Scalable, Integrated, Multimission Simulation Suite (SIMSS)

User's Guide
Release 7.0

October 2002

National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Scalable, Integrated, Multimission Simulation Suite (SIMSS)

User's Guide
Release 7.0

October 2002

Prepared Under Contract NAS9-98100
CSOC Task Order SODA GM-03
By Computer Sciences Corporation

Prepared by:

J. K. Carlson Date
Developer, SIMSS, CSC

Approved by:

E. Noone Date
CSOC Customer Service Representative, SIMSS, CSC

Reviewed by:

M. McCoy Date
Task Lead, SIMSS, CSC

F. Schlosser Date
Task Manager, SIMSS, GSFC, Code 583

Goddard Space Flight Center
Greenbelt, Maryland

Preface

This document contains the user's guide for the Scalable, Integrated, Multimission Simulation Suite (SIMSS). This user's guide, once baselined, will be controlled by the SIMSS Configuration Control Board.

This user's guide was prepared by

SIMSS Project
Code 583
Goddard Space Flight Center
Greenbelt, Maryland 20771

Contributors to this document include the following:

Tina Bilmanis
Pat Burrows
Jean K. Carlson
Chris Cohoon
Sandy Dempewolf
Dave Green
Leonid Kaminsky
John Kim
Matt McCoy
Estelle Noone
Kevin Parker
Bill Parlock
Ernest Quintin
Adrienne Siskind
Phong Tran
Paul Yeh
Jian-Rong Zhou

Abstract

About This Document

This document is the System User's Guide for the Scalable, Integrated, Multimission Simulation Suite (SIMSS) developed for the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC). The simulator is one of the deliverables of the Next Generation Simulator and Test Tool Development Task under the CSOC contract. The System User's Guide provides an overview of operational concepts and procedures for this simulator. The Table of Contents is arranged in an alphabetical order of module names. This document will be maintained by Computer Sciences Corporation (CSC) under the CSOC contract and will be updated for every SIMSS release. For any questions or recommendations for enhancement regarding the System User's Guide, please contact Matt McCoy (mmccoy@csc.com).

Who Should Use This Document

This document is for anyone who wishes to use the SIMSS. To use this simulator effectively you should already be familiar with the basics of the Windows NT Operating System as well as the basic spacecraft command and telemetry data formats.

Keywords: Scalable, Integrated, Multimission Simulation Suite (SIMSS); simulators; telemetry; commanding; spacecraft

Table of Contents

How to Run a SIMSS-Based Simulator

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

SIMSS-1.0 Terminology

SIMSS runs on one or more Windows NT 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 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

To install the SIMSS client, first run jdk1_2_2-win.exe located in the root of the delivery CD. After JDK1.2.2 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.
properties	Contains client configuration files.
save	This directory is where client information is stored when projects are saved. The user may want to delete old files from this directory. (Files with the same names should also be deleted from the Server save 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 four additional directories: elog, 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.
properties	Contains the server property.txt configuration file.
save	This directory is where server information is stored when projects are saved. The user may want to delete old files from this directory. (Files with the same names should also be deleted from the Client save directory)
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).
ConcurrentScenario	This flag applies to scenario modules in the project. The scenario mode may be set to 0 for serial operation or 1 for concurrent operation. It is normally set to 1. This parameter applies only to scenario files that are not under direct operator GUI control, such as scenario files started by other scenario files. If too many scenarios run concurrently, there could be system resource issues.

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.

Paragraph SIMSS-4.1 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.

SIMSS-4.1 Application Startup Sequence

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". Or, a "Restore From" selection can 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 7.
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.

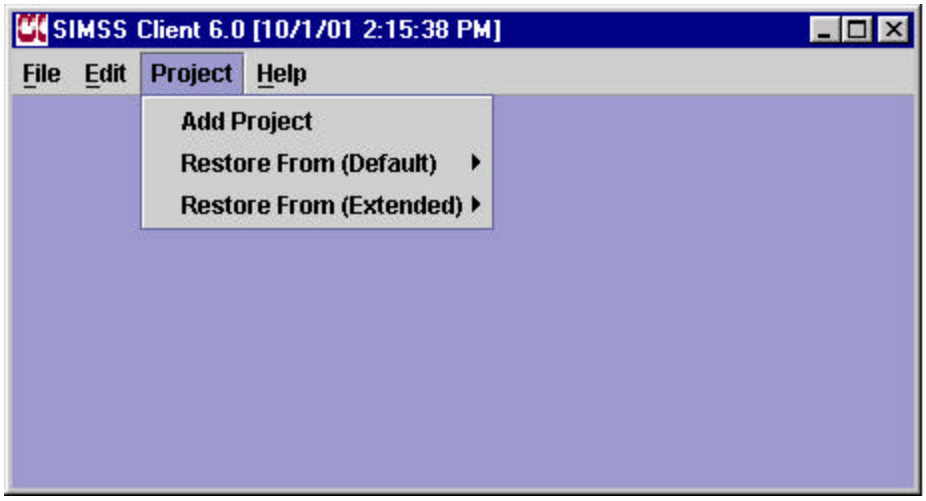
6) Create Links Note: Refer to the module specific chapters for input and output channel data descriptions in order to link the output channels to the appropriate input channels!	In the project window, click the Module menu and select "Create Links". Click on the sending module border then move the cursor to the receiving module border and click again. If either module has multiple links, a popup window will appear for specification of the exact links. Repeat this step to create all desired links. 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. 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) Lock the Project	In the project window, click the Run menu and "Lock".
9) Save the Project (optional)	Click the project window's System menu and select a "Save Project" option. The Default option saves to the local PC. The Extended option allows navigation and saving to a networked PC. Specify a name for the project save file and apply it from the save or browse window.
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

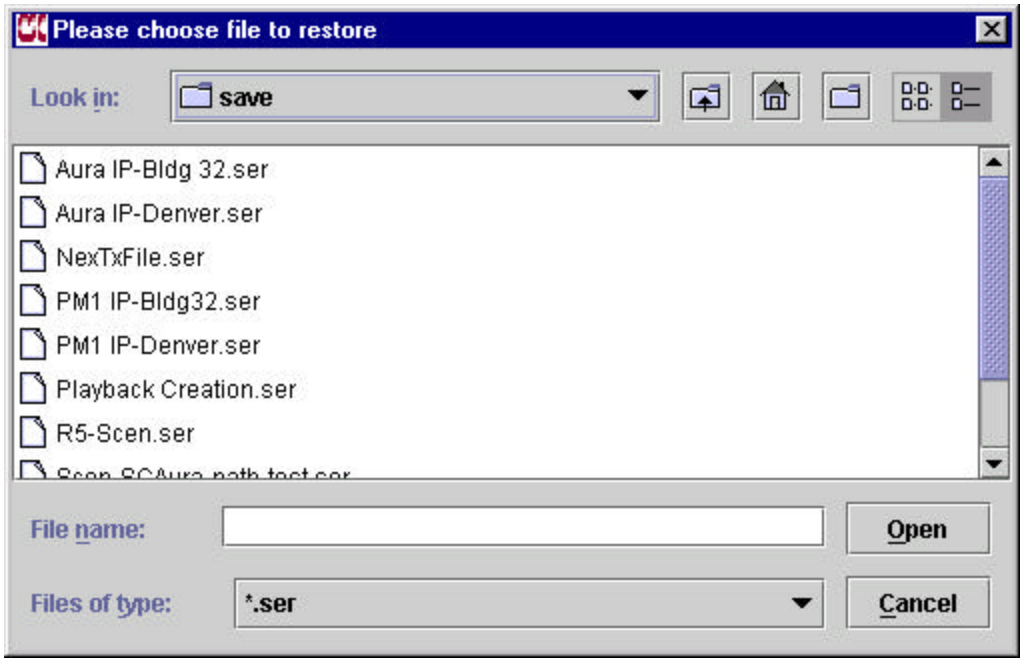
The Project menu contains three selections, "Add Project", "Restore From (Default)", "Restore From (Extended)".

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 From (Default)

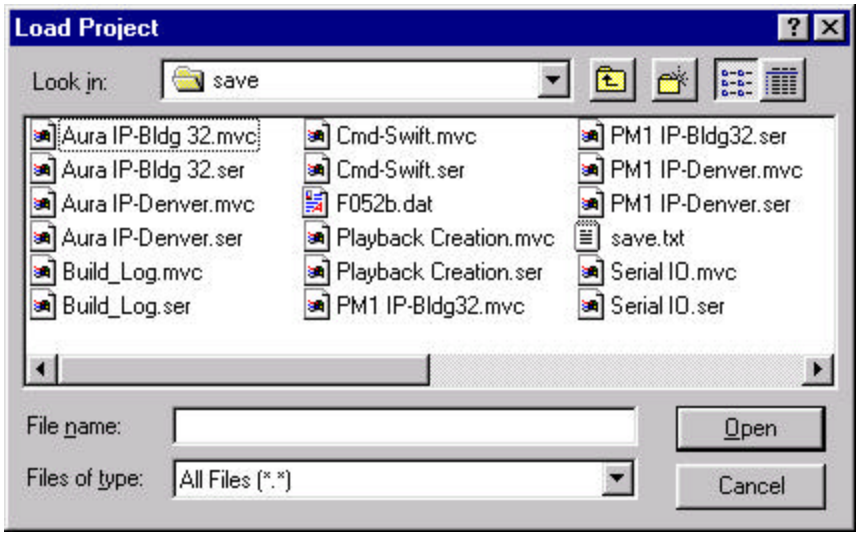
To restore an existing project from the local PC, select the “Restore From (Default)” item from the Project menu. A file selection screen like the following will appear.



The system will search the save directory by default but another directory may be selected by clicking the Save Folder button. Select a file and click the **Open** button. Click on the **Cancel** button to dismiss the display without restoring a project.

SIMSS-5.1.3.3 Restore From (Extended)

To restore an existing project from another PC within the same network, select the "Restore From Extended" item from the Project menu. A file selection screen like the following will appear. This file selection window allows navigation to directories within the network neighborhood.

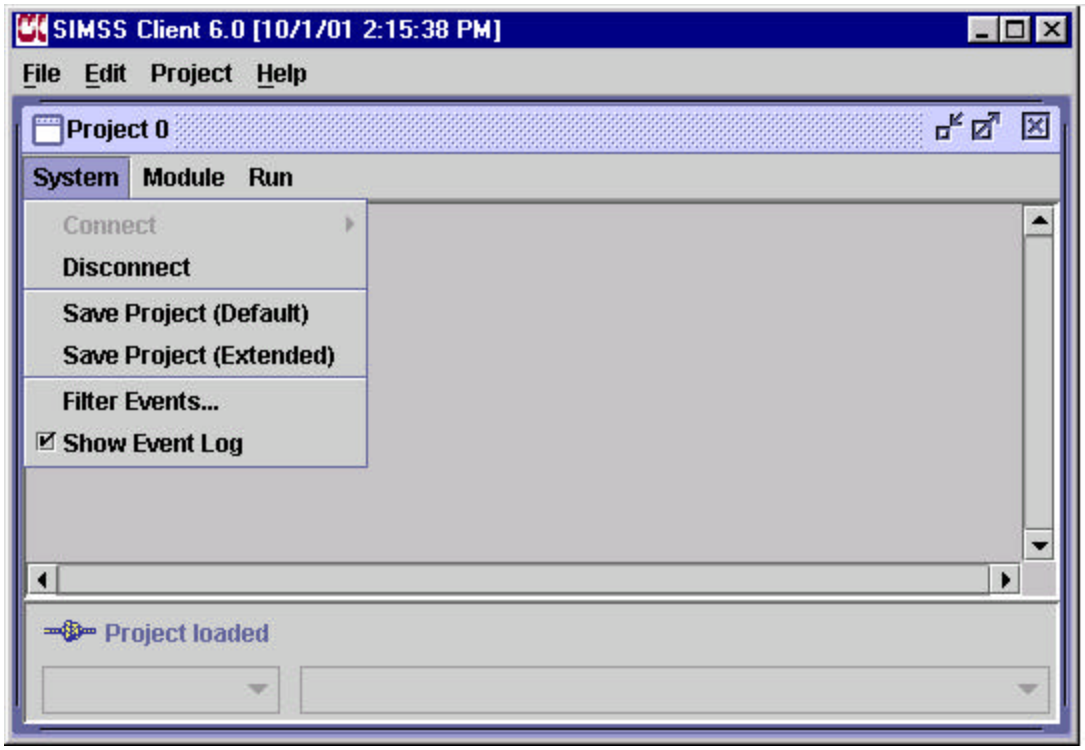


SIMSS-5.1.4 Help Menu

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. At the bottom right is the directive region of the screen.

