the **Apply** button puts the settings into effect. Clicking the **Close** button exits the display.

Data Entry Field	Description
APID	Enter Application ID of the packet to be modified.
Enable	If flag is set, packet is internally formatted and sent.
Disable	If flag is set, the packet is not formatted and transmitted.
Ext. Enable	If flag is set and the Disable flag is set, packet may be overwritten with external packet data. If Ext. Enable and Enable are set, the packet is internally formatted and sent.
Ext. Disable	If flag is set, packet cannot receive external packet data.
Change Rate	This box must be checked in order to change the rate field
Rate	Enter the interval in seconds and fractions of seconds between successive packet transmissions.

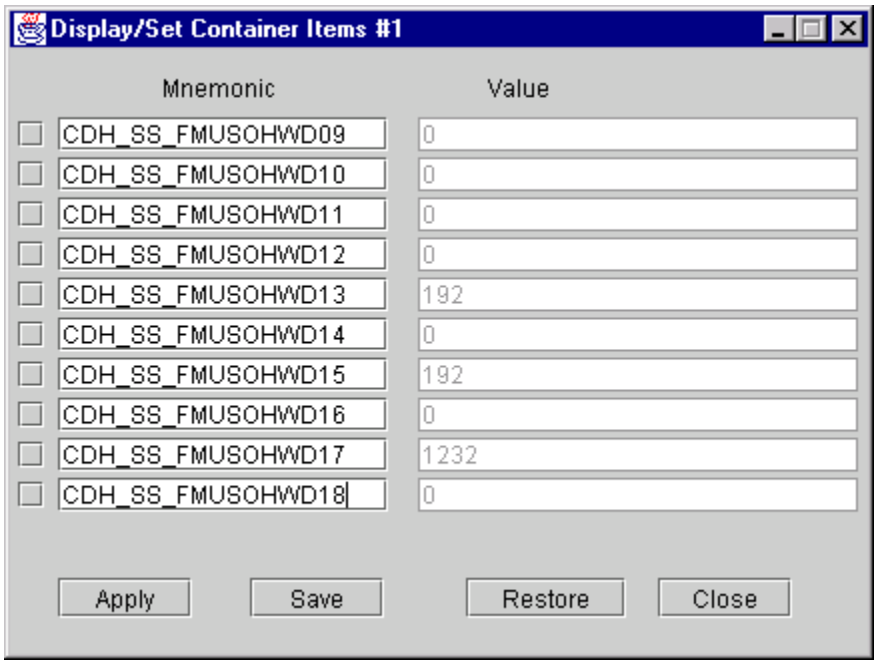
ETSAura-5.2.1.1.5 Initiate Dump Packets



This display allows the operator to specify dump packet header values and initiate the sending of dump packets. The dump data words are kept in the container buffer SCMemory. This buffer may be modified using the **setBuffer** directive on the directive line or within a scenario file. Only one dump may be specified at a time. See Paragraph ETSAura-4.1.5, Telemetry Dump Packet Container Items, for more information.

Data Entry Field	Description
Packet APID	Application ID to store in dump packet header
Packet Length	Maximum size of dump packet. The last packet may be smaller.
Interval	Time between dump packet transmissions, in milliseconds
Start Address	Starting memory address
Number of Words	Number of words to be dumped
Dump Secondary Header	If this box is checked, the dump packets will have a spacecraft-format secondary header containing a CCSDS unsegmented time code. If not checked, packets will not have a secondary header.

ETSAura-5.2.1.1.6 Display/Set Container Items



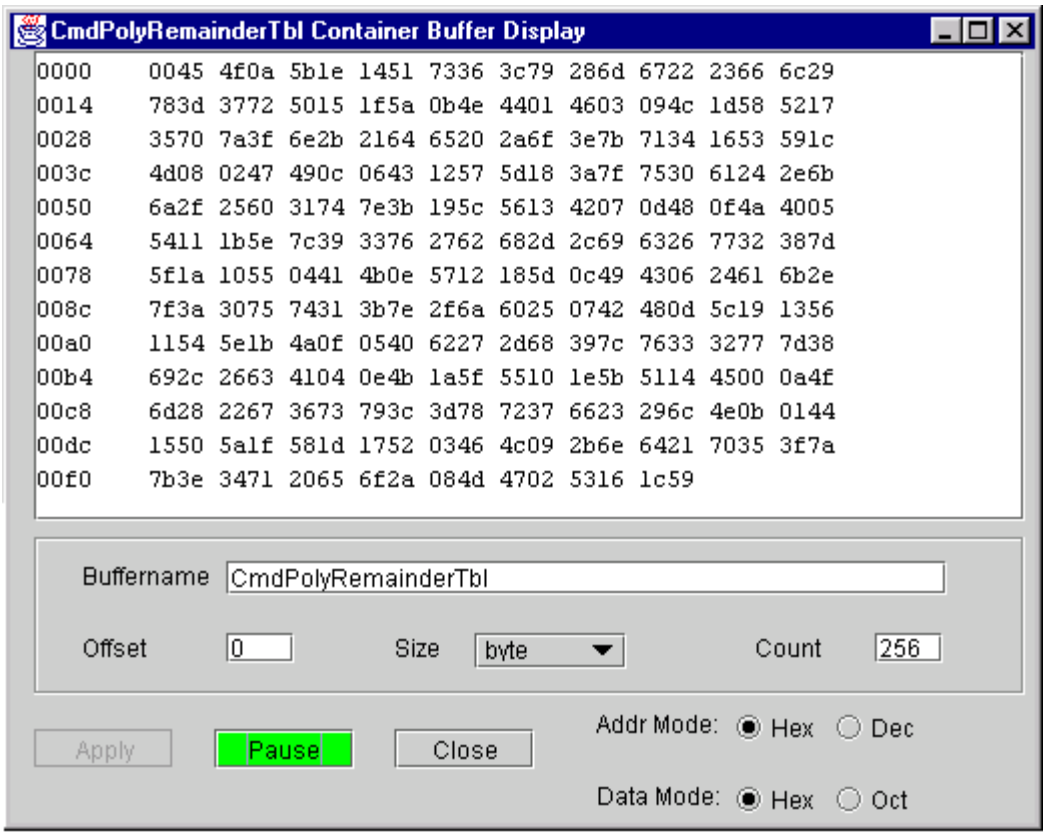
This display allows the user to view and modify individual items from the ETSAura module's container. Since telemetry points are stored as container items, telemetry points may be monitored and overwritten with this display. To request an item for display or editing, enter its name in the mnemonic field and click the **Apply** button. If there are any

problems with displaying or modifying items, error messages are sent to the event message area of the screen. Multiple copies of this display are allowed, but each container item should only appear on one screen for properly updated values. The list of displayed container item names may be saved to a file using the **Save** button. Similarly, a saved set of item names may be loaded into this display by using the **Restore** button and selecting a saved file.

Field	Description
Checkbox	Click on this box to modify an item's value.
Mnemonic	This field identifies a container item for display. Refer to section "ETSAura-4.0 Container Items" for the container names of specific items. This field is not case-sensitive.
Value	If the associated checkbox is not checked, this is the current decimal value of the container item. If the checkbox is checked, this is a data entry field for the value of the item. Values may be entered in decimal, octal, or Hexadecimal. Negative values for Signed Integer or Floating Point container items may be entered by preceding the number with a minus sign. Octal and Hexadecimal entries will be converted to decimal.

ETSAura-5.2.1.1.7 Display Container Buffer

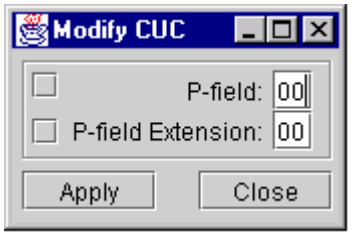
This display may be used to display container buffers that do not have a customized display available for them. Enter the buffer name, offset, size and count and click on the **Apply** button. If there are problems displaying a buffer, error messages are sent to the event message area of the screen.



Only one copy of this display may be active at a time. When switching from the display of one buffer to another, it may be helpful to toggle the **Pause/Cont** button on and off to force a refresh of the buffer contents.

Field	**Description**
Buffername	This field identifies the container buffer for display. Refer to section "ETSAura-4.0 Container Items" for names of container buffers.
Offset	This field specifies the first byte to be displayed. This display is limited to 1400 data bytes at a time. This field is used with the count field to specify the portion of the buffer to display.
Size	Size of the data items to display (choices are byte, word, double)
Count	Number of data items to display (limited to 1400 total characters)
Addr Mode	Addresses of the data may be displayed in hexadecimal or decimal
Data Mode	Data may be displayed in hexadecimal or octal formats
Apply	Use **Apply** button to submit Buffername, Offset, Size and Count fields.
Pause/Cont	If the buffer's contents are updated frequently, the **Pause** button may be used to display a "snapshot" of the buffer. (Buffer contents will continue to be updated offscreen.) The button's name is changed to **Cont**. To resume updates of the buffer, press the **Cont** button.
Close	Use the **Close** button to dismiss the display.

ETSAura-5.2.1.1.8 Modify CCSDS Unsegmented Time Code

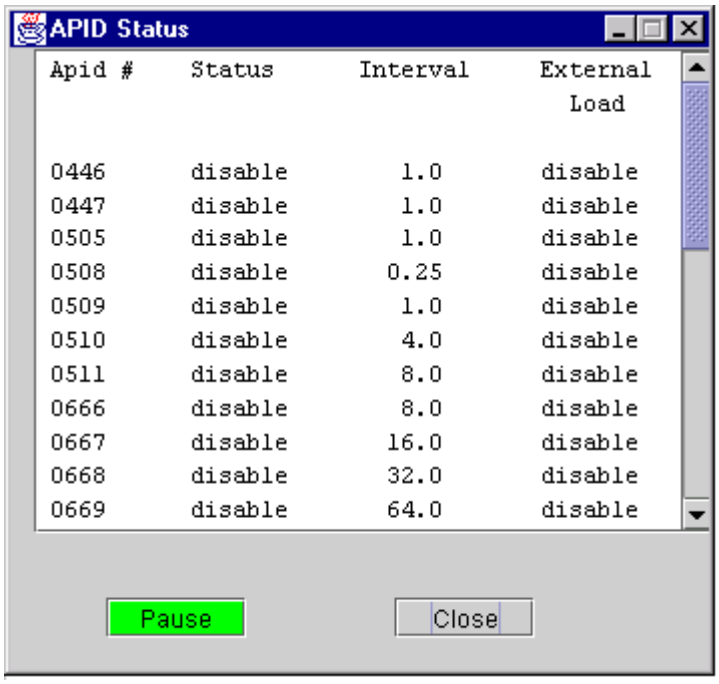


This display allows the user to modify the fixed fields of the CCSDS Unsegmented Time Code (CUC) in the telemetry packet headers. Clicking the enable boxes at left indicate whether or not the given field should be modified. Clicking the **Apply** button puts the settings into effect for all APIDs.

Data Entry Field	Description
P-field	P-field flags in the CUC. Enter hexadecimal value.
P-field Extension	Second byte of p-field, containing number of leap seconds since 1958. Enter hexadecimal value.

ETSAura-5.2.1.1.9 APID Status Display

This updating display provides a scrollable list of the current status of all telemetry packets created from the project database. The first column provides the APID number in decimal. Next the packet's internal "disabled" or "ENABLED" status is displayed.



The Interval column provides the time between packet transmissions in seconds. The External column lists whether the packet may be overwritten with externally generated data.

Clicking on the **Pause** button will interrupt the display's updates. The button's title will change to **Cont**. Click on the **Cont** button to resume status updates.

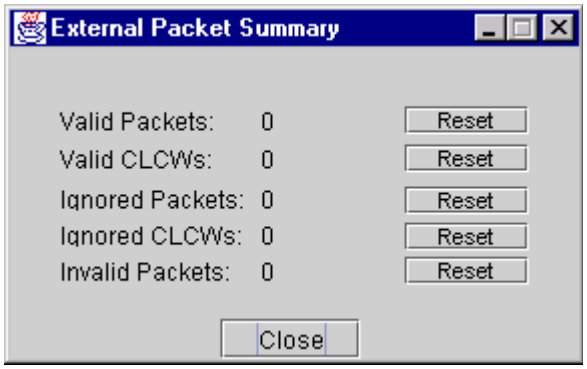
Use the **Close** button to dismiss the display.

The "Control Packet…" telemetry menu item may used to modify any packet's status or interval. Directives and scenario files may also be used to change packet status. The container item names for these fields are TlmPacket<APID>Enabled, TlmPacket<APID>Interval, and TlmPacket<APID>ExternalEnabled, where the <APID> portion is replaced with the packet's APID in 4 decimal digits. For example, the following directives change the interval of packet 220 to 8 seconds and enable its transmission.

```
Set TlmPacket0220Interval 8.0
Set TlmPacket0220Enabled 1
```

Entering a directive to change a TlmPacket<APID>Interval, or applying a new interval via the Control Packet Display, causes the telemetry APID to be entered into the simulator's transmission schedule at Interval seconds from the time the directive is entered. Setting an APID's interval at arbitrary times is discouraged. It is best to set APID intervals using a scenario script because proper relative transmission timing may be achieved.

ETSAura-5.2.1.1.10 External Packet Summary Display



This display provides information about external telemetry packet processing.

Click any **Reset** button to set its associated counter to zero.

Click **Close** to exit the display.

Summary Field	Description
Valid Packets	Count of external APID packets that were received and stored into the appropriate simulator packet buffer.
Valid CLCWs	Count of external CLCW packets that were received and stored into the appropriate simulator CLCW buffer.
Ignored Packets	Count of external APID packets received that were ignored because external loading of these packets was not enabled.
Ignored CLCWs	Count of external CLCW packets received that were ignored because external loading of these packets was not enabled.
Invalid Packets	Total count of external packets received with invalid formats. These packets did not match the APIDs of any database-defined packets and did not match the VCIDs of expected CLCW packets.

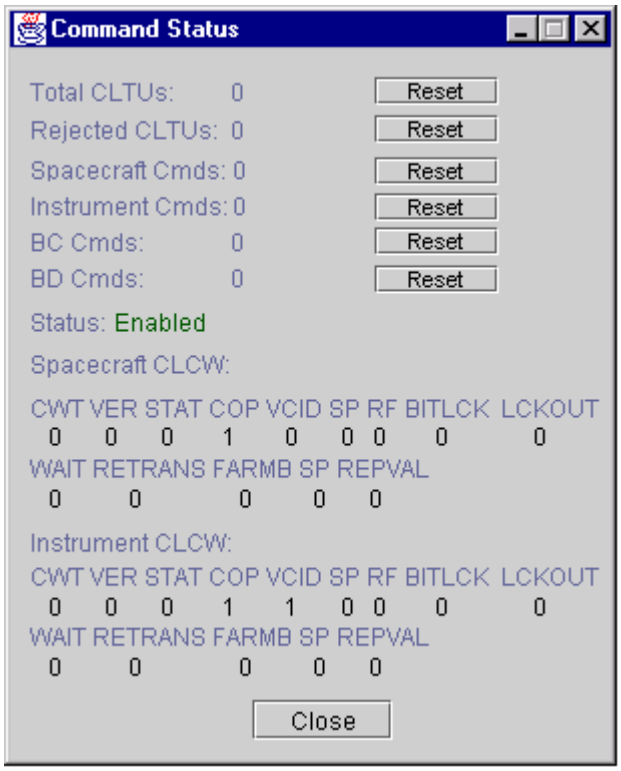
ETSAura-5.2.1.2 Command Menu

Clicking the **Command** button on the main display brings up the command menu.

Menu item	Description
Start	Enable command reception
Stop	Disable command reception
Display Status…	Enable/disable status, command counts, CLCWs by virtual channel
Modify Validation Criteria…	Allows modification of validation options
Display Spacecraft Packet…	Displays spacecraft virtual channel packet contents

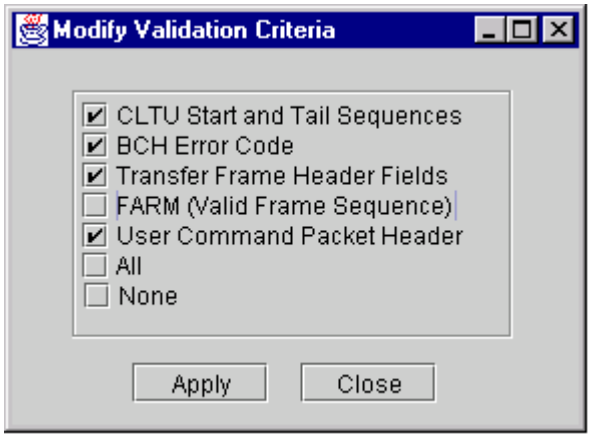
Display Instrument Packet…	Displays instrument virtual channel packet contents
Override CLCWs	Edit the fields of the Command Link Control Words
Display Spacecraft Load…	Display the Spacecraft Memory Load Buffer

ETSAura-5.2.1.2.1 Display Command Status



When the "Display Status…" option is selected, this display is shown. The command status display shows command counters, command enabled or disabled status and the Command Link Control Words (CLCWs) for the spacecraft and instrument virtual channels. Clicking the **Reset** button sets the counter in that row to zero.

ETSAura-5.2.1.2.2 Modify Validation Criteria



When the "Modify Validation Criteria" option is selected, this screen is displayed. The operator may select any combination of the validation options. The operator may enable all of the validation tests by selecting the "All" option or disable all of the validation tests by selecting the "None" option. The **Apply** button is used to activate the new settings. The **Close** button is used to dismiss the screen without changes.

ETSAura-5.2.1.2.2.1 None Validation Option

Even when the "None" option is selected the following validation tests are performed.

- A received CLTU must be long enough to contain a start sequence, a single codeblock, and a tail sequence or it will be rejected and an error event message will be generated.

- A CLTU's length will be checked to see there are enough bytes for a start sequence, tail sequence, and an even multiple of codeblocks. If there are extra bytes, they will be reported in a warning event message. The CLTU will then be processed as if there were no extra bytes. The existence of "extra" bytes may indicate a problem with the source system's formatting of the CLTU.
- The transfer frame header length field must report a value big enough to contain a transfer frame header and at least one byte of data or it will be rejected and an error event message will be generated.
- The transfer frame header length field is compared to the actual length of the passed frame buffer. If there are more data bytes than are reported in the header, these bytes are compared to the fill data byte. If there are any "extra" bytes that are not fill data, a warning event message will be generated. The transfer frame will then be processed and the "extra" bytes will be ignored. The existence of "extra" non-fill bytes may indicate a problem with the source system's formatting of the Transfer Frame.
- The transfer frame header VCID is used to determine whether the frame should be processed as a critical TIE command or passed to the spacecraft or instrument virtual channel Frame Acceptance and Reporting Mechanism (FARM). The VCID must match one of the four expected VCID values stored in the container. When a VCID does not match any of the expected values, the transfer frame is rejected and an error event message is generated.

ETSAura-5.2.1.2.2.2 Validation of CLTU Start and Tail Sequences

Since a received CLTU must contain a valid start and tail sequence to be recognized by the software, disabling validation of these items is not an option. Rather, the software will report appropriate event error messages if the start sequence cannot be located, or if the start sequence is located but the tail sequence is not found.

ETSAura-5.2.1.2.2.3 BCH Error Code Validation Option

When Bose-Chaudhuri-Hocquenghem (BCH) Error Code validation is enabled, the parity byte of each received codeblock is compared to the parity value calculated from the codeblock data area. If a parity comparison fails, both parity bytes are reported in an error event message and the current CLTU is rejected.

ETSAura-5.2.1.2.2.4 Transfer Frame Header Validation Option

When transfer frame validation is enabled, the following validations are done.

- The transfer frame header version field must contain 0 or the frame is rejected and an error event message is generated.
- The transfer frame header spare field must contain 0 or the frame is rejected and an error event message is generated.
- The transfer frame header length is tested to be within the range specified in the ICD. If the length is invalid, the "Incorrect Frame Length" status is stored in the CLCW,

the frame is rejected, and an error event message is generated.

- The transfer frame header length is compared to the actual length of the received buffer. If the frame contains fewer bytes than reported in the header, the "Incomplete Frame" status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- The transfer frame header mode flags are checked. If the mode is "AC", the "Illegal frame type (Type-AC)" status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- If the transfer frame header mode flags specify "BC", the frame data is checked for a valid transfer frame control command. If the frame data does not contain a valid frame control command, the "Illegal Type-BC frame" status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- The transfer frame header VCID field is compared to the VCID field from the FARM's CLCW. If the VCID fields do not match, the "Illegal VCID" status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- If the transfer frame header SCID field does not match the Aura SCID stored in container point CmdSCID, the "Illegal SCID" status is stored in the CLCW, the frame is rejected, and an error event message is generated.

ETSAura-5.2.1.2.2.5 Farm (Valid Frame Sequence) Validation Option

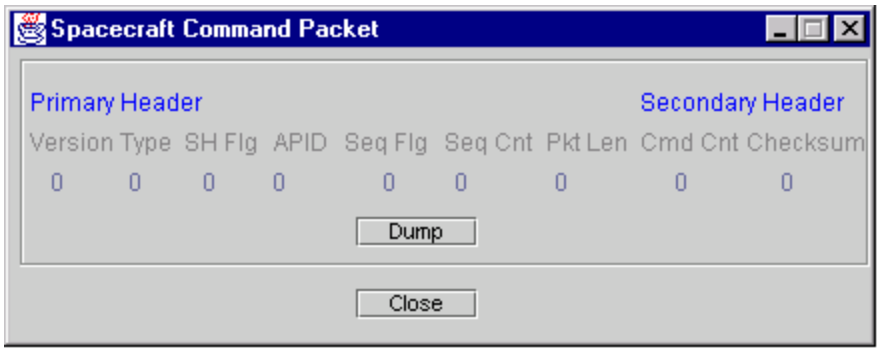
When FARM validation is enabled, the transfer frame header sequence field is subjected to the FARM-1 protocol and the CLCW flags, FARM-B Counter, and Report Value fields will be updated accordingly. Event messages are generated for every transfer frame that fails the acceptance test.

ETSAura-5.2.1.2.2.6 User Command Packet Header Validation Option

When packet validation is enabled, fields in the packet header will be validated as specified in the EOS Aura Spacecraft to Ground ICD.

ETSAura-5.2.1.2.3 Display Spacecraft Packet

This display shows the spacecraft virtual channel command packet header. A dump of the full packet buffer may be requested by pressing the **Dump** button.



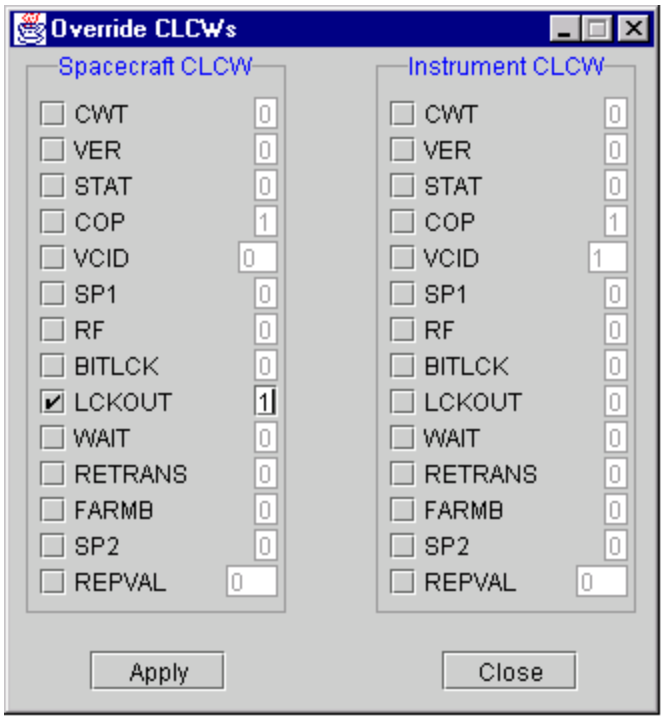
The Dump address region may be displayed in decimal or hexadecimal. The data region may be displayed in hexadecimal or octal. Since this display normally updates as the buffer contents are changed, use the **Pause** button to freeze the current contents. Use of the **Pause** button does not affect command packet reception. When the **Pause** button has been used, the button's label is changed to **Cont** (for continue). Press the **Cont** button to resume screen updates. Note that the display will be updated to show the most recent command packet received. Intermediate packets received while the display was frozen cannot be displayed.

ETSAura-5.2.1.2.4 Display Instrument Packet

This display shows the instrument virtual channel packet header. Optionally, a dump of the full packet buffer may be requested. They are very similar in appearance and function to the spacecraft packet header and packet dump displays that are described in the previous section.

ETSAura-5.2.1.2.5 Override CLCWs

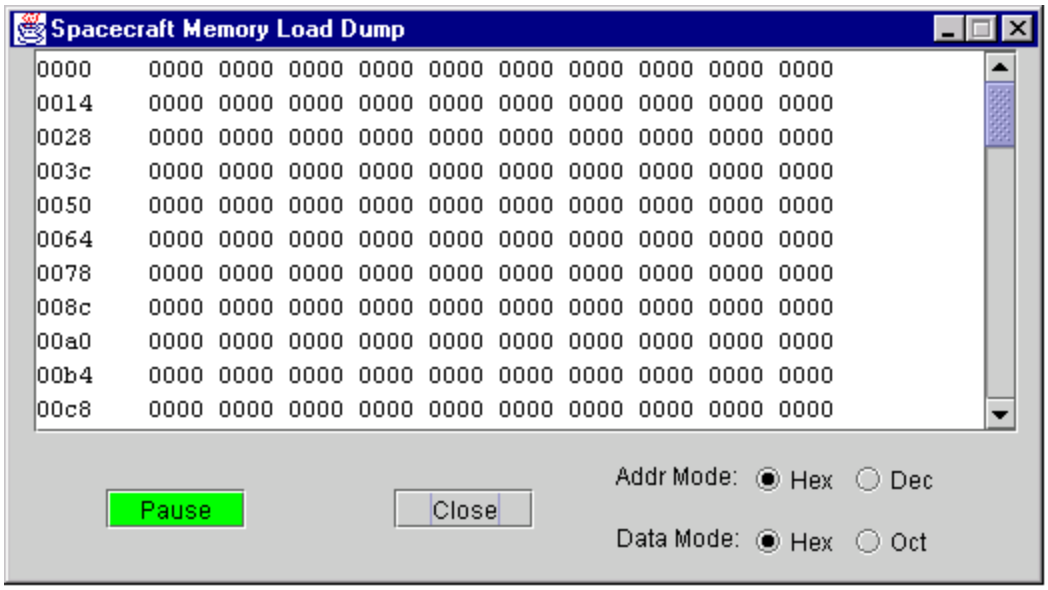
This display shows the Command Link Control Words (CLCWs) for the two virtual channels broken out into bit fields. Any field may be overridden by checking the field's selection box and then typing in the new decimal value. The new value takes affect when the **Apply** button is pressed. Note that since these are bit fields, attempts to assign values that are too large will result in truncation of the value. Changes may be verified by observing the CLCW portion of the Command Status Display. Event messages also show the values assigned.



CLCW Field	Description
CWT	Control Word Type (1 bit)
VER	Version (2 bits)
STAT	Status (3 bits)
COP	Command Operations Procedure (COP) in Effect (2 bits)
VCID	Virtual Channel Identification (6 bits)
SP1	Spare field 1 (2 bits)
RF	No RF Available Flag (1 bit)
BITLCK	No Bit Lock (1 bit)
LCKOUT	Lockout Flag (1 bit)
WAIT	Wait Flag (1 bit)
RETRANS	Retransmission Flag (1 bit)
FARMB	FARM-B Counter (2 bits)
SP2	Spare field 2 (1 bit)
REPVAL	Report Value (8 bits)

ETSAura-5.2.1.2.6 Display Spacecraft Load

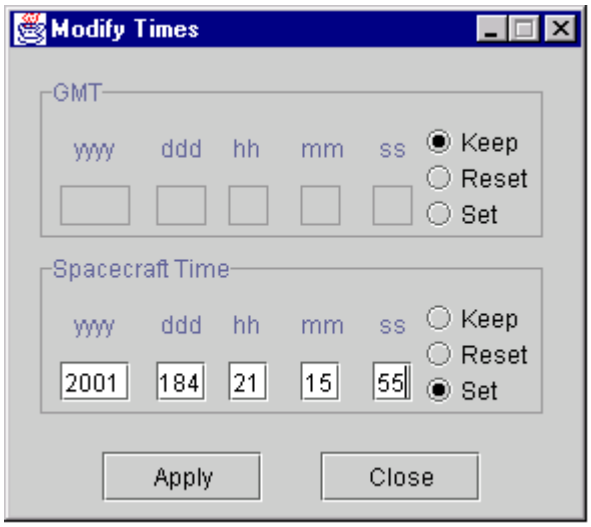
When the "Display Spacecraft Load…" option is selected, the buffer containing the accumulated spacecraft load packet data is displayed.



ETSAura-5.2.1.3 Time

The time area of the main display shows the current Greenwich Mean Time (GMT) and spacecraft time fields. The GMT is a reflection of the simulated time of the ETSAura module, i.e., what time will be used to model external events (transitions to day or night, for example) within the module. It is not currently used. The spacecraft time is the time that the spacecraft thinks it is. In particular, it is the time (in the appropriate format) that goes into the secondary header of the telemetry packets generated by the ETSAura module. Clicking the **Time** button brings up a menu with one selectable item, "Modify Times…".

ETSAura-5.2.1.3.1 Modify Times Display



This display allows the user to set the GMT or spacecraft time manually, retain the time currently being used, or reset the time to the current system time.

Clicking the **Apply** button will cause the requested actions to take effect.

Clicking the **Close** button closes the window with no further action taken.

Field	Description
yyyy	Year
ddd	Day of year
hh	Hour of day
mm	Minute of hour
ss	Second of minute
Keep	Don't update time
Reset	Change time to current system time
Set	Set time to value supplied

ETSAura-5.3 Remove

Clicking the "Remove" option from the module pop-up menu causes this module to be removed from the project. This option is not available during run-time.

ETSAura-5.4 About

Clicking the "About" option from the module pop-up menu requests a display that lists generic information about the module such as the numbers of input and output channels.

ETSAura-6.0 Special Operating Instructions

ETSAura-6.1 Serial Mode Operation

In order to run the ETSAura module in serial mode, select serial mode from the Select Simulation Mode Display during project configuration. ETSAura output channel 3 must be linked to a serial output module. A serial I/O card is required on the server platform.

ETSAura-6.2 Triggering Scenarios with Commands

A powerful feature of the ETSAura module is to automatically trigger the execution of specified scenario files when specific commands are recognized from the database. The project must contain a scenario module. The output channel from the scenario module must be connected to input channel 1 of the ETSAura module.

ETSAura-6.2.1 Using the Default Scenario Directory

Create a file named **Command-Scenario.txt** in the release's scenario directory. (The scenario directory should be at the same level as the properties, elog and save directories.) The scenario files to be executed should be placed in this directory. The command trigger definition file is read immediately after the database is loaded. The trigger information is stored in memory. Event messages will be generated for any invalid command mnemonics and any invalid scenario filenames in the definition file.

Format of Command-Scenario.txt File
Command-scenario pairs are specified by the following format: `command mnemonic	scenario filename` One pair is specified per line with the "|" character as the delimiter. Leading and trailing spaces and tab characters will be removed so these characters may be used to align the data. The scenario filename may be specified as a full or relative pathname. Relative pathnames will start from the scenario directory. Each command mnemonic should only appear in the definition file once.
Any line starting with a semicolon ";" in column one is ignored as a comment. Blank lines are also ignored.

ETSAura-6.2.2 Using an Alternate Scenario Directory

If a user wishes to use an alternate set of scenario files, place them in a specific directory on the disk, for example, D:\My-Scenarios. Place the command trigger definition file, Command-Scenario.txt, in this directory as well. Set the ScenarioPath variable to this directory.

Every reference to a scenario file, from within the command trigger definition file or from within a scenario file may either use the file's fully specified path name or a relative file name. For example, a fully specified entry in the definition file might be:

```
CDH_EXECUTE_TIENOP | D:\My-Scenarios\tienop.txt
```

If any scenario file calls tienop.txt, the fully specified format for the calling statement within the scenario file is

```
Start scenario D:\My-Scenarios\tienop.txt
```

When the simulator is started, the default command trigger definition file will be read from the default directory. To force replacement of the default trigger information with the alternate definition file send directives similar to the following to the ETSAura module. Do not use quotes around the alternate directory name. The ending slash is required syntax.

```
Set scenarioPath D:\My-Scenarios\
Set readScenFile 0
```

These set directives must be done each time the simulator is started. To make this easier, place both directives in a scenario file in the alternate directory. After the simulator is running, use the Scenario module GUI file browser to navigate to the alternate directory and execute this file.