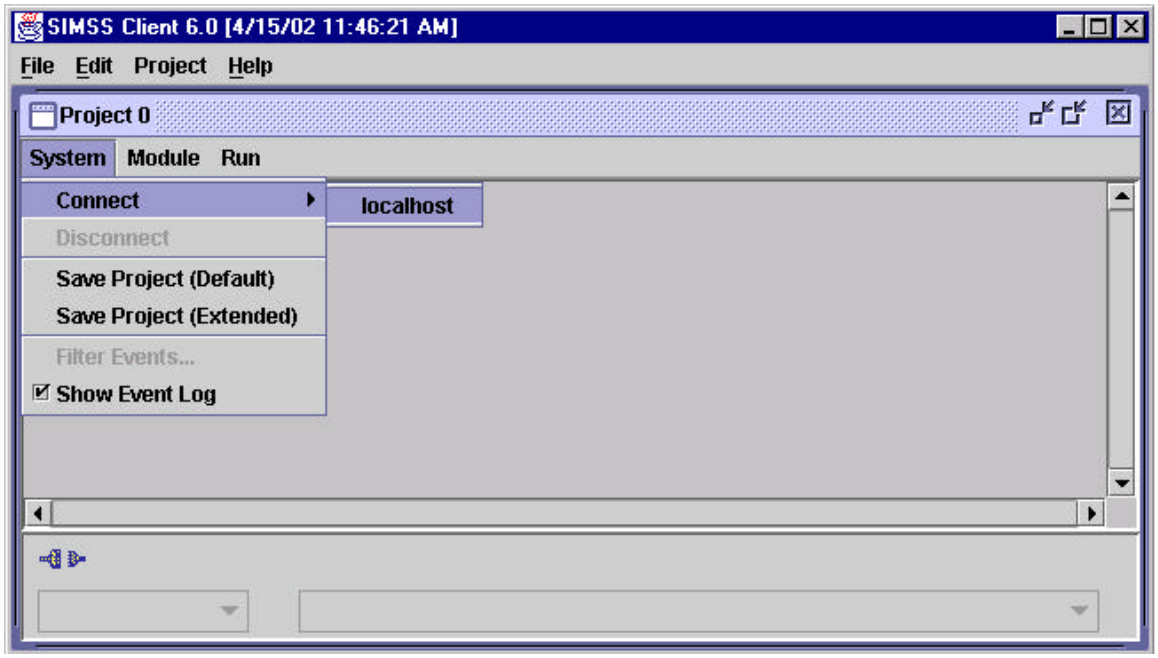
SIMSS-5.2.1 Project System Menu

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 (Default)	Saves project design in file on local PC
Save Project (Extended)	Saves project design to file in PC network
Filter Events…	Change event message filtering
Show Event Log	When this option is enabled, the project event message log window is displayed.

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.



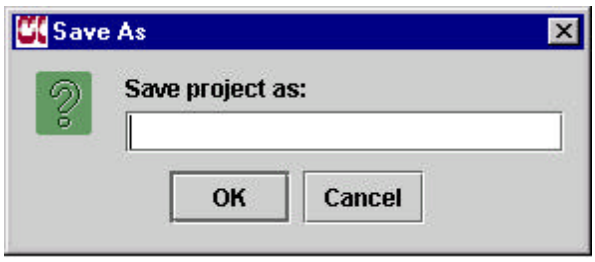
After the client has been connected to a server, the "Disconnect" option will be enabled on the System menu and the "Connect" option will be disabled.

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 (Default)

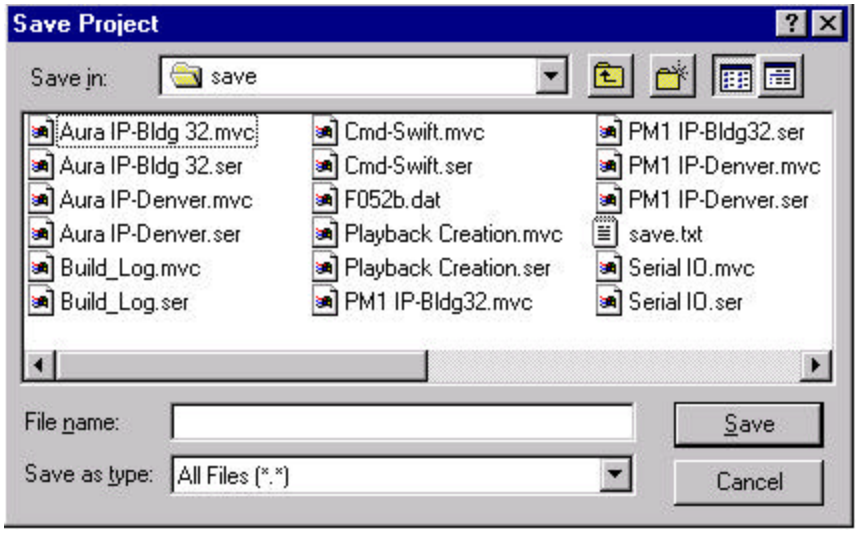
This option allows the user to store a created and configured project to a file on the local PC. The following screen will appear.



Enter the file name for storage of the current project and click the **OK** button to have project information written into it. The file name will have ".ser" appended to the end. Clicking **Cancel** will close this window without performing a save.

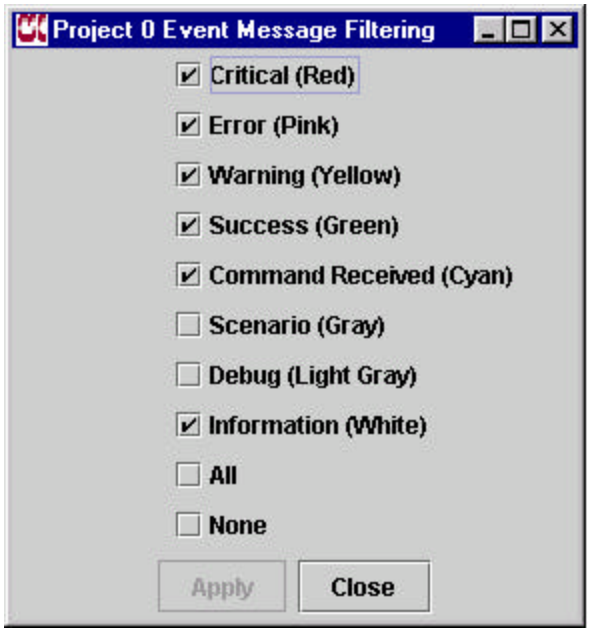
SIMSS-5.2.1.4 Save Project (Extended)

To save a project on another PC within the same network, select the "Save Project (Extended)" option from the Project System menu. A file selection screen like the following will appear. This file selection window allows navigation to directories within the network neighborhood.



SIMSS-5.2.1.5 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-5.2.1.6 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.

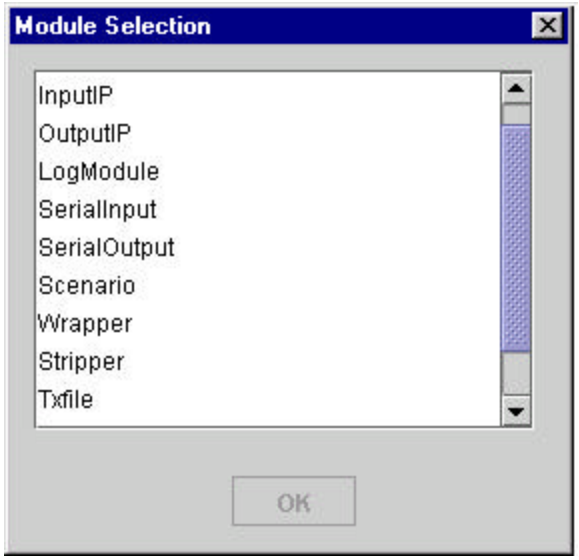
SIMSS-5.2.2 Project Module Menu

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 to select a module and then click the **OK** button.

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 cyan border while in Design mode.

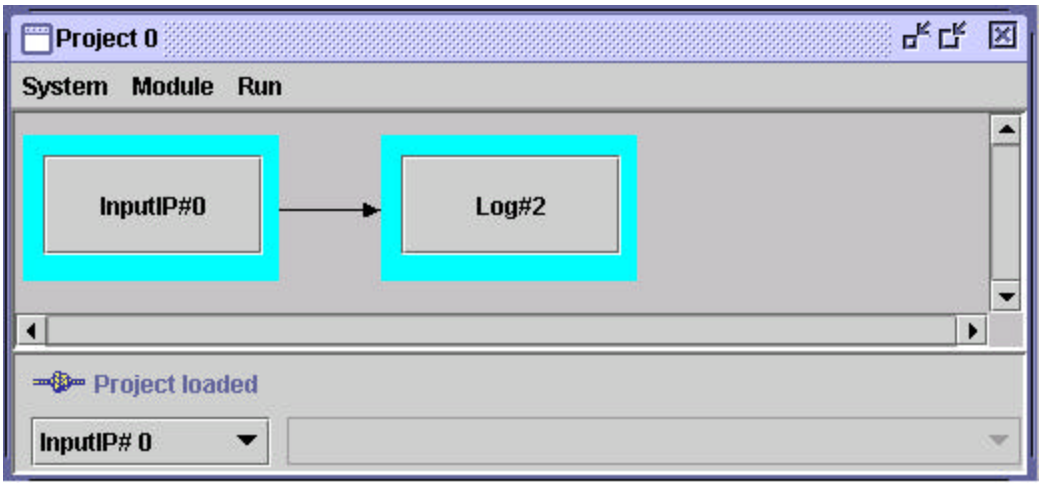
If a module is added that accepts operator directives, the directive region of the project window becomes accessible.

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 can have 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.

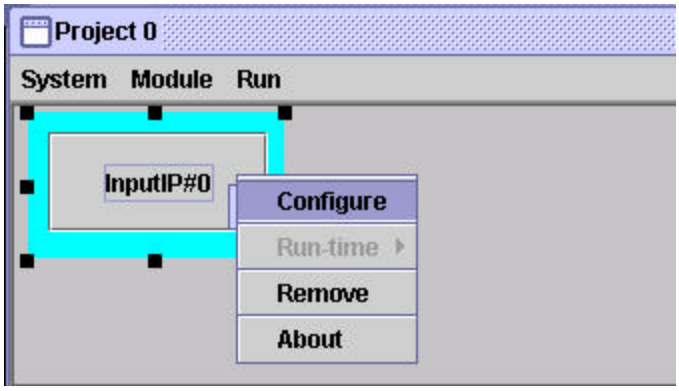
SIMSS-5.2.2.3 Design Mode

After the "Create Links" option has been used, 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 items of the design diagram 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.

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.
About	Provides information about the module.



The actions initiated by the "Configure" and "Run-time" options are different for every module and are described in the other chapters of this system user's guide that are dedicated to specific modules.

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 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 have been created in part 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 (^). 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.
- Arguments to the Boolean functions must be in decimal.
- The Boolean functions perform a bit-wise AND, OR, etc.

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)

Function Description	Set directive example
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)
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
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"

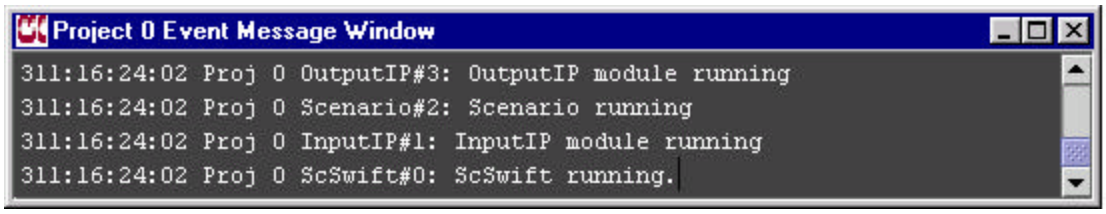
will inform the operator. If unsuccessful, verify that the module indicator is correct, and that the buffer name spelling is correct.

SIMSS-5.3 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 also 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.



ASCII to Binary (ascii2bin) Module

ASCII2BIN-1.0 Overview

The ascii2bin module is used to convert an ASCII text database file into multiple binary files to be used by the SIMSS CCSDS Generic Telemetry (GenTlm) module. To use this module select an input file and output base file name and lock-run the project. If there are no errors a message will appear in event log saying that conversion is complete. If there are errors a message will appear in the event log saying there were errors. In the event of errors, consult the log file for details of each error.

ASCII2BIN-2.0 Input ASCII Database File

The input file must be an ASCII text file. This file must be made up of records (defined below). The fields of a record are deliminated by the pipe character '|'. The record indicators (first field of each record) are not case sensitive. If there is no data for a field, leave the field empty (do not include any white space).

The white space between the last pipe character of a record and the first character of the next record is discarded. This allows the user to set up a file with new line characters after each record. While a new line is not required, it makes editing and viewing the file much easier.

The telemetry record types are:

'WholeVCDU'	defines the WholeVCDU flag
'CrossPackets'	defines the CrossPackets flag
'VERSION'	defines the version
'SCID'	defines the spacecraft identification (SCID) number
'SIZE'	defines the size
'VCDU'	defines a Virtual Channel Data Unit (VCDU)
'PACKET'	defines a packet
'SKEY'	defines a secondary key for a packet
'MNEMONIC'	defines a telemetry mnemonic
'LOCATION'	defines a packet location for a telemetry mnemonic
'COMMENT'	Defines a comment field

ASCII2BIN-2.1 WholeVCDU Record

Wholevcdu|<FLAG>|

Field	Name	Description
1	Record type	Field must contain 'WholeVCDU'.
2	Flag	This is a flag for determining the rate of VCDUs. If this field is set to 'true', the SIMSS TlmGen module will wait for the VCDU to be filled with packet data before transmitting the VCDU (VCDU rate is dependent on packet rate). If this flag is set to

		'false', the VCDU rate will be .25. This field must be entered as 'true' or 'false'.

ASCII2BIN-2.2 CrossPackets Record

CrossPackets|<FLAG>|

Field	Name	Description
1	Record type	Field must contain 'CrossPackets'.
2	Flag	This is a flag for determining if packets are allowed to be split over VCDUs. If true, the packets are allowed to be split. If false, only whole packets are inserted. This field must be entered as 'true' or 'false'.

ASCII2BIN-2.3 VERSION record

VERSION|<VERSION>|

Field	Name	Description
1	Record type	Field must contain 'VERSION'.
2	Version	Version of data (1=VCDU, 0=Transfer Frame). This utility currently only supports VCDU, so the only valid value is 1.

ASCII2BIN-2.4 SCID Record

SCID|<SCID>|

Field	Name	Description
1	Record type	Field must contain 'SCID'.
2	SCID	This field is the spacecraft identification number. This must be a decimal integer between 0 and 255 (inclusive).

ASCII2BIN-2.5 SIZE Record

SIZE|<SIZE>|

Field	Name	Description
1	Record type	Field must contain 'SIZE'.
2	SIZE	This field specifies the byte size of a VCDU/Frame. This must be a positive decimal integer. This size does not include the sync value (4 bytes) or Reed-Solomon encoding size (varies). This size does include the primary header (6 bytes for VCDU) and MPDU header (2 bytes).

ASCII2BIN-2.6 VCDU Record

The VCDU record defines a VCDU. Values for the VCDU primary header, and error controls are set using this record.

VCDU|<VCID>|<CHANNEL>|<REPLAY>|<HEADER ERROR CONTROL> |<INSERT SIZE>|<OP CONTROL>|<ERROR CONTROL>|<REED-SOLOMON SIZE>|<COMMAND CHANNEL>|

Field	Name	Description
1	Record type	Field must be set to 'VCDU'.
2	VCID	This is the telemetry virtual channel identification number. This entry should be a decimal integer between 0 and 63 (inclusive).
3	Channel	This field specifies the physical output channel from the SIMSS GenTlm module. This should be a decimal integer between 0 and 2 (inclusive). If this field is empty the channel defaults to 0.
4	Replay	This field determines the value of the replay flag in telemetry. This should be 'true' or 'false'. A value of 'true' means this is replay telemetry. A value of 'false' means this is realtime telemetry. If this field is empty, the field defaults to false (realtime).
5	Header Error Control	This field specifies whether VCDU Header Error Control is present in the VCDU Primary Header or not. This entry should be 'true' or 'false'. If this field is empty, the default value is false (no VCDU Header Error Control).
6	Insert size	This field specifies the number of bytes allocated for the insert zone. This should be a decimal integer. If no insert zone is present, use '0' or leave this field empty.
7	Op Control	This field should be set to 'true' if the Operational Control Field is present in the VCDU trailer. It should be 'false' if the field is not present. When this field is empty, the default value is false (no Operational Control Field).
8	Error Control	This field should be set to 'true' if the VCDU Error Control Field is present in the VCDU trailer. It should be 'false' if the field is not present. If this field is empty, the default value is true (VCDU Error Control Field in trailer).
9	Reed-Solomon size	This field specifies the size in bytes of the Reed-Solomon (RS) encoding. This should be a decimal integer. If no RS encoding is present, enter '0'. If this field is empty, the default value is 0 (no RS encoding).
10	Command channel	This field specifies whether the VCDU will contain a command link control word (CLCW) and which command virtual channel the CLCW will be reporting. Valid values for this field are 0 to 63 or blank if no CLCW will be present. If a CLCW is desired, the value in this field will designate which command virtual channel (and its CLCW) will be associated with and reported by the telemetry virtual channel id in this VCDU record. When this field is empty, the VCDU will not contain a CLCW.

ASCII2BIN-2.7 PACKET Record

The PACKET record defines a telemetry packet. Each VCID used in a PACKET record must have a corresponding VCDU record. There can be multiple packets per VCDU.

PACKET|<APID>|<VCID>|<SECONDARY HDRTYPE>|<SIZE>|<INTERVAL>|<OFFSET>|

Field	Name	Description
1	Record type	Field must be set to 'PACKET'.
2	APID	This field is the packet application identification number. This must be a decimal integer between 0 and 2047 (inclusive).
3	VCID	This field specifies the virtual channel that this packet will be placed in. The virtual channel named here must be described with a VCDU record.
4	Secondary header type	This field specifies whether a secondary header will be inserted in telemetry packets for this APID. The only valid values are 0 (no secondary headers) and 1 (secondary headers). Currently, the only valid secondary header format supported is an 8 byte CUC (1 byte pfield, 1 byte pfield extension, 4 bytes coarse time, 2 bytes fine time). The pfield defaults to 0X9E but may be modified via "set" directive using the mnemonic, "TlmCUCpfield". The pfield extension defaults to 0 but may be modified via "set" directive using the mnemonic, "TlmCUCpfieldextension". The coarse time field uses an epoch of 1/1/58. The fine time field has a range of 0 – 65K, each unit equal to 15.528 microseconds.
5	Size	This field specifies size of the packet data in bytes. This does not include the packet primary header (six bytes). This should be a positive decimal integer.
6	Interval	This field specifies the interval in seconds that the packet will be transmitted. This must be a non-negative decimal floating point. The format of this field is a floating point number using only digits (can not have 4.0E01
7	Offset	This field specifies the time period in seconds that will pass before the first packet is sent out. This must be a non-negative decimal floating point. The format of this field is a floating point number using only digits (can not have 5.0E01).

ASCII2BIN-2.8 SKEY Record

Note: The SKEY Record is not currently implemented by the GenTlm module.

The SKEY record is used to define a secondary key. Each APID in an SKEY record must be previously defined by a PACKET record. There can be multiple skeys for a single packet. The secondary key is used to map multiple groups (formats) of telemetry points into a given application packet.

SKEY|<APID>|<SKEY>|<OFFSET>|<SIZE>|

Field	Name	Description
1	Record type	Field must be set to ‘SKEY’.
2	APID	This field identifies the application packet that contains the secondary key. The APID referenced here must have been previously declared within a PACKET record.
3	SKEY	This field specifies the SKEY value.
4	OFFSET	This field specifies the secondary key's byte offset into the packet.
5	SIZE	This field specifies the secondary key's size in bytes.

ASCII2BIN-2.9 MNEMONIC Record

The MNEMONIC record defines a telemetry data item. Each mnemonic name (field 2) must be unique.

MNEMONIC|<NAME>|<SIZE>|<TYPE>|<DESCRIPTION>|

Field	Name	Description
1	Record type	Field must be set to ‘MNEMONIC’.
2	NAME	This field specifies the unique mnemonic name of a telemetry data item. This field must be a string of at most 32 characters.
3	SIZE	This field specifies the bit size of the telemetry item in decimal.
4	TYPE	This field specifies the type of the telemetry item. This field is a string of up to 4 characters. This field is not required, and defaults to an empty string. The GenTlm module does not currently use this field.
5	Description	This field contains a description of the telemetry item. This field is limited to 63 characters. This field is not required and defaults to an empty string.

ASCII2BIN-2.10 LOCATION Record

The LOCATION record defines a location for a telemetry item within a packet. Each telemetry name (field 2) in the LOCATION record must be previously defined by a MNEMONIC record. There can be multiple locations for a single telemetry item in the same packet. A telemetry item may also have locations in multiple packets.

LOCATION|<NAME>|<APID>|<SKEY>|<OFFSET>|

Field	Name	Description
1	Record type	Field must be set to ‘LOCATION’.
2	Name	The mnemonic name of a telemetry item. The name given here must be previously declared with a MNEMONIC record.
3	APID	This field identifies the application packet where this telemetry value will be stored. The APID given here must be declared with a PACKET record.
4	SKEY	If the packet identified in field 3 uses secondary keys, this field specifies the secondary key for this location. This must be an decimal integer between 0 and 63 (inclusive). If this field is used,

		the skey must be declared using a SKEY record. If the packet does not use secondary keys, this field should be set to –1. If this field is empty, the default value is -1 (no secondary key).
5	OFFSET	This field is the bit offset of the telemetry item's location within the packet's data zone. An offset of zero indicates that the telemetry item starts in the first bit following the packet's primary header. This entry must be a non-negative decimal integer.

ASCII2BIN-2.11 COMMENT Record

The COMMENT record defines a comment. All comments are written to the log file, allowing easier debugging of any problems in the input file.

COMMENT|<COMMENT|

Field	Name	Description
1	Record type	Field must be set to 'COMMENT'.
2	Comment	A comment. This will be written to the log file.

ASCII2BIN-3.0 Output Binary Database Files

This utility will create six output files based on the name given by the user. The file names will be:

<NAME>
<NAME>.frame
<NAME>.locs
<NAME>.pts
<NAME>.pkts
<NAME>.log

where <NAME> is the binary output file name entered by the user. These files must be kept together in the same directory. When selecting a database from the GenTlm module, use the file name that does not have extensions (the first one in the list).

If the binary files do not contain the desired information, delete them, edit the ASCII text file and convert it again using the ascii2bin utility.

NOTE: DO NOT EDIT THE BINARY FILES UNDER ANY CIRCUMSTANCES.

ASCII2BIN-4.0 Container Items

There are no container items available to the user.

ASCII2BIN-5.0 Displays

To access 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

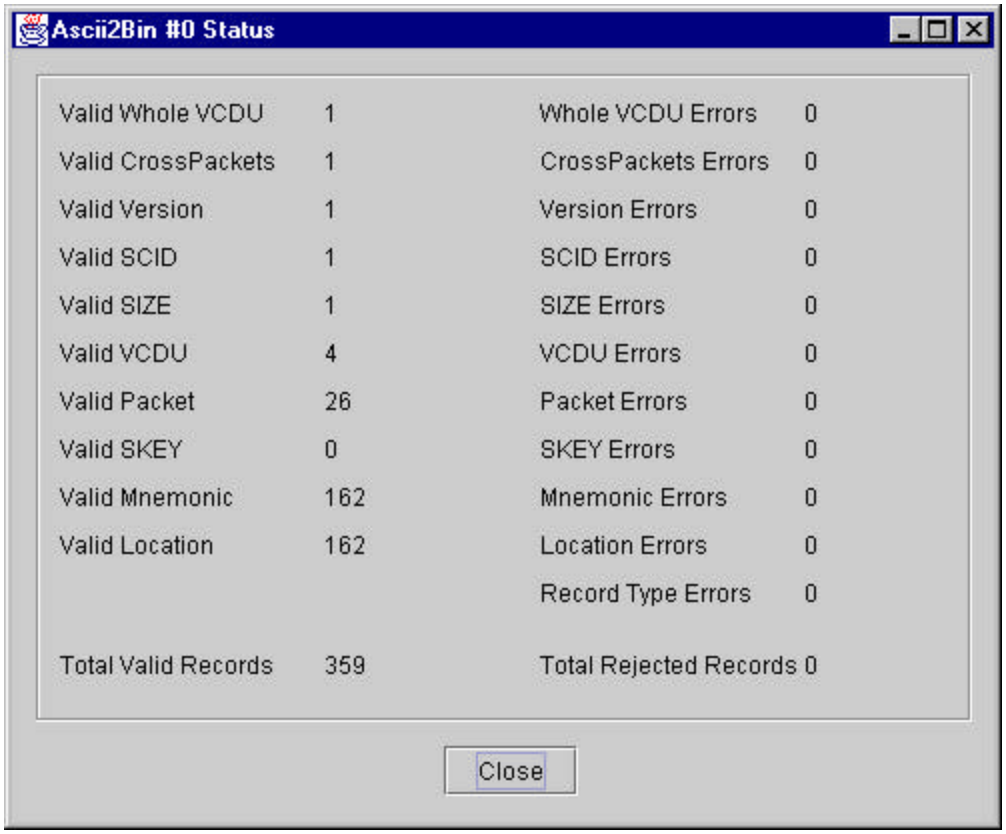
ASCII2BIN-5.1 Configuration

Two things must be configured before running the Ascii2Bin module. The user must select an input file and an output file base name.

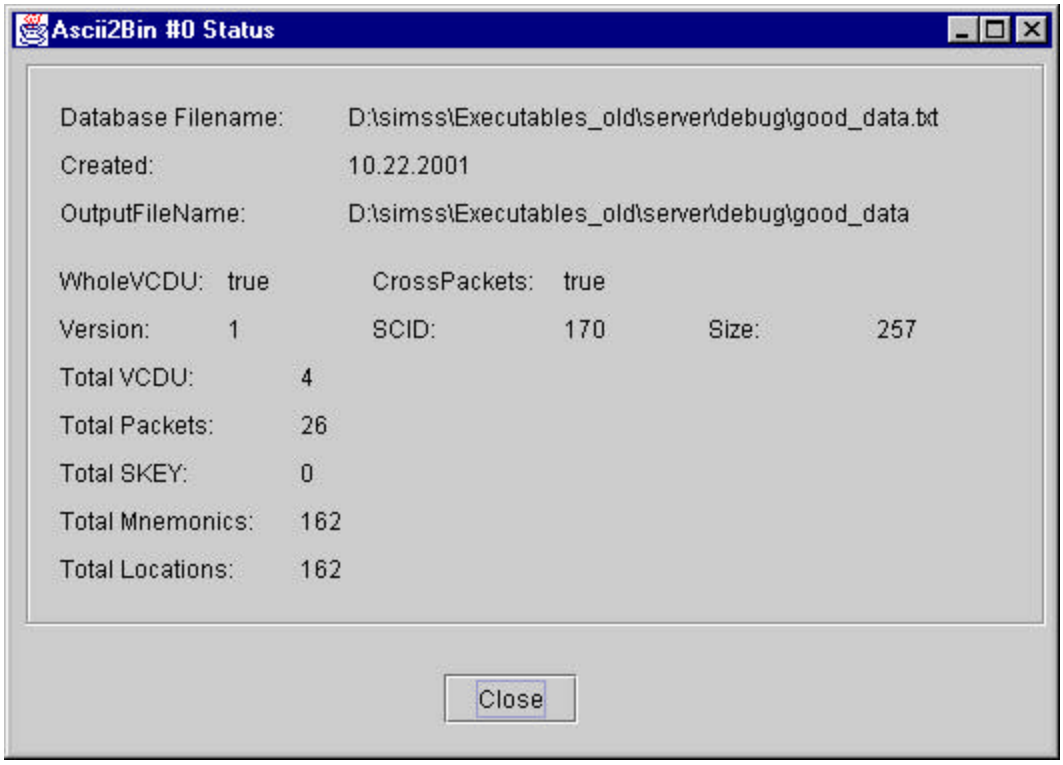
Remember that the name given for the output file name is just the base name. Six output files will be made using this base name (see ASCII2BIN-3.0 for more details on the output files).