ETSAura-6.2.3 Update/Replace Command Trigger Definitions

During run-time the operator may request that the Command-Scenario.txt definition file be read in again replacing the previous definitions. On the directive line to the ETSAura module, enter the following. Check the event message log for status and error messages.

```
Set ReadScenFile 0
```

ETSAura-6.2.4 Using Full Path Scenario Filenames

Filenames starting with a slash and filenames containing the colon character are considered to be full path filenames. Full path filenames may be used in the Command-Scenario.txt definition file and within scenario script files to precisely identify script files to be executed.

ETSAura-6.2.5 Using Relative Path Scenario Filenames

Filenames that do not contain the colon character and do start with a slash are considered to be relative path filenames. Within the ETSAura module, the directory specified by the ScenarioPath variable will be added to the beginning of any relative filenames in the Command-Scenario.txt file. If one scenario script starts another script using a relative path, the directory of the executing script is used to complete the path. Note in the example that relative paths may contain subdirectory names.

Example 1: D:/Scenarios/Command-Scenario.txt contains the following lines:

```
CDH_COMMAND_A     |     cdh_command_a.txt
INSTR1_COMMAND_Y  |     instr1/command_Y.txt
INSTR2_COMMAND_X  |     instr2/command_x.txt
```

ScenarioPath must be set to D:/Scenarios/ before reading in this Command-Scenario.txt file. Full path filenames are generated during the reading of this file so that existence of the specified script files may be verified. The full path names of these scenario files are stored and their full path names are passed to the Scenario module when triggered by commands. The fully specified names of the scenario scripts in the lines above are D:/Scenarios/cdh_command_a.txt, D:/Scenarios/instr1/command_Y.txt, and D:/Scenarios/instr2/command_x.txt

Example 2: Scenario file D:/Scenarios/Scenario1.txt contains the following line.

```
Start Scenario initialize_variables.txt
```

When the scenario1.txt script is executed, it will attempt to execute a file with the full path name of D:/Scenarios/initialize_variables.txt.

ETSAura-6.3 1553 Bus Interface Operation

The ETSAura module may be configured to function as an instrument or spacecraft subsystem emulator by connecting it to a 1553 Bus interface module. This 1553 module should be linked to pass command packets to ETSAura's input channel 2 and linked to receive telemetry packets from ETSAura's output channel 1. The simulator only interfaces with bus A. Bus B is not currently supported. The simulator must be operated in IP mode. Other than that, the only special configuration setup has to do with command receipt.

Commands received from a 1553 Bus are not in the CCSDS CLTU format. The CLTU codeblock and transfer frame level processing must be bypassed. To configure the ETSAura module for 1553 Bus operation, set the Command Input Type to 1553 (or any non-zero value) during project initialization, or enter the directive

```
Set CommandInputMode 1553
```

Since commands received from the 1553 bus do not have transfer frame headers, the APID from the packet header must be used to determine the appropriate virtual channel.

ETSAura-6.4 Synchronization with External Packet Time

The ETSAura module may be configured to receive external telemetry packets from input channel 3. In another configuration, the ETSAura module may be configured to receive command packets from a 1553 bus module. In both cases, the ETSAura module may be configured to synchronize its spacecraft time with a received CCSDS Unsegmented Time Code (CUC) time. There are two important container items that control the time.

The UseExternalPacketTime variable is set to one, indicating enabled, by default. If synchronization with external time is not desired, this variable should be set to zero.

The last external time received is stored in the variable ExternalCUCTime.

ETS Ground Station Module (ETSGS)

ETSGS-1.0 Overview

The EOS Test System Ground Station (ETSGS) module is responsible for receiving telemetry, adding an EDOS Service header (ESH) to telemetry packets and transmitting the data as EDOS Data Units (EDUs). The ETSGS module provides a user interface to set EDOS Service header fields. The ETSGS module also receives Command Data Blocks (CDB), validates and strips off the Ground Message Header (GMH), and transmits Command Link Transmission Units (CLTUs).

ETSGS-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	EDOS Command Data Blocks	Checks GMH fields	Ground Message Header removed, resulting data transmitted through output channel 1
2	Telemetry packets	None	EDOS Service Header added, resulting data transmitted through output channel 2
3	Command Link Control Words (CLCWs)	None	EDOS Service Header added, resulting data transmitted through output channel 3

ETSGS-3.0 Outputs

Ch.	Description
1	Command stream with acquisition sequence and CLTUs
2	EDOS Data Units (EDUs) containing telemetry packets
3	EDOS Data Units (EDUs) containing CLCWs

ETSGS-4.0 Container Items

The ETSGS module accepts operator directives and is capable of receiving directives from a Scenario module. Use the Set and Get directives to access items with a fixed type. Use the SetBuffer and GetBuffer directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case-sensitive.

ETSGS-4.1 Counters

Name	Type	Description
GSCmdReceiveCount	Fixed	Number of command blocks received
GSTlmCh1TransmitCount	Fixed	Number of channel 1 blocks sent
GSClcwTransmitCount	Fixed	Number of channel 2 blocks sent

ETSGS-4.2 Buffers

Name	Type	Description
GSCmdBuffer	Buffer	Most recent command block received
GSTlmCh1Buffer	Buffer	Most recent telemetry block sent
GSClcwBuffer	Buffer	Most recent CLCW block sent

ETSGS-4.3 Ground Message Header (GMH) fields

Name	Type	Description
GSCmdEnableHeaderValidation	Fixed	GMH validation flag (1 = enabled, 0 = disabled)
GSCmdExpectedDestination	Fixed	GMH expected destination (1)
GSCmdExpectedMsgType	Fixed	GMH expected message type (3)
GSCmdExpectedSCID	Fixed	GMH expected SCID (0x00)
GSCmdExpectedSource	Fixed	GMH expected source (4)
GSCmdSequenceCount	Fixed	GMH expected sequence count (0)

ETSGS-4.4 EDOS Data Unit (EDU) Header fields

There are two sets of EDU container items, one set for each telemetry channel. Each EDU has an EDOS Service Header (ESH). To access fields from telemetry channel 1, insert "GsTlmch1" for "<ch>" in the names provided in the following table. To access fields from telemetry channel 2, insert "GsClcw" for "<ch>". For example, the container name for the ESH version field of the most recent channel 1 block is "EduHeaderGsTlmCh1Version". The container name for the same field in the most recent channel 2 block is "EduHeaderGsClcwVersion".

Name	Type	Description
EduHeader<ch>Version	Fixed	ESH version field
EduHeader<ch>SDUtype	Fixed	ESH SDU type field
EduHeader<ch>PortId	Fixed	ESH Port Id
EduHeader<ch>SourceDiscontinuityFlag	Fixed	ESH Source VCDU Discontinuity Flag
EduHeader<ch>PlaybackDataFlag	Fixed	ESH Contains Playback Data Flag
EduHeader<ch>RecoveryProcessingFlag	Fixed	ESH Recovery Processing Indicator
EduHeader<ch>TestDataFlag	Fixed	ESH Test Data Indicator
EduHeader<ch>CRCfailureFlag	Fixed	ESH CRC Failure Indicator
EduHeader<ch>PathDiscontinuityFlag	Fixed	ESH Path SDU Discontinuity Flag
EduHeader<ch>PktLengthErrorFlag	Fixed	ESH Packet Length Indicator
EduHeader<ch>PacketFillFlag	Fixed	ESH Packet Fill Indicator
EduHeader<ch>FirstSpare	Fixed	ESH First Spare Field
EduHeader<ch>SCID	Fixed	ESH SCID
EduHeader<ch>VCID	Fixed	ESH VCID

EduHeader<ch>FillLocation	Fixed	ESH Fill Data Location
EduHeader<ch>SecondSpare	Fixed	ESH Second Unused Field
EduHeader<ch>RSstatus	Fixed	ESH RS Validation Status
EduHeader<ch>HeaderErrorCount	Fixed	ESH VCDU Header Error Count
EduHeader<ch>TotalErrorCount	Fixed	ESH Total Error Count

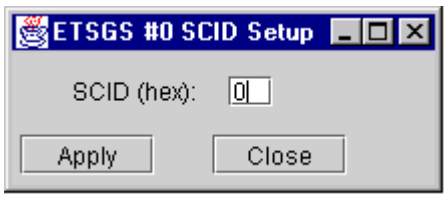
ETSGS-5.0 Displays

To access the displays for this module, first click on the center of the module in the project window. The module pop-up menu will appear.

Module Pop-Up Menu Item	Description
Configuration	Setup expected Spacecraft ID (SCID)
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

ETSGS-5.1 Configuration

There is one display available at configuration time.



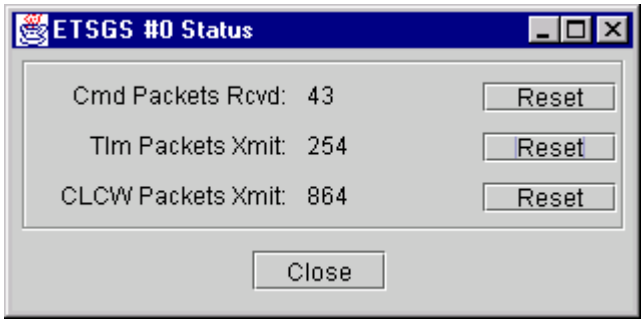
This display is used to specify the expected Spacecraft Identifier (SCID) field for Ground Message Header validation. This value can be saved and restored as part of a project. Click **Apply** for the entered hexadecimal value to take effect. Use **Close** to dismiss the display. This value will also be used in the SCID fields of outgoing EDOS Service Headers for housekeeping and CLCW telemetry unless overridden. See sections ETSGS-5.2.4 Modify Tlm Header and ETSGS-5.2.5 Modify CLCW Header for more information.

ETSGS-5.2 Run-time

There are six displays available from the Run-time menu.

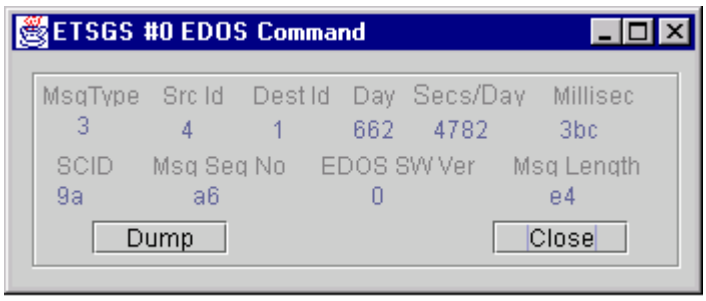
Run-time Menu Item	Description
Show Status	Show counts of commands and telemetry packets
Show Cmd Packet	Display the command buffer
Show Tlm CH1 Packet	Display the housekeeping telemetry buffer
Modify Tlm Header	Modify header fields in housekeeping telemetry
Modify CLCW Header	Modify command link control word headers
Set/Display GMT	Display time fields for possible modification

ETSGS-5.2.1 Show Status

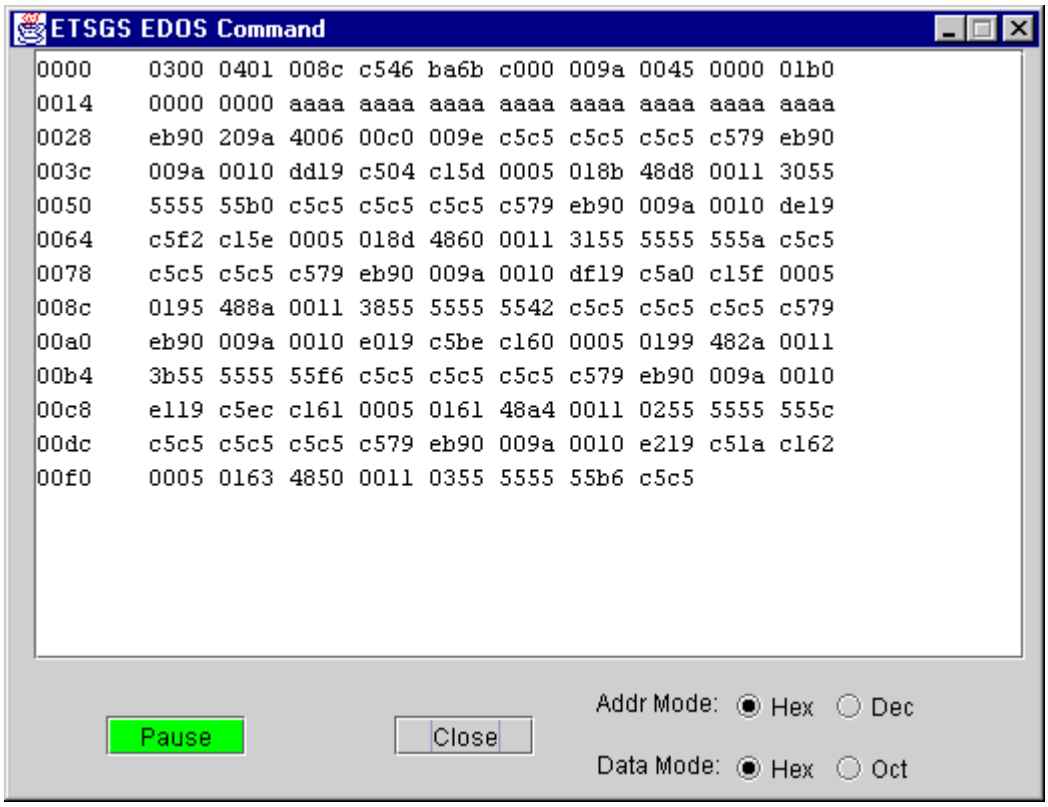


The status display shows the number of command blocks received and the number of housekeeping telemetry blocks transmitted. The **Reset** button may be used to zero out the counter in the same row.

ETSGS-5.2.2 Show Command Packet

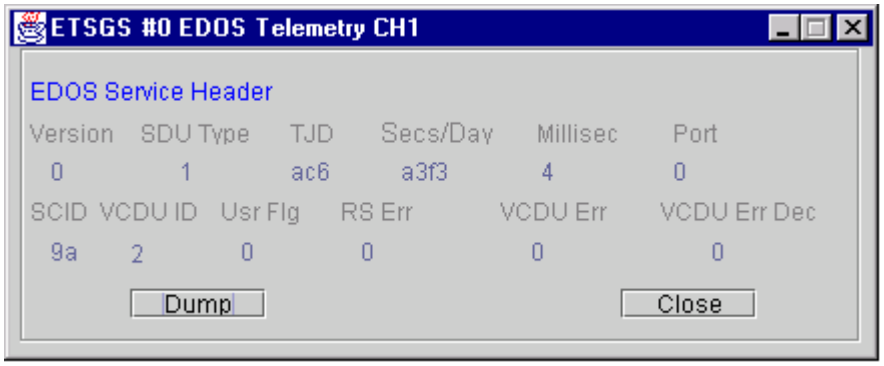


The command packet display shows the header and, optionally (by pressing the **Dump** button), the contents of the most recent command block received, including the Ground Message Header.

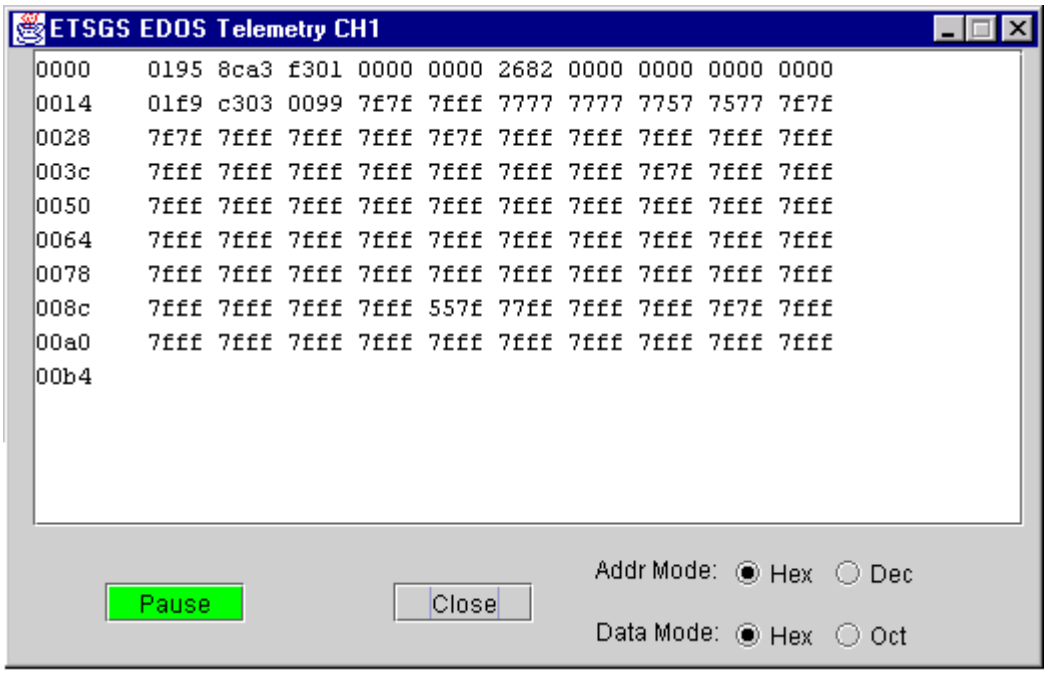


The address field of the dump display may be toggled between hexadecimal and decimal display formats. The data portion may be toggled between hexadecimal and octal formats.

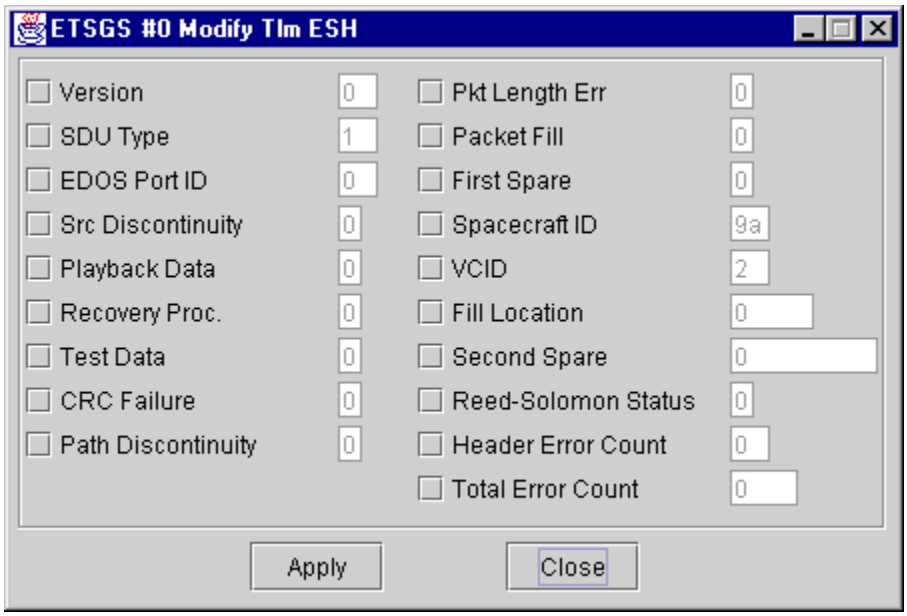
ETSGS-5.2.3 Show Telemetry Channel 1 Packet



The telemetry channel 1 packet display shows the header and, optionally (by pressing the **Dump** button), the contents of the most recent telemetry block as transmitted through telemetry channel 1 (module output channel 2).



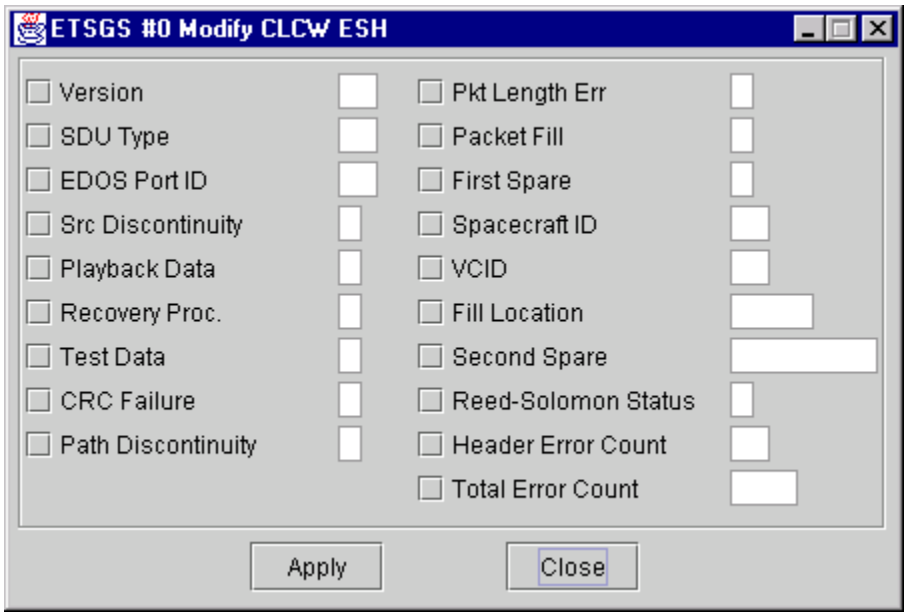
ETSGS-5.2.4 Modify Telemetry Header



The Modify Tlm ESH display allows the operator to change values in the EDOS service header of housekeeping telemetry. Some these changes are in effect for a single packet. Fields marked with an asterisk (*) in the following table are non-sticky. That is, they reset to zero after the transmission of one packet. To change the PB5 Time field, use the Set/Display GMT display.

Modify Tlm ESH Fields	Description
Version	ESH Version Number
SDU Type	SDU Type
EDOS Port ID	EDOS Port ID
Src Discontinuity*	Source VCDU Sequence Counter Discontinuity
Playback Data	VCDU Contains Playback Data
Recovery Proc	Recovery Processing Indicator
Test Data	Test Data Indicator
CRC Failure*	CRC Failure Indicator
Path Discontinuity*	Path SDU Source Seq. Counter Discontinuity
Pkt Length Err*	Packet Length Error
Packet Fill*	Packet Fill Indicator
First Spare	Spare field
Spacecraft ID	Spacecraft identification code
VCID	VCDU Id
Fill Location	Location 1st byte Fill Data For Path SDU
Second Spare	Spare field
Reed-Solomon Status*	RS Error Control Flag
Header Error Count	Source VCDU Header Error Decode Results
Total Error Count	Source VCDU Error Decode Results

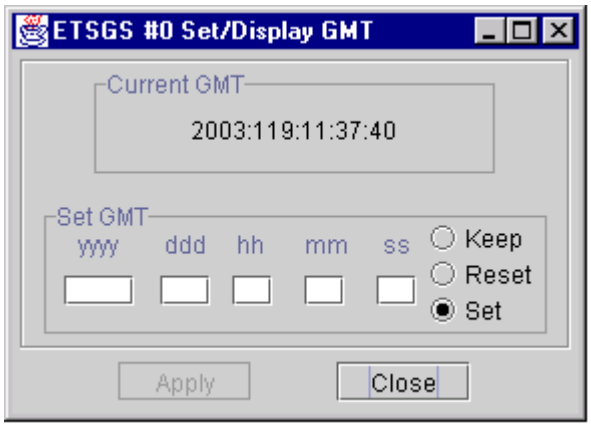
ETSGS-5.2.5 Modify CLCW Header



The ModifyCLCW ESH display allows the operator to change values in the EDOS service header of command link control words. Some these changes are in effect for a single packet. Fields marked with an asterisk (*) in the following table are non-sticky. That is, they reset to zero after the transmission of one packet. To change the PB5 Time field, use the Set/Display GMT display.

Modify CLCW ESH Fields	Description
Version	ESH Version Number
SDU Type	SDU Type
EDOS Port ID	EDOS Port ID
Src Discontinuity*	Source VCDU Sequence Counter Discontinuity
Playback Data	VCDU Contains Playback Data
Recovery Proc	Recovery Processing Indicator
Test Data	Test Data Indicator
CRC Failure*	CRC Failure Indicator
Path Discontinuity*	Path SDU Source Seq. Counter Discontinuity
Pkt Length Err*	Packet Length Error
Packet Fill*	Packet Fill Indicator
First Spare	Spare field
Spacecraft ID	Spacecraft identification code
VCID	VCDU Id
Fill Location	Location 1st byte Fill Data For Path SDU
Second Spare	Spare field
Reed-Solomon Status*	RS Error Control Flag
Header Error Count	Source VCDU Header Error Decode Results
Total Error Count	Source VCDU Error Decode Results

ETSGS-5.2.6 Set/Display GMT



This display shows the GMT time that is used in the EDOS Server Header. The user may **Set** the GMT time fields manually, **Keep** the time currently being used, or **Reset** time to the current system time.

Click the **Apply** button for the actions to take effect. Click the **Close** button to dismiss the window with no further action taken.

ETSGS-5.3 About

To display generic information about the ETSGS module, choose the "About" option from the module pop-up menu.

ETSGS-6.0 Special Operating Instructions

There are no special operating instructions for the current release.

EOS Packet Extractor (EOSXtract) Module

EOSXtract-1.0 Overview

The EOS Packet Extractor (EOSXtract) module receives CADUs and extracts and transmits telemetry packets. It also builds and transmits EOS enhanced CLCW packets. The SCID, VCID and VCDU counter fields are validated. CADUs with invalid SCID or VCID fields are dropped. Fill CADUs are counted only. If the VCDU counter is not in sequence, the partial packet data is dropped and processing resumes with the beginning of the next packet. EOS Enhanced CLCWs are created from the CLCW from the data zone and the VCDU primary header in each CADU. While this module may be applicable to other missions, it was written specifically for parsing EOS formatted CADUs, extracting EOS telemetry APID packets and creating EOS Enhanced CLCWs.

EOSXtract-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	CADUs (256 bytes each)	SCID, VCID and VCDU counter fields must match expected values.	EOS telemetry APID packets and CLCWs are extracted from the CADUs. EOS Enhanced CLCWs are built.

EOSXtract-3.0 Outputs

The extracted telemetry and EOS Enhanced CLCW packets are transmitted on separate output channels. These packets do not have EDOS Service Headers.

Ch.	Description
1	EOS telemetry application packets
2	EOS Enhanced Command Link Control Words (CLCWs)

EOS Enhanced CLCW Format

Fill		VCDU	Identifier		CLCW
Unsigned integer	Version Number	Spacecraft Identifier (SCID)	Virtual Channel ID (VCID)	VCDU Counter	CCSDS CLCW
24 bits	2 bits	8 bits	6 bits	24 bits	32 bits
3 bytes	5 bytes				4 bytes

EOSXtract-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the **Set** and **Get** directives to access items with a fixed type. Use

the **SetBuffer** and **GetBuffer** directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

EOSXtract-4.1 Configurable Parameters

Name	Type	Description
SCID	Fixed	Spacecraft Identifier (defaults to 0xCC)
DataVCID	Fixed	Data Virtual Channel (defaults to 2)
FillVCID	Fixed	Fill Virtual Channel (defaults to 63)

EOSXtract-4.2 Processing Buffers

Name	Type	Description
CaduBuffer	Buffer	CADU Buffer (256 bytes)
ApidBuffer	Buffer	Telemetry packet buffer (256 bytes)
CLCWBuffer	Buffer	EOS Enhanced CLCW Buffer (12 bytes)

EOSXtract-4.3 Processing Counters

Name	Type	Description
VcduCount	Fixed	VCDU counter from CADU primary VCDU header
CaduCount	Fixed	Number of CADUs received
ErrsCount	Fixed	Number of CADUs dropped due to validation errors
FillCount	Fixed	Number of fill CADUs received
ClcwCount	Fixed	Number of CLCWs extracted
ApidCount	Fixed	Number of APID packets extracted

EOSXtract-4.4 Debug Flags

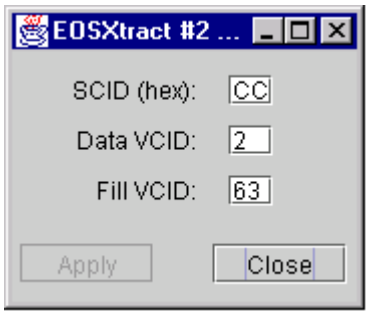
Name	Type	Description
DebugEnabled	Fixed	When this debug flag (0=disabled, 1=enabled) is enabled, all of the following debug event messages are generated.
DebugCADU	Fixed	When this debug flag (0=disabled, 1=enabled) is enabled, debug event messages are generated about CADU header fields and partial packet processing.
DebugAPID	Fixed	When this debug flag (0=disabled, 1=enabled) is enabled, debug event messages are generated to show the APID and length in bytes of transmitted packets.
DebugCLCW	Fixed	When this debug flag (0=disabled, 1=enabled) is enabled, debug event messages are generated to show the VCID for transmitted CLCW packets.

EOSXtract-5.0 Displays

To access the displays, click in the center of the module in the project window.

Module Pop-Up Menu Item	Description
Configure	Access the module's configuration screen
Run-time	Access the module's Run-time menu
Remove	Remove module from the project
About	Display generic module information

EOSXtract-5.1 Configuration Display



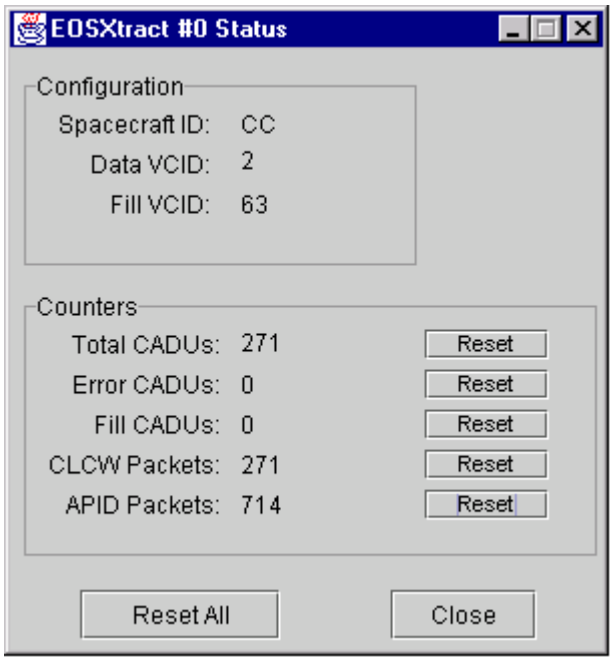
This module processes a single CADU stream at a time. The operator may modify the expected SCID and expected VCID fields. The SCID should be entered as a hexadecimal number. The VCIDs should be entered as decimal numbers.

Use the **Apply** button for changes to take effect.
Use the **Close** button to dismiss this display.

EOSXtract-5.2 Run-Time Menu

Run-Time Menu Item	Description
Show Status	Request module status display
CADU Buffer Dump	Request dump display of the CADU buffer
APID Buffer Dump	Request dump display of the APID processing buffer
CLCW Buffer Dump	Request dump display of the EOS Enhanced CLCW buffer
Resume	Resume the module's execution
Pause	Pause the module's execution

EOSXtract-5.2.1 Show Status Display



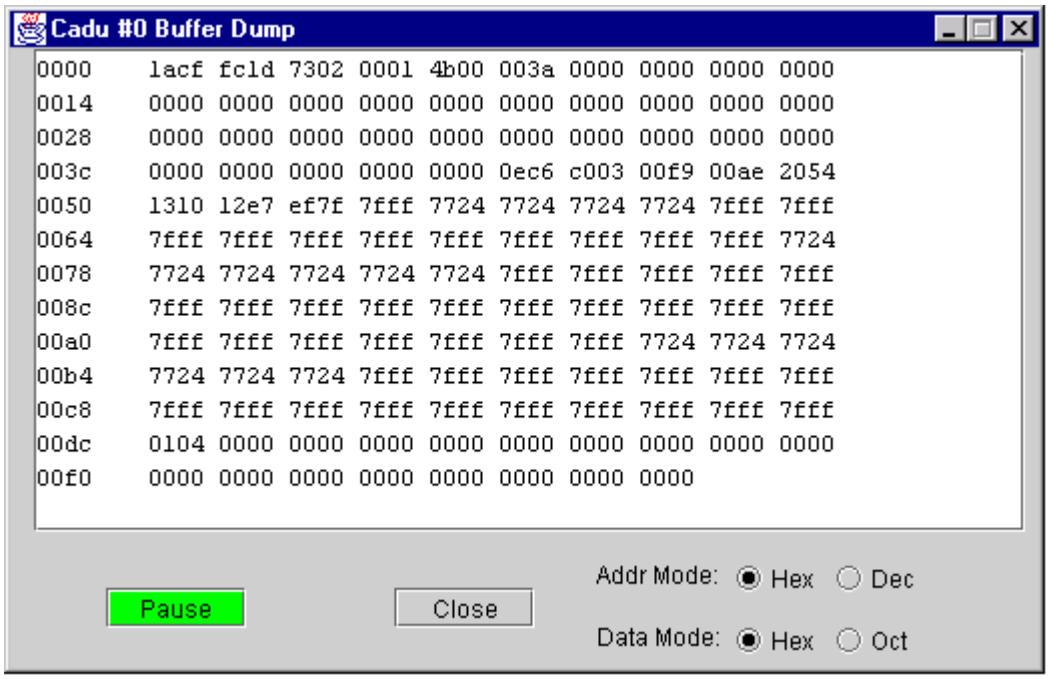
This display shows the current configuration and processing status of the module.