ASCII2BIN-5.2 Runtime

There are two status displays windows. The Record Status is a summary of the types of records encountered in the input file. This is a syntax counter and does not take into account any semantic errors (duplicate records, Mnemonics with no locations, etc.).



The Database Summary is a summary of the database. This display reflects the database that is written to the output files.



ASCII2BIN-6.0 Sample ASCII Database Files

ASCII2BIN-6.1 Quickstart Example

In order to configure some basic telemetry streams from the GenTlm module, it is not necessary to define all 10 types of database records in the ASCII input file. In the following example, three packets are defined for three VCDUs on three separate physical channels. No skey, mnemonic or location records are defined and several fields in the VCDU records are empty in order to use the default values.

```
wholevcdu|true|
crosspackets|true|
version|1|
scid|168|
size|252|
vcdu|0|0|||||||0|
vcdu|1|1|||||||1|
vcdu|2|2|||||||16|
packet|1|0||248|0.125|0.0|
packet|2|1||248|0.125|0.0|
packet|3|2||248|0.125|0.0|
```

ASCII2BIN-6.2 Complete Example

In order to provide higher fidelity telemetry from the GenTlm module, the user may wish to define telemetry mnemonics and their locations within packets. Values of database-defined mnemonics may then be modified in realtime via scenario files with the changed values appearing in the database-defined locations in the telemetry streams.

The following example defines 163 mnemonics which are located within 27 packets within 6 VCDUs mapped to 3 physical channels. The *blue italicized words* are comments and should not appear in the ASCII file.

```
wholevcdu|true|
crosspackets|true|
version|1|
scid|168|
size|252|
```
Set up physical channel 0 to contain VCID 1 with APIDs 1 and 2. APID 1 has 36 (8 bit) telemetry mnemonics. APID 2 has 18 (16 bit) telemetry mnemonics.
Define the VCDU with VCID 1 on physical channel 0
```
vcdu|1|0||||||||
```
Define packet 1 on VCID 1
```
packet|1|1||288|5|0|
```
Define the telemetry mnemonics for packet 1.
```
mnemonic|p1_00|8||description of p1_00|
mnemonic|p1_01|8||description of p1_01|
mnemonic|p1_02|8||description of p1_02|
mnemonic|p1_03|8||description of p1_03|
mnemonic|p1_04|8||description of p1_04|
mnemonic|p1_05|8||description of p1_05|
mnemonic|p1_06|8||description of p1_06|
mnemonic|p1_07|8||description of p1_07|
mnemonic|p1_08|8||description of p1_08|
mnemonic|p1_09|8||description of p1_09|
mnemonic|p1_10|8||description of p1_10|
mnemonic|p1_11|8||description of p1_11|
mnemonic|p1_12|8||description of p1_12|
mnemonic|p1_13|8||description of p1_13|
mnemonic|p1_14|8||description of p1_14|
mnemonic|p1_15|8||description of p1_15|
mnemonic|p1_16|8||description of p1_16|
mnemonic|p1_17|8||description of p1_17|
mnemonic|p1_18|8||description of p1_18|
mnemonic|p1_19|8||description of p1_19|
mnemonic|p1_20|8||description of p1_20|
mnemonic|p1_21|8||description of p1_21|
mnemonic|p1_22|8||description of p1_22|
mnemonic|p1_23|8||description of p1_23|
mnemonic|p1_24|8||description of p1_24|
```

```
mnemonic|p1_25|8||description of p1_25|
mnemonic|p1_26|8||description of p1_26|
mnemonic|p1_27|8||description of p1_27|
mnemonic|p1_28|8||description of p1_28|
mnemonic|p1_29|8||description of p1_29|
mnemonic|p1_30|8||description of p1_30|
mnemonic|p1_31|8||description of p1_31|
mnemonic|p1_32|8||description of p1_32|
mnemonic|p1_33|8||description of p1_33|
mnemonic|p1_34|8||description of p1_34|
mnemonic|p1_35|8||description of p1_35|
```

Define packet 2.

```
packet|2|1||288|5|2|
```

Define the telemetry mnemonics for packet 2.

```
mnemonic|p2_00|16||description of p2_36|
mnemonic|p2_01|16||description of p2_37|
mnemonic|p2_02|16||description of p2_38|
mnemonic|p2_03|16||description of p2_39|
mnemonic|p2_04|16||description of p2_40|
mnemonic|p2_05|16||description of p2_41|
mnemonic|p2_06|16||description of p2_42|
mnemonic|p2_07|16||description of p2_43|
mnemonic|p2_08|16||description of p2_44|
mnemonic|p2_09|16||description of p2_45|
mnemonic|p2_10|16||description of p2_46|
mnemonic|p2_11|16||description of p2_47|
mnemonic|p2_12|16||description of p2_48|
mnemonic|p2_13|16||description of p2_49|
mnemonic|p2_14|16||description of p2_50|
mnemonic|p2_15|16||description of p2_51|
mnemonic|p2_16|16||description of p2_52|
mnemonic|p2_17|16||description of p2_53|
```

Define the locations for the telemetry mnemonics in packet 1.

```
location|p1_00|1||0|
location|p1_01|1||8|
location|p1_02|1||16|
location|p1_03|1||24|
location|p1_04|1||32|
location|p1_05|1||40|
location|p1_06|1||48|
location|p1_07|1||56|
location|p1_08|1||64|
location|p1_09|1||72|
location|p1_10|1||80|
location|p1_11|1||88|
location|p1_12|1||96|
location|p1_13|1||104|
```

```
location|p1_14|1||112|
location|p1_15|1||120|
location|p1_16|1||128|
location|p1_17|1||136|
location|p1_18|1||144|
location|p1_19|1||152|
location|p1_20|1||160|
location|p1_21|1||168|
location|p1_22|1||176|
location|p1_23|1||184|
location|p1_24|1||192|
location|p1_25|1||200|
location|p1_26|1||208|
location|p1_27|1||216|
location|p1_28|1||224|
location|p1_29|1||232|
location|p1_30|1||240|
location|p1_31|1||248|
location|p1_32|1||256|
location|p1_33|1||264|
location|p1_34|1||272|
location|p1_35|1||280|
```

Define the locations for the telemetry mnemonics in packet 2.

```
location|p2_00|2||0|
location|p2_01|2||16|
location|p2_02|2||32|
location|p2_03|2||48|
location|p2_04|2||64|
location|p2_05|2||80|
location|p2_06|2||96|
location|p2_07|2||112|
location|p2_08|2||128|
location|p2_09|2||144|
location|p2_10|2||160|
location|p2_11|2||176|
location|p2_12|2||192|
location|p2_13|2||208|
location|p2_14|2||224|
location|p2_15|2||240|
location|p2_16|2||256|
location|p2_17|2||272|
```

Set up physical channel 1 to contain VCID 2 and 3. VCID 2 contains packets 21 and 22. VCID 3 contains packets 31 and 32. APIDs 21 and 31 have 36 (8-bit) telemetry mnemonics. APIDs 22 and 32 have 18 (16-bit) telemetry mnemonics.

Define VCDU 2.

```
vcdu|2|1|||||||
```

Define packet 21.

```
packet|21|2||288|1|.5|
```
Define the telemetry mnemonics in packet 21.
```
mnemonic|p21_00|8||description of p21_54|
mnemonic|p21_01|8||description of p21_55|
mnemonic|p21_02|8||description of p21_56|
mnemonic|p21_03|8||description of p21_57|
mnemonic|p21_04|8||description of p21_58|
mnemonic|p21_05|8||description of p21_59|
mnemonic|p21_06|8||description of p21_60|
mnemonic|p21_07|8||description of p21_61|
mnemonic|p21_08|8||description of p21_62|
mnemonic|p21_09|8||description of p21_63|
mnemonic|p21_10|8||description of p21_64|
mnemonic|p21_11|8||description of p21_65|
mnemonic|p21_12|8||description of p21_66|
mnemonic|p21_13|8||description of p21_67|
mnemonic|p21_14|8||description of p21_68|
mnemonic|p21_15|8||description of p21_69|
mnemonic|p21_16|8||description of p21_70|
mnemonic|p21_17|8||description of p21_71|
mnemonic|p21_18|8||description of p21_72|
mnemonic|p21_19|8||description of p21_73|
mnemonic|p21_20|8||description of p21_74|
mnemonic|p21_21|8||description of p21_75|
mnemonic|p21_22|8||description of p21_76|
mnemonic|p21_23|8||description of p21_77|
mnemonic|p21_24|8||description of p21_78|
mnemonic|p21_25|8||description of p21_79|
mnemonic|p21_26|8||description of p21_80|
mnemonic|p21_27|8||description of p21_81|
mnemonic|p21_28|8||description of p21_82|
mnemonic|p21_29|8||description of p21_83|
mnemonic|p21_30|8||description of p21_84|
mnemonic|p21_31|8||description of p21_85|
mnemonic|p21_32|8||description of p21_86|
mnemonic|p21_33|8||description of p21_87|
mnemonic|p21_34|8||description of p21_88|
mnemonic|p21_35|8||description of p21_89|
```
Define packet 22.
```
packet|22|2||288|1|.5|
```
Define the telemetry mnemonics in packet 22.
```
mnemonic|p22_00|16||description of p22_90|
mnemonic|p22_01|16||description of p22_91|
mnemonic|p22_02|16||description of p22_92|
mnemonic|p22_03|16||description of p22_93|
mnemonic|p22_04|16||description of p22_94|
mnemonic|p22_05|16||description of p22_95|
```

```
mnemonic|p22_06|16||description of p22_96|
mnemonic|p22_07|16||description of p22_97|
mnemonic|p22_08|16||description of p22_98|
mnemonic|p22_09|16||description of p22_99|
mnemonic|p22_10|16||description of p22_100|
mnemonic|p22_11|16||description of p22_101|
mnemonic|p22_12|16||description of p22_102|
mnemonic|p22_13|16||description of p22_103|
mnemonic|p22_14|16||description of p22_104|
mnemonic|p22_15|16||description of p22_105|
mnemonic|p22_16|16||description of p22_106|
mnemonic|p22_17|16||description of p22_107|
```

Define the locations of the telemetry mnemonics in packet 21.

```
location|p21_00|21||0|
location|p21_01|21||8|
location|p21_02|21||16|
location|p21_03|21||24|
location|p21_04|21||32|
location|p21_05|21||40|
location|p21_06|21||48|
location|p21_07|21||56|
location|p21_08|21||64|
location|p21_09|21||72|
location|p21_10|21||80|
location|p21_11|21||88|
location|p21_12|21||96|
location|p21_13|21||104|
location|p21_14|21||112|
location|p21_15|21||120|
location|p21_16|21||128|
location|p21_17|21||136|
location|p21_18|21||144|
location|p21_19|21||152|
location|p21_20|21||160|
location|p21_21|21||168|
location|p21_22|21||176|
location|p21_23|21||184|
location|p21_24|21||192|
location|p21_25|21||200|
location|p21_26|21||208|
location|p21_27|21||216|
location|p21_28|21||224|
location|p21_29|21||232|
location|p21_30|21||240|
location|p21_31|21||248|
location|p21_32|21||256|
location|p21_33|21||264|
```

```
location|p21_34|21||272|
location|p21_35|21||280|
```
Define the location of the telemetry mnemonics in packet 22.
```
location|p22_00|22||0|
location|p22_01|22||16|
location|p22_02|22||32|
location|p22_03|22||48|
location|p22_04|22||64|
location|p22_05|22||80|
location|p22_06|22||96|
location|p22_07|22||112|
location|p22_08|22||128|
location|p22_09|22||144|
location|p22_10|22||160|
location|p22_11|22||176|
location|p22_12|22||192|
location|p22_13|22||208|
location|p22_14|22||224|
location|p22_15|22||240|
location|p22_16|22||256|
location|p22_17|22||272|
```
Define VCDU 3.
```
vcdu|3|1||||||||
```
Define packet 31.
```
packet|31|2||288|1|.5|
```
Define the mnemonics in packet 31.
```
mnemonic|p31_00|8||description of p31_108|
mnemonic|p31_01|8||description of p31_109|
mnemonic|p31_02|8||description of p31_110|
mnemonic|p31_03|8||description of p31_111|
mnemonic|p31_04|8||description of p31_112|
mnemonic|p31_05|8||description of p31_113|
mnemonic|p31_06|8||description of p31_114|
mnemonic|p31_07|8||description of p31_115|
mnemonic|p31_08|8||description of p31_116|
mnemonic|p31_09|8||description of p31_117|
mnemonic|p31_10|8||description of p31_118|
mnemonic|p31_11|8||description of p31_119|
mnemonic|p31_12|8||description of p31_120|
mnemonic|p31_13|8||description of p31_121|
mnemonic|p31_14|8||description of p31_122|
mnemonic|p31_15|8||description of p31_123|
mnemonic|p31_16|8||description of p31_124|
mnemonic|p31_17|8||description of p31_125|
mnemonic|p31_18|8||description of p31_126|
mnemonic|p31_19|8||description of p31_127|
mnemonic|p31_20|8||description of p31_128|
```

```
mnemonic|p31_21|8||description of p31_129|
mnemonic|p31_22|8||description of p31_130|
mnemonic|p31_23|8||description of p31_131|
mnemonic|p31_24|8||description of p31_132|
mnemonic|p31_25|8||description of p31_133|
mnemonic|p31_26|8||description of p31_134|
mnemonic|p31_27|8||description of p31_135|
mnemonic|p31_28|8||description of p31_136|
mnemonic|p31_29|8||description of p31_137|
mnemonic|p31_30|8||description of p31_138|
mnemonic|p31_31|8||description of p31_139|
mnemonic|p31_32|8||description of p31_140|
mnemonic|p31_33|8||description of p31_141|
mnemonic|p31_34|8||description of p31_142|
mnemonic|p31_35|8||description of p31_143|
```
Define packet 32.
```
packet|32|2||288|1|.5|
```
Define the telemetry mnemonics in packet 32.
```
mnemonic|p32_00|16||description of p32_144|
mnemonic|p32_01|16||description of p32_145|
mnemonic|p32_02|16||description of p32_146|
mnemonic|p32_03|16||description of p32_147|
mnemonic|p32_04|16||description of p32_148|
mnemonic|p32_05|16||description of p32_149|
mnemonic|p32_06|16||description of p32_150|
mnemonic|p32_07|16||description of p32_151|
mnemonic|p32_08|16||description of p32_152|
mnemonic|p32_09|16||description of p32_153|
mnemonic|p32_10|16||description of p32_154|
mnemonic|p32_11|16||description of p32_155|
mnemonic|p32_12|16||description of p32_156|
mnemonic|p32_13|16||description of p32_157|
mnemonic|p32_14|16||description of p32_158|
mnemonic|p32_15|16||description of p32_159|
mnemonic|p32_16|16||description of p32_160|
mnemonic|p32_17|16||description of p32_161|
```
Define location of the telemetry mnemonics in packet 31.
```
location|p31_00|31||0|
location|p31_01|31||8|
location|p31_02|31||16|
location|p31_03|31||24|
location|p31_04|31||32|
location|p31_05|31||40|
location|p31_06|31||48|
location|p31_07|31||56|
location|p31_08|31||64|
location|p31_09|31||72|
```

```
location|p31_10|31||80|
location|p31_11|31||88|
location|p31_12|31||96|
location|p31_13|31||104|
location|p31_14|31||112|
location|p31_15|31||120|
location|p31_16|31||128|
location|p31_17|31||136|
location|p31_18|31||144|
location|p31_19|31||152|
location|p31_20|31||160|
location|p31_21|31||168|
location|p31_22|31||176|
location|p31_23|31||184|
location|p31_24|31||192|
location|p31_25|31||200|
location|p31_26|31||208|
location|p31_27|31||216|
location|p31_28|31||224|
location|p31_29|31||232|
location|p31_30|31||240|
location|p31_31|31||248|
location|p31_32|31||256|
location|p31_33|31||264|
location|p31_34|31||272|
location|p31_35|31||280|
```

Define the location of the telemetry mnemonics in packet 32.

```
location|p32_00|32||0|
location|p32_01|32||16|
location|p32_02|32||32|
location|p32_03|32||48|
location|p32_04|32||64|
location|p32_05|32||80|
location|p32_06|32||96|
location|p32_07|32||112|
location|p32_08|32||128|
location|p32_09|32||144|
location|p32_10|32||160|
location|p32_11|32||176|
location|p32_12|32||192|
location|p32_13|32||208|
location|p32_14|32||224|
location|p32_15|32||240|
location|p32_16|32||256|
location|p32_17|32||272|
```

Set up physical channel 2 to contain VCID 4. VCID 4 contains packets 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000. There are no defined telemetry mnemonics for any of these packets.
Define VCDU 4.

```
vcdu|4|2|||||||
```

Define the packets that will go in VCDU 4.

```
packet|100|4||288|2|0.100000|
packet|200|4||288|2|0.200000|
packet|300|4||288|2|0.300000|
packet|400|4||288|2|0.400000|
packet|500|4||288|2|0.500000|
packet|600|4||288|2|0.600000|
packet|700|4||288|2|0.700000|
packet|800|4||288|2|0.800000|
packet|900|4||288|2|0.900000|
packet|1000|4||288|2|1.000000|
packet|1100|4||288|2|1.100000|
packet|1200|4||288|2|1.200000|
packet|1300|4||288|2|1.300000|
packet|1400|4||288|2|1.400000|
packet|1500|4||288|2|1.500000|
packet|1600|4||288|2|1.600000|
packet|1700|4||288|2|1.700000|
packet|1800|4||288|2|1.800000|
packet|1900|4||288|2|1.900000|
packet|2000|4||288|2|2.000000|
```

ASCII to Binary Command (ASCII2BINCMD) Module

ASCII2BINCMD-1.0 Overview

The ascii2bin_cmd module is used to convert an ASCII text database file into multiple binary files to be used by the SIMSS command module. To use this module select an input file and output base file name and lock-run the project. If there are no errors a message will appear in the event log saying that conversion is complete. If there are errors a message will appear in the event log saying there were errors. In the event of errors, consult the log file for details of each error.

ASCII2BINCMD-2.0 Input ASCII Database File

The input file must be an ASCII text file. This file must be made up of records (defined below). The fields of a record are deliminated by the pipe character '|'. The record indicators (first field of each record) are not case sensitive. If there is no data for a field, leave the field empty (do not include any white space).

The white space between the last pipe character of a record and the first character of the next record is discarded. This allows the user to set up a file with new line characters after each record. While a new line is not required, it makes editing and viewing the file much easier.

The telemetry record types are:

'Version'	Defines the version number.
'CMD'	Defines a command record.
'SUBCMD'	Defines a sub-command record.
'COMMENT'	Defines a comment field

ASCII2BINCMD-2.1 VERSION Record

Version|<NUMBER>|

Field	Name	Description
1	Record type	Field must contain 'Version'.
2	Number	This field determines the version of the ascii file being used. Only version 1 is currently accepted. If this record is omitted, a version of 1 will be used.

ASCII2BINCMD-2.2 CMD Record

CMD|<Mnemonic>|<APID>|<VCID>|<Data Type>|<Data Length>|<Data>|

Field	Name	Description
1	Record type	Field must contain 'CMD'.
2	Mnemonic	The name of the command parameter. This field must be no more than 32 characters.

3	APID	The application process identifier uniquely identifying the type of packet to be decommutated. Valid range is from 0 through 2047 (inclusive).
4	VCID	The virtual channel supported. Valid range is from 0 through 63 (inclusive).
5	Data Type	The specific type for spacecraft normal commands. Valid values are 'fixed' or 'variable'.
6	Data Length	The number of bytes in the <Data> field.
7	Data	A hexidecimal value indicating the fixed bit pattern of the data word. Spaces are allowed within the hex value to make the data easier to read.

ASCII2BINCMD-2.3 SUBCMD record

SUBCMD|<Mnemonic>|<Sub-Name>|<Offset>|<Length>|

Field	Name	Description
1	Record type	Field must contain 'SUBCMD'.
2	Mnemonic	The name of the command parameter. Each SUBCMD record must have a matching CMD record with identical <Mnemonic> fields. This field must be no more than 32 characters.
3	Sub-Name	The name of a sub command associated with a variable type command. This field must be no more than 32 characters.
4	Offset	The first bit of the command bit pattern where the sub command value will be inserted in the command data message.
5	Length	The number of bits constituting the sub command value within the command bit pattern. Valid range is from 1 through 32 (inclusive).

ASCII2BINCMD-2.4 COMMENT Record

The COMMENT record defines a comment. All comments are written to the log file, allowing easier debugging of any problems in the input file.

COMMENT|<COMMENT|

Field	Name	Description
1	Record type	Field must be set to 'COMMENT'.
2	Comment	A comment. This will be written to the log file.

ASCII2BIN_CMD-3.0 Output Binary Database Files

This utility will create three output files based on the name given by the user. The file names will be:

<NAME>
<NAME>.var
<NAME>.log

where <NAME> is the binary output file name entered by the user. The first file will contain binary data to be used by the command ingest module. The second file (.second file (.var) will contain the variable command data. The third file (.log) will be a text file containing logged information from the ascii2bin_cmd module. Refer to the log file for information about any errors from this module.

If the binary files do not contain the desired information, delete them, edit the ASCII text file and convert it again using the ascii2bin utility.

NOTE: DO NOT EDIT THE BINARY FILES UNDER ANY CIRCUMSTANCES.

ASCII2BIN_CMD-4.0 Container Items

There are no container items available to the user.

ASCII2BIN_CMD-5.0 Displays

To access 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.

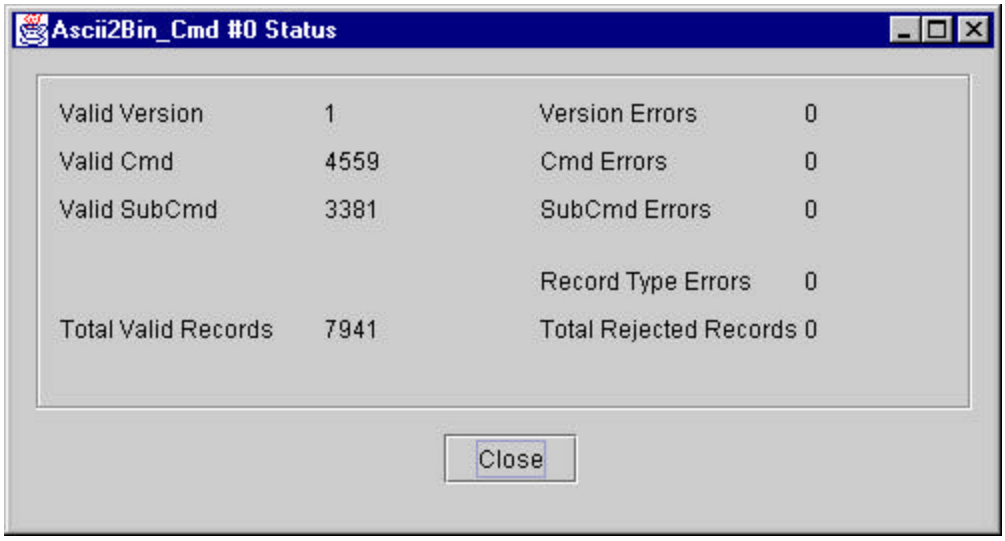
Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display generic module information

ASCII2BIN_CMD-5.1 Configuration

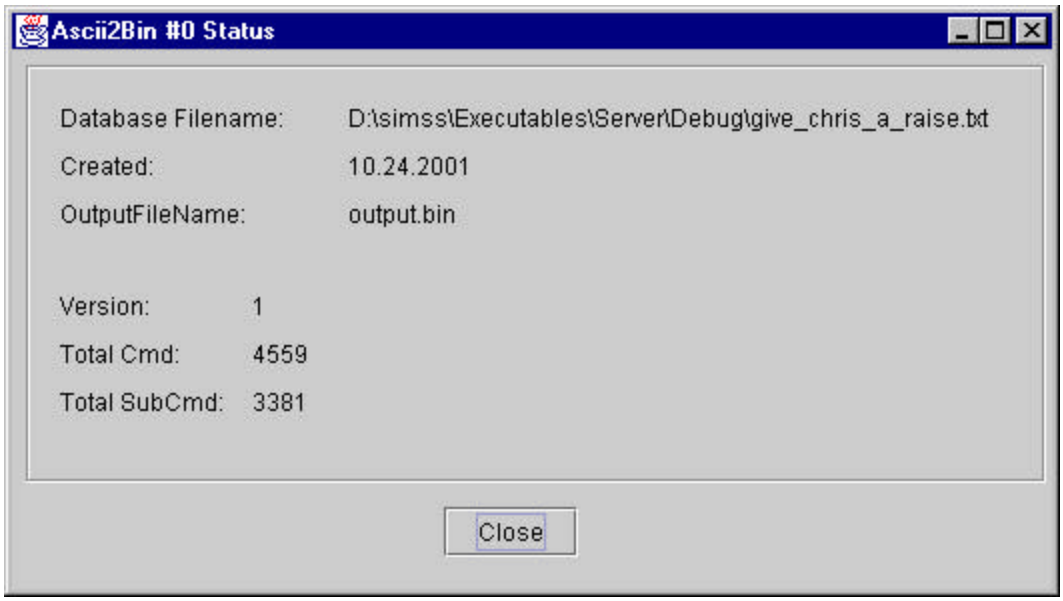
Two things must be configured before running the Ascii2Bin_Cmd module. The user must select an input file and an output file base name.

ASCII2BIN_CMD-5.2 Runtime

There are two status displays windows. The first is a summary of the types of records encountered in the input file. This is a syntax counter and does not take into account any semantic errors (duplicate records, SUBCMDs with no CMDs, etc.).