Use any **Reset** button to clear its associated individual counter.

The **Reset All** button may be used to reset all of the counters to zero.

Use the **Close** button to dismiss the display.

EOSXtract-5.2.2 Dump CADU Buffer Display



This display shows the most recently received CADU. This a standard SIMSS dump display with the following functionality. Addresses of the data bytes may be displayed in hexadecimal or decimal by setting the "Addr Mode" field. Data may be displayed in hexadecimal or octal formats by setting the "Data Mode" field. Clicking on the **Pause** button will interrupt the display's updates. The button's title will change to **Cont**. Click on the **Cont** button to resume status updates. Use the **Close** button to dismiss the display.

EOSXtract-5.2.3 Dump APID Buffer Display

This is a standard SIMSS dump display of the APID buffer. Its format is similar to the CADU display in the previous section. Bytes are copied to this buffer from the CADU as APID packets are extracted. Since there are multiple complete and partial APID packets within each CADU, this buffer is constantly changing and rarely shows a complete packet.

EOSXtract-5.2.4 Dump CLCW Buffer Display

This is a standard SIMSS dump display of the EOS Enhanced CLCW Buffer. Its format is similar to the CADU and APID dump displays in the previous sections. Since only one EOS Enhanced CLCW is built from each CADU, this display is not as volatile as the APID buffer display.

EOSXtract-5.2.5 Resume

Clicking the "Resume" option causes this module to restart execution.

EOSXtract-5.2.6 Pause

Clicking the "Pause" option causes this module to stop execution.

EOSXtract-5.3 Remove

Clicking the "Remove" option from the module pop-up menu causes this module to be removed from the project. This option is not available during run-time.

EOSXtract-5.4 About

Clicking the "About" option from the module pop-up menu requests a display that lists generic information about the module such as the numbers of input and output channels.

EOSXtract-6.0 Special Operating Instructions

There are no special operating instructions for this release.

1553 Bus Interface (ETS1553Bus) Module

ETS1553Bus-1.0 Overview

The 1553 Bus Interface (ETS1553Bus) module receives command packets from a 1553 bus and sends telemetry packets to the bus. It provides a configurable interface for four Remote Terminals (RTs) on the bus.

While this module may be suitable for other applications, it was written specifically for the EOS Aura mission. The ETSAura module should be linked to receive instrument command packets from the ETS1553Bus module and should also be linked to send instrument telemetry packets to the ETS1553Bus module.

ETS1553Bus-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Telemetry packets	APID is compared to the RT configuration.	APID is used to determine the RT for transmission. If APID is not mapped to an RT, it is not sent out on the bus.

ETS1553Bus-3.0 Outputs

Ch.	Description
1	Command and time synchronization packets received from the 1553 bus

ETS1553Bus-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the **Set** and **Get** directives to access items with a fixed type. Use the **SetBuffer** and **GetBuffer** directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

ETS1553Bus-4.1 Configurable Parameters

Name	Type	Description
Rmt001	Fixed	First remote terminal bus address
RmtAPIDfrom001	Fixed	Start of APID range processed by first RT
RmtAPIDto001	Fixed	End of APID range processed by first RT
Rmt002	Fixed	Second remote terminal bus address
RmtAPIDfrom002	Fixed	Start of APID range processed by second RT
RmtAPIDto002	Fixed	End of APID range processed by second RT
Rmt003	Fixed	Third remote terminal bus address
RmtAPIDfrom003	Fixed	Start of APID range processed by third RT

RmtAPIDto003	Fixed	End of APID range processed by third RT
Rmt004	Fixed	Fourth remote terminal bus address
RmtAPIDfrom004	Fixed	Start of APID range processed by fourth RT
RmtAPIDto004	Fixed	End of APID range processed by fourth RT

ETS1553Bus-4.2 Processing Counters

In the container item names below, substitute 1 through 5 for the "<row>" characters. Rows 1 through 4 correspond to the RTs configured previously. The 5th row counters are for unexpected packets.

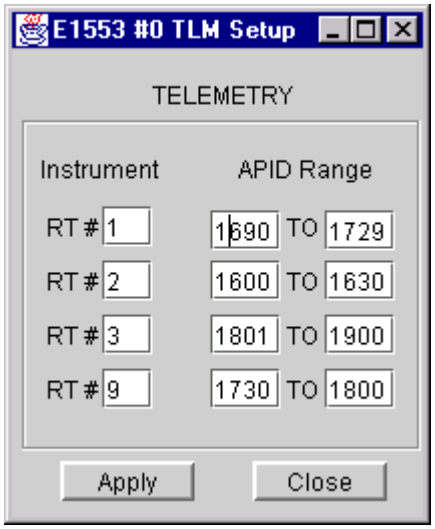
Name	Type	Description
CmdRec<row>	Fixed	Count of command packets received on referenced RT
CmdXmitted<row>	Fixed	Count of command packets processed on referenced RT
CmdAPID00<row>	Fixed	Most recent command APID received on referenced RT
CmdPkt00<row>	Fixed	Most recent command packet length on referenced RT
RmtRec<row>	Fixed	Count of telemetry packets received for referenced RT
RmtXmitted<row>	Fixed	Count of telemetry packets transmitted on referenced RT
RmtAPID00<row>	Fixed	Most recent telemetry APID received on referenced RT
RmtPkt00<row>	Fixed	Most recent telemetry packet length on referenced RT

ETS1553Bus-5.0 Displays

To access the displays, click in the center of the module in the project window.

Module Pop-Up Menu Item	Description
Configure	Access the module's configuration screens
Run-time	Access the module's Run-time menu
Remove	Remove module from the project
About	Display generic module information

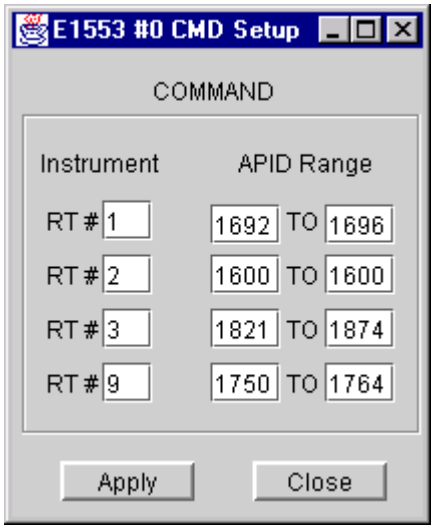
ETS1553Bus-5.1 Configuration Displays



For each of the RT rows, enter the 1553 bus address number and the full range of telemetry APIDs associated with an instrument. Each row must contain a unique range.

Use the **Apply** button for changes to take effect.

Use the **Close** button to dismiss this display.



For each of the RT rows, enter the 1553 bus address number and the full range of Command APIDs associated with an instrument. Each row must contain a unique range.

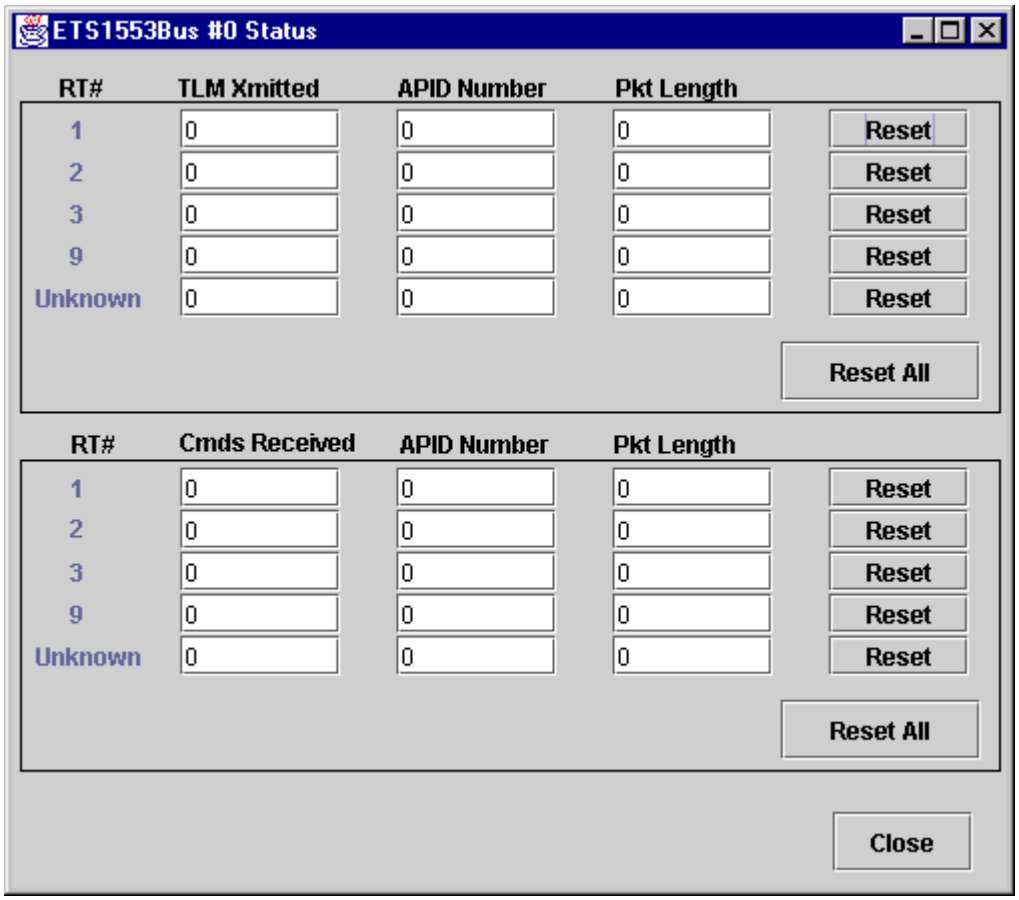
Use the **Apply** button for changes to take effect.

Use the **Close** button to dismiss this display.

ETS1553Bus-5.2 Run-Time Menu

Run-Time Menu Item	Description
Show Status	Request module status display
Resume	Resume the module's execution
Pause	Pause the module's execution

ETS1553Bus-5.2.1 Status Display



This display shows the current processing status of the module. While the simulator is running, this display will show counts of all packets transmitted and received over each Remote Terminal. If a command packet is received from the bus or a telemetry packet is received from another module that does not fall within any of the defined APID ranges, it will be displayed in the appropriate "Unknown" row and will not be forwarded by the ETS1553Bus module.

Use any **Reset** button to clear its associated row's counters. The **Reset All** button may be used to reset all of the counters in its region to zero. Use the **Close** button to dismiss the display

ETS1553Bus-5.2.2 Resume

Clicking the "Resume" option from the module run-time menu causes the execution of this previously paused module to restart.

ETS1553Bus-5.2.3 Pause

Clicking the "Pause" option from the module run-time menu causes this module to stop its execution. The "Configure" option from the module pop-up menu becomes available.

ETS1553Bus-5.3 Remove

Clicking the "Remove" option from the module pop-up menu causes this module to be removed from the project. This option is not available during run-time.

ETS1553Bus-5.4 About

Clicking the "About" option from the module pop-up menu requests a display that lists generic information about the module such as the numbers of input and output channels.

ETS1553Bus-6.0 Special Operating Instructions

Only the A-side bus is supported.

Frame Monitor Module

FrameMonitor-1.0 Overview

The Frame Monitor module provides the capability to receive asynchronous data from the upper module, synchronize data based on sync patterns and frame size parameters, and output data to next modules.

FrameMonitor-2.0 Input

Channel	Description
1	Receive data from upper module through this channel

FrameMonitor-3.0 Output

Channel	Description
1	Synchronous data is output through this channel

FrameMonitor-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the **Set** and **Get** directives to access items with a fixed or string type. Use the **SetBuffer** and **GetBuffer** directives on buffer types. Although names in the following table contain upper and lower case, directive lines are not case sensitive.

Name	Type	Description
Bitrate	Fixed	Bit rate of the input data, averaged over the last second.
FMDataType	String	Current expected input data type. 0 implies telemetry; 1 implies command
FMPatternField	String	Frame Sync pattern of the data being received.
FMSyncParamLength	String	Expected number of bits of Frame Sync length
FMPolarity	String	Polarity of the expected data. 0 implies True, 1 implies Inverted, and 2 enables the Automatic Polarity Correction logic
FMIDLocation	String	Offset in bytes from the start of the data frame to the start of the subframe sync pattern.

FMIDLength	String	Length of the subframe sync in bits.
FMMaxID	String	Maximum value allowed for the subframe sync.
FMMinID	String	Minimum value allowed for the subframe sync.
FMSubFrameFlag	String	1 indicates a subframe is present; 0 indicates there is no subframe
FMFrameParamLength	String	Length of the data frame in bytes.
FMErrorMode	String	Number of bit errors allowed in the frame sync pattern before synchronization is lost. 0 means no errors allowed, 1 means one error allowed, 2 means two errors allowed, and 3 means three errors allowed.
FMBitRate	String	The expected bit rate of the incoming data. If different bit rates are expected, enter the lowest bit rate expected. This will help to prevent time-out and loss of sync.
FMPDFStreamName	String	Name of the Pre-defined Configruation File from which configuration parameters are read during initialization.

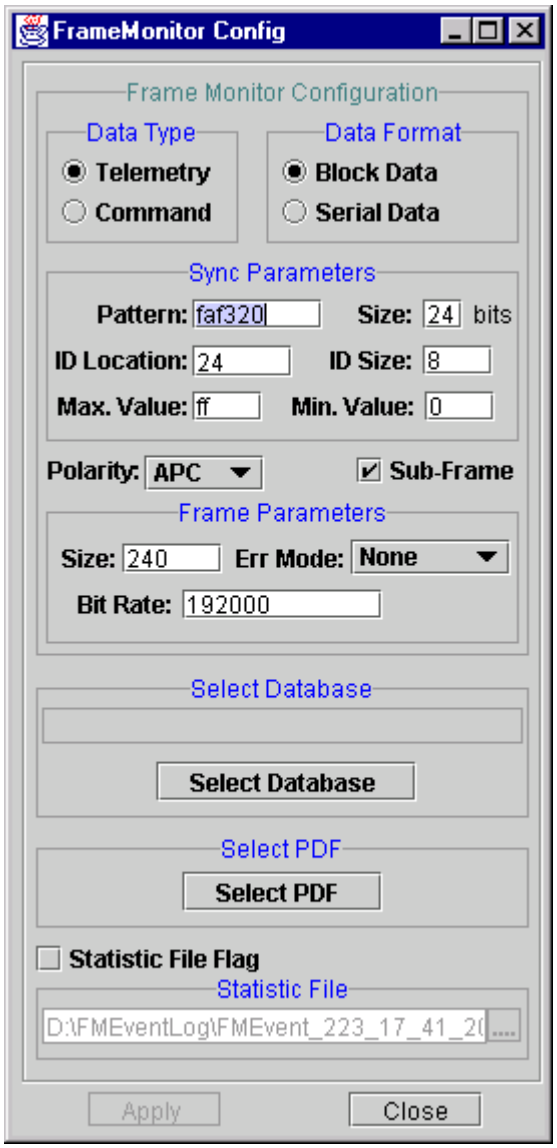
FrameMonitor-5.0 Displays

To access the displays for this module, click on the center of the module in the project window. The module pop-up menu will appear with the following choices: "Configure", "Run Time", "Remove", and "About". The "Remove" options are only available during project design. The "Run-time" option is only available when the project is running.

Module Pop-up Menu Item	Description
Configure	Access the module' s configuration window
Run-time	Access to module's Run-time menu
Remove	Remove the module from the project
About	Display generic module information

FrameMonitor-5.1 Configuration

Select the "Configure" item of the module pop-up menu to access the configuration window of the module. The configuration screen is shown in the following picture.



This display is used to set the FrameMonitor configuration. The display may be accessed while the Project is stopped or running. Every setting except for the Statistics file may be changed while the Project is running.

Parameter	Description
Data Type	Choose either Telemetry or Command, as appropriate.
Data Format	Choose Block Data to input data from an IP module. Choose Serial Data to input from a Serial Input module.
Pattern	Enter the expected Sync. Pattern for the module to search for and synchronize data
Size	Enter the sync size in bits
ID Location	If the incoming data contains a subframe, enter the byte location (starting from byte 0) of the frame ID
ID Size	Enter the ID size in bits
Max. Value	Enter maximum subframe ID value
Min. Value	Enter minimum subframe ID value

Subframe Check	Check this box to monitor subframe status, uncheck to ignore
Size	Enter the frame size in bytes
Err Mode	Select the number of errors allowed in sync patterns for synchronization to be achieved
Bit Rate	Enter the expected data rate of the received data stream
Select Database	Select data base name for configuration parameters
Select PDF	Select pre-defined configuration parameters from a PDF name
Statistics File	If it is desired to save a statistics file, check the Statistics File Flag then enter the path and filename of the desired statistics file.

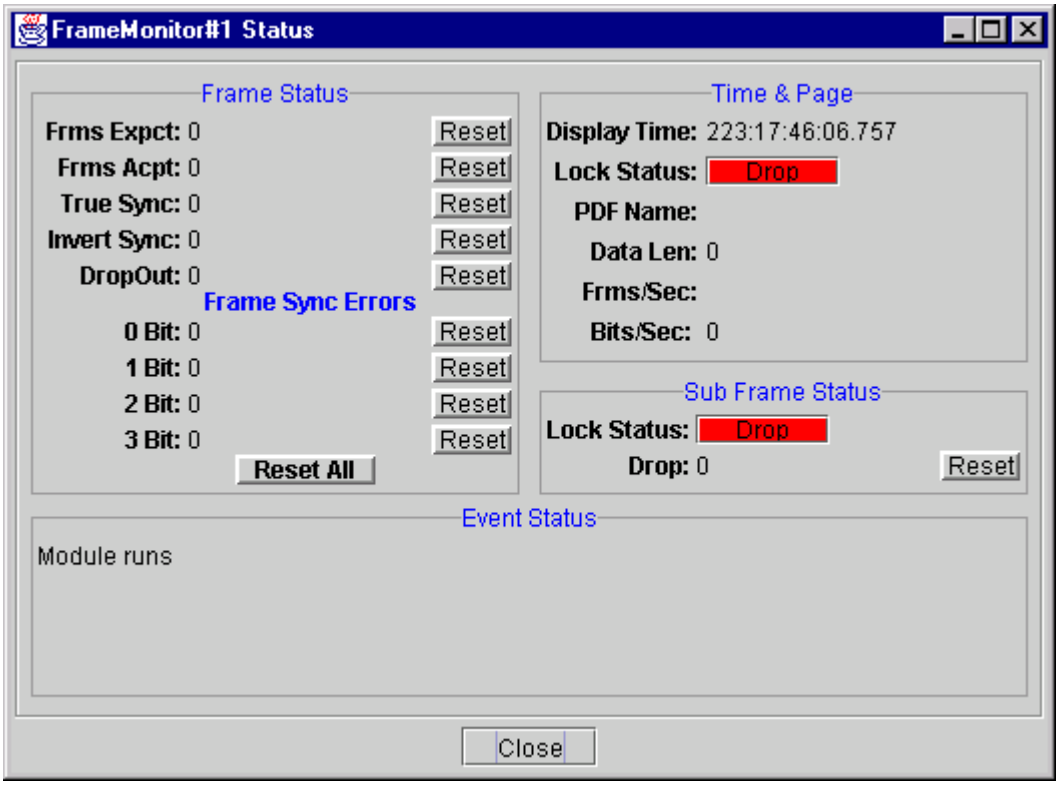
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

FrameMonitor-5.2 Run-time

Click on the module pop-up menu "Run-time" option to request the Run-time menu.

Run-time Menu Item	Description
Show Status	Show the status window of the module
Show Output Block	Show the output block data
Resume	Restart the module
Pause	Stop the module

FrameMonitor-5.2.1 Display Status



The status window shows information about data rate, lock status, data length, and frame statistics.

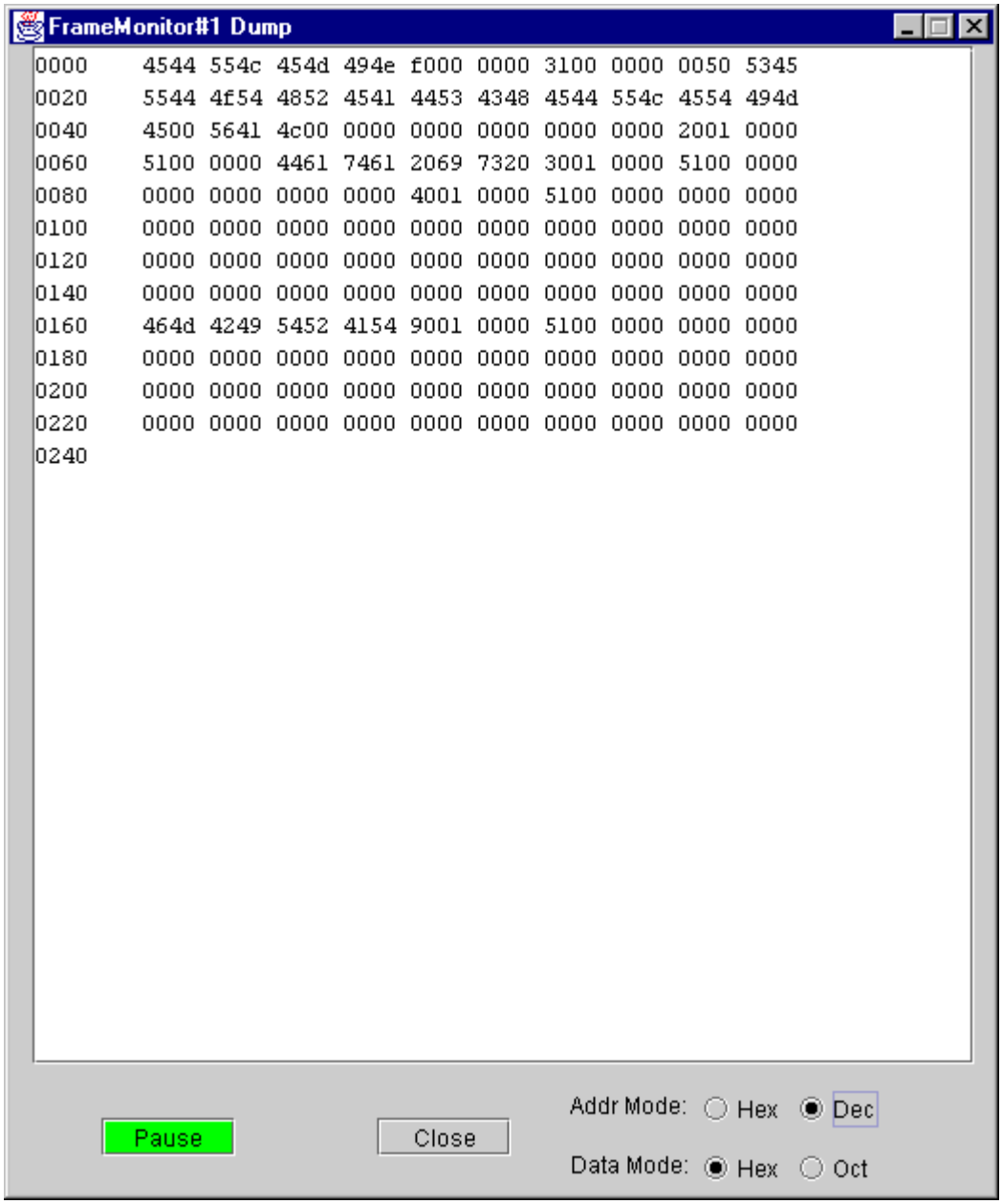
- The "Reset All" button is used to reset all counters.
- The "Reset" button associated with each counter is used to reset that counter.
- Frames with sync pattern errors falling within errors allowed will be accepted and counted.
- The "Lock" status indicates that detected frames are back-to-back frames, otherwise, the "Lock" status will switch to "Drop".
- The subframe "Lock" status indicates that subframe ID is in sequence, otherwise, it switch to "Drop" and the subframe drop counter will increase by one.

FrameMonitor-5.2.2 Display Dump Menu

Select the "Show Output Block" option to view the available dump formats.

Show Output Block Menu Item	Description
Show Output Block	To display output block

When this option is selected, the content of the output data is displayed.



Buttons	Description
Addr Mode	Select "Hex" to display the memory address in hexadecimal format Select "Dec" to display the memory address in decimal format
Data Mode	Select "Hex" to display data in hexadecimal format Select "Oct" option to display data in octal format
Pause	Freeze the data window contents. "Pause" button is renamed "Cont".
Cont	Resume updating the data window contents. "Cont" button is renamed "Pause".
Close	Close the window

FrameMonitor-5.2.3 Resume

After an individual module has been stopped, select "Resume" from the Run-time menu to restart the module. The color around the module's border will change from red-striped (indicates stopped state) to green-striped (indicate run state).

FrameMonitor-5.2.4 Pause

Select "Pause" from the Run-time menu to temporarily stop the module. The color around the module's border will change from green-striped (indicates run state) to red-striped (indicates stopped state). To return to run mode, select "Resume" from the Run-time menu.

FrameMonitor-5.3 Remove

Select this option to remove the module from the project.

FrameMonitor-5.4 About

Selecting the "About" option from the module pop-up menu produces a display that lists the number of input channels, number of output channels, and whether directives are allowed.

FrameMonitor-6.0 Special Operating Instructions

The input channel of this module is commonly connected to the output channel 1 of the Block Monitor module. In other cases, it can be connected to any module to synchronize asynchronous data if the sync patterns and frame size are known, provided that the orientation of the incoming data stream is Most Significant Bit (MSB) first.

Internet Protocol (InputIP and OutputIP) Modules

InternetProtocol-1.0 Overview

The Input and Output Internet Protocol modules receive/send data packets from/to other sources/destinations using one of several IP types (TCP/IP-Client, TCP/IP-Server, UDP Multicast, or UDP Unicast). The Input IP module receives data from an external source and passes that data to another module. The Output IP module receives data from any module and can pass that data to another module and send it to an external destination. Both IP modules are described in this chapter because of their great similarity.

InternetProtocol-2.0 Inputs

The Input IP module does not have any input channels. The Output IP module has a single input channel, which is listed below.

Channel	Data expected	Validation performed	Processing performed
1	Packets	None	Received data is passed to connected modules and transmitted to configured external IP destination

InternetProtocol-3.0 Outputs

Both the Input IP and Output IP modules have a single output channel.

Channel	Description
1	Input IP module passes received data to all connected modules. Output IP module passes received data to connected modules (usually the Log Module) and sends it to an external destination

InternetProtocol-4.0 Container Items

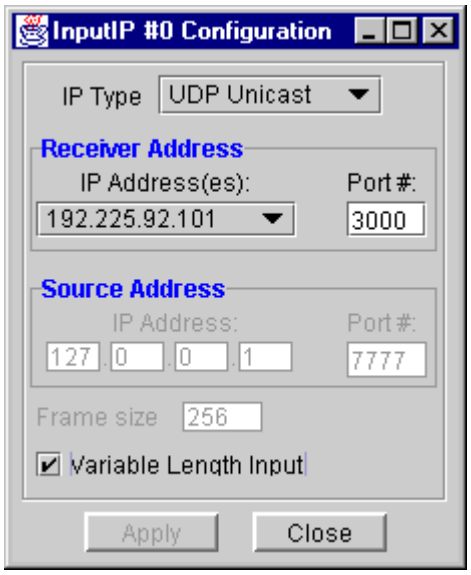
IP container items are not accessible via operator directives, so they are not listed here.

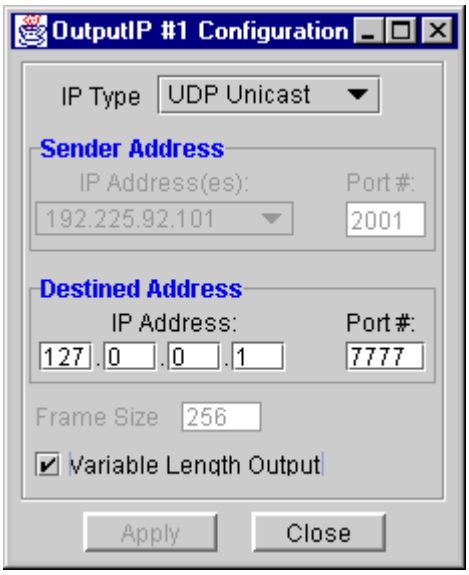
InternetProtocol-5.0 Displays

Click in the center of the module in the project window to access displays. A module pop-up menu will appear. The "Remove" option can be used during project design to remove this module. The "Configure" option must be used prior to running the project and is unavailable at run-time.

Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

InternetProtocol-5.1 Configuration Menu





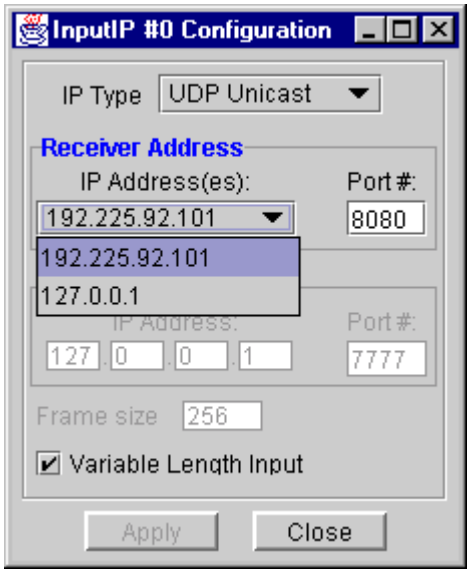
Clicking on the "Configure" item of the Input (or Output) IP module pop-up menu produces a display similar those shown above.

InternetProtocol-5.1.1 IP Type Field

There are 4 possible IP Types available from a drop down menu: UDP Multicast, TCP/IP-Client, TCP/IP-Server, and UDP Unicast.

InternetProtocol-5.1.2 Receiver (Sender) Address Box

Various options will be available depending on the configuration of the host PC running the server.



Step 1. If there is more than one Ethernet Card on the host machine where the server is running, the user can select an IP address from the drop down menu (similar to that shown on the left) as the network interface for the host machine.

Step 2. Certain IP Types will require a Port Number. If this option is sensitized, enter a Port Number for the host machine to bind to, to either receive or offer a service.

Note: The IP address **127.0.0.1** is a loopback address. It is used to test the network program in a single machine (except in UDP Multicast mode). **Don't** select **127.0.0.1** as a receiver (or sender) IP address in UDP Multicast mode.

InternetProtocol-5.1.3 Source (Destination) Address Box

There are different settings for the different IP types available.

For use in **TCP/IP-Client** mode:

Step 1. Enter the four fields of the Server IP Address to which the client will attempt to connect.
Step 2. Enter the Port Number where the server is offering the service.
In **TCP/IP-Server** mode the user is not allowed to configure this box. The four fields of IP address and 1 field of Port Number are desensitized.

For use in **UDP Unicast** mode (Output IP module only):

Step 1. Enter the four fields of the destination IP address of the remote host where the packets will be sent.
Step 2. Enter the Port Number where the packets will be sent.
The user is not allowed to configure this box when using an Input IP module in **UDP Unicast** mode. The fields will be desensitized.

For use in **UDP Multicast** mode (Input IP module):

Enter the four fields of the Multicast Group Address. This is the multicast group address that the Input IP module will join and receive packets from. The Port Number is not configurable. It will be desensitized. Enter the port number of the host PC in the Receiver Address box.

For use in **UDP Multicast** mode (Output IP module):

Step 1. Enter the four fields of the Multicast Group Address. This is the multicast group address where the packets will be sen.
Step 2. Enter the Port Number where the packets will be sent.

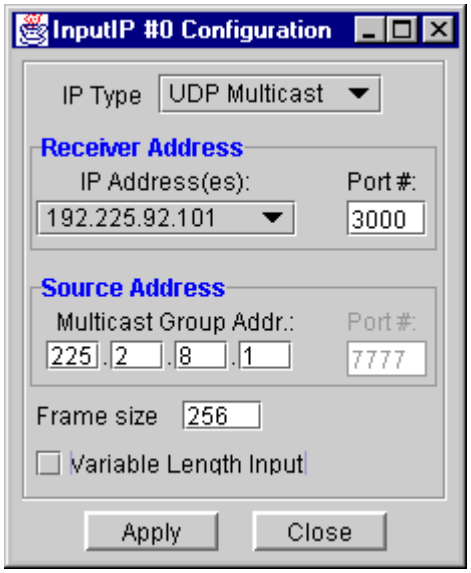
Note: The valid range of port numbers is 1000 to 65535. The valid range of multicast group IP addresses is 224.0.1.0 through 239.255.255.255.

InternetProtocol-5.1.4 Variable Length Input (Output) Check Box

When this option is selected, the box will be checked and the Frame Size field will be desensitized. The actual length of each data packet will be used without truncation or padding. This is the default configuration.

InternetProtocol-5.1.5 Frame Size Field

When the Variable Length Input (Output) Check Box is deselected, the Frame Size field becomes active. The user should enter the desired number of bytes to receive (transmit). The default is 256 bytes.



If the size selected is larger than the actual data, the additional bytes will be filled with binary zero.

If the Frame Size selected is smaller than the actual data size, the data bytes received (transmitted) will be truncated and the additional bytes will be lost.

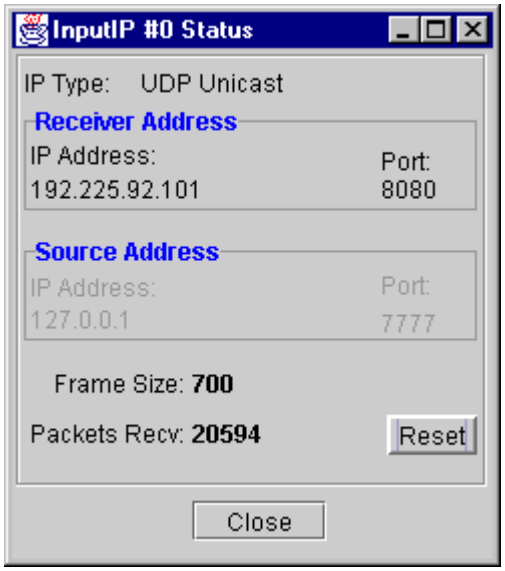
InternetProtocol-5.2 Run-time Menu

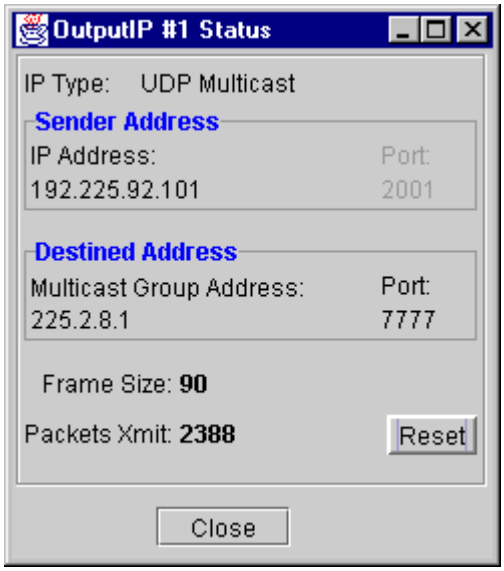
The Run-time menu for the IP modules contains the following four items.

Run-time Menu Item	Description
Show Status	Request Status Display
Show Raw Packet	Request Raw Packet Display
Pause (or Resume)	Pause (or Resume) the module
Stop (or Restart)	Stop (or Restart) the module

InternetProtocol-5.2.1 Show Status

Select the "Show Status" item to request the module's status. This produces a display similar to one shown below.

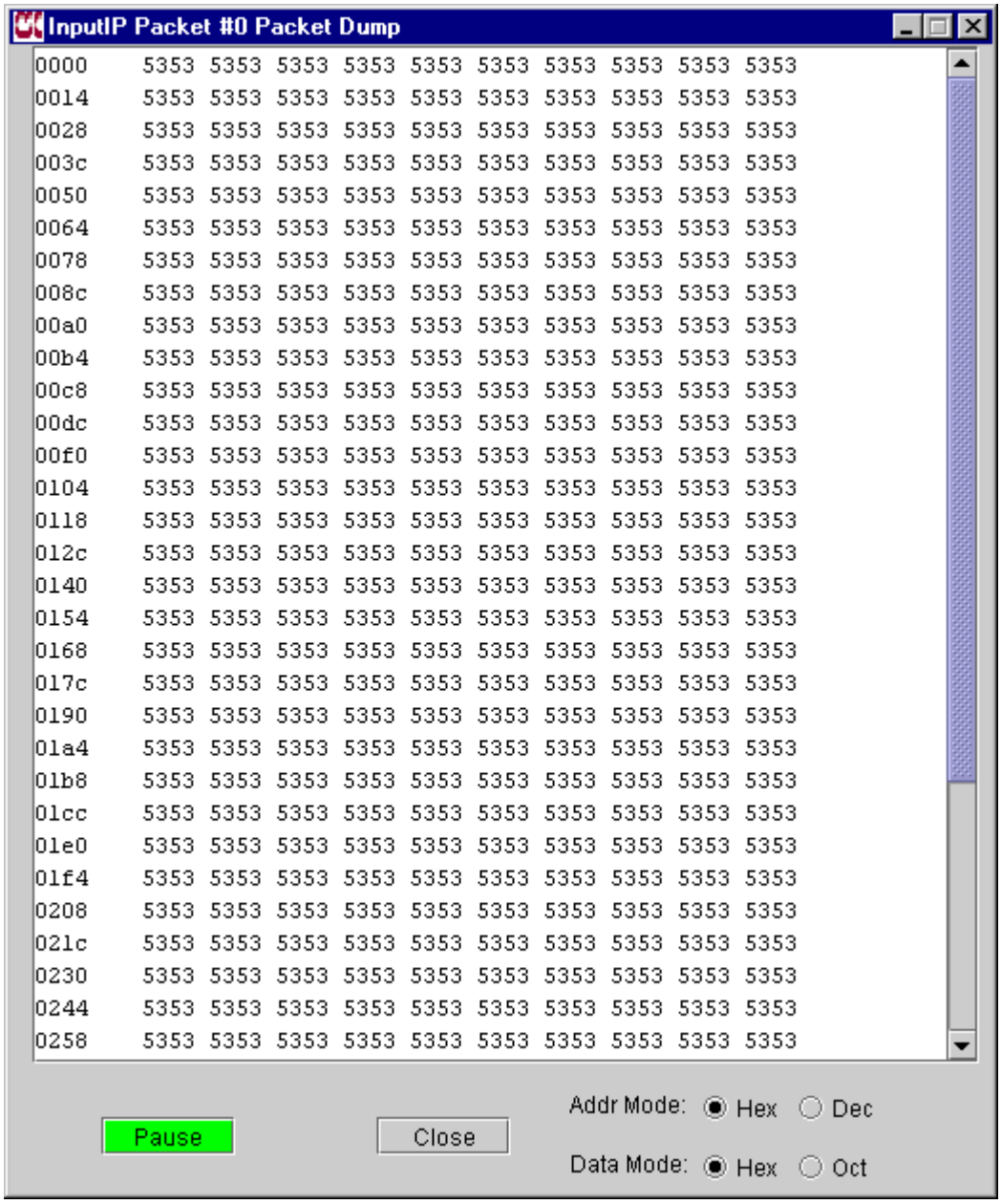




This display shows the Input (or Output) IP module's IP Type, Receiver (Sender) Address (including the port number, if appropriate for the mode), Source (Destination) Address (IP address, or multicast group IP address and port number), Frame Size, and number of packets received (or transmitted). If variable length items are being received, the Frame Size field will show the size of the most recent packet received (transmitted). The **Reset** button will reset the count of packets received. Click the **Close** button to dismiss the display.

InternetProtocol-5.2.2 Show Raw Packet

Select "Show Raw Packet" from the Run-time Menu to view the most recent data packet received by the module. An example of the packet display follows.



The address region may be displayed in decimal or hexadecimal by clicking on the respective radio button. Likewise, the data region may be displayed in hexadecimal or octal. Since this display updates as the buffer contents are changed; the **Pause** button may be used to freeze the current contents. Use of the **Pause** button does not affect data transmission or reception. When the **Pause** button has been used, its label is changed to **Cont** for continue. Press the **Cont** button to resume screen updates. Note that the display will update with the most recent packet processed. Packets processed by the module while the display was frozen cannot be displayed. Click the **Close** button to dismiss the display.

InternetProtocol-5.2.3 Pause

Select "Pause" from the Run-time Menu to pause the IP module's processing. The module's border will change from green (indicates run state) to red (indicates non-run state). This option's text will change to "Resume". When an Input IP module is paused, a limited amount of data received will be queued up. When an Output IP module is paused, all data passed to it is dropped.

InternetProtocol-5.2.4 Resume

After a module has been paused, select "Resume" from the Run-time Menu to restart its execution. The color of the module's border will change from red to green. This option changes to "Pause". The Input IP module first processes any data queued up while paused before forwarding new inputs. The Output IP module processes new data only.

InternetProtocol-5.2.5 Stop

Select "Stop" from the Run-time Menu to stop an IP module's processing. The module's border changes from green to red. This option's text changes to "Restart". Once a module has been stopped, the "Configure" option of the module pop-up menu is available again so the module may be reconfigured.

InternetProtocol-5.2.6 Restart

After a module has been stopped, select "Restart" from the Run-time Menu to restart it. The module's border changes from red to green. This option's text changes back to "Stop". Since their configurations may have changed, the IP modules reset all counters.

InternetProtocol-5.3 About

Selecting the "About" option from the module pop-up menu provides information on the number of channels, whether directives are allowed and the version number.

InternetProtocol-6.0 Special Operating Instructions

There are no special operating instructions for this release.

Log Module

Log-1.0 Overview

The Log module writes the data it receives to a specified log file.

Viewing or printing of log files may be accomplished by means of an external program. The software used must be capable of displaying binary data in an ASCII representation. The shareware utility, Hexedit, which is available over the Internet from Alexander Reidel Informations-Systeme, is one such program.

Log-2.0 Inputs

Ch	Data expected	Validation performed	Processing performed
1	Bytes	None	If logging is enabled, received log data is written into the log file.

Log-3.0 Outputs

The Log module does not have any output channels.

Log-4.0 Container Items

The Log module is capable of receiving directives from the operator from the directive line or from a scenario file. Container item names are not case-sensitive. Values for string type variables should not include quote characters. When configuring the Log module from a scenario, set the "OpenNewLog" container item to one as the last parameter to force the opening of a log file with the previously specified parameters.

Name	Type	Project Saved?	Description
PROJECTFLAG	Number	Saved	Optional project flag (1=project information is included in generated file name, 0=field not used)
MODULEFLAG	Number	Saved	Optional module flag (1=module information is included in generated file name, 0=field not used)
TIMEFLAG	Number	Saved	Optional timestamp flag (1=timestamp is included in generated file name, 0=field not used)
DESCFLAG	Number	Saved	Optional description flag (1=description field is included in

			generated file name, 0=field not used)
DESCFLD	String	Saved	Configured description field
DIRNAME	String	Saved	Configured log directory
EXTFLAG	Number	Saved	Optional extension flag (1=file extension is included in generated file name, 0=field not used)
EXTFLD	String	Saved	Configured file extension field
FILECOUNT	Number	Not saved	Current full file count
FILESIZE	Number	Not saved	Current log file size
FIXEDFLAG	Number	Saved	Fixed format flag (1=every log entry will be the same fixed size, 0=log entries have variable lengths)
FIXEDSIZE	Number	Saved	Log item size in bytes when the Fixed format is selected
FORMATFLAG	Number	Saved	Optional format flag (1=format characters are included in generated file name, 0=field not used) When the format flag is set, the following letters are inserted in the generated file name: H if log headers are used N if log headers are not used V if variable length entries are written F# if fixed length entries are written # is the fixed record size in bytes
HEADERFLAG	Number	Saved	Log header flag (1=log with header, 0=no header)
ITEMSLOGGED	Number	Not Saved	Number of records logged
ITEMSTRUNCATED	Number	Not Saved	Number of truncated records
LOGGINGENABLED	Number	Saved	Logging enabled flag (1=enabled, 0=disabled)
MAXFILECOUNT	Number	Saved	Maximum full file count limit (0=no limit, 2 thru 999=file limit)
MAXFILESIZE	Number	Saved	Maximum size of each log file in bytes.
OPENNEWLOG	Number	Not Saved	Open new log request (0 = open if parameters have changed, 1 = always open new log file)

Log-5.0 Displays

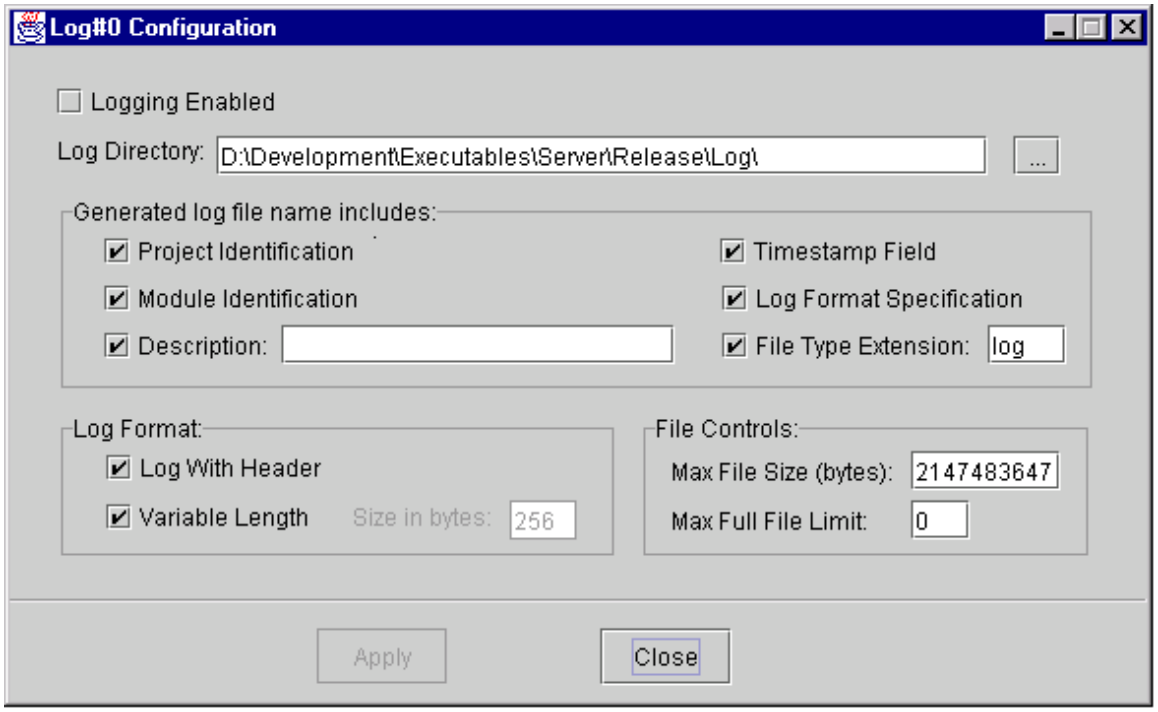
To access displays for the Log module, click in the center of the module in the project window. The module pop-up menu will appear. The "Remove" option can be used during project design to remove this module. The "Configure" option may be used prior

to running the project and at run-time. The Run-time option is available only when the project is running.

Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display Information about the module

Log-5.1 Configuration

Selecting the "Configure" pop-up menu option produces a display similar to the following screen. Default values are shown for all parameters. The Log module will generate a name for the log file that includes the project number, module name, module number and a time field. This prevents file name collisions between any two Log modules on the same server PC in the same or different projects. By including a time stamp field, there should be no file name collisions with recently logged data.



Field	Default Value	Default Description
Logging Enabled	Disabled	This field MUST be toggled to enabled to start data logging.
Log Directory	Path from server properties DefaultLogDirectory.txt file or "Log" directory below installed server release directory	Sample display shows a development directory.

Project Identification	Selected	Project information will be added to the generated log file name.
Module Identification	Selected	Module information will be added to the generated log file name.
Description Field Box	Selected	The associated description field will be added to generated log file name.
Description Field String	Empty	No default description field
Timestamp Field	Selected	A timestamp will be added to the generated log file name.
Log Format Specification	Selected	Letters "HV" will be included in generated file name to indicate use of Log Headers and Variable length entries.
File Type Extension Box	Selected	The associated extension field will be added to the end of the generated file name.
File Type Extension String	"log"	Associated file type extension field
Log With Header Box	Selected	Data will be logged with log headers. Log headers record the length of the entry and the logged time.
Variable Length Box	Selected	Variable length records will be written into the log.
Size in bytes	256	This field is not applicable for the variable length format.
Maximum File Size	2147483647 bytes	When the current log file reaches this size, it is closed and a new log file is opened.
Maximum Full File Count	0 indicating no limit	By default, there is no limit on the number of log files that may be created.

Log-5.1.1 Logging Enabled Flag

The Logging Enabled flag allows the operator to control when a Log module logs data. Since this flag is initially disabled, several Log modules may be added to a project without being configured. Later during runtime, these modules may be configured. At the very least this flag MUST be changed to enabled to start logging data.

NOTE: When a Log module is paused, stopped or logging disabled, the Log module box on the project window has a red border. When a Log module is running and logging is enabled, the module's box has a green border.

Log-5.1.2 Log Directory

A default location for the log files may be specified in the DefaultLogDirectory.txt file in the installation server/properties directory. If a default was not provided in the properties directory, the default location for the log files is in the "Log" directory below the directory where the server executables were installed. The operator may optionally configure a different directory location. A file browse button is provided.

Log-5.1.3 Generated Log File Name Fields

The Log module generates the names of its output log files based on a flexible set of naming conventions. The intent is to avoid naming conflicts with existing files, which could cause loss of data. The Log module cannot append to an existing file because the existing file may not be a log file. If the existing file is a log file, its log format may not be consistent with the currently configured format. In either case appending data to the existing file could corrupt it. It is also never advisable to have the Log module automatically delete or overwrite an existing file of unknown origin.

The naming conventions provide unique log file names by default, but the operator may add a description field or change the file extension. The operator may disable any of these naming conventions. Once an operator has configured the set of naming rules, the Log module can open new log files as needed without additional operator input.

The order of the fields of the generated log file name is as follows. If project information is enabled, it appears first in the file name. If module information is enabled, it appears next in the name. If the operator provided a description field, it appears next in the name. If the timestamp is enabled, it appears next. If the format specification is enabled it appears next. Underscore characters separate these fields in the name. If enabled, the last field in the file name is the file type extension.

NOTE: At least one of the file naming fields must be enabled in order for a log file name to be generated. Should the configuration of the naming conventions generate a file naming conflict with an existing file, the timestamp option will be automatically enabled and a new log file name generated.

NOTE: Should the log module be unable to open a new log file in two attempts, the logging status will be automatically changed to disabled and the border of the Log module box will change to red. Event messages will provide information on the specific file open errors.

Log-5.1.3.1 Project Identification Field

When this box is checked, the generated log file name starts with the letters "Proj" and end with the project's number. Since multiple projects may run on a PC, this option

distinguishes log files by project. When many log files are written to a common directory, all of the log files using this option in the same project will sort together.

Log-5.1.3.2 Module Identification Field

When the Module Identification box is checked, the generated log file name contains module information. Since there may be multiple Log modules in a project, this options distinguishes the log files from different Log modules. If the Log module has not been renamed, the module information starts with the letter "Log#" and ends with the module's number in the project. When the module has been renamed, this field contains the Log module's current name. When many log files are written to a common directory and this option is being used, the log files for the same module will sort together.

Log-5.1.3.3 Description Fields

When the Description Field Box is checked, the characters in the Description Field are included in the generated log file name.

Log-5.1.3.4 Timestamp Field

When the Timestamp Field Box is checked, the current system time is read and put into the format of MMDD-HHMMSS where MMDD is the month and day and HHMMSS is the hour, minute and second. The use of this field is highly recommended for the generation of unique file names.

Log-5.1.3.5 Log Format Specification Field

When this option is selected, characters are added to the generated log file name that indicate whether the data was logged with headers or not and whether the log entries are fixed or variable in length.

Knowing how the data was logged makes a log file easier to use with other SIMSS utilities. Log files with headers may be retransmitted using the same timing intervals as originally logged. Variable length format logs must have log headers to be usable in this manner. The fixed length record size is needed to process fixed format files if log headers were not used.

When the Log Format Specification option is specified the following letters are added to the file name.

Format Characters	Format Description
H	Indicates logged With Log Headers
N	Indicates logged with No Log Headers
V	Indicates Variable length records
Fn	Indicates Fixed length records of n bytes. For example F256 indicates Fixed format with 256 byte records.

Log 5.1.3.6 File Type Extension Fields

When the File Type Extension Box is checked, the characters in the File Type Extension String are included at the end of the generated log file name. It is not necessary to precede this field with a period character.

Log 5.1.4 Log With Header Format

When the Log With Header box has been checked, all log records will begin with a log header of ten bytes. This header consists of eight bytes of system time followed by two bytes of data length. The following example shows two CLCW packets. The header in each packet has been circled for clarity. When the Log With Header box is not checked, all records will be written without a header.

```
hexedit1
00000000  01 BF 85 59 4F 6D 6A A0 00 20 02 8C 8C EA DA 13 00 00 00 00 26
00000015  80 00 00 00 00 00 00 00 00 00 00 00 66 82 00 00 01 01 00 00 00
0000002A  01 BF 85 59 4F 7F C0 E0 00 20 02 8C 8C EA DA 13 00 00 00 00 26
0000003F  80 00 00 00 00 00 00 00 00 00 00 00 66 82 00 00 02 01 04 00 00
```

Log-5.1.5 Variable Length Format

When the Variable Length box has been checked, varying length records are written to the log file. No truncation or padding of received data is done. When this field is checked, the fixed size field is desensitized. When the Variable Length box is not checked, the fixed length format is selected and the fixed size field is enabled for input.

Log-5.1.6 Size in bytes

The Size field defines the fixed format length of the data record to be written to the log file. A size that is shorter than the actual data received will result in truncation of the data that is logged. A size that is longer than the actual data received will be zero filled to the specified size and then written to the log file.

Log 5.1.7 Maximum File Size (bytes)

The Maximum File Size in bytes limits the size of individual log files. When the next item to log would exceed the maximum size, the current log file is closed and a new log file is opened. The new log file will use the currently configured file naming options but with a newer timestamp field. The item is then logged in the new log file. This parameter may not be set smaller than 12,000 bytes.

Log 5.1.8 Maximum Full File Count

The Maximum Full File Count limits the total number of "full" log files generated by a Log module. Each time the Maximum File Size is reached, the current log file is closed

and the full path name of the closed file is saved. When the number of "full" log file names reaches the maximum file count, the oldest log files are deleted. When the maximum full file count is set to zero, there is no limit on the number of files created by the Log module. When this parameter is non-zero, it must be a value between 2 and 999.

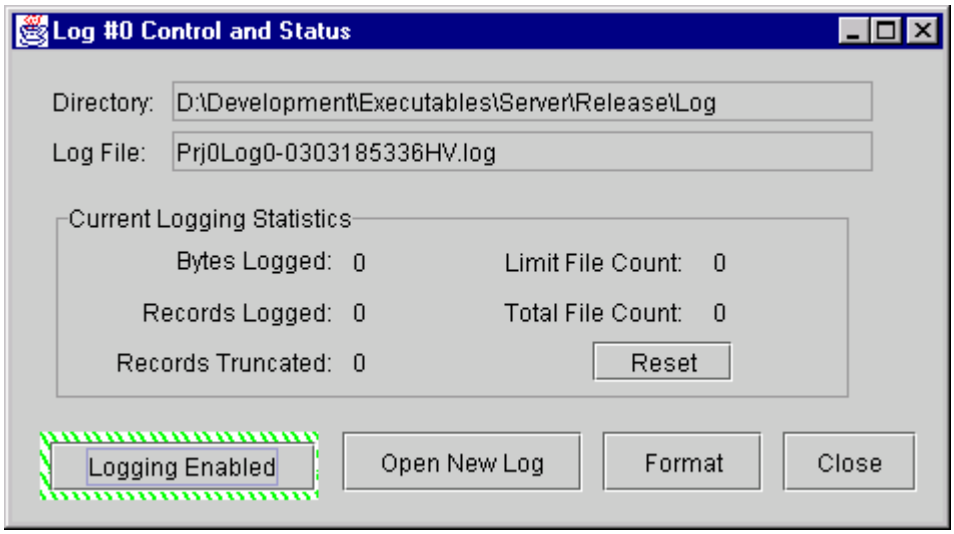
Log-5.2 Run-time

Clicking the "Run-time" option of the module pop-up menu produces a Run-time menu.

Run-time Menu Item	Description
Configure	Request Configuration Display (see Log-5.1)
Show Status	Request Control and Status Display
Pause (or Resume)	Pause (or Resume) the module
Stop (or Restart)	Stop (or Restart) the module

Log-5.2.1 Control and Status Display

When the "Show Status" option is selected from the Run-time Menu, a screen like the following is displayed. These fields do not accept data entry.



Status Fields	Description
Directory	This field shows the directory for the current log file.
Log File	This field shows the current log file name.
Bytes Logged	Total bytes written to the current log file. This count includes log headers when headers are being used.
Records Logged	Number of records written to the current log file.
Records Truncated	Number of records truncated because the data was larger than the configured fixed record size. This is an indication that all data is not being captured in the log. This statistic is only applicable to the Fixed log format.

Limit File Count	Number of "full" files that count toward the maximum full file limit. A file is considered "full" when writing the current record would exceed the maximum file size. The record to be logged is never split between files. The software automatically closes a "full" log file without exceeding the maximum file size. This number only increments when there is a maximum full file limit.
Total File Count	Total number of non-empty files that have been created and closed by this Log module. This number includes files closed due to the maximum file size limit and files closed by the Open New Log button. This number is reduced when file limit processing causes file deletions.

Log 5.2.1.1 Reset Button

The Records Truncated count shows the number of records that were truncated prior to being logged. The first time a truncation occurs; a warning event message is also generated. The reset button is only associated with this field. Each time this count is reset, the next truncation produces another warning event message.

Log 5.2.1.2 Logging Enabled/Disabled Button

The Logging status button is used to toggle the logging enabled flag. When logging is enabled, this button is labeled "Logging Enabled" and has a green border. When logging is disabled, this button is labeled "Logging Disabled" and has a red border.

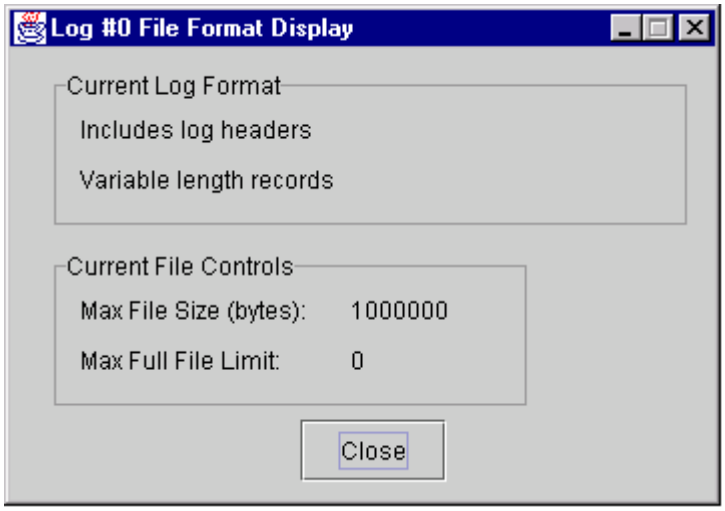
NOTE: The current logging status affects the color of the Log module box in the project window. When a Log module is paused, stopped or logging disabled, the Log module box has a red border. When the Log module is running and logging is enabled, the module's box has a green border.

Log 5.2.1.3 Open New Log Button

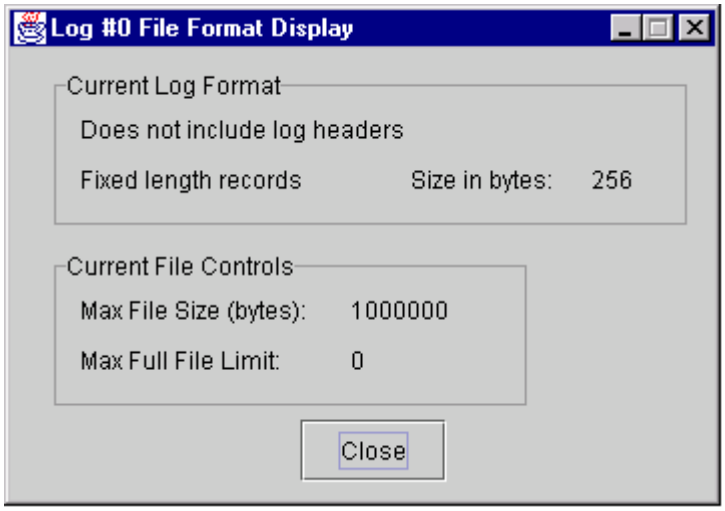
The Open New Log button is used to force the closing of the current log file and open a new log file. The new log file will use the latest configured log parameters. Use of this button has no immediate effect on maximum file limit processing. When a maximum full file limit has been specified, the names of ALL non-empty closed log files are saved. These partial files are automatically deleted only after the maximum amount of new data has been logged.

Log 5.2.1.4 Format Button

The Format button on the Control and Status Display is used to request a display of the current logging format and file limits. Some sample Format displays are shown below. These fields do not accept data entry.



When a fixed format is used, this display also shows the fixed record size.



Log-5.2.2 Pause/Resume

Select "Pause" from the Run-time Menu to pause the Log module's processing. The module's border changes to red. The text of this menu option changes to "Resume". While the Log Module is paused, it does not receive or log data. After a Log module has been paused, select "Resume" from the Run-time Menu to restore processing. If logging is enabled when the module is resumed, the border color changes to green.

Log-5.2.3 Stop/Restart

Select "Stop" from the Run-time Menu to stop the Log module's processing. The module's border changes to red. The text of this menu option changes to "Restart". After a Log module has been stopped, select "Restart" from the Run-time Menu to restart the module's execution. If logging is enabled when the module is restarted, the border color changes to green.

Log-5.3 About

Selecting the "About" option from the module pop-up menu produces a display that lists information about the module's number of input links, number of output links, whether directives are allowed.

Log-6.0 Special Operating Instructions

Log-6.1 Optional Project Default Log Directory

A default directory may be defined for all of the Log modules in a project by providing a text file with the name of "DefaultLogDirectory.txt" in the \server\release\properties directory. The single line of data in this file is a fully specified directory path name. The directory name should not have quotation marks around it. For example, the contents of this file could be the following:

D:\Log

If the DefaultLogDirectory.txt file does not exist or it does not specify a valid directory, the installed server\release\log directory will be used as the default. Event messages will inform the operator of these actions.

The DefaultLogDirectory.txt file is processed only once when a Log module is added or restored to a project. If this file is modified, the new directory name will be used the next time a Log module is added to the project or the next time a project is restored.

This file specifies the log directory to use only when the user does not configure a directory. Whenever the user specifies a valid directory from the Configuration screen that directory will be used.

Packet Processor Module

Packet Processor-1.0 Overview

The Generic CCSDS Packet Processor module is used to monitor and verify packets and to filter user-selectable packets on a CCSDS data stream.

Packet Processor-2.0 Inputs

The Packet Processor module has a single input channel.

Channel	Description
1	Accepts an external data stream of CCSDS data. Data must be CCSDS telemetry format in either continuous or burst data mode.

Packet Processor-3.0 Outputs

The packet Processor module has only one output channel. The output channel is user configurable (see below for configuration instructions).

Channel	Description
1	Sends user-defined packets to other modules.

Packet Processor-4.0 Container Items

None.

Packet Processor-5.0 Displays

Click in the center of the module within the project window. The module pop-up menu will appear. The "Remove" option may be used during project design to remove the module. The "Configure" option should be used prior to running the project and is unavailable at run-time. The "Run-time" option is available only when the project is running.