The second display is a summary of the database. This display reflects the database that is written to the output file.



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.

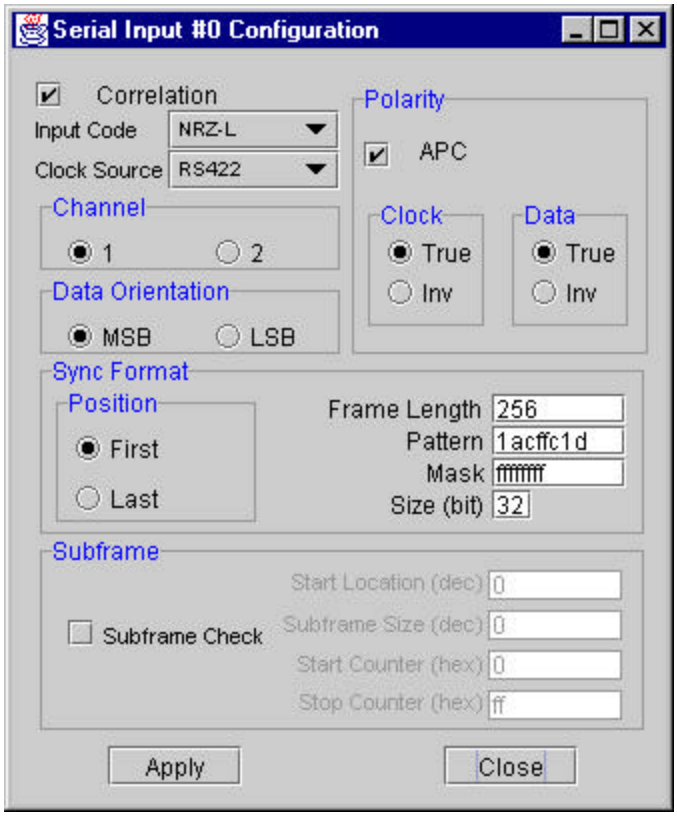
AvtecSerialInput-5.0 Displays

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)

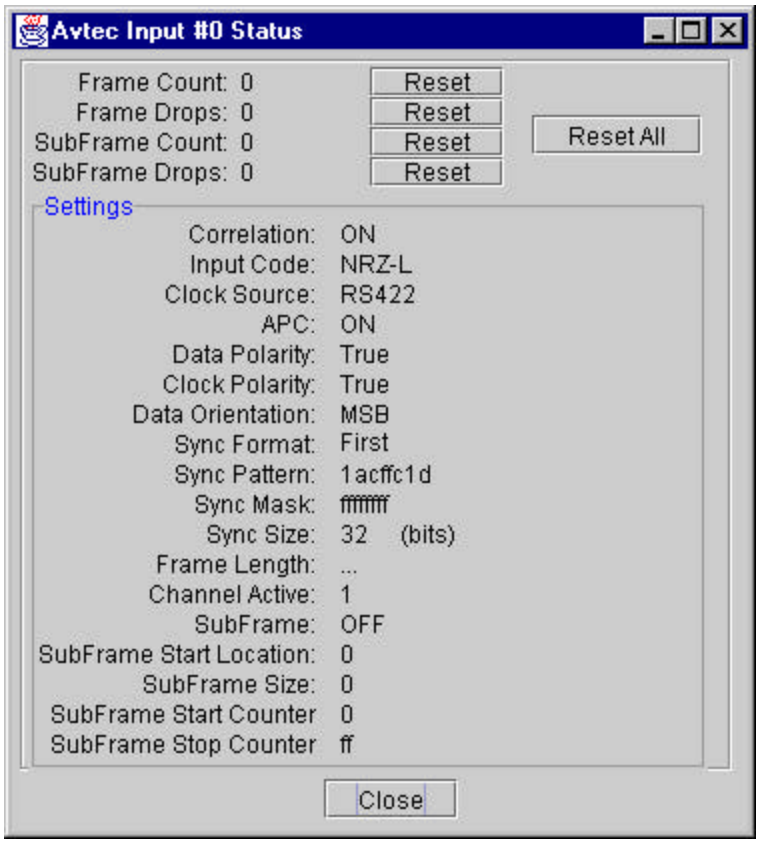
Click **Apply** button to configure any changes. Click **Close** button to dismiss the display.

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.

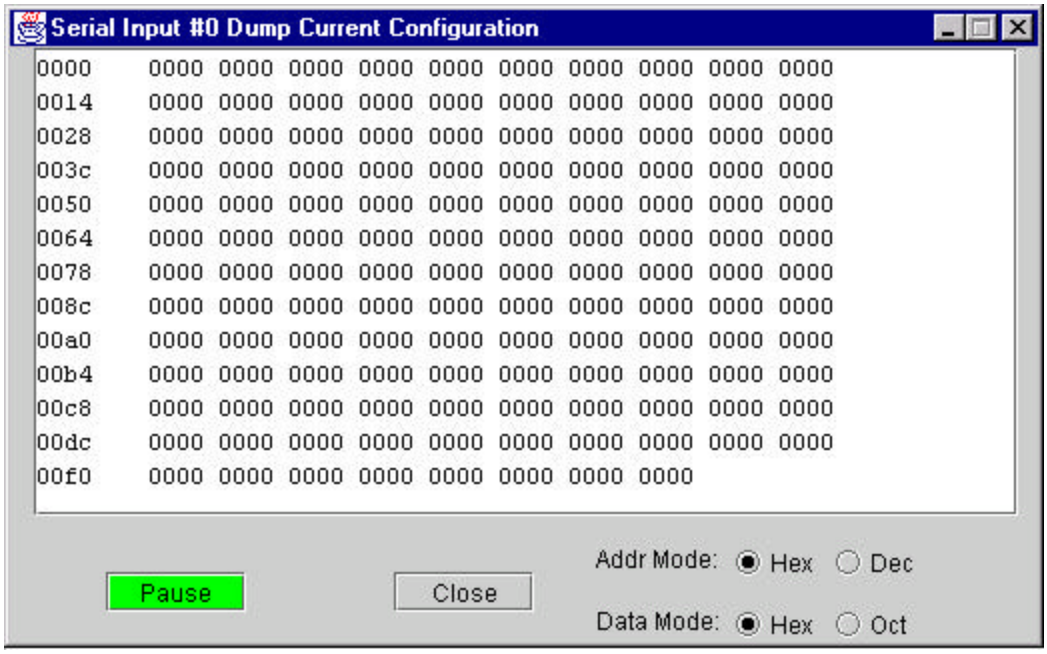
AvtecSerialInput-5.2.2 Display Dump Menu

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.

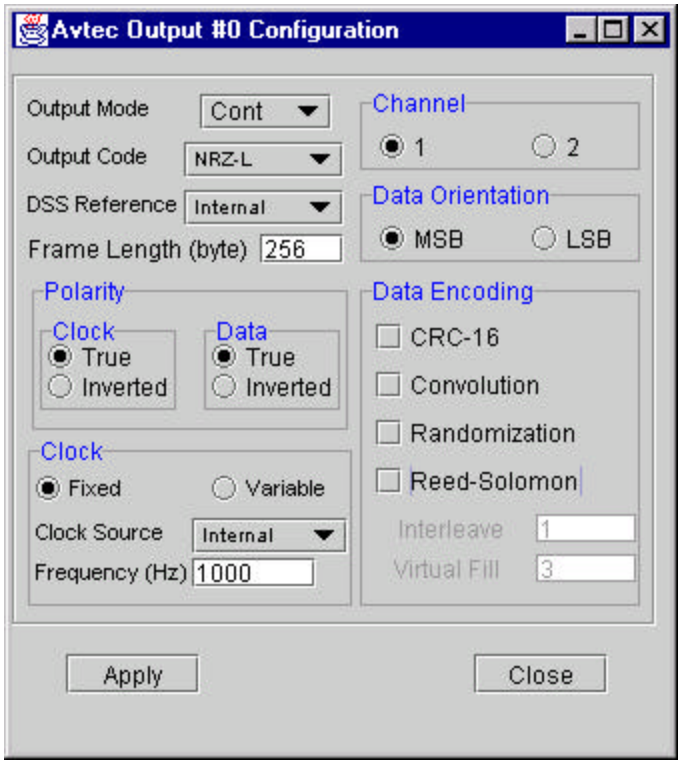
AvtecSerialOutput-5.0 Displays

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)

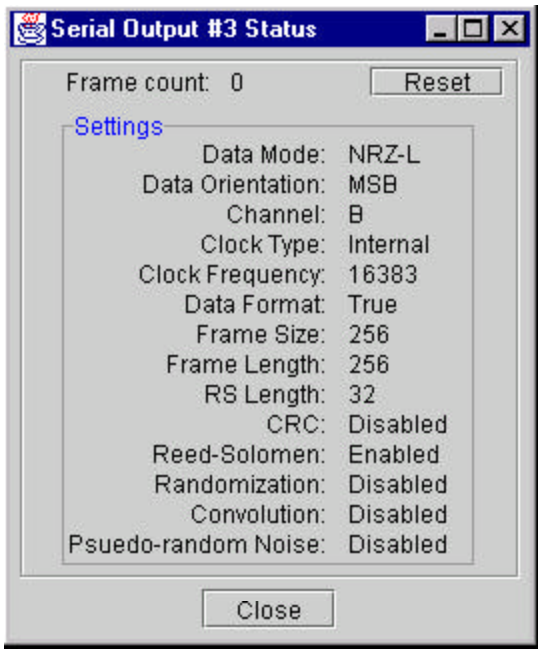
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

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

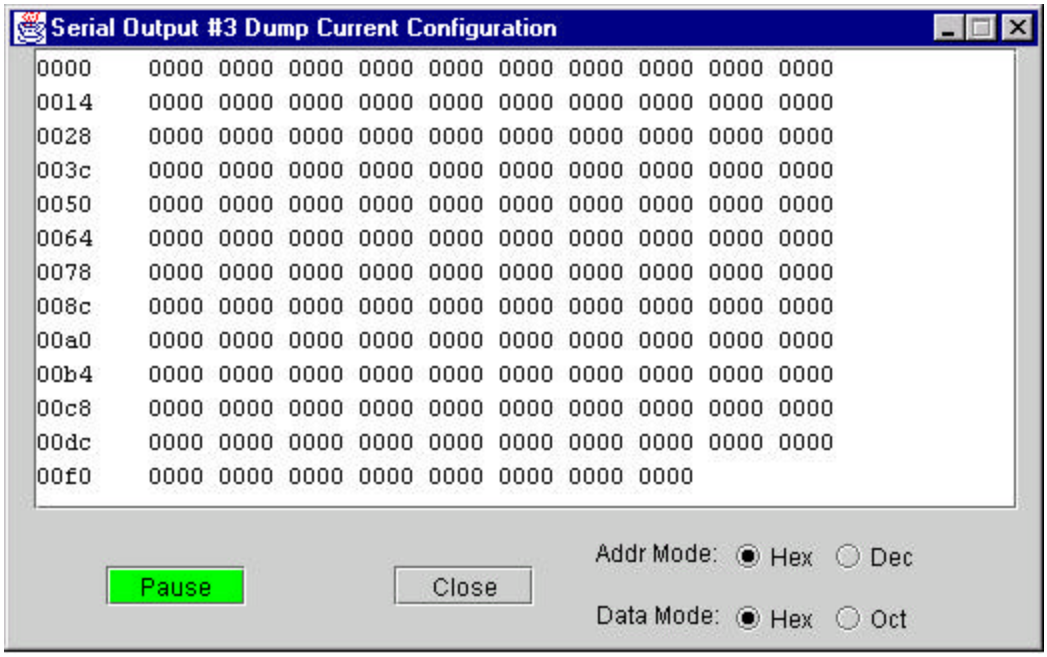
Select the "Display Dump" option to view the available dump formats.

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

AvtecSerialOutput-5.2.2.1 Display Buffer Current Configuration

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.

Block Monitor Module

BlockMonitor-1.0 Overview

The Block Monitor module provides the capability to receive synchronized NASCOM blocks from the upper module, extract NASCOM header and trailer information, and output data to next modules

BlockMonitor-2.0 Input

Channel	Description
1	Receive data from upper module through this channel. Upper module definition: any module that sends NASCOM block. An example would be serial modules. Most of time, serial modules receive NASCOM blocks.

BlockMonitor-3.0 Output

Channel	Description
1	NASCOM data (without header and trailer) is output through this channel. This channel is commonly used with Frame Monitor module
2	NASCOM data (with header and trailer) is output through this channel. This channel is commonly used with Log module
3	NASCOM data, source ID, and destination ID information are output through this channel. This channel is commonly used with Shuttle Frame Monitor module

BlockMonitor-4.0 Container Items

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

BlockMonitor-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

BlockMonitor-5.1 Configuration

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

Parameter	Description
Bits/Sec	Not implemented in this release
Delta T Exp	Not implemented in this release
DT Variance	Not implemented in this release
DT Variance (%)	Not implemented in this release
Type	Enter "Type" information for Block Monitor to filter the NASCOM block that has the same "Type" information in the header. Blank is "don't care"
SCID	Enter "Spacecraft ID" information for Block Monitor to filter the NASCOM block that has the same "Spacecraft ID" information in the header. Blank is "don't care"
DSID	Enter "Data Stream ID" information for Block Monitor to filter the NASCOM block that has the same "Data Stream ID" information in the header. Blank is "don't care"
SrcID	Enter "Source ID" information for Block Monitor to filter the NASCOM block that has the same "Source ID" information in the header. Blank is "don't care"
DestID	Enter "Destination ID" information for Block Monitor to filter the NASCOM block that has the same "Destination ID" information in the header. Blank is "don't care"
NASCOM Header	To define the NASCOM header type which the Block Monitor module is receiving.
Select Database	To select pre-defined configuration parameters from selected database

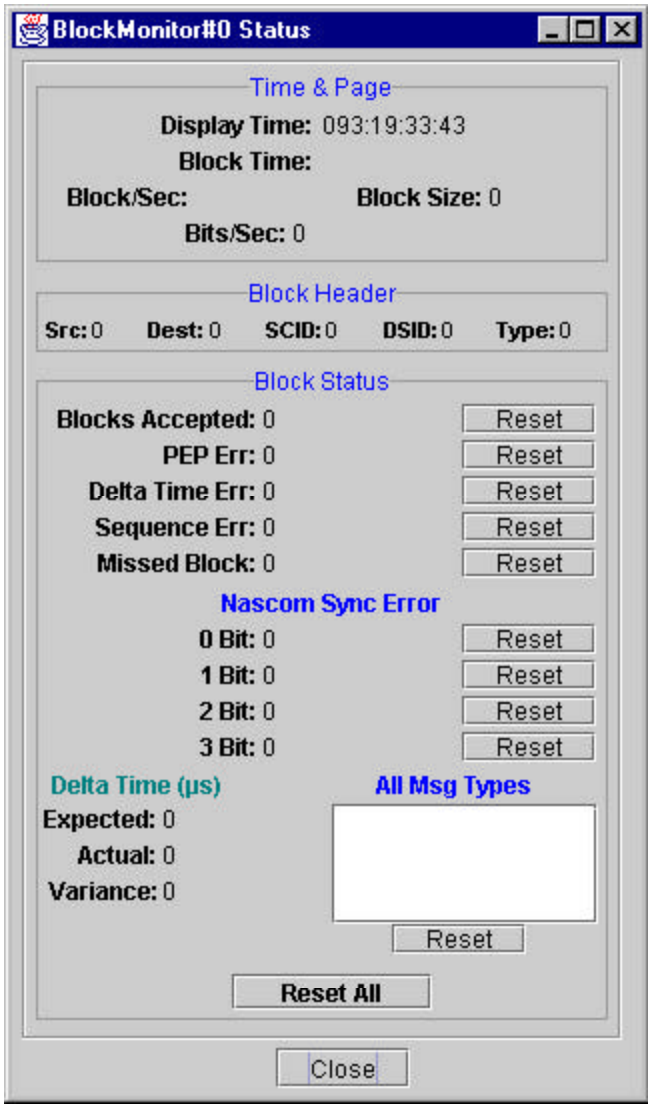
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

BlockMonitor-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. Because there are three output channels, just the data of channel 2 is displayed
Resume	Restart the module
Pause	Stop the module

BlockMonitor-5.2.1 Display Status



The status window of Block Monitor module shows information of NASCOM header, data rate, and block statistic.

Use the **Reset All** button to reset all counters.

Use the **Reset** buttons to reset individual associated counters.

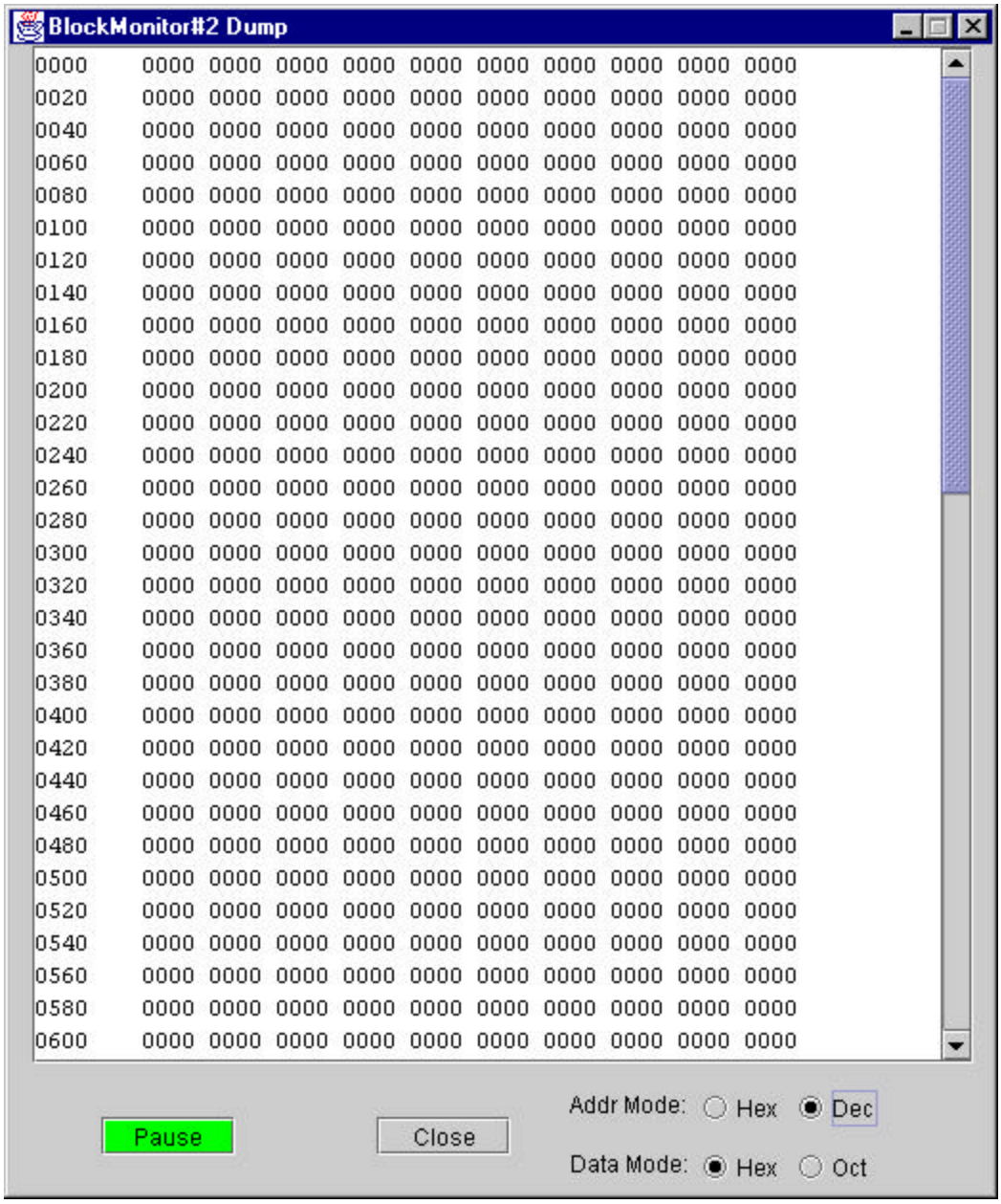
The "All Msg Types" window is used to show numbers of each message type in the incoming NASCOM block stream. The counters of the "All Msg Types" window will be reset when "Reset" or "Reset All" button is clicked. The list of all message types will be cleared when the Block Monitor module is re-configured during run-time.

The "Delta Time", Expected, Actual, and Variance statistics are not implemented in this version.

BlockMonitor-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 of channel 2



When this option is selected, the content of the output data is displayed. This display window has the following standard 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

BlockMonitor-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).

BlockMonitor-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.

BlockMonitor-5.3 Remove

Select this option to remove the module from the project.

BlockMonitor-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.

BlockMonitor-6.0 Special Operating Instructions

None

CmdEcho Module

CmdEcho-1.0 Overview

The function of the CmdEcho module is to take an incoming NASCOM block, swap the source and destination fields of the header, recalculate the PEP, and output the block to an IP module. The source and destination field byte locations are defined in the "cmdecho.txt" file. If the file cannot be read, the default byte location 3 is used for the source and byte location 4 is used for the destination. The CmdEcho module also checks for a valid NASCOM block length.

CmdEcho-2.0 Inputs

Channel	Description
1	One NASCOM block

CmdEcho-3.0 Outputs

Channel	Description
1	One NASCOM block

CmdEcho-4.0 Container Items

Name	Type	Description
NascomBlockBuffer	Buffer	Contains the NASCOM block

CmdEcho-5.0 Displays

To access displays for this module, click the center of the module in the project window. The pop-up menu choices will appear. The "Remove" option can be used during the project design to remove the module. The "About" option provides generic module information.

Module Pop-up Menu	Description
Remove	Remove the module from the project
About	Display generic module information

CmdEcho-6.0 Special Operating Instructions

The CmdEcho module reads the NASCOM block source and destination byte locations from the "cmdecho.txt" file. The first entry in the file is the source field byte location, the second entry is the destination field byte location. In the event this file cannot be read, the default byte location for the source field is 3 and the default byte location for the destination field is 4. In either case, the byte locations used for the source and destination fields are written to the event log. The CmdEcho module checks for a valid NASCOM block length. If there is an error in the block length, the CmdEcho module will write an error message to the event log and drop the NASCOM block.

CMDXmit Module

CmdXmit-1.0 Overview

The CDMXMIT module serves as a source for commands. It provides the user with CCSDS templates for ease of creating commands. Any command can be generated, not only CCSDS commands. For instance, the user can read in any data into the Command Data template. The Command Data template can be used to enter commands free form.

CmdXmit-2.0 Inputs

None.

CmdXmit-3.0 Outputs

Channel	Description
1	Command data sent from internal output buffer

CmdXmit-4.0 Container Items

Name	Type	Description
CmdXmitRun	Fixed	Run/pause flag
CmdXmitTFCodeBlockSize	Fixed	Code Block Size
CmdXmitTFPoly	Fixed	0 = no poly, 1 = poly
CmdXmitCmdsSent	Fixed	Total number of commands transmitted
CmdXmitDBScid	Fixed	Spacecraft Identification
CmdXmit Interval	Fixed	Interval in milliseconds between command transmits

CmdXmit-5.0 Displays

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

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

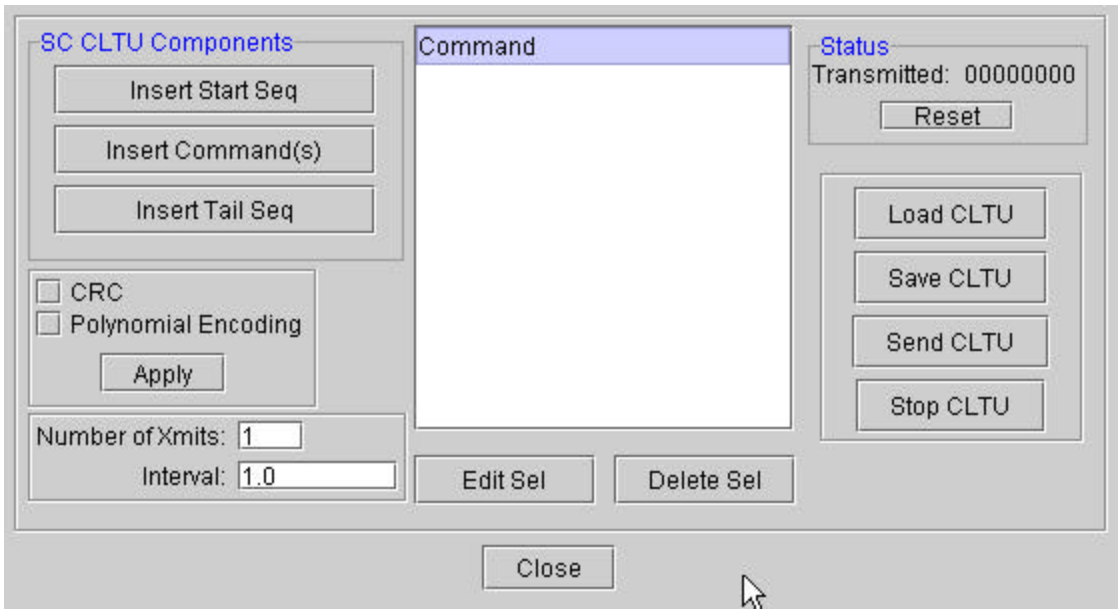
CmdXmit-5.1 Run-time

The Run-time menu for CmdXmit module contains the following items.

Run-time Menu Item	Description
Build CLTU	Create a CLTU
DataBase	Create/retrieve a configuration database

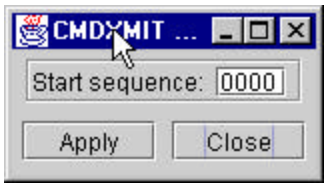
CmdXmit-5.1.1 Build CLTU

This allows the user to create/retrieve/control transmission of CLTUs. Please note that the user can create/retrieve/control transmission of ANY type of command data. The CLTU is merely a template to facilitate CCSDS command building. The user can add as many start sequences, tail sequences and commands as desired. No limit checks are made as these are merely templates.