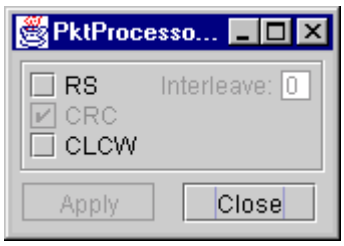
Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display generic module information

Packet Processor -5.1 Configure

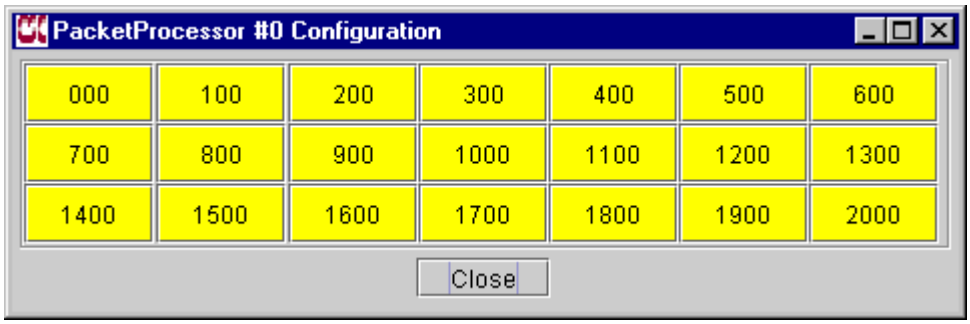
There are two static configuration windows for this module. The module must be configured for the CCSDS encoding method being used via the VC Configuration

Window. This will ensure correct packet locations and lengths. Before configuring, the user must know the specific spacecraft's CCSDS format. Perform packet configuration via the Packet Configuration Window after configuring the VC encoding method. This module may be re-configured in runtime (see 5.2 below).

Packet Processor-5.1.1 VC Configuration Window



Packet Processor-5.1.2 Packet Configuration Window



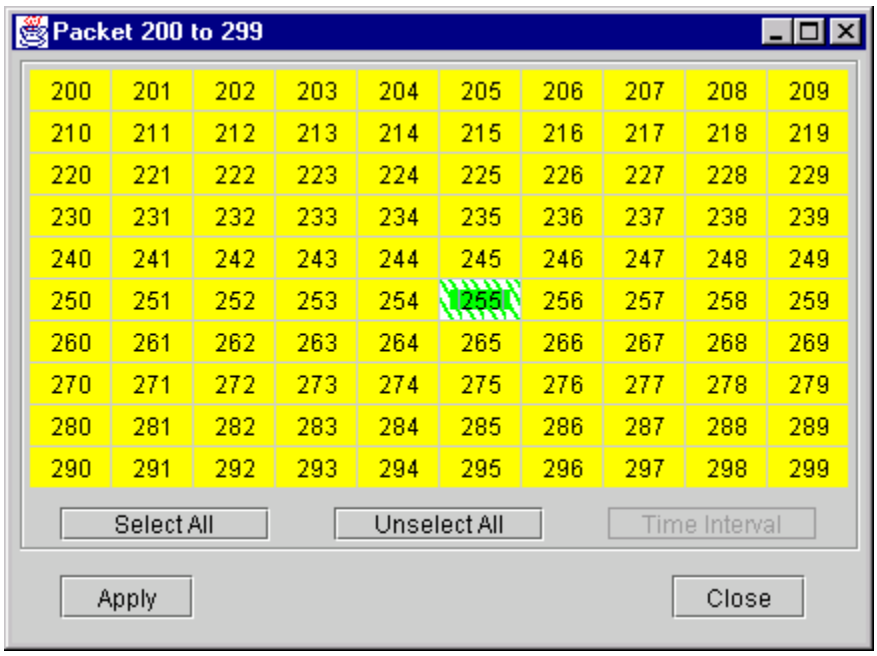
This is the main configuration window. By selecting one of the configuration buttons, the user will be provided with an APID Filtering Configuration window. Use the APID Filtering Configuration windows to select the APIDs that will be forwarded to other modules.

Button Name	Description
000	Selects APID 000-099
100	Selects APID 100-199
200	Selects APID 200-299
300	Selects APID 300-399
400	Selects APID 400-499
500	Selects APID 500-599
600	Selects APID 600-699
700	Selects APID 700-799
800	Selects APID 800-899
900	Selects APID 900-999
1000	Selects APID 1000-1099
1100	Selects APID 1100-1199
1200	Selects APID 1200-1299
1300	Selects APID 1300-1399
1400	Selects APID 1400-1499

1500	Selects APID 1500-1599
1600	Selects APID 1600-1699
1700	Selects APID 1700-1799
1800	Selects APID 1800-1899
1900	Selects APID 1900-1999
2000	Selects APID 2000-2099

Packet Processor-5.1.3 Select APID

The following window is an example of an APID filtering configuration window. This window was selected by clicking on the APID 200 button in the Packet Configuration Window. Any number of APIDs, displayed above, may be enabled for transmission to the module's output channel. The user should select the desired APIDs by clicking on the appropriate buttons.



Packet Processor-5.2 Run

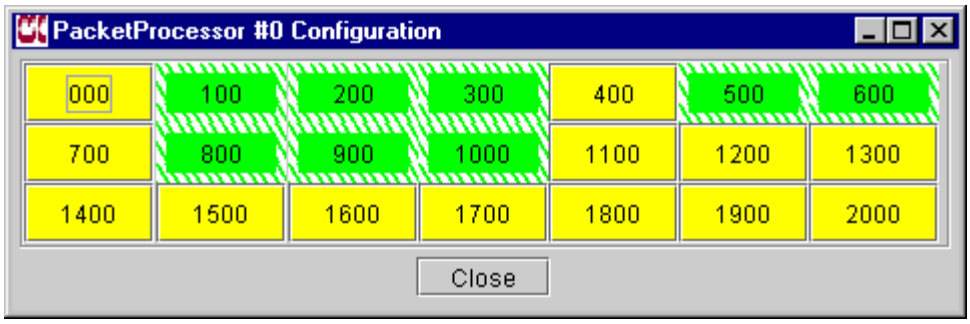
During runtime, the Packet Processor module will perform an analysis of the CCSDS data stream going through it. When the module encounters a new packet in the stream, it will inform the user through status windows and report on the status of each active APID.

The Run-time menu for Packet Processor contains the following items.

Run-time Menu Item	Description
Show Status	Display the status of each active APID.
Configure	Allows user to reconfigure on the fly

Resume	Resumes module.
Pause	Pauses module.

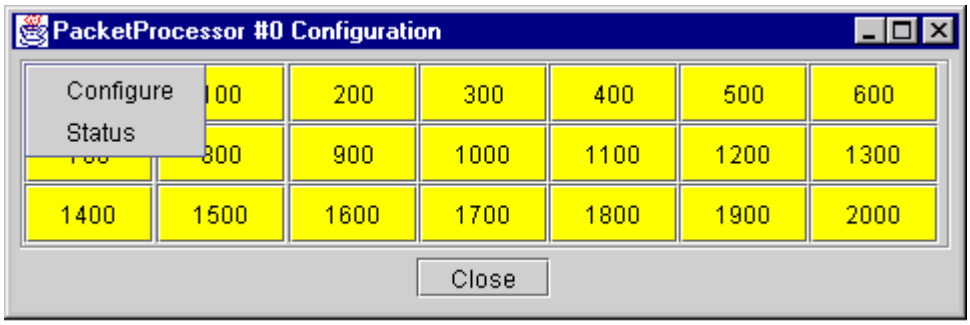
Packet Processor-5.2.1 Show APID status



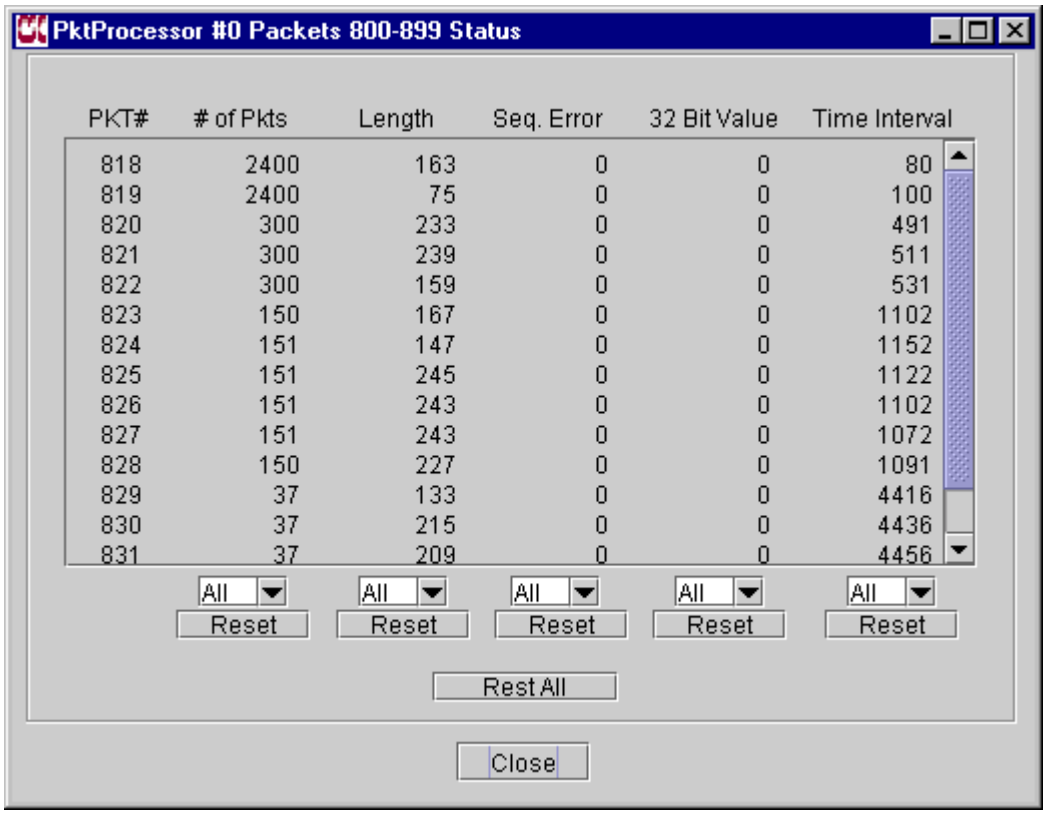
This is a "quick" display of the status of all APID in 100's. The following color code applies:

Yellow	All APIDs in the range are inactive.
Green	One or more APIDs in the range are active and no errors have been detected.
Red	One or more APIDs in the range are active and one or more have experienced sequence errors

All APIDs start out as inactive. When a Packet is detected in the data stream, the corresponding box turns green or red, depending on quality of data. The user can click on any box and see the status of the APID indicated in that range. To do so, click on the Status option of the drop-down menu that appears. A Packet Status window will appear.



Packet Processor-5.2.2 Packet Status Window



This window shows the status of each active APID.

Status Column Title	Description
PKT#	The Packet APID
# of Pkts	The number of APIDs received within range
Length	Packet length
Seq. Error	Number of APID sequence number encountered for packet
32 Bit Value	Shows packet offset value
Time Interval	Time in msec between packets

Packet Processor-5.2.3 Configure

This option allows the user to re-configure the Packet Processor module during run-time. Neither the module nor the project needs to be stopped in order do the reconfiguration. Clicking on the Configure option causes a Packet Configuration Window to appear. During run time each APID button in this window has an additional option – Configure. Selecting the Configure option results in a Select APID screen like the one shown in Section "Packet Processor 5.1.3."

Scenario Module

Scenario-1.0 Overview

The Scenario module reads directives from specified scenario files line by line and passes them to down-linked modules for processing. The user interface allows the operator to control and monitor the execution of up to five scenarios at a time. The scenario module may also receive scenario file names from a downlinked module. In addition, any scenario file being processed may start the execution of other scenario files.

Scenario-2.0 Inputs

Scenario-2.1 Server/Properties/Property.txt

The Scenario module does not currently use any settings from the Property.txt file. All scenarios are run concurrently.

Scenario-2.2 Module Triggered Scenarios

The Scenario module does not technically have any input channels. Its four output channels are actually bi-directional, providing the downlinked module(s) a way to pass names of scenario files to be executed. Processing specific to the downlinked module may trigger the passing of scenario file names to the scenario module for immediate execution.

Scenario-2.3 Scenario File Input

Much of the input to this module is via scenario files, which are read during run-time. There are no naming conventions for scenario files. Scenario files are expected to be in the default scenario directory, which is below the server executable directory, unless a full pathname is specified. Full pathnames contain a drive letter and colon or start with a slash character. Relative pathnames are assumed to start from the scenario directory.

Scenario-2.3.1 Scenario File Contents

The format of scenario files is straightforward. Each line is limited to 260 characters of ASCII text. Each line contains one of the items defined in the following table. Tab characters and blanks may be used to indent scenario lines for readability. Although Scenario module-specific keywords are shown below in uppercase, they are not case sensitive. The keywords shown in upper case are reserved for the Scenario module.

Scenario Script Line	Description
comment	A semicolon defines a comment. A comment may be on a separate line or added to the end of any directive line. All characters on a line after the semicolon are ignored.
blank line	A line containing only blank or tab characters is treated as a comment line.
SLEEP milliseconds	The scenario module pauses execution of the file for the number of milliseconds specified. When *milliseconds* are specified as -1, the directive is a SLEEP forever. A SLEEP forever requires operator intervention in order to continue the scenario's execution. To cancel a SLEEP forever or any SLEEP period in a GUI controlled scenario, click the Pause button and then the Resume button. For non-GUI controlled scenarios, the SLEEP forever instruction is ignored.
START SCENARIO name	The scenario module starts execution of a nested scenario file. If a full path is not provided for the file name, the same directory as the executing scenario file is used.
STOP SCENARIO name	The scenario module stops execution of any scripts with the specified file name. If a full path is not provided for the file name, any scripts with that file name from any directory are stopped.
directive	Any line that is not a comment or processed by the scenario module is assumed to be a directive for the linked module. The receiving module performs any validation done on this line. The format for a directive is the same as if the operator had entered it in the project directive entry line.
IF (expression)	Specifies the conditional expression that must be true in order for the directives between this line and the next ELSE IF, ELSE or ENDIF statement to be executed.
ELSEIF (expression) ELSE IF (expression)	Specifies a conditional expression that must be true in order for the directives between this line and the next ELSE, ELSE IF or ENDIF statement to be executed.
ELSE	Specifies the beginning of the directives to be executed when the preceding IF or ELSE IF expression is false.
ENDIF END IF	Specifies the end of any optional ELSE clauses and the end of the current IF statement.
WHILE (expression)	Specifies the conditional expression that must be true in order for the directives between this line and the ENDWHILE statement to be executed. One of the directives within the loop body should be a SLEEP directive.
ENDWHILE END WHILE	Specifies the end of the current WHILE loop.

Scenario-2.3.1.1 Specifying Modules to Receive Scenario Directives

A scenario file may contain directives for more than one module. Any executable line within a scenario file may start with an optional module number field with the format:

#x <directive line text>

where x is a number from 1 to 4. A directive labeled with #2 is sent to the module(s) connected to Scenario output channel 2. Subsequent directives are also sent to channel 2 until the end of the scenario file or another channel is specified. When present, the module number must be followed by a blank character and there must be a directive on the same line. At the beginning of each scenario's execution, the module number is initialized to 1. In the absence of any module numbers, directives are sent to the module(s) linked to the Scenario module's output channel 1.

Note that specifying a module number in the middle of a WHILE loop can result in unintended consequences. On subsequent iterations of the loop all directives prior to the module specification number will be sent to that module. That may not be the intended behavior.

Scenario-2.3.1.2 Specifying Conditional Expressions

The conditional expression must be specified on a single line of up to 260 characters (including the return key). The conditional expression must be entered on the same line as the keywords IF and WHILE.

The special characters used to describe conditional expressions for IF and WHILE statements are parentheses, binary math operators (+, -, *, /, %), binary logical operators (<, <=, ==, >=, >, !=, &&, ||), unary operators (!, -) and trigonometric operators (rsin, rcos, rtan, rasin, racos, ratan, sin, cos, tan, asin, acos, atan, dtor, rtod, sqrt, ln, exp). Additional operators (such as XOR and NAND) may be used in set directives but not in conditional expression clauses.

At least one set of parentheses must be used to enclose a conditional expression. Additional sets are recommended for proper evaluation of complex algorithms. Proper matching of parentheses is very important. If any parentheses are missing, a warning event message will be generated and the conditional expression will be evaluated as false.

Scenario-2.3.1.3 Simple Scenario Example

The following 4 lines are a simple example of a scenario file. The first line is a comment. The second line is a directive to the Scenario module to pause for one second. The third line is a directive to the linked module to change the value of the container item named "TransmitMode" to one. The fourth line starts execution of a nested scenario file named OtherScenario.txt.

```
;This is a comment line in a scenario file
SLEEP 1000          ; Sleep for one second
set TransmitMode 1
Start scenario D:/Scenario/OtherScenario.txt
```

Scenario-2.3.1.4 IF Scenario Example

```
if ((mnemonic0 + rsin(mnemonic1)) > mnemonic2)
     set mnemonic4 321
else if (!mnemonic3)
     set mnemonic5 20
     set mnemonic6 += 100
else
     set mnemonic7 512
endif
```

Scenario-2.3.1.5 While Scenario Example

The following is an example of a scenario file using a while statement. A sleep delay within the loop is recommended to prevent CPU-intensive wait loops from depleting system resources. If no sleep statement is included, the Scenario module will delay one second each time through the loop.

```
while ((mnemonic0 < 5)&& (mnemonic2 > sqrt(mnemonic3)))
     set mnemonic2 -= sqrt(mnemonic3)
     set mnemonic0 += 1
     Sleep 1000
endwhile
```

Scenario-3.0 Outputs

The Scenario module has 4 output channels. Each output channel may be linked to an input channel of the module(s) to receive the directives.

Channel	Description
1-4	Scenario module sends the directives to the linked module.

Scenario-4.0 Container Items

The Scenario module accepts operator directives and is capable of receiving directives from other SIMSS modules. Use the **Set** and **Get** directives to access these items. Although container variable names in the following sections contain upper and lower case, directive lines are not case-sensitive.

Scenario-4.1 Scenario Script Identification Numbers

Every script executed by the scenario module is assigned an ID number. The GUI controlled scripts use their slot number as their ID number. Generated threads are

assigned a one-up number between six and 65535. After 65535 is reached, the ID number wraps around to the first free number greater than five. When applicable, event messages generated by the Scenario module include the Scenario ID number and the current scenario script file line number.

Scenario-4.2 Request Script Information (ScriptStatus)

Each time the ScriptStatus variable is set, a summary of all of the currently accessible scenario threads is written to a file in the default scenario directory. The name of the file is "Scenario#<module>ScriptStatus<value>.txt" where the <module> field is replaced by the module number and <value> is replaced by the ScriptStatus variable's value. To get multiple status files, vary the value when issuing the ScriptStatus directive. The status file includes a time stamp and information about each script by ID number.

Scenario-4.3 Terminate Scenarios (KillThread)

The KillThread variable may be used to kill one or all of the scenario scripts at a time. To stop all threads, set the killthread variable to zero. To stop an individual script, set the killthread to that scenario script's ID.

Scenario-4.4 Debugging Aids

The following flags may be used to assist in debugging scenario scripts. While all messages generated when these flags are enabled will be written to the Event log file, display of Information, Scenario and Debug Event messages must be enabled before the messages generated by these flags may be viewed in the Event window. See section SIMSS-5.2.1.5, Event Message Filtering, for instructions on enabling display of Event messages.

When either of these flags are enabled, the Scenario module adds delays to the script processing so that the Event messages generated may be more easily seen, and so that the Event queue is not flooded with messages.

Scenario-4.4.1 Request Echoing of Directive Lines (EchoThread)

The EchoThread variable controls the listing of lines executed from scenarios to the event message window. To request echoing for a scenario, set this variable to the scenario's ID number. If the debug flag for this scenario is also set, these event messages are the information color (white). If this flag is set by itself, the event messages are the scenario color (dark gray). To disable echoing for all threads, set this variable to zero.

Scenario-4.4.2 Request Additional Debug Messages (DebugThread)

The DebugThread variable controls the generation of additional debug event messages while executing scripts. To enable this flag for a scenario, set it to the scenario's ID

number. These event messages have the debug color (light gray). Setting this variable to zero disables the additional debug messages for all threads.

Scenario-4.5 Default Scenario Directory (ScenarioPath)

The ScenarioPath string variable specifies a default path for scenario scripts. When GUI requests to start scenarios do not include a full pathname, this directory is used to complete the name. This variable also specifies where files created by the ScriptStatus variable are stored. This field MUST end with a slash character. Quotes are not used in setting this field. This variable is initially set to the scenario directory below the server executable directory.

Scenario-5.0 Displays

To access displays for a module, click in the center of the module in the project window. The following module pop-up menu will appear. There are no configuration screens for the Scenario module. The Run-time option is available only when the project is running. The "Remove" option can be used during project design to remove the module. The "About" option is for the display of generic module information.

Module Pop-Up Menu Item	Description
Configure	Access the Configuration menu for this module
Run-time	Access the Run-time menu for this module
Remove	Remove the module from the project
About	Display generic module information

Scenario-5.1 Configure

There are no configuration displays for this module.

Scenario-5.2 Run-time

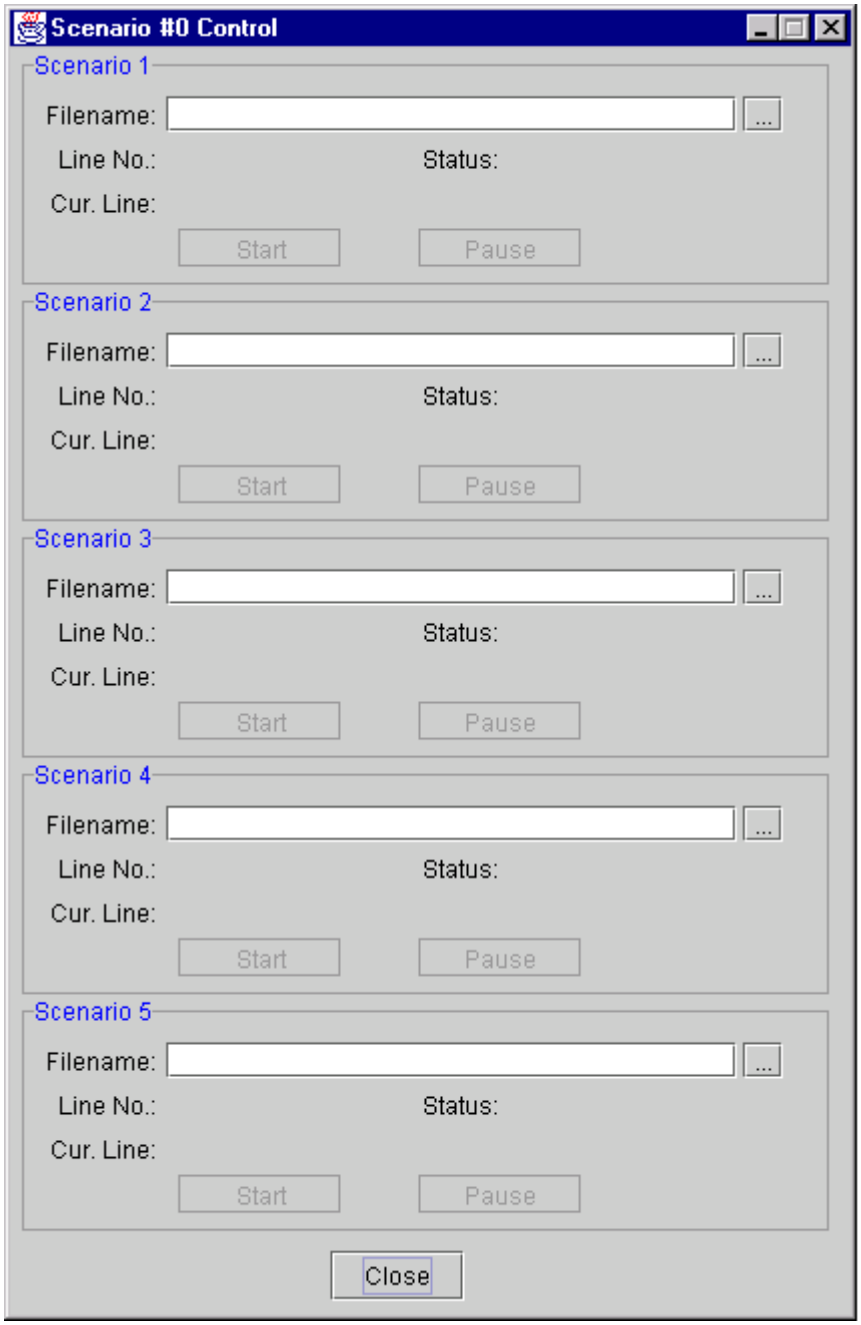
Click on the module pop-up menu "Run-time" option to request the Run-time menu.

Run-time Menu Item	Description
Control	Request Scenario Control Display

Scenario-5.2.1 Scenario Control Display

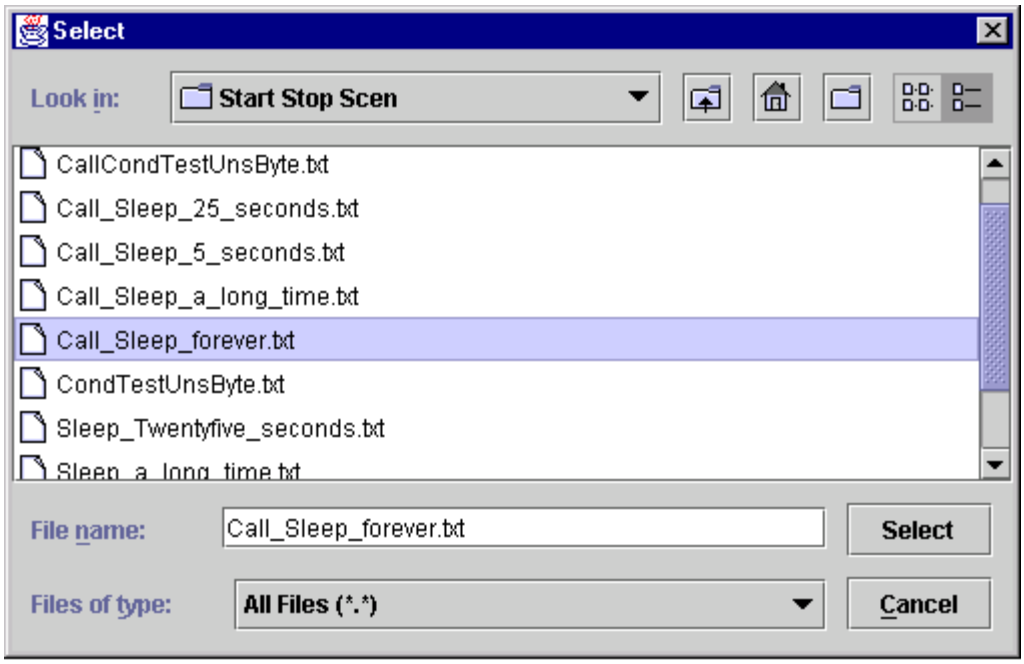
Select "Control" from the Run-time menu to show the Scenario Control Display. Up to five scenario files may be controlled and monitored with this display. The fields associated with each file are described in the following table. Click the **Close** button to dismiss the display.

Field	Description
Filename	Enter the full or relative pathname for the scenario file in this field.
File browser ...	Click on this button to the right of the filename field to invoke file browsing to select a file.
Start/Stop Button	This button's title changes to Start or Stop as appropriate. To start processing an entered file, click its **Start** button. To stop an executing file, click its **Stop** button.
Pause/Resume Button	This button's title changes to Pause or Resume as appropriate. To temporarily stop the processing of a scenario file, click its **Pause** button. To resume a file after pausing, click its **Resume** button.
Line No.	This field shows the line position within the file.
Cur. Line	This field shows the text of the most recently executed line.
Status	This field shows "Running", "Paused", or "Finished".



Scenario-5.2.2 Scenario File Browser

When any of the file browser buttons is selected from the Scenario Control Display, a file selection display similar to the next screen will be displayed. The current search directory will be shown in the "Look in:" field. Double click on the appropriate folder icons to navigate to the desired directory. Double click on a named folder to display its contents. Click to select a scenario file name. The file name should then appear in the File name field. Use the **Select** button to transfer the selected file name to the control display and dismiss the file selection window. Use the **Cancel** button to dismiss the display without selecting a file.



Scenario-5.3 About

Selecting the "About" option from the Scenario module pop-up menu produces a display that lists generic module information.

Scenario-6.0 Special Operating Instructions

Scenario-6.1 Addition of Delays to Scenario Files

Because of the fast processing of scenario files (e.g. 300 directives or more per second), the GUI Event Message window may become jammed by the volume of event messages produced in a very short time. Consequently, other displays may also be "hung up" by this undesirable effect. To prevent undesirable behavior it is recommended that delays be built into the scenario files so that no more than 20 directives are executed each second. If scenario event messages are being filtered from display in the event window, up to 50 directives may be executed each second.

```
; Sample scenario file with sleep directives
Set MOD_SS_DRNADSTEP 1
SET CEF_NR_ETRQ_1 2
SLEEP 1000    ; Sleep one second
SET AS2_TS_DETECTOR 3
SET AMR_NR_MWASPEED 65
SET AMR_SS_BITPKT1 1
SET AMR_SS_BITPKT4 1
SLEEP 5000    ; Sleep five seconds
SET EPS_SS_MAINBUSVR 0XFF
```

Serial Input Module

SerialInput-1.0 Overview

The Serial Input module provides the capability to receive serial data through a port on an ICS serial card in the host computer and to pass the data received to other modules.

SerialInput-2.0 Input

The Serial Input module does not have any input channels.

SerialInput-3.0 Output

Channel	Description
1	Data received from the serial port is passed on this channel

SerialInput-4.0 Container Items

The Serial Input module's container items are not accessible via operator directives, so they are not listed here.

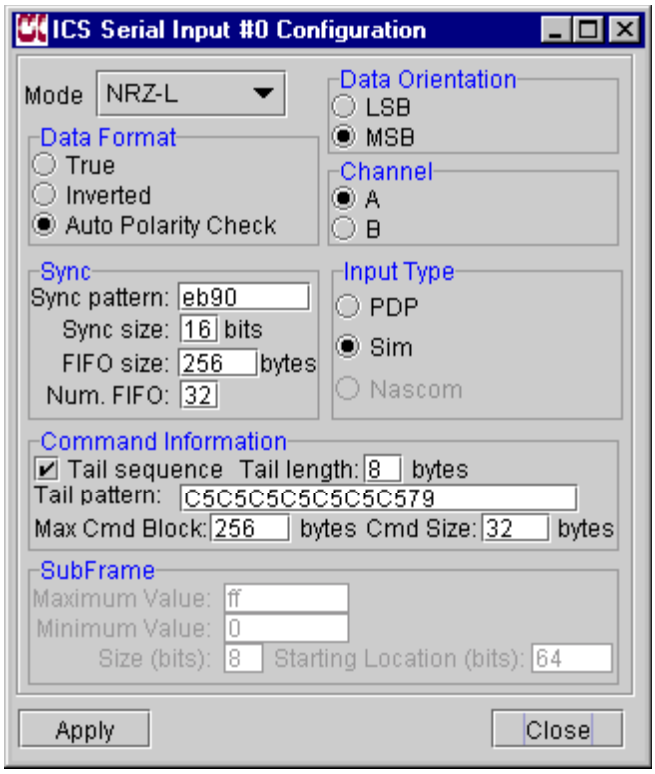
SerialInput-5.0 Displays

To access displays, click in the center of the module in the project window. The module pop-up menu will appear. The "Configure" and "Remove" options are only available during project design. The "Run-time" option is only available when the project is running.

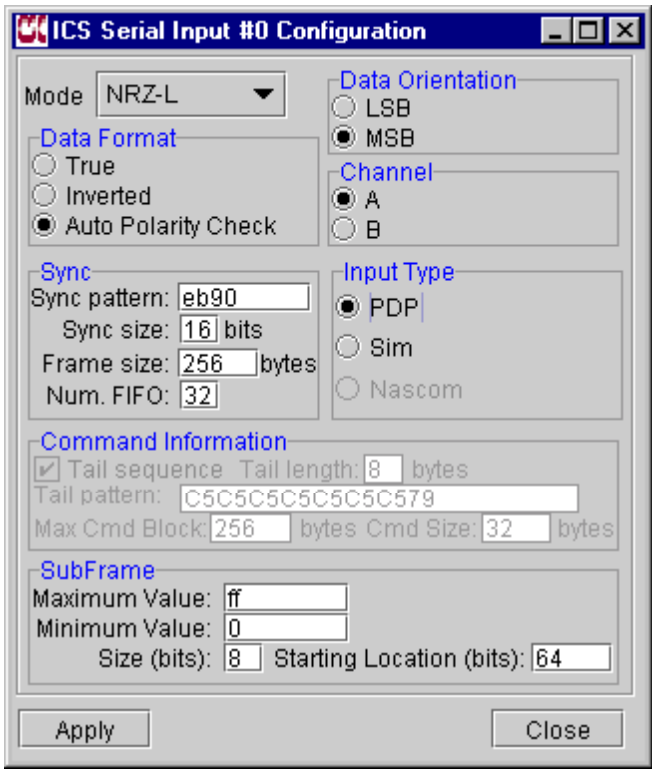
Module Pop-up Menu Item	Description
Configure	Access the module's configuration window
Run-time	Access the module's run-time menu
Remove	Remove the module from the project
About	Display generic module information

SerialInput-5.1 Configuration

Select the "Configure" item from the module pop-up menu to access the configuration window for the Serial Input module.



The default configuration is for the Sim Input Type. When the Input Type field is set to Sim, the Command Information section is enabled for entry and the SubFrame section is disabled.



When "PDP" is selected as the Input Type, the "SubFrame" options will be enabled, "Command Information" will be disabled, and "Frame Size" replaces "FIFO Size" within the Sync section of the display.

Parameter	Description
Mode	Select the Pulse Code Modulation (PCM) decoding method of the input stream from NRZ-L, NRZ-M, or NRZ-S
Data Orientation	Select "LSB" if the data stream is received with the least significant bit first. Select "MSB" if the data stream is received with the most significant bit first.
Data Format	Select "True" if the polarity of the data stream is normal. Select "Inverted" if the polarity of the data stream is inverted. Select "Auto Polarity Check (APC)" if the polarity is unknown.
Channel	Select "Channel A" or "Channel B" for the operation. "Channel A" refers to the first ICS board installed in the system, while "Channel B" refers to the second one.
Sync Pattern	Enter the synchronization pattern in hexadecimal format for the serial I/O card to receive data (maximum 8 digits).
Sync Size	Enter the size of the synchronization pattern in bits (maximum 32 bits)
Frame Size	Enter number of bytes per frame to be received (maximum 4096 bytes).
FIFO Size or Frame Size	Enter the FIFO' s size (maximum 4096) in "SIM" mode. Enter the data frame size (maximum 4096 bytes) in "PDP" mode
Num. FIFO	Enter number of FIFO to receive data (maximum 99)
Input Type	Select "PDP" if the input stream is telemetry. Select "SIM" if the input stream is command.
Tail Sequence	If the command bit stream has a tail sequence, check this box.
Tail Length	Enter the command tail sequence size in bytes (maximum 8 bytes)
Tail Pattern	Enter the command tail sequence pattern in hexadecimal digits (maximum 16 digits).
Max Cmd Size	Enter the maximum command size expected in bytes (maximum 4096 bytes)
Cmd Size	Enter command size in bytes (1 to Max Cmd Size). Serial Input module will separately pass to next module sync patterns, bytes of command data (specified by "Cmd Size"), and post-amble. If Cmd Size equals Max Cmd Size, Serial Input module will pass to next module sync patterns, command(s) data, and post-amble in one packet.
Maximum Value	Enter telemetry subframe's maximum value.
Minimum Value	Enter telemetry subframe's minimum value.
Size	Enter telemetry subframe size in bits (maximum 16 bits)
Starting Location	Enter the starting location for the telemetry subframe in bits (greater than 32)

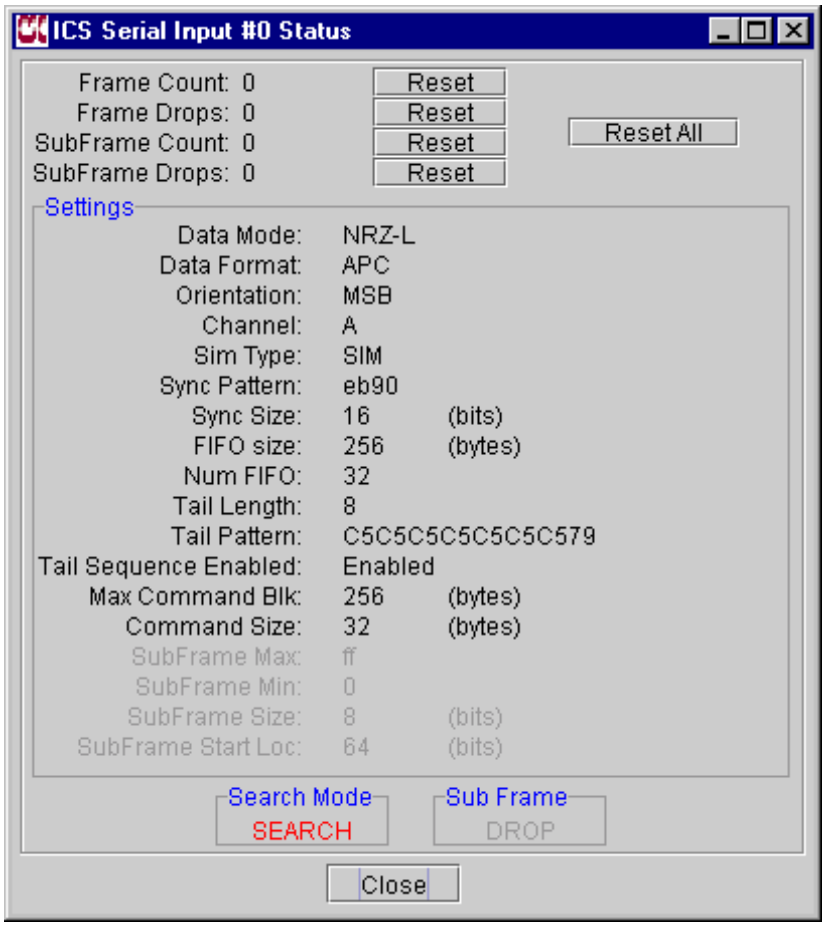
Click **Apply** button to configure any changes. Click **Close** button to dismiss the display.

SerialInput-5.2 Run-time

Click the module pop-up menu "Run-time" option to access the Run-time menu with the following choices.

Run-time Menu Item	Description
Display Status	Show the Status window of the Serial Input module
Display Dump	Show the Dump formats available
Resume	Restart the module
Pause	Stop the module

SerialInput-5.2.1 Display Status



This display shows the number of frames and sub-frames that have been received and dropped. It also shows the current configuration settings. Please refer back to the table in section SerialInput-5.1 for descriptions of the settings.

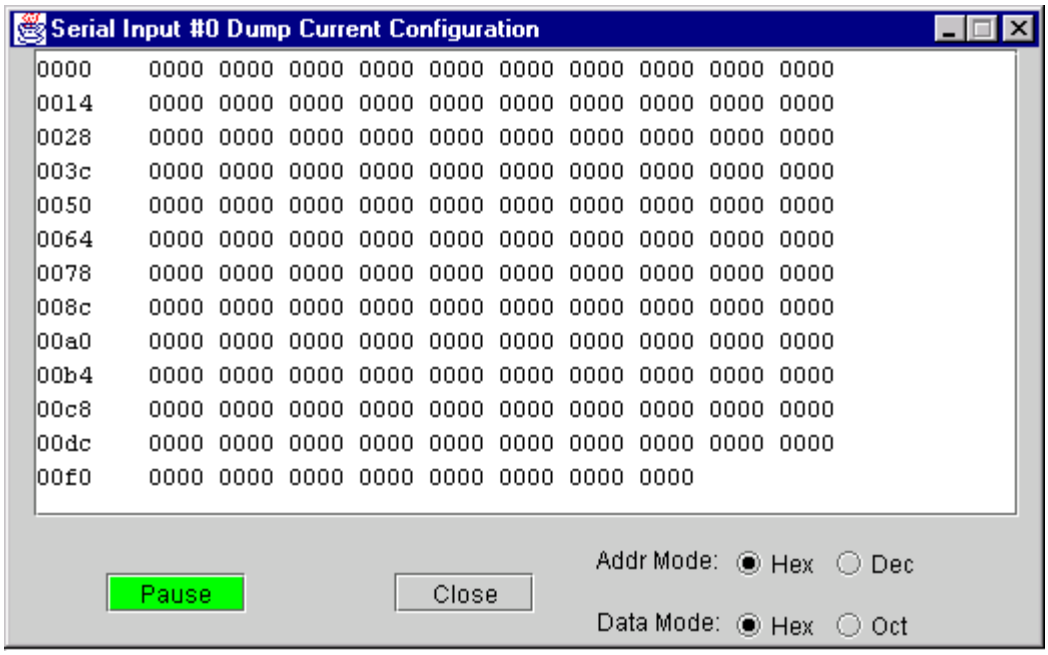
Search Mode and Sub Frame status fields are only applicable when receiving telemetry data.

Click any **Reset** button to change its associated counter to zero.

Click the **Close** button to dismiss the display.

SerialInput-5.2.2 Display Dump

Select the "Display Dump" option to view the available dump options. Currently there is only one format available. When this option is selected the data received from the serial I/O port is displayed.



Button	Description
Addr Mode	Select "Hex" to display the memory address in hexadecimal format Select "Dec" to display the memory address in decimal format
Data Mode	Select "Hex" to display data in hexadecimal format Select "Oct" to display data in octal format
Pause	Pause updating the data contained in the window. The button name changes to **Cont**.
Cont	Continue updating a previously paused display. The button name changes to **Pause**.
Close	Close the window

SerialInput-5.2.3 Resume

After an individual module's execution has been paused, select "Resume" from the Run-time menu to restart the module. The color around the module's border will change from red (indicates stopped state) to green (indicates run state).

SerialInput-5.2.4 Pause

Select the "Pause" option from the Run-time menu to temporarily stop the Serial Input module's processing. The color around the module's border will change from green (indicates run state) to red (indicates stopped state). The Serial Input module can be re-configured by selecting the "Configure" option of the module pop-up menu. To return to run mode, select the "Resume" option.

SerialInput-5.3 Remove

Select this option to remove the Serial Input module from the project.

SerialInput-5.4 About

Selecting the "About" option from the module pop-up menu produces a display that lists the number of input channels, number of output channels, whether directives are allowed, names of authors and the module's version number.

SerialInput-6.0 Special Operating Instructions

SerialInput-6.1 "Max Cmd Block" and "Cmd Size" Configuration

In "Sim" mode, the Serial Input module has two different ways to pass received command data to next module, depending on how "Max Cmd Block" and "Cmd Size" are configured. "Max Cmd Block" is the maximum size of a block of command data including the sync patterns and tail sequence. "Cmd Size" is the size of each command within a block of command data. A block of command has from one to many commands.

1. If a block of command data has multiple commands and user wants individual command to be passed to next module whenever it is fully received, then the "Cmd Size" should be set to exactly the size of one command. In this case, Serial Input module will pass sync patterns, first command, second command, third command… and finally tail sequence to next module in separate packets.

2. If user wants the whole block of command data from sync patterns to tail sequence to be passed to next module in one packet (no matter how many commands included within the block), then the "Cmd Size" must be equal to the "Max Cmd Block". In this case, Serial Input module will pass to next module the whole block of data from sync patterns to tail sequence or to "Max Cmd Block" bytes, whichever comes first.

Serial Output Module

SerialOutput-1.0 Overview

The Serial Output module provides the capability to transmit serial data through ports on an ICS serial card in the host computer.

SerialOutput-2.0 Input

Channel	Description
1	Receive data from upper module through this channel

SerialOutput-3.0 Output

Channel	Description
1	Pass transmitted data to next module through this channel

SerialOutput-4.0 Container Items

The Serial Output module's container items allow for changing the output frequency while the project is running. It is suggested that data transmission be halted prior to changing the output frequency to prevent loss of data.

Name	Type	Description
SerialFrequency	Fixed	Clock frequency in hertz at which the data will be transmitted.
SerialZZChannel	Fixed	Set to re-enable output after changing the transmit frequency. The argument is zero for channel A and one for channel B. NOTE: This does not re-enable output from the upstream module.

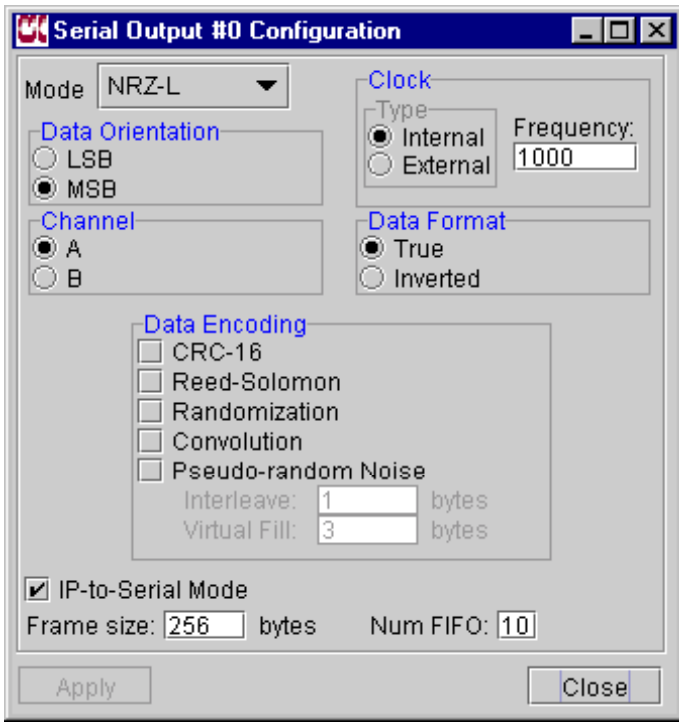
SerialOutput-5.0 Displays

To access displays, click in the center of the module in the project window. The module pop-up menu will appear. The "Configure" and "Remove" options are only available during project design mode. The "Run-time" option is only available when the project is running.

Module Pop-up Menu Item	Description
Configure	Access the Serial Output configuration window
Run-time	Access to module's Run-time menu
Remove	Remove the module from the project
About	Display generic module information

SerialOutput-5.1 Configuration

Select the "Configure" item of the module pop-up menu to access the configuration window of the Serial Output module.



Parameter	Description
Mode	Select the PCM encoding method of the data stream from NRZ-L, NRZ-M, NRZ-S, BIO-L, BIO-M, or BIO-S.
Clock Type	In this version, only the internal clock is used.
Clock Frequency	Enter the frequency in Hz to transmit data out.
Data Orientation	Select "LSB" if the data stream is transmitted with the least significant bit first. Select "MSB" if the data stream is transmitted with the most significant bit first
Channel	Select "Channel A" or "Channel B" for the operation. "Channel A" refers to the first ICS board installed in the system while "Channel B" refers to the second one.
Data Format	Select "True" if the polarity of the data stream is normal. Select "Inverted" if the polarity of the data stream is inverted.
Data Encoding	Click the appropriate boxes to enable "CRC-16", "Reed-Solomon", "Randomization", "Convolution" and "Pseudo-random Noise" encoding.
Interleave	If "Reed-Solomon" encoding is enabled, enter the length in bytes for interleave.
Virtual Fill	If "Reed-Solomon" encoding is enabled, enter number of bytes for virtual fill.

IP-to-Serial Mode	Check this option if the Serial Output module is configured to receive data from an Input IP module.
Frame Size	Enter total number of bytes per frame to be transmitted (maximum 4096 bytes)
Num. FIFO	Enter number of FIFO to contain transmitting data (maximum 99)

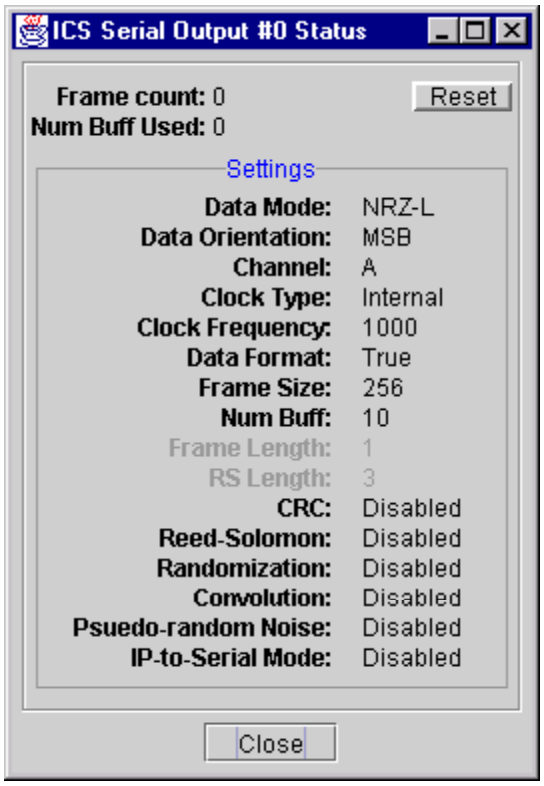
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

SerialOutput-5.2 Run-time

Click on the module pop-up menu "Run-time" option to request the Run-time menu.

Run-time Menu Item	Description
Display Status	Show the status window of the module
Display Dump Menu	Show options for displaying data within a frame
Resume	Restart the module
Pause	Stop the module

SerialOutput-5.2.1 Display Status



This display shows the current frame counter, number of FIFO used, and the currently configured settings. Please refer to the table in section SerialOutput-5.1 for descriptions of the settings.

Click **Reset** to clear the frame counter to zero.

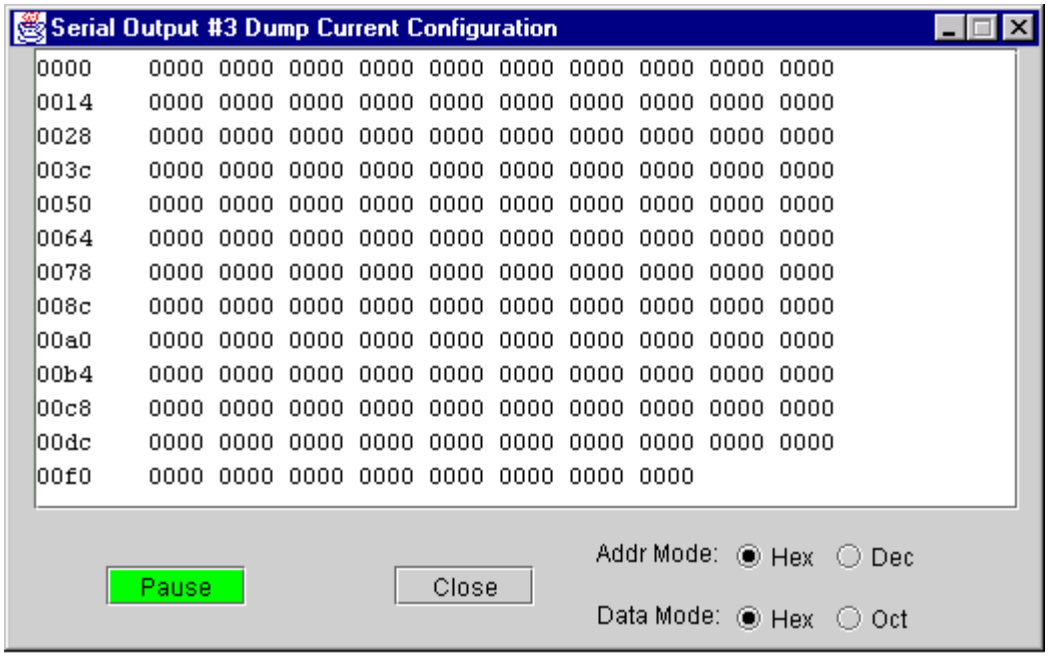
Click **Close** to close the window.

Note: if the Serial Output module is configured to receive data from an Input IP module, the "Num FIFO Used" value varies from 0 to "Num FIFO" value, defending on the difference of data rate between Input IP module and Serial Output module. In any other configuration the "Num FIFO Used" is always at "Num FIFO" value.

SerialOutput-5.2.2 Display Dump

Select the "Display Dump" option to view the available dump formats. Currently there is only one format available. When this option is selected, the output data buffer is displayed. This display has the following standard dump buttons.

Buttons	Description
Addr Mode	Select "Hex" to display the addresses in hexadecimal format Select "Dec" to display the addresses in decimal format
Data Mode	Select "Hex" option to display data in hexadecimal format Select "Oct" option to display data in octal format
Pause	Freeze the data window contents. The **Pause** button is renamed **Cont**.
Cont	Resume updating the data window contents. The **Cont** button is renamed **Pause**.
Close	Close the window



SerialOutput-5.2.3 Resume

After an individual module has been stopped, select "Resume" from the Run-time menu to restart the module. The module's border will change from red (indicates stopped state) to green (indicating it has entered the run state).

SerialOutput-5.2.4 Pause

Select "Pause" from the Run-time menu to temporarily stop the Serial Output module's processing. The module's border will change from green (indicates run state) to red (indicates stopped state). While Paused, the Serial Output module can be re-configured by selecting the "Configure" option of the module pop-up menu. To return to run mode, select "Resume" from the Run-time menu.

SerialOutput-5.3 Remove

Select this option to remove the Serial Output module from the project.

SerialOutput-5.4 About

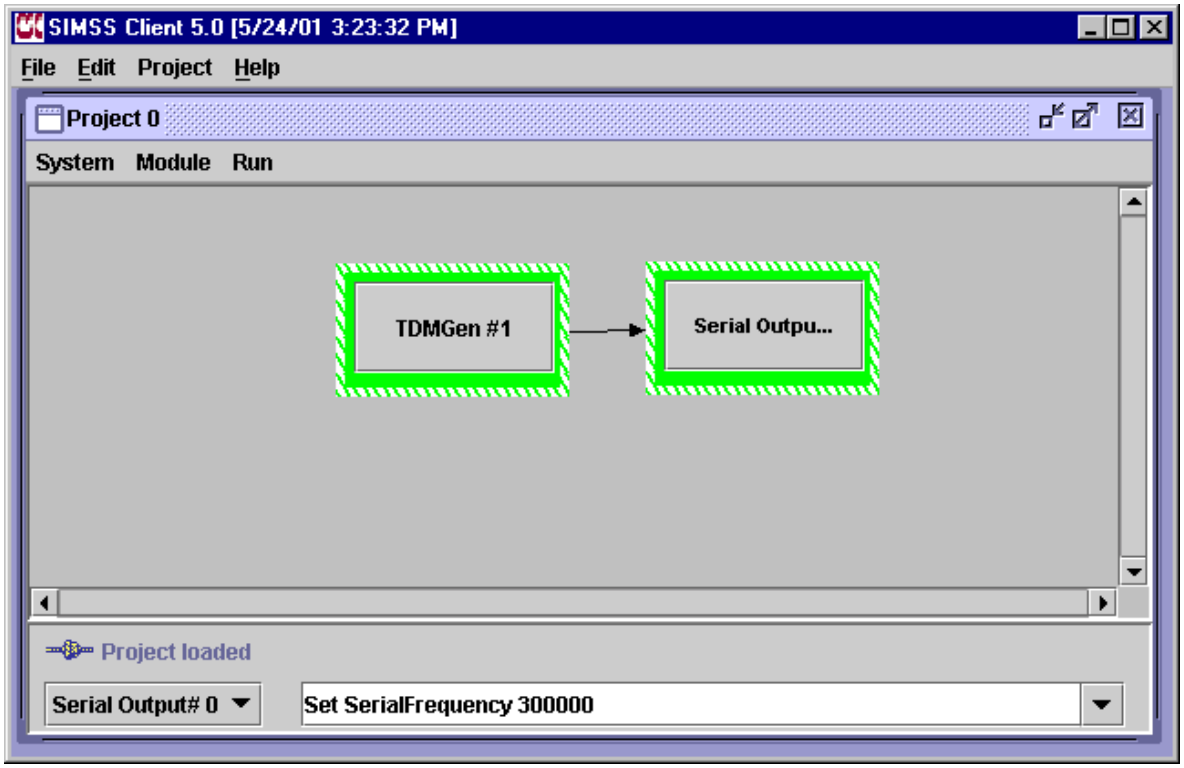
Selecting the "About" option from the module pop-up menu produces a display that lists the number of input channels, number of output channels, whether directives are allowed and the module's version number.

SerialOutput-6.0 Special Operating Instructions

SerialOutput-6.1 Operation with Input IP Module

When the Serial Output module is configured to receive data from an Input IP module, the "IP-to-Serial Mode" option should be checked. When the project is running, bring up the Status window of the Serial Output module to monitor the "Num FIFO Used", and look at the event messages to know the input data rate of IP data. By looking at "Num FIFO Used", buffer under flow or buffer over flow can be detected. It is necessary to set the data rate of Serial Output module about the same as the data rate of the IP data. See "Setting Frequency on the Fly" in this section for the instructions on how to set a new frequency during run-time, or manually stop the Serial Output module to set a new frequency and re-start it again.

SerialOutput-6.2 Setting Frequency on the Fly



The directive window of the project GUI can be used to change the serial output frequency during run-time. Follow the steps given below.

1. Click on the drop-down menu on the bottom left of the directive window to select the Serial Output module for which to change the frequency value

2. In the directive input field, type "Set SerialFrequency " + integer value in Hz, then press the <Enter> key. For example, to set the new frequency to 300000 Hz:

   Set SerialFrequency 300000 + <Enter>

3. Type "Set SerialZZChannel " + current channel value in integer (A = 0, B = 1), and then press the <Enter> key. For example, if the current channel is channel B:

   Set SerialZZChannel 1 + <Enter>

4. Check that the serial output channel is operating at the new frequency.

AVTEC Serial Input Module

AvtecSerialInput-1.0 Overview

The Avtec serial Input module provides the capability to receive serial data through a port on an Avtec serial card in the host computer and to pass the data received to other modules.

AvtecSerialInput-2.0 Input

The AvtecSerialInput module does not have any input channels.

AvtecSerialInput-3.0 Output

Channel	Description
1	Data received from the serial port is passed on this channel

AvtecSerialInput-4.0 Container Items

The AvtecSerialInput module's container items are not accessible via operator directives, so they are not listed here.

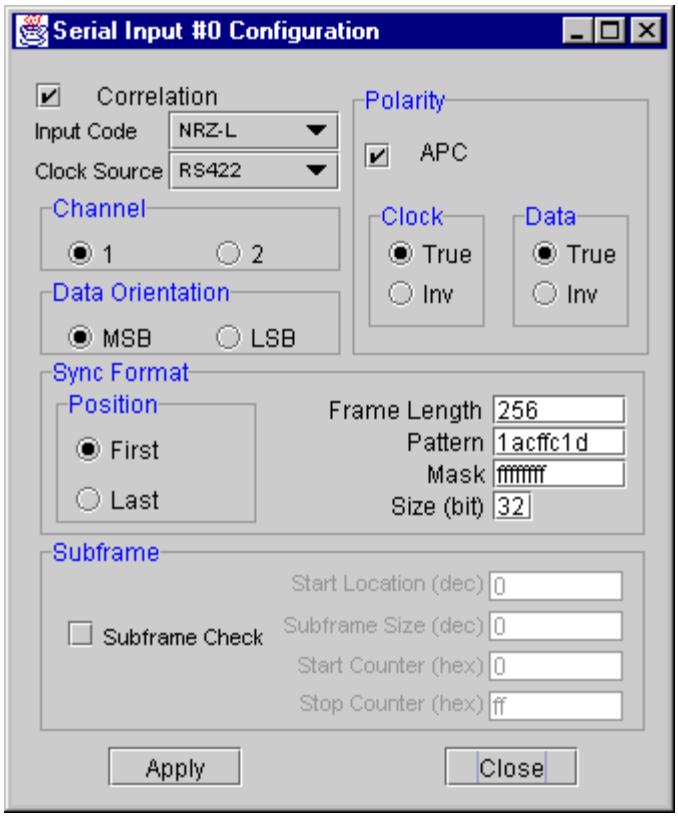
AvtecSerialInput-5.0 Displays

To access displays for this module, click in the center of the AvtecSerialInput module in the project window. The following module pop-up menu choices will appear. The "Configure" and "Remove" options are only available during project design. The "Run-time" option is only available when the project is running.

Module Pop-up Menu Item	Description
Configure	Access the Serial Input configuration window
Run-time	Access the module's run-time menu
Remove	Remove the module from the project
About	Display generic module information

AvtecSerialInput-5.1 Configuration

Select the "Configure" item from the module pop-up menu to access the configuration window of the Serial Input module.



Parameter	Description
Code	Select the Pulse Code Modulation (PCM) decoding method of the input stream from NRZ-L, NRZ-M, or NRZ-S (Bi0_L, Bi0_M, and Bi0_S are available in only output mode).
Data Orientation	Select LSB if the data stream is received with the least significant bit first. Select MSB if the data stream is received with the most significant bit first.
Clock & data Polarity	Select True if the polarity of the data stream is normal. Select Inverted if the polarity of the data stream is inverted. Select Auto Polarity Check (APC) if the polarity is unknown.
Correlation	If selected, performs frame sync. Check. Otherwise throughput
Channel	Select channel 1 or channel 2 for the operation.
Sync Pattern	Enter the synchronization pattern in hexadecimal format for the serial I/O card to receive data (maximum 8 digits).
Sync Size	Enter the size of the synchronization pattern in bits (maximum 32 bits).
Frame Size	Enter number of bytes per frame to be received (maximum 4096 bytes).
Subframe check	If checked, subframe search is performed.
Stop counter	Enter telemetry subframe's maximum value.
Start counter	Enter telemetry subframe's minimum value.

Subframe size	Enter telemetry subframe size in bits (maximum 16 bits)
Starting Location	Enter the starting location for the telemetry subframe in bits (greater than 32)

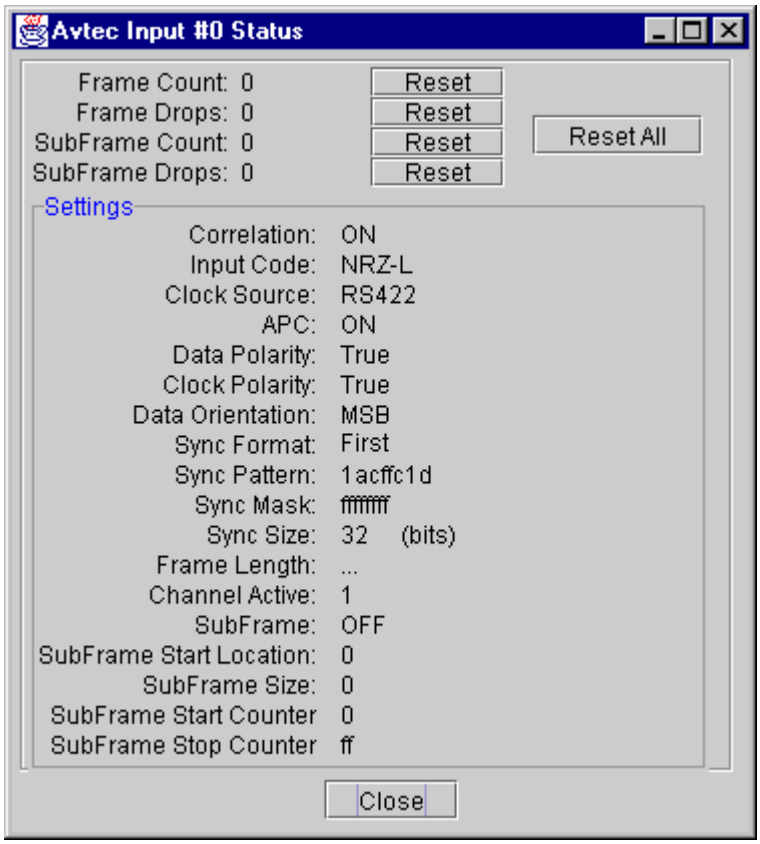
Click **Apply** button to configure any changes. Click **Close** button to dismiss the display.

AvtecSerialInput-5.2 Run-time

Click the module pop-up menu "Run-time" option to access the Run-time menu with the following choices.

Run-time Menu Item	Description
Display Status	Show the Status window of the Serial Input module
Display Dump Menu	Show options for displaying data within a frame
Resume	Restart the module
Pause	Stop the module

AvtecSerialInput-5.2.1 Display Status



This display shows the number of frames and sub-frames that have been received and dropped. It also shows the current configuration settings. Please refer back to the table in section AvtecSerialInput-5.1 for descriptions of the settings.

Search Mode and Sub Frame status fields are only applicable when receiving telemetry data**.**

Click any Reset button to change its associated counter to zero.

Click the Close button to dismiss the display.

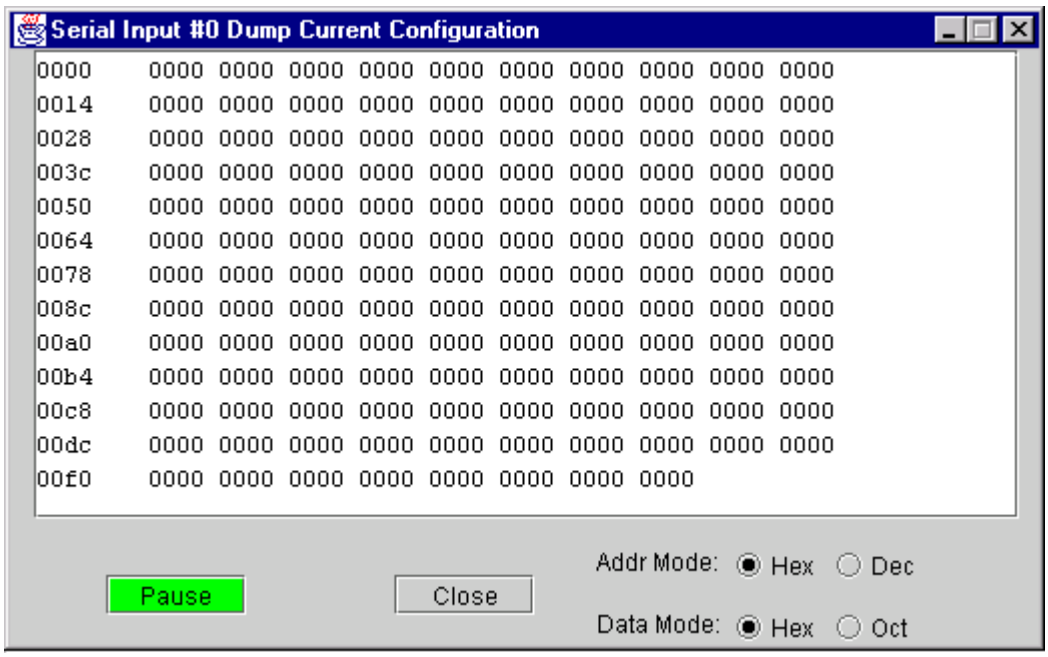
AvtecSerialInput-5.2.2 Display Dump Menu

Select the "Display Dump" option to view the available dump options.

Display Dump Menu Item	Description
Display Buffer	The input data buffer is displayed as received.

AvtecSerialInput-5.2.2.1 Display Buffer Current Configuration

When this option is selected the data received from the serial I/O port is displayed exactly as it is received.



Button	Description
Addr Mode	Select **Hex** to display the memory address in hexadecimal format Select **Dec** to display the memory address in decimal format
Data Mode	Select **Hex** to display data in hexadecimal format Select **Oct** to display data in octal format
Pause	Pause updating the data contained in the window. The button name changes to **Cont**.
Cont	Continue updating a previously paused display. The button name changes to **Pause**.
Close	Close the window

AvtecSerialInput-5.2.3 Resume

After an individual module has been stopped, select "Resume" from the Run-time menu to restart the module. The color around the module's border will change from red-striped (indicates stopped state) to green-striped (indicates run state).

AvtecSerialInput-5.2.4 Pause

Select the "Pause" option from the Run-time menu to temporarily stop the Serial Input module's processing. The color around the module's border will change from green-striped (indicates run state) to red-striped (indicates stopped state). The Serial Input module can be re-configured by selecting the "Configure" option of the module pop-up menu. To return to run mode, select the "Resume" option.

AvtecSerialInput-5.3 Remove

Select this option to remove the Serial Input module from the project.

AvtecSerialInput-5.4 About

Selecting the "About" option from the module pop-up menu produces a display that lists the number of input channels, number of output channels, whether directives are allowed, names of authors and the module's version number.

AvtecSerialInput-6.0 Special Operating Instructions

There are no special operating instructions for this release.

AVTEC Serial Output Module

AvtecSerialOutput-1.0 Overview

The AVTEC Serial Output module provides the capability to transmit serial data through ports on an Avtec serial card in the host computer.

AvtecSerialOutput-2.0 Input

Channel	Description
1	Data to be transmitted through the serial port.

AvtecSerialOutput-3.0 Output

Channel	Description
1	Data being transmitted through the serial port.

AvtecSerialOutput-4.0 Container Items

The module's container items are not accessible via operator directives, so they are not listed here.

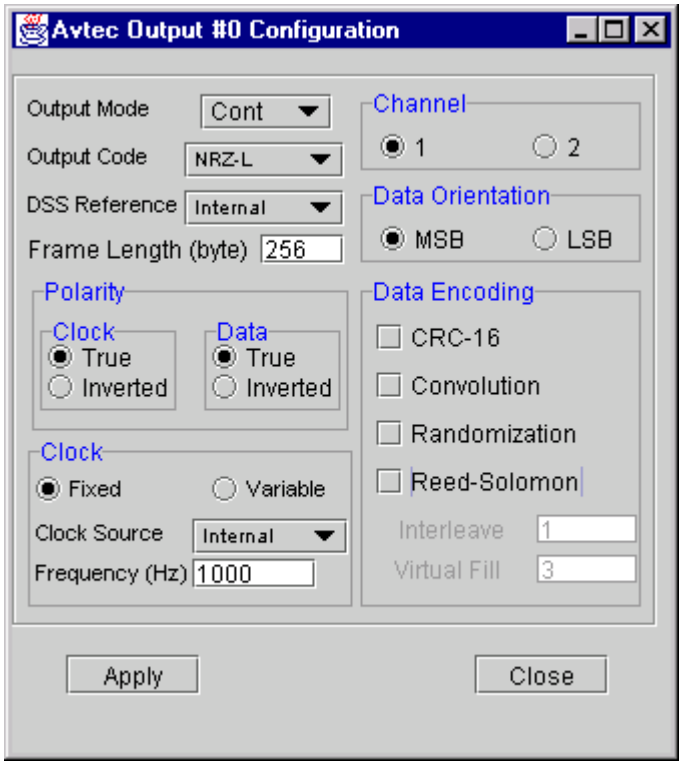
AvtecSerialOutput-5.0 Displays

To access the displays for this module, click in the center of the module in the project window. The "Configure" and "Remove" options are only available during project design. The "Run-time" option is only available when the project is running.

Module Pop-up Menu Item	Description
Configure	Access the Serial Output configuration window
Run-time	Access to module's Run-time menu
Remove	Remove the module from the project
About	Display generic module information

AvtecSerialOutput-5.1 Configuration

Select the "Configure" item of the module pop-up menu to access the configuration window of the Serial Output module.



Parameter	Description
Mode	Cont, Burst, Toggle Cont – put out continuous data stream Burst – put out burst data Toggle – put out toggle data with fill
Code	Select the PCM encoding method of the data stream from **NRZ-L, NRZ-M, NRZ-S, BIO-L, BIO-M,** or **BIO-S**.
Clock Type	In this version, only the internal clock is used.
Clock Frequency	Enter the frequency in Hz to transmit data out.
Data Orientation	Select **LSB** if the data stream is transmitted with the least significant bit first. Select **MSB** if the data stream is transmitted with the most significant bit first
Channel	Select channel **A** or channel **B** for the operation
Fixed/Variable clock	If fixed selected, output data with fixed clock. If variable selected, Frequency changes to prevent underflow/overflow FIFO. The variable selection is used for IP to serial conversion.
Data Format	Select **True** if the polarity of the data stream is normal. Select **Inverted** if the polarity of the data stream is inverted.
Data Encoding	Click the appropriate boxes to enable **CRC**, **Reed-Solomon**, **Randomization**, **Convolution** or **Pseudo-random Noise** encoding.
Frame Length	Enter the frame length in bytes before encoding (maximum 4096 bytes)
RS Length	If Reed-Solomon encoding is enabled, enter the length in bytes of RS data to be stored in the frame.

Frame Size	Enter total number of bytes per frame to be transmitted (maximum 4096 bytes)

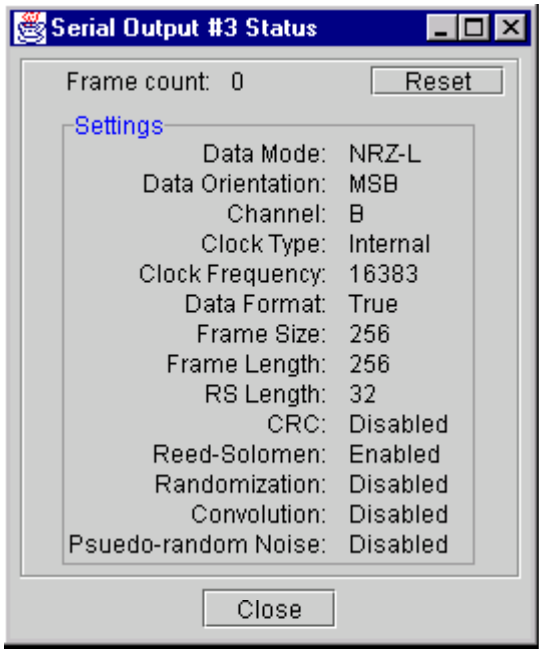
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

AvtecSerialOutput-5.2 Run-time

Click on the module pop-up menu "Run-time" option to request the Run-time menu.

Run-time Menu Item	Description
Display Status	Show the status window of the Serial Output module
Display Dump Menu	Show options for displaying data within a frame
Resume	Restart the module
Pause	Stop the module

AvtecSerialOutput-5.2.1 Display Status



This display shows the current frame counter and the currently configured settings. Please refer to the table in section AvtecSerialOutput-5.1 for descriptions of the settings.

Click **Reset** to clear the frame counter to zero.

Click **Close** to close the window.

AvtecSerialOutput-5.2.2 Display Dump Menu

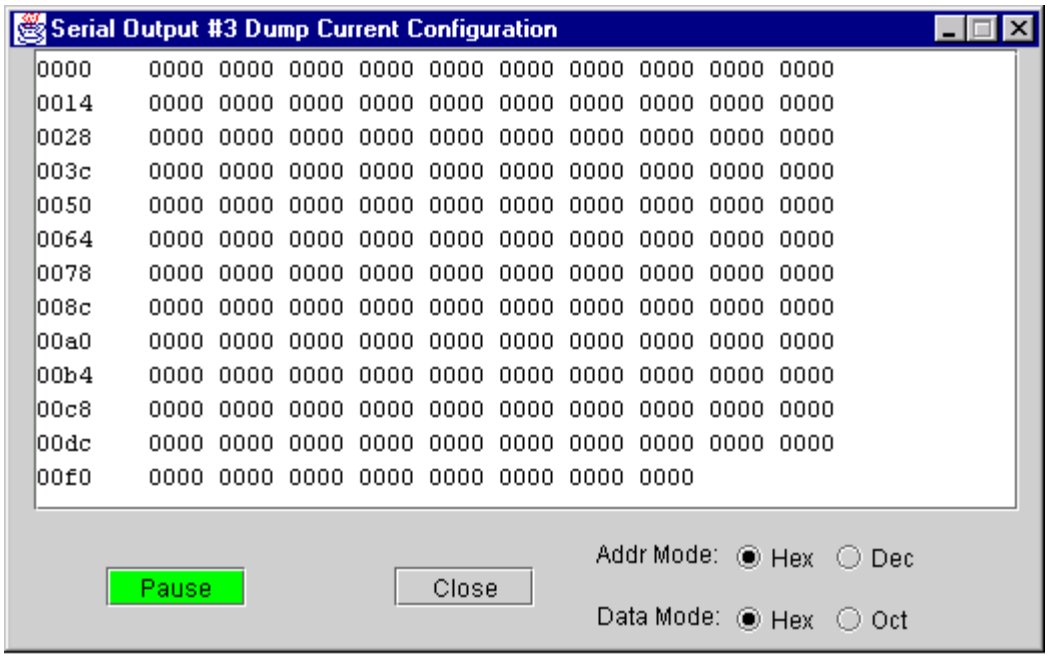
Select the "Display Dump" option to view the available dump formats.

Display Dump Menu Item	Description
Display Buffer Current Configuration	The data buffer is displayed as configured.

<u>AvtecSerialOutput-5.2.2.1 Display Buffer Current Configuration</u>

When this option is selected, the output data buffer is displayed as configured. This display has the following standard dump buttons.

Buttons	Description
Addr Mode	Select **Hex** to display the memory address in hexadecimal format Select **Dec** to display the memory address in decimal format
Data Mode	Select **Hex** to display data in hexadecimal format Select **Oct** option to display data in octal format
Pause	Freeze the data window contents. **Pause** button is renamed **Cont**.
Cont	Resume updating the data window contents. **Cont** button is renamed **Pause**.
Close	Close the window



AvtecSerialOutput-5.2.3 Resume

After an individual module has been stopped, select "Resume" from the Run-time menu to restart the module. The color around the module's border will change from red-striped (indicates stopped state) to green-striped (indicate run state).

AvtecSerialOutput-5.2.4 Pause

Select "Pause" from the Run-time menu to temporarily stop the Serial Output module's processing. The color around the module's border will change from green-striped (indicates run state) to red-striped (indicates stopped state). The Serial Output module can be re-configured by selecting the "Configure" option of the module pop-up menu. To return to run mode, select "Resume" from the Run-time menu.

AvtecSerialOutput-5.3 Remove

Select this option to remove the Serial Output module from the project.

AvtecSerialOutput-5.4 About

Selecting the "About" option produces a display listing the number of input and output channels, whether directives are allowed and the module's version number.

AvtecSerialOutput-6.0 Special Operating Instructions

There are no special operating instructions for this release.

Transmit File (TxFile) Module

TxFile-1.0 Overview

The Transmit File (TxFile) module reads in data from a file according to user-defined parameters and sends the data out unformatted. It operates in either manual mode, where data blocks are loaded and sent out one at a time under user control, or automatic mode, where data blocks are loaded and sent out under user-defined parameters. TxFile can transmit the data in IP transmission mode or in serial transmission mode. TxFile can process a file in normal order or reversed order (the first byte of the file is read and sent out as last byte, the last byte of the file is read and sent out as first byte). If the data file was created by the Log Module and has log headers, it is possible to retransmit the file with the same relative timing between blocks.

TxFile-2.0 Inputs

The TxFile module does not technically have any input channels. It is not possible to create an input link to this module. Its lone output channel is actually bi-directional, providing the down-linked module an optional method for controlling the flow of output data. This method is only used when the TxFile is connected to a module requesting serial transmission of data.

TxFile-3.0 Outputs

Channel	Description
1	Data read from file

TxFile-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the Set and Get directives to access items with a fixed, integer, float, or string types. Use the SetBuffer and GetBuffer directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

Name	Type	Description
TxFileName	String	Pathname of file to send
TxFileStart	Integer	The position of the first block begin to read
TxFileSize	Fixed	Size of file in bytes
TxFilePosition	Integer	Current file pointer position
TxFileTransmitCount	Integer	Number of blocks transmitted
TxFileTransmitBuffer	Buffer	Most recent buffer loaded from file
TxFileBlockSize	Integer	Size of block to load and then transmit
TxFileReversed	Integer	Reversed order flag

TxFileMode	Integer	File Read Modes: (1) Offset (2) Log file (playback a logged file with header) (3) Sync (4) Length **Note:** (3) & (4) not implemented yet
TxFileModeOffset	Integer	Offset from the start position of previous block
TxFileLogTiming	Integer	Use log file timing instead of the interval
TxFileAuto	Integer	Automatic or manual mode flag
TxFileLoad	Integer	Flag to load a block
TxFileFileLoopFlag	Integer	Flag for number of times a whole file to be read and sent out
TxFileLoopCount	Integer	Number of times a whole file to be read and sent out. **Note:** a whole file would be read and sent out indefinite times if the counter is set to be equal or less than 0.
TxFileBlockFlag	Integer	Flag for number of blocks to be read and sent out
TxFileReadCount	Integer	Number of blocks to be read
TxFileSendCount	Integer	Number of times each block to be sent out
TxFileInterval	Float	Interval between two blocks being sent in milliseconds
TxFileSend	Integer	Flag to start transmission
TxFileContinueSend	Integer	Flag to continuously transmit
TxFileStop	Integer	Flag to stop transmission
TxFileTransmitMode	Integer	Output transmission mode (0=IP (default), 1=serial, 2=force IP mode)
TxFileSerialStatus	Integer	Serial transmission status (0=stopped automatically, 1=stopped by operator, 2=running, 3=waiting for send key)

TxFile-5.0 Displays

To access the displays for this module, first click on the center of the module in the project window. The following items will appear in a pop-up menu.

Module Pop-Up Menu Item	Description
Configure	Access the configuration menu for the module
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

TxFile-5.1 Configuration Displays

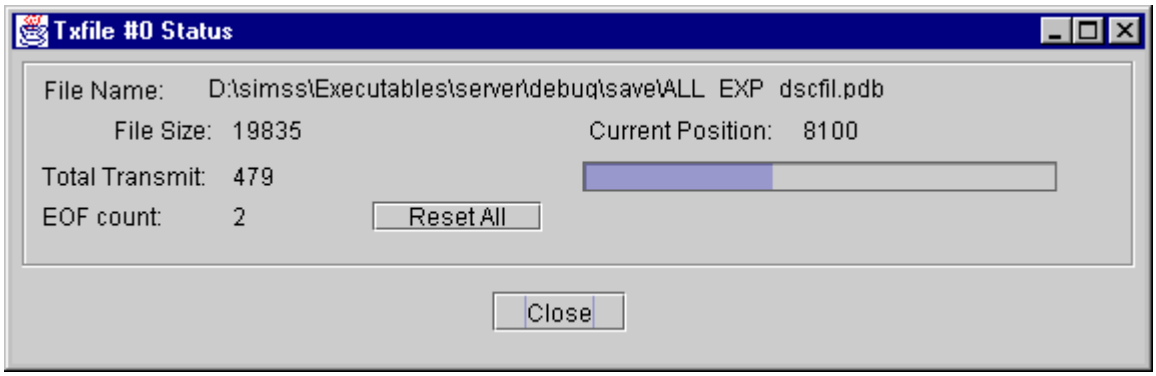
There are no configuration displays for this module. However, if during run-time TxFile receives a request for more data signal from the down-linked module, the transmission mode will be automatically reconfigured to serial and an event message will be generated to inform the operator.

TxFile-5.2 Run-time Displays

Run-time Menu Item	Description
Display Status	Display TxFile module status
Display Transmit Buffer	Display current block for transmission
Send From File	Control TxFile module processing

TxFile-5.2.1 Status Display

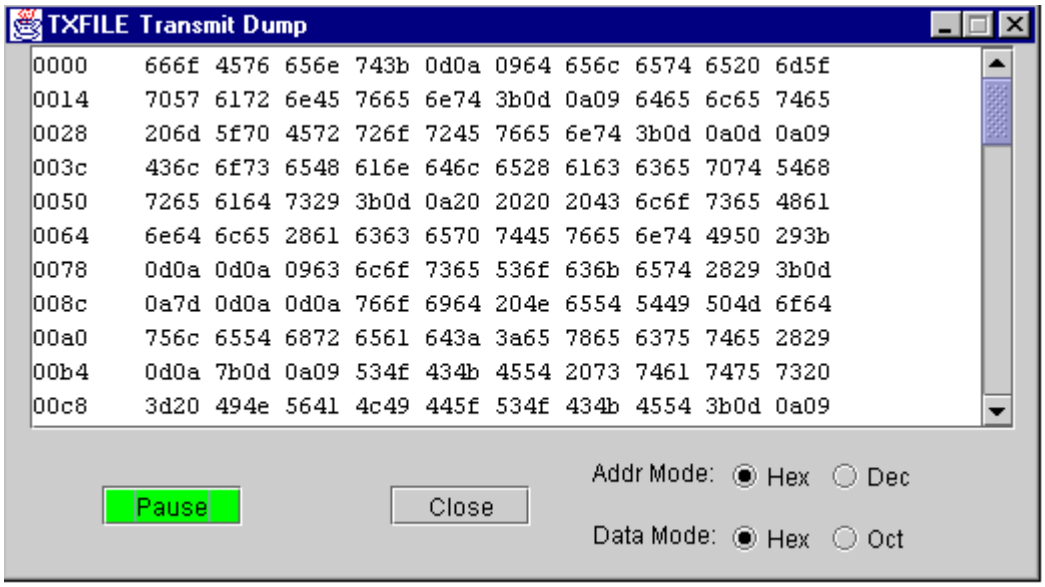
The status display shows the current filename, the size of the file, current file pointer position, and the total number of blocks transmitted.



Status Display Field	Description
File Name	Pathname of the file being transmitted
File Size	Size (in bytes) of the file being transmitted
Current Position	Current byte position in file
Total Transmit	Total number of blocks transmitted
EOF Count	Total number of End Of File reached.
Reset All	Click to clear the **Total Transmit** count and **EOF Count**

TxFile-5.2.2 Transmit Buffer Display

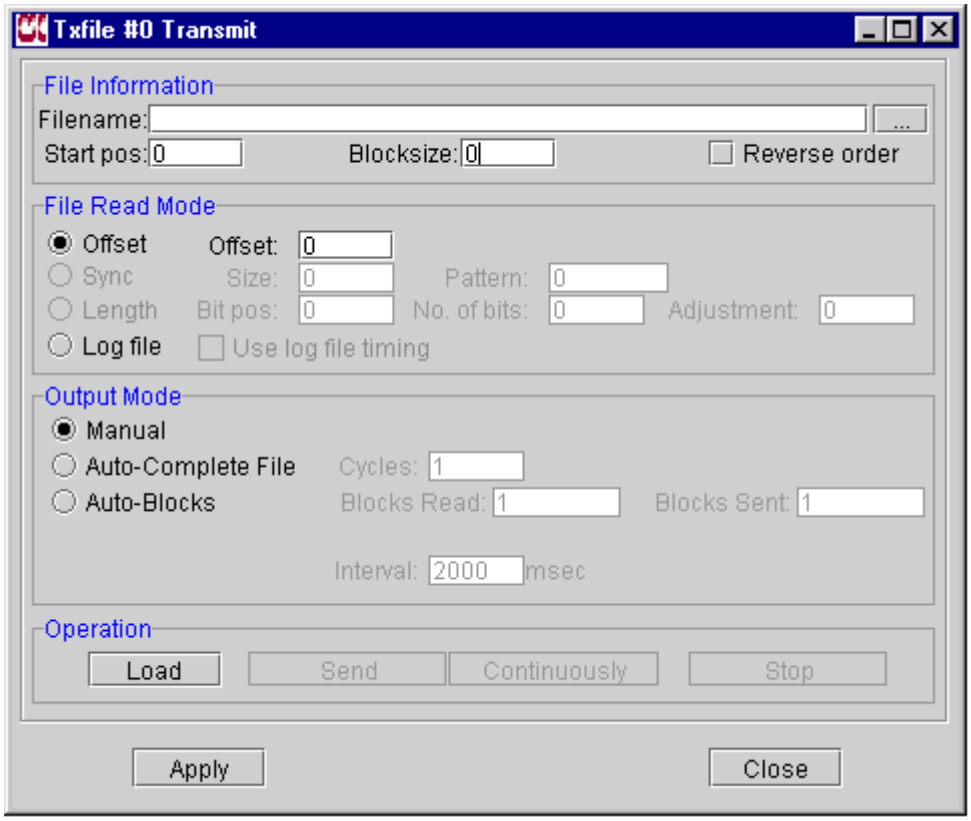
The transmit buffer display shows the contents of the currently loaded block.



The address field of the dump display may be toggled between hexadecimal and decimal display formats. The data portion may be toggled between hexadecimal and octal formats. The screen contents may be "frozen" using the **Pause** button. Paused data may be "unfrozen" with the **Cont** button.

TxFile-5.2.3 Send From File

The send from file display is the main operations display. This display allows the user to specify data files, specify read mode parameters, specify output mode parameters and control the data transmission. The following description mainly applies to the default IP transmission mode. For a description of this display's operation from serial transmission mode, see section TxFile-5.2.3.7 Example 5 Serial Transmission Mode.



This is the initial Send from File Display before any parameters have been set. The default interval (for automatic transmission) is 2000 milliseconds (2 seconds) between blocks. The default read mode is Offset. The default output mode is Manual.

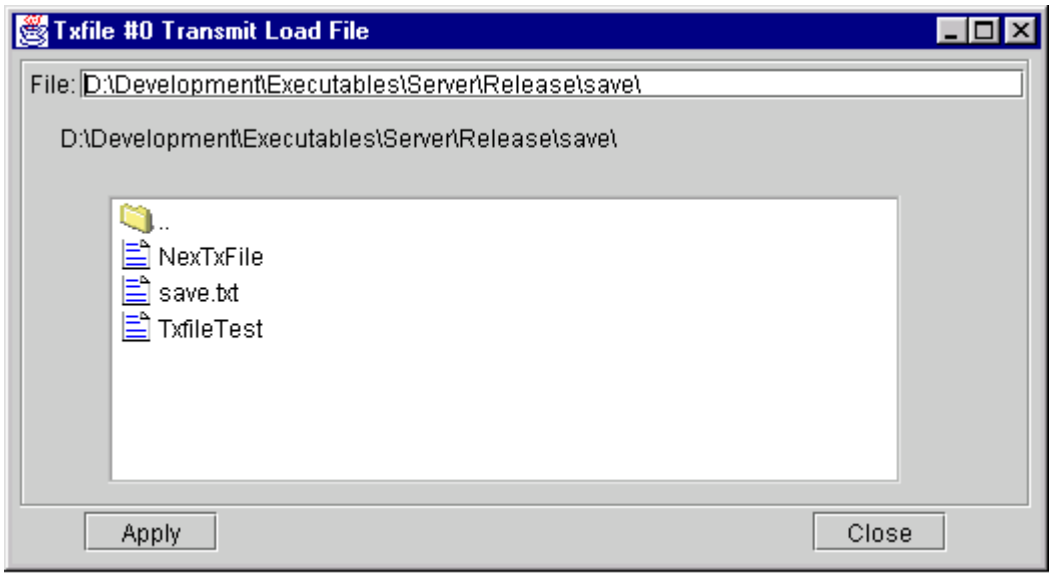
File Information	**Description**
Filename	Pathname of the file to transmit can be relative or absolute (leading or trailing white space is allowed, but not embedded spaces).
File browser ...	Click on this button to the right of the filename field to invoke file browsing to select a file.
Start pos	Starting position (in bytes) within the file of the first block to send.
Blocksize	Size of block to read (in bytes).
Reverse Order	If set, the file is transmitted in reversed order (the first bit of file is transmitted as the last bit, last bit of file is transmitted as first bit).
File Read Mode	**Description**
Offset button	Click to select Offset read mode
Offset field	Increment (in bytes) between the start of consecutive blocks. Should normally be the same as the block size. If zero, the same block would be read and sent out repeatedly.
Sync	**(Not yet implemented)** Sync mode -- Look for a sync pattern in the file as given by the bit size and the bit pattern and send out

	the number of bytes given by block size starting with and including the sync pattern.
Length	**(Not yet implemented)** Length mode -- Look for a length value in the file as given by the bit offset after the end of the previous block. Pick up the number of bits given, add the adjustment, and use that as the length of the block to send.
Log file	Specify a Log Module file with log headers
Use log file timing	Use relative timing from the log headers instead of the **Interval** to transmit data blocks
Output Mode	**Description**
Manual	(Manual mode) Data blocks are loaded and sent out one at a time under user control
Auto-Complete File	(Automatic mode) Transmit a whole file the specified number of **Cycles.**
Cycles	To transmit the file a fixed number of times, set **Cycles** to a nonzero value (default value is 1). To transmit the file an indefinite number of times, set **Cycles** to zero. **Note:** If the "Offset field" is set to zero, the end of file is never reached and the same block will be transmitted repeatedly until stopped.
Auto-Blocks	(Automatic mode) Transmit a portion of a file, indicated by setting **Blocks Read** and **Blocks Sent** (both are 1 by default).
Blocks Read	Number of blocks to read
Blocks Sent	Number of times each read block is sent out. **Note:** If this field is less than or equal to 0, the blocks read are sent out repeatedly until stopped.
Interval	Interval between block transmission (in milliseconds).
Operation	**Description**
Load	**(Manual mode only)** Loads the next block from the file.
Send	**Manual mode**: Sends the currently loaded block one time. **Automatic modes**: Starts the specified automatic transmission.
Continuously	**(Auto-Blocks mode only)** Pressing the **Send** button sends out the current block. Pressing the **Continuously** button will make the next block the current block and send it out.
Stop	**(Automatic modes only)** Stop sending blocks.
Apply	Configuration values are used only after this button is pressed**.**
Close	Close this window.

TxFile-5.2.3.1 File Browsing

When the file browser button is selected from the TxFile Transmit Display, a file selection display similar to the next screen will be displayed. The current search directory will be shown in the File field. Double click on the appropriate folder icons to navigate to the desired directory. Double click on the ".." folder to go up the directory structure. Double click on a named folder to display its contents. Double click to select a scenario file name. The file name should then appear in the File field. Use the **Apply** button to

transfer the selected file name to the control display. Use the **Close** button to dismiss the display without selecting a file. When a file is selected in this manner, its directory then becomes the current search directory for the next file browse action.



Note: The file browser cannot navigate to a disk drive other than the drive where the server software is installed.

TxFile-5.2.3.2 File Types

Any type of data file may be used with varying degrees of success. Depending on the type of file, different modes of operation should be used. Currently this module only recognizes two basic file types.

Raw data:	Any file that contains data bytes only. It contains no additional bytes preceding or following the data bytes. An example of a raw data file is any log file created by the SIMSS Log Module without the log header option. Raw data files may be received from many other sources.
Log file:	Any file created by the SIMSS Log Module with the log header option enabled. The log header bytes contain data length and log time information.

A file may have fixed or variable length data.

Fixed data:	Every data entry in the file is the same size. When the Log Module creates fixed sized files, some entries may be truncated and others may be padded with zero bytes.
Variable data:	Data entries in the file may vary in size. This is usually the most flexible and efficient way to log data.

The following table summarizes read and transmission modes for the different file types and provides a map to applicable examples in the following sections.

File Type	Data Size	Read Modes	Transmit Modes	Examples
Raw data	Fixed	Offset	Manual mode	Example 1
Raw data	Fixed	Offset	Auto-Complete File	Example 2
Raw data	Fixed	Offset	Auto-Blocks	Example 3
Raw data	Varying	None apply	None apply	None apply, see note
Log file	Fixed or Varying	Log file	Manual mode	Example 1
Log file	Fixed or Varying	Log file	Auto-Complete File	Example 2
Log file	Fixed or Varying	Log file	Use Log file timing	Example 4

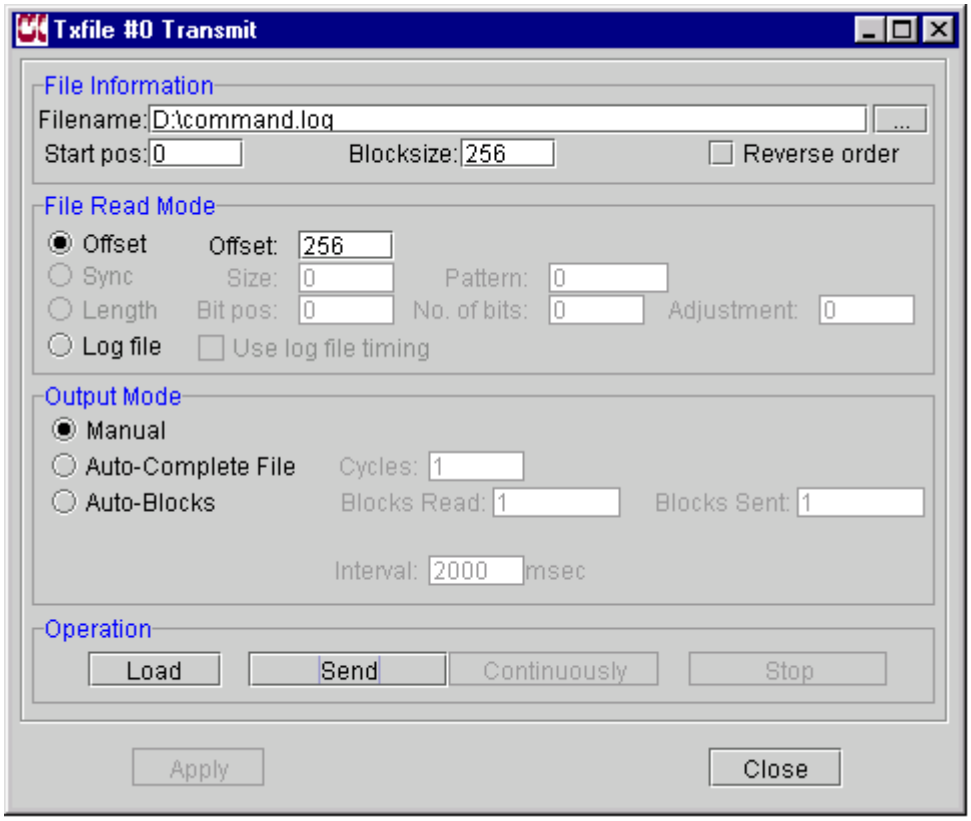
Note Since the Sync and Length read modes are not yet implemented, TxFile can not properly process raw files with varying size data. Every data byte can be read and transmitted using modes valid for fixed raw files, but the timing and grouping of the data may not be valid for the receiving system. These files should be converted to fixed data size files. When varying length data is recorded with the Log module for later TxFile playback, the log header option must be enabled.

The next five sections provide examples for using the different output modes. The steps given in each example are the minimum steps required. Review the parameter description table at the beginning of section 5.2.3 for more information.

TxFile-5.2.3.3 Example 1 Manual Mode Transmission

Step	Manual Mode Transmission Description
1	Enter the file name in the Filename field.
2	Select Manual Output Mode if it is not already on. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, click to set Offset File Read Mode and enter the fixed data size in bytes in the Blocksize and Offset fields. For log files, click to set the Log file option.
4	Click Apply button and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
5	Click Load button to read in the next entry of the file.
6	Click Send button to transmit the loaded entry one time.
	Repeat steps 5 and 6 as often as needed.

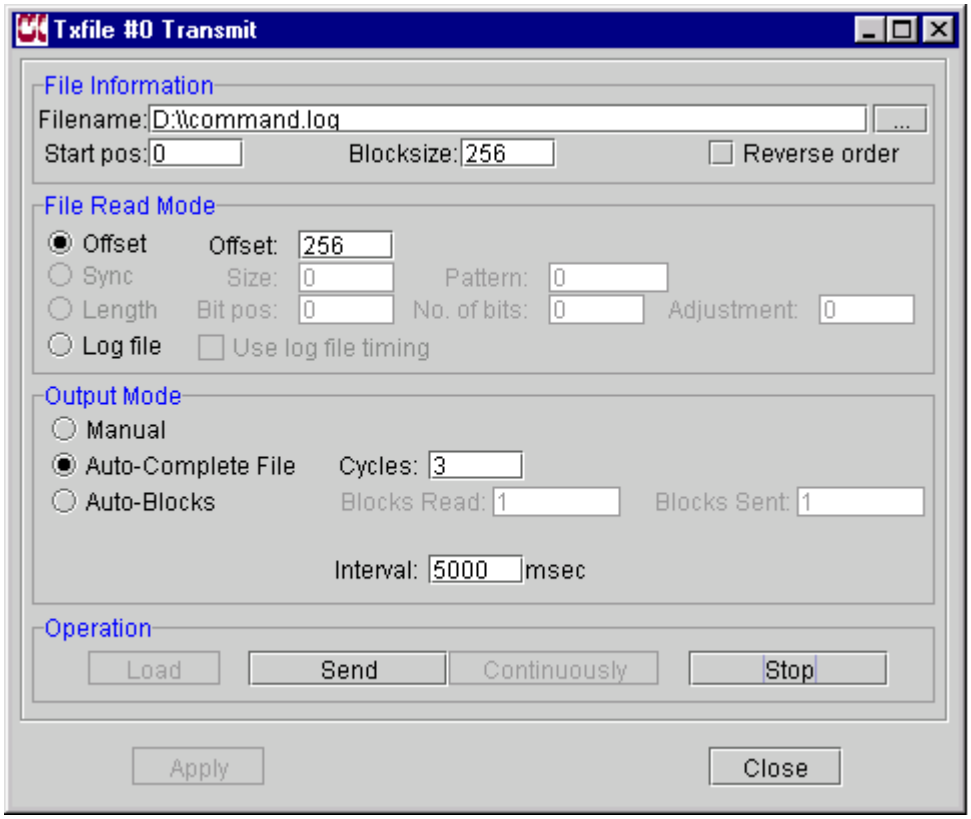
The following screen shows TxFile configured for manual output mode using a raw data file with entries of 256 bytes. Steps 1 through 4 (Apply) have been done. Steps 5 and 6 must be executed to transmit data.



TxFile-5.2.3.4 Example 2 Auto-Complete File Transmission

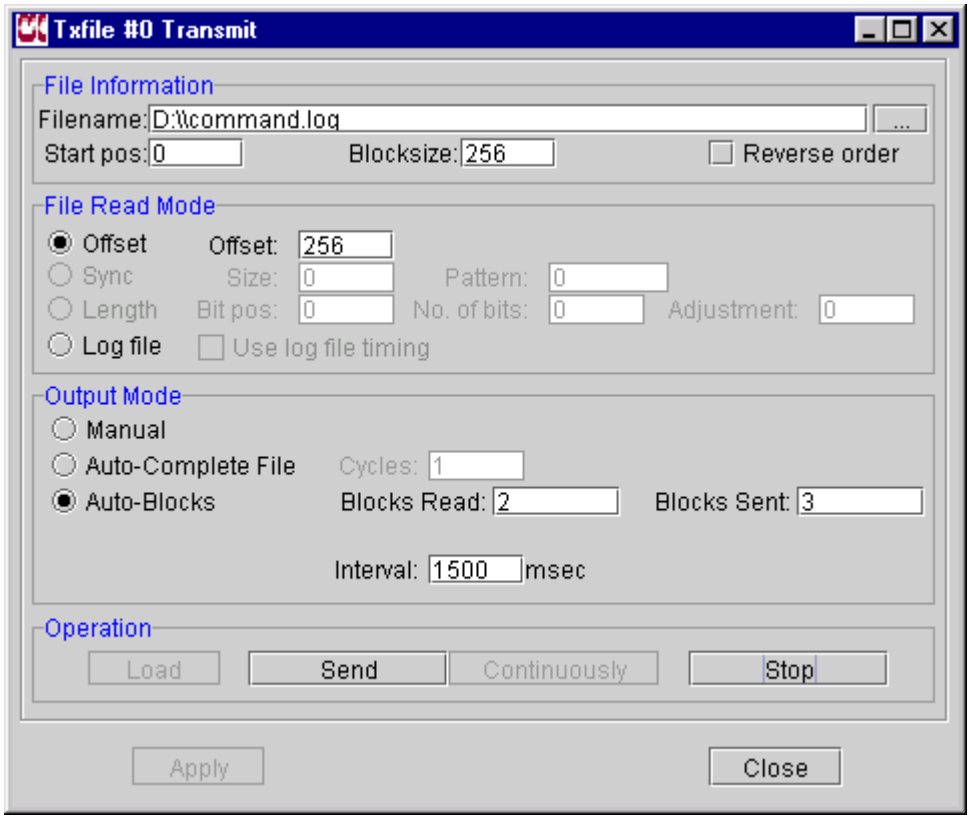
Step	Auto-Complete File Transmission Description
1	Enter the file name in the Filename field.
2	Select Auto-Complete File in Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, select Offset File Read Mode and enter the fixed size in bytes in the Blocksize and Offset fields. For log files, click to set the Log file option.
4	Enter the desired timing between blocks in the Interval field. (1000 = 1 second)
5	For raw data files only, the Cycles field may be modified to specify the number of times to transmit the file. Set Cycles to zero for an indefinite number of times.
6	Click Apply button and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
7	Click Send button to start the automatic transmission. Transmission will stop when the file has been transmitted the specified number of cycles. Transmission does not automatically stop for raw files when the Offset field or the Cycles field is set to zero. Log files are only transmitted one time.
8	Click Stop button to stop the automatic transmission at any time.

The following screen shows TxFile configured for Auto-Complete File output mode using a raw data file with entries of 256 bytes. The entire file will be transmitted 3 times with 5 seconds between blocks. Steps 1 through 6 (Apply) have been done. A single click on the Send button will start the transmission.



TxFile-5.2.3.5 Example 3 Auto-Blocks Transmission (Raw Fixed Data Files Only)

Step	Auto-Blocks Transmission (Raw Fixed Data Files Only)
1	Enter the file name in the Filename field.
2	Select Auto-Blocks in the Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	Set Offset File Read Mode if not already set and enter the fixed data size in bytes in the Blocksize and Offset fields.
4	Enter the desired timing between blocks in the Interval field. (1000 = one second)
5	Enter the number of blocks to read in the Blocks Read field.
6	Enter the number of times to transmit each block in the Blocks Sent field. If this field is set to zero, the same block will be sent until the Continuously button is used to force loading of the next block.
7	Click Apply and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
8	Click Send to start the automatic transmission. If Blocks Read specifies a finite value, the transmission will stop after the specified blocks have been sent the specified number of times.
9	If the Blocks Read field was set to zero, click on Continuously button when desired to cause loading and sending of the next block.
10	Click Stop to stop the automatic transmission at any time.

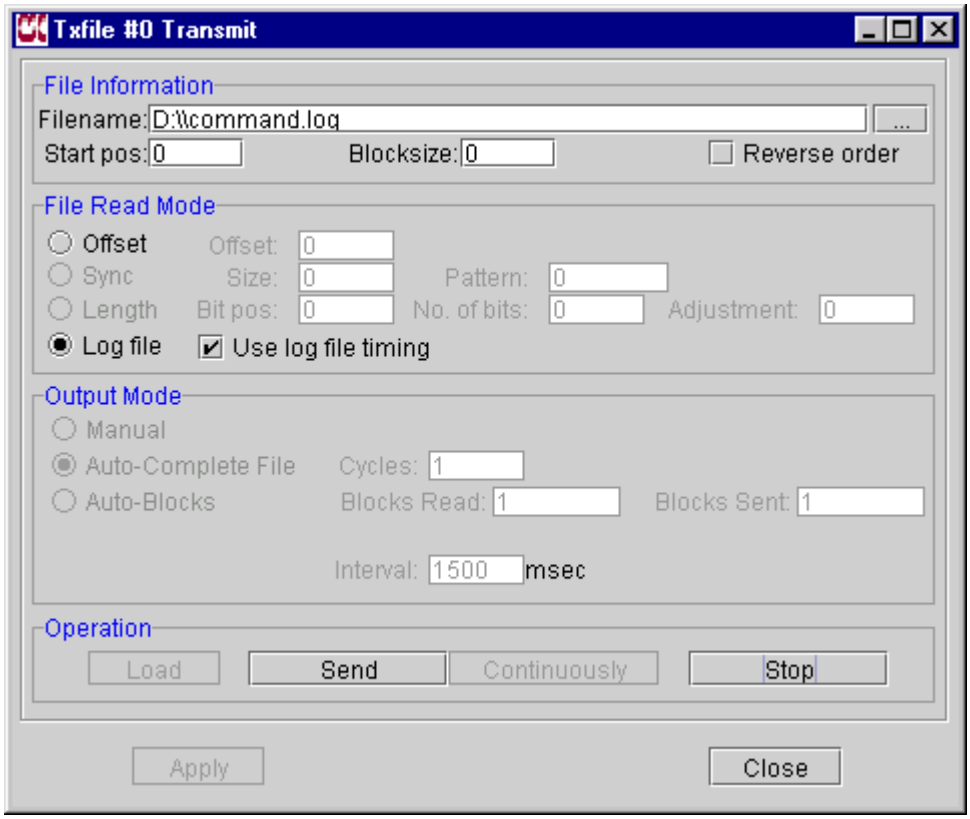


The above screen shows TxFile configured for Auto-Blocks output mode using a raw data file with entries of 256 bytes. The first 2 blocks of the file will be transmitted 3 times with 1500 milliseconds between blocks. Steps 1 through 7 (Apply) have been done. A single click on the Send button will start the transmission.

TxFile-5.2.3.6 Example 4 Use of Log File Timing (Log Files Only)

Step	Use of Log File Timing (Log Files Only)
1	Enter the log file name in the Filename field.
2	Select Auto-Complete File in Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	Click to select the Log file option.
4	Click to select the Use log file timing option. Relative timing between blocks will be calculated from the time tags in the log headers.
5	Click Apply and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
6	Click Send to start the automatic transmission of the file using log file timing. Transmission will stop when the file has been sent one time.
7	Click Stop to halt the automatic transmission before completion.

The following screen shows TxFile configured for Auto-Complete File mode using log file timing. Steps 1 through 5 (Apply) have been done. A single click on the Send button will start the transmission.



TxFile-5.2.3.7 Example 5 Serial Transmission Mode

When TxFile is linked to a module requesting serial output, the transmission mode is automatically changed to serial transmission mode. When an **Automatic** output mode is selected, the module requesting serial data will control the timing of the transmissions. The operator may specify the file to read, read parameters and the number of times to transmit the file. Specification of log file timing or an interval will be ignored. Event messages will inform the operator when the mode is changed to serial, when transmission starts, when transmission stops, and each time the end-of-file is reached.

Note: Use of **Manual** mode with serial transmit mode is the same as with IP transmit mode. Refer to the previous Manual example for instructions.

Note: Should the operator want to override the serial timing and use the operator-specified interval or log file timing options, use the following directive to force the transmit mode to stay in IP mode. Then follow any of the other examples.

Set TxFileTransmitMode 2

Step	Serial Transmission Mode
1	Enter the file name in the Filename field.
2	Select Auto-Complete File in the Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, select Offset File Read Mode and enter the fixed size in bytes

	in the Blocksize and Offset fields. For log files, click to set the Log file option.
4	Set the Cycles field to the number of times to transmit the file. Set Cycles to zero to specify an indefinite number of times.
5	Click Apply and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
6	Click Send to start the serial transmission. Transmission will automatically stop when the file has been sent the specified number of times.
7	Click Stop to halt the serial transmission before completion.
8	Optionally click Send to resume or repeat the serial transmission. If the transmission was previously stopped, it will be resumed at the halted position in the file. If the end of transmission was reached, the original request will be repeated. If the Send button is not available, click on the Stop button first.

TxFile-5.3 About Display

The "About" option requests a display with information on the numbers of input and output channels and whether the module accepts operator directives.

TxFile-6.0 Special Operating Instructions

- It is important to specify the appropriate read mode for the data file type (see TxFile-5.2.3.2 File Types). If a raw data file is read using the Log file mode, data bytes will be interpreted as log header fields and may cause the TxFile module to crash. If a log file is read using the Offset mode, the log header fields will be incorrectly transmitted as part of the data.

- When using the Auto-Complete File and Auto-Blocks Output modes, it is critical to set the Interval field properly. This field is in milliseconds. If a very short interval is used, an entire file may be transmitted multiple times in a brief period. In some configurations, the automatic transmission does not stop without operator intervention. This may overwhelm the resources of the sending system or that of the receiving system.

- There are several limitations when using Log files. When some of the following options are selected, they currently do not work and there are no error event messages.

  1. The Auto-Blocks output mode does not work with log files. (If Auto-Blocks is selected with use log file timing, TxFile processes the request as Auto-Complete File with log file timing.)
  2. When using the Auto-Complete File output mode with log files, the file is only transmitted once. It does not matter how the Cycles field is set.
  3. A starting offset (Start pos) cannot be used with log files. If specified, it will be ignored. The log file will be read from the beginning.
  4. Reverse order should not be used with log files.

  - Auto-Blocks output mode does not work in Serial Transmit mode.

VC Processor Module

VC Processor-1.0 Overview

The CCDSD Generic CCSDS Virtual Channel (VC) Processor is used to monitor and verify virtual channels on a CCSDS data stream.

VC Processor-2.0 Inputs

The VC Processor module has a single input channel.

Channel	Description
1	Accepts external data stream of CCSDS data. Data must be synchronous and contain exactly one VC packet per data burst.

VC Processor-3.0 Outputs

The VC Processor module has three separate output channels. The output for each channel is user configurable (see below for configuration instructions).

Channel	Description
1, 2, 3	Sends user-defined channels to other modules.

VC Processor-4.0 Container Items

Container Name	Description
VCCRCCheck	Enable/disable CRC checking. (0 = disable, 1 = enable)
VCRSCheck	Enable/disable RS checking. (0 = disable, 1 = enable)
Interleave	The value for interleave. This is only used if RS checking is enabled.
Virtualfill	The value for virtual fill. This is only used if RS checking is enabled.
Displyarray	The status of each channel. This updates the colored boxes in the runtime display (0 = yellow, 1 = green, 2 = red).
Container Name for each VC. *XX* represents a decimal number (00 – 63)	**Description – each container item is specific to a single VC**
NumVCXX	The number of VCDU/Transfer Frames.
VCSeqErrsXX	The number of sequence errors.
VCCRCXX	The number of CRC errors.
VCRSErr	The number of RS errors.
VCUncorrRSErrXX	The number of uncorrectable RS errors. Note: this is not yet implemented.
VCvallOffset	The telemetry value starting at the offset byte.

OffsetLocation	The offset byte where telemetry extraction will begin.

VC Processor-5.0 Displays

To access the displays for a module, click in the center of the module in the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove the module. The "Configure" option should be used prior to running the project and is unavailable at run-time. The Run-time option is available only when the project is running.

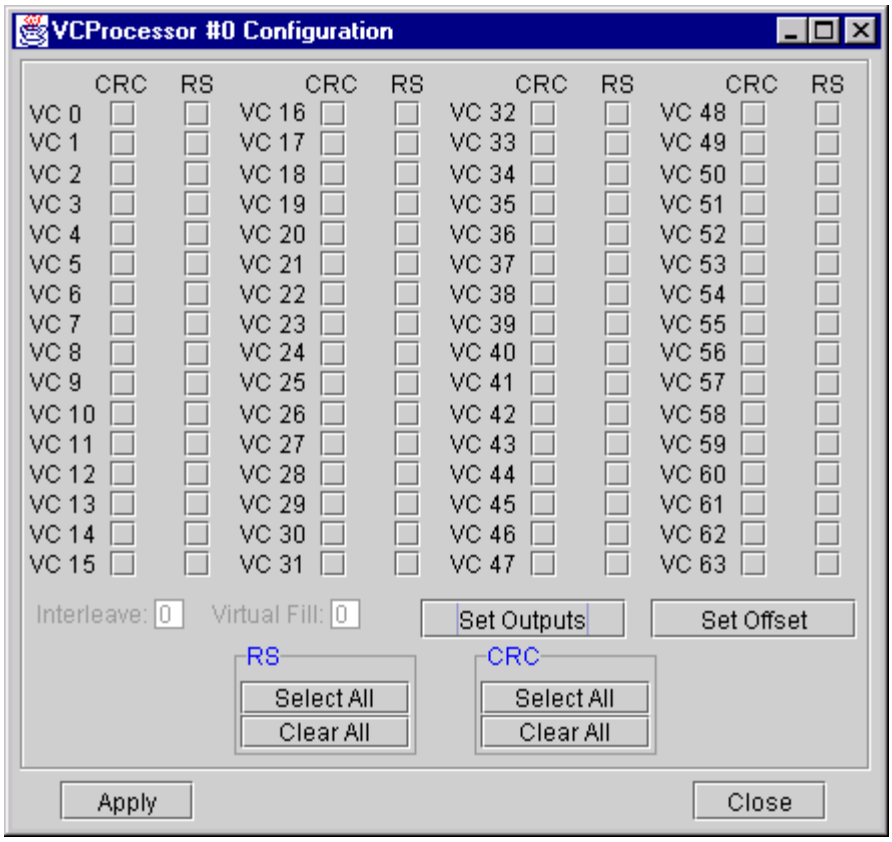
Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display generic module information

VC Processor -5.1 Configure

There are three different configuration windows: Main Configuration, Channel Selection, and Display Offset. This module can also be re-configured during runtime.

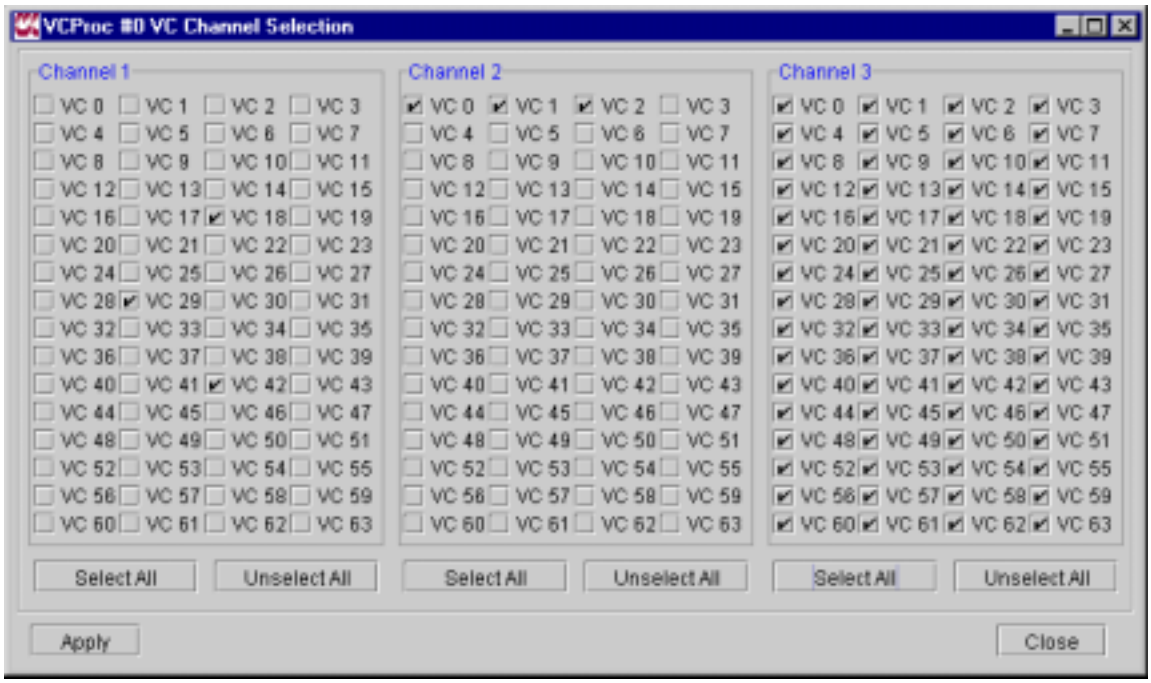
VC Processor-5.1.1 Main Configuration Window

This is the main configuration window. CRC and RS checking is enabled/disabled from this window. The two other configuration windows are accessible from this window.



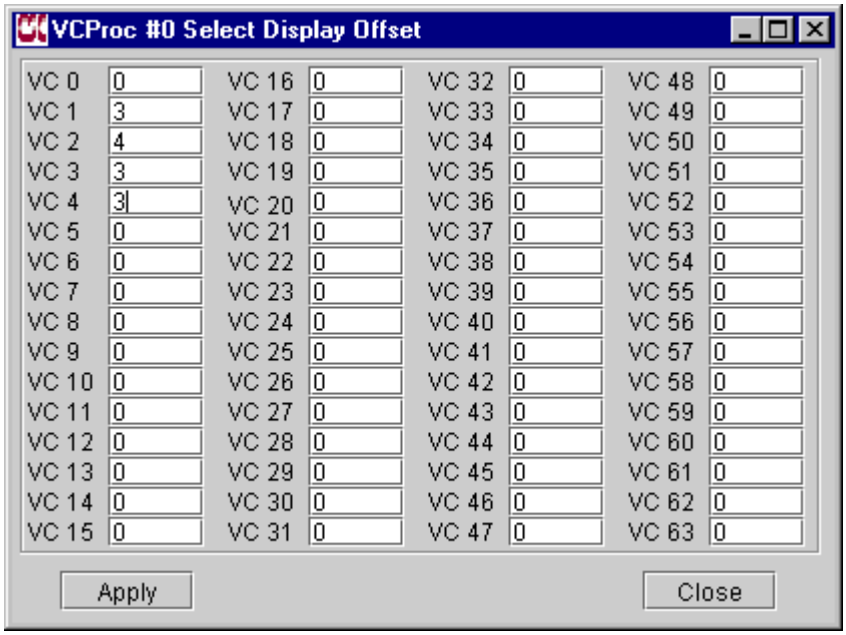
Button Name	Description
CRC	Enable/disable CRC checking for each Virtual Channel
RS	Enable/disable RS encoding checking for each Virtual Channel.
Interleave	Only accessible when doing RS check. Input is a decimal integer.
Virtual Fill	Only accessible when doing RS check. Input is a decimal integer.
Set Outputs	Call up the display to set virtual channels to transmit on each of the three output channels.
Set Offset	Call up the display to set the start byte location for display value (one per virtual channel).

VC Processor-5.1.2 Set Outputs



This window is the output channel configuration window. Any number of virtual channels can be transmitted over any of the three module output channels. The checked VCs will be transmitted on the specified channel. In the above picture, module output channel 1 will transmit VCs 18, 29, and 42.

VC Processor-5.1.3 Set Offsets



Each VC can display a 32-bit telemetry value in the runtime Status window (5.2.1). The byte offset (first byte to display) is configured in this window. The first byte in a VC is

byte zero. For example, an offset value of 3 will display bytes 3, 4, 5, and 6 in the Status window when the module is running.

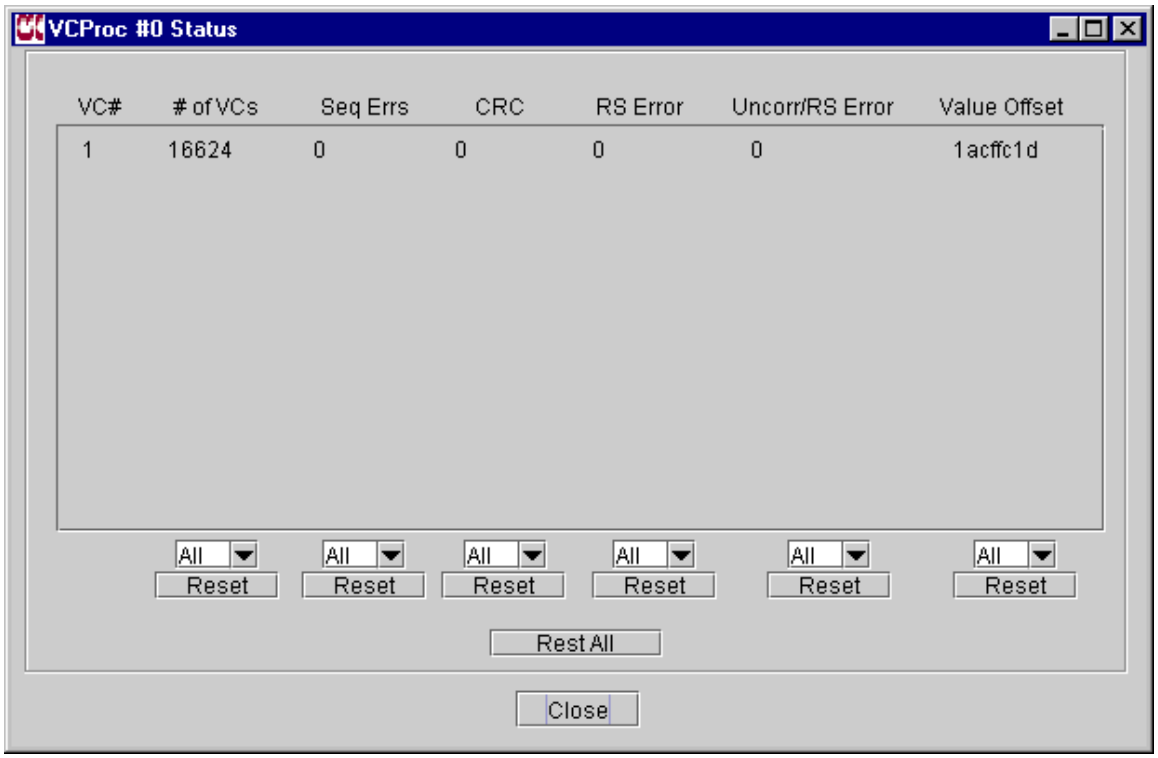
VC Processor-5.2 Run

During runtime, the VC Processor module will perform an analysis of the CCSDS data stream going through it. When the module encounters a new VC in the stream, it will inform the user through status windows and report on the status of each active VC.

The Run-time menu for the VC Processor contains the following five items.

Run-time Menu Item	Description
Show Status	Display the status of each active VC.
Display VC	A "quick" display of all VCs.
ReConfigure	Allows user to reset configuration variables.
Resume	Resumes module.
Pause	Pauses module.

VC Processor-5.2.1 Show Status

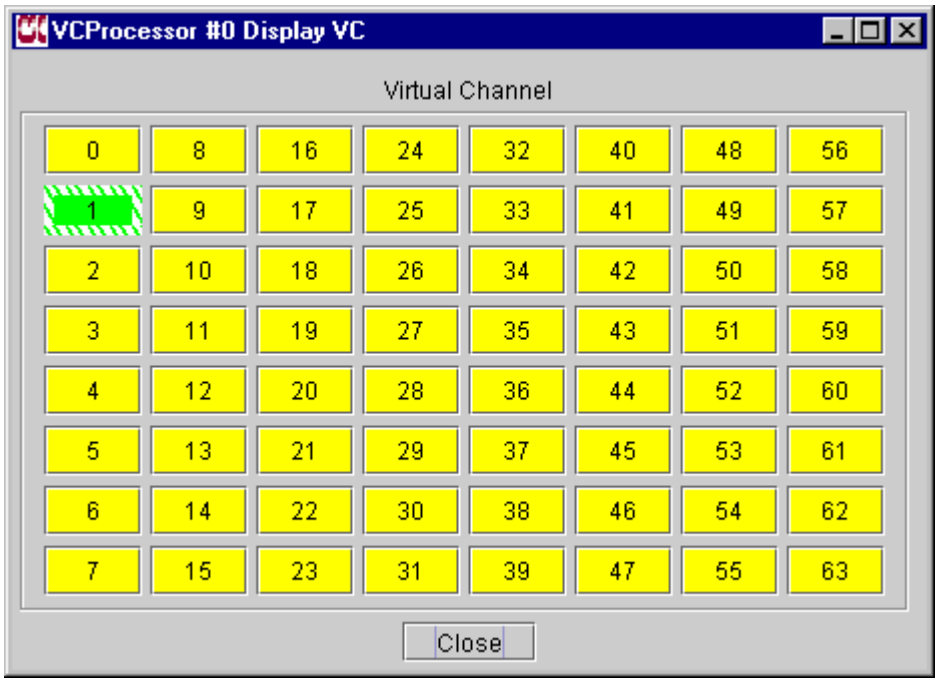


This window shows the status of each active VC.

Status Column	Description
VC#	The VC ID (0 – 63).
# of VCs	The number of VCDU/Transfer Frames received for this virtual channel.

SeqErrs	The number of sequence errors encountered for this virtual channel.
CRC	The number of CRC errors encountered for this virtual channel. This will be zero if CRC checking is not enabled.
RS Error	The number of RS errors encountered for this virtual channel. This will be zero if RS checking is not enabled
Uncorr/RS Error	This functionality is not implemented yet.
Value Offset	The 32 bit value taken from telemetry. The location to start extracting data from is found in the configuration window

VC Processor-5.2.2 Display VC



This is a "quick" display of the status of all VCs. The following color code applies:

Yellow	Inactive VC
Green	Active VC with no errors
Red	Active VC with errors

All VCs start out as inactive. When a VC is encountered in the data stream, the corresponding box turns green or red, depending on quality of data. If a box turns red, the user can click on it, turning it yellow (inactive). A VC box will not change from red to any other color unless the user clicks on it.

VC Processor-5.2.3 ReConfigure

This option allows the user to re-configure the VC Processor module during run-time. Neither the module nor the project needs to be stopped in order to reconfigure. Refer to the configuration section (VC Processor 5.1) for details.

ACRONYMS

APID	Application Identifier
BCH	Bose-Chaudhuri-Hocquenghem
CADU	Channel Access Data Unit
CCSDS	Consultative Committee on Space Data Systems
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
COP	Command Operations Procedure
COTS	Commercial, off-the-shelf
CSC	Computer Sciences Corporation
CUC	CCSDS Unsegmented Time Code
EDOS	EOS Data and Operations System
EDU	EDOS Data Unit
EGS	EOS Ground System
EMOS	EOS Mission Operations System
EOSGS	SIMSS module for the EOS Ground Station
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information Systems
EPGS	EOS Polar Ground Stations
ESH	EDOS Service Header
ETS	EOSDIS Test System
FARM	Frame Acceptance and Reporting Mechanism
FTP	File Transfer Protocol
GMH	Ground Message Header
GMT	Greenwich Mean Time
GUI	Graphical User Interface
GS	SIMSS Ground Station Module
ICD	Interface Control Document
IP	Internet Protocol
MPS	Multimode Portable Simulator (legacy simulator for AM-1)
MPS/Aura	Common name for the SIMSS/Aura Simulator
NT	New Technology
PC	Personal Computer
PDB	Project Database
SC	SIMSS Spacecraft Module
SCAura	SIMSS Spacecraft Module with Aura extensions
SIMSS	Scalable Integrated Multimission Simulation Suite
SIMSS/Aura	SIMSS-based Aura Simulator (also known as MPS/Aura)
TCP	Transmission Control Protocol
TIE	Transponder Interface Electronics
UDP	User Datagram Protocol
VCDU	Virtual Channel Data Unit
VCID	Virtual Channel Identification