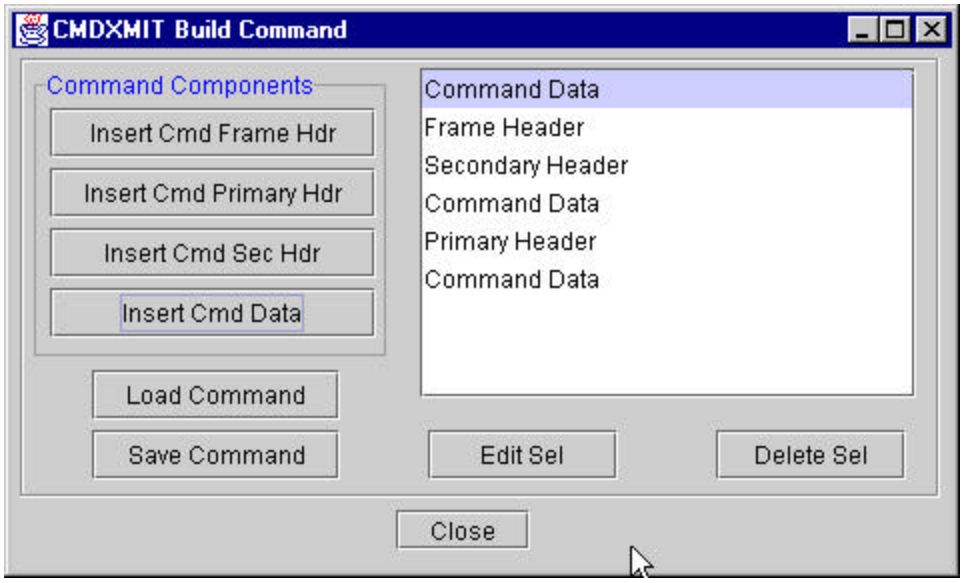
CmdXmit-5.1.1.1 Start Sequence

The Start Sequence allows the user to enter two bytes of command data.



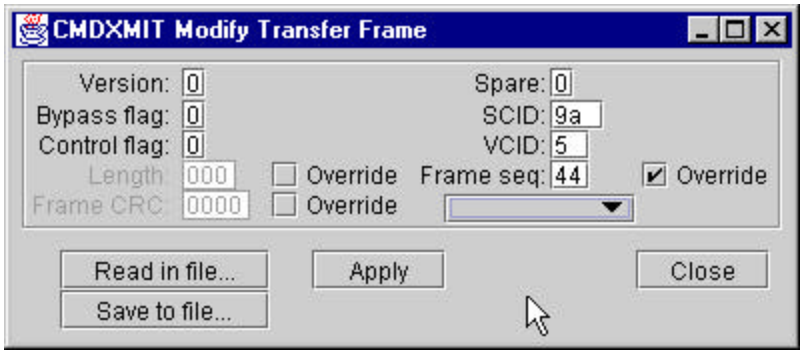
CmdXmit-5.1.1.2 Command

The Command template allows the user to build a command using other templates. Again, the user can use any number of and in any sequence the templates. The user may also save a command or retrieve a previously built command.



CmdXmit-5.1.1.2.1 Frame Header

The user can use the frame header to create and/or modify a transfer frame header.

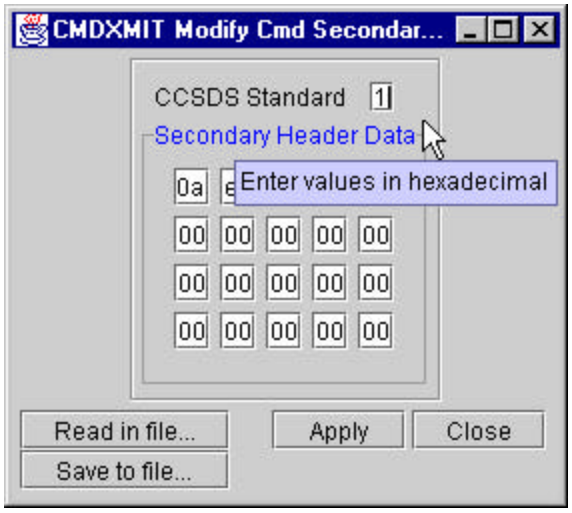


CmdXmit-5.1.1.2.2 Primary Header

The user can use the primary header to create and/or modify a packet header.

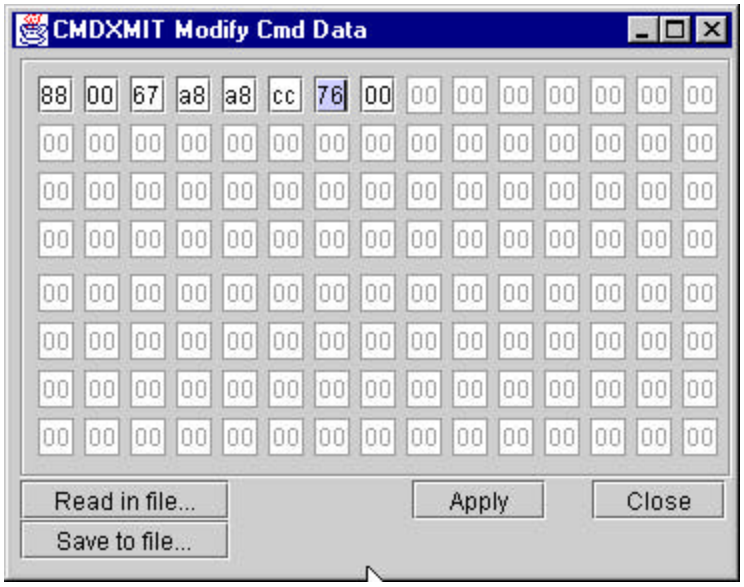
CmdXmit-5.1.1.2.3 Secondary Header

The user can use the secondary header to create and/or modify the packet secondary header.



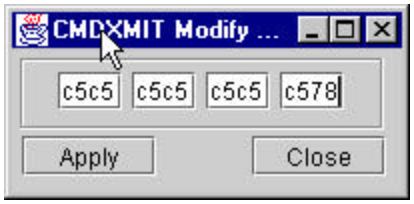
CmdXmit-5.1.1.2.4 Command Data

The user can use the command data template to create and/or modify command data. This template can be used to create/restore any type of command data wanted, such as a NSSC 48 bit command. As the user types in data, the next field is desensitized. The total number of bytes are the number of bytes that the user actually typed in. In the example below there would be 7 bytes entered.



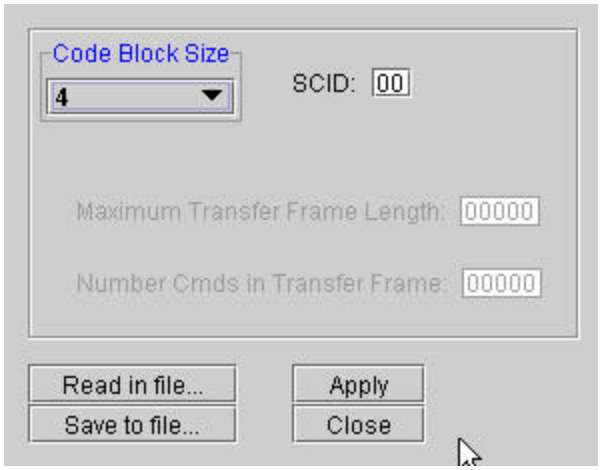
CmdXmit-5.1.1.3 Tail Sequence

The Tail Sequence allows the user to enter eight bytes of command data.



CmdXmit-5.1.2 Database Layout

The database layout allows the user to save/retrieve the code block size and spacecraft identification from session to session.



Encoding Module

Encoding-1.0 Overview

The Encoding module provides the capability to receive data from other modules and encode data before passing to next module.

Encoding -2.0 Input

Channel	Description
1	Receive data from upper module through this channel

Encoding -3.0 Output

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

Encoding -4.0 Container Items

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

Encoding -5.0 Displays

To access the displays for this module, click in the center of the module in the project window. The module pop-up menu will appear with the following choices. 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 Encoding 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

Encoding -5.1 Configuration

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

Parameter	Description
Frame Size Enable	Uncheck this option for variable size of incoming data block.
Frame Length	Select the size of block data to be encoded, in bytes.
Data Encoding	Click the appropriate boxes to enable "CRC-16", "Convolution", "Randomization", or "Reed-Solomon".
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.
Sync Size	Enter the size of the sync pattern in bytes so that sync pattern is not encoded.

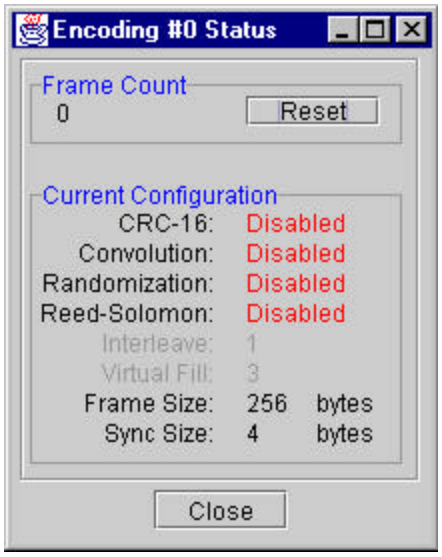
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

Encoding -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 Encoding module
Display Dump Menu	Show options for displaying data within a frame
Resume	Restart the module
Pause	Stop the module

Encoding-5.2.1 Display Status



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

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

Click **Close** to close the window.

Encoding -5.2.2 Display Dump Menu

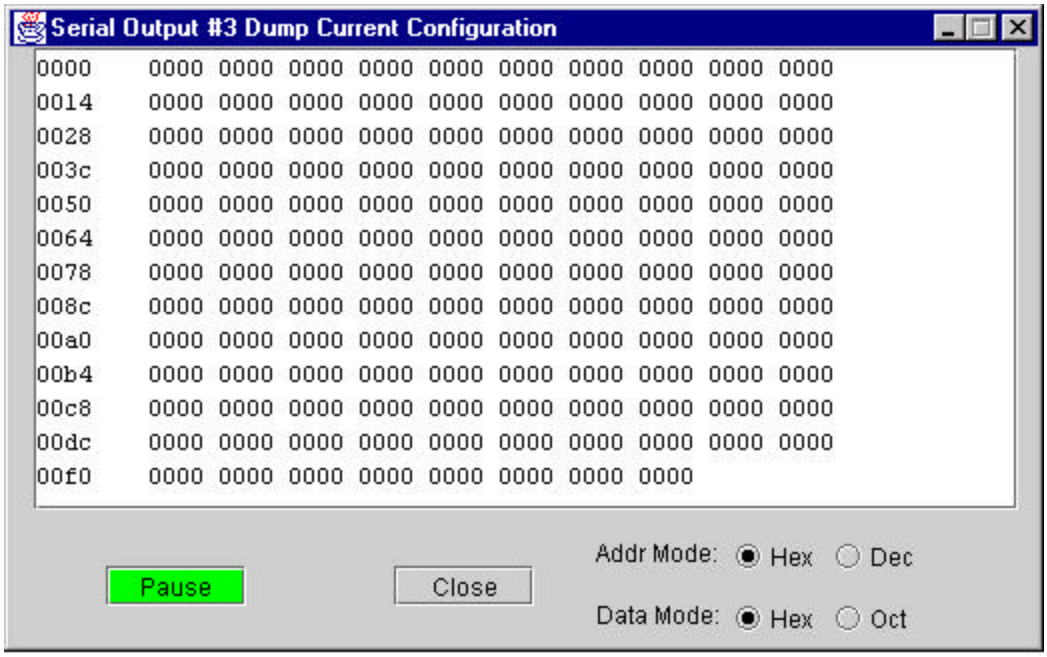
Select the “Display Dump” option to view the available dump formats.

Display Dump Menu Item	Description
Display Buffer Current Configuration	To display data within a frame.

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 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



Encoding -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).

Encoding -5.2.4 Pause

Select “Pause” from the Run-time menu to temporarily stop the Encoding 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 Encoding 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.

Encoding-5.3 Remove

Select this option to remove the Encoding module from the project.

Encoding-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.

Encoding-6.0 Special Operating Instructions

Because the Encoding module does not generate the “tic” as the Serial Output module does, configuring the Encoding module to receive data from other modules should be the same as configuring the Output IP module. The Encoding module can relay the “tic” from Serial Output module to upper module.

Flat File to ASCII (FF2AS) Database Utility

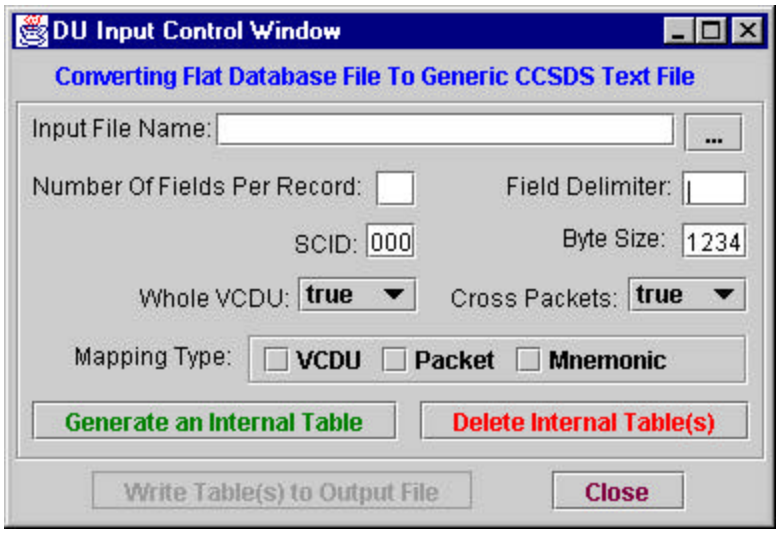
FF2AS-1.0 Overview

This offline utility will convert project specific database flat files into a single ASCII text file that is in the generic SIMSS CCSDS Database format.

Note: The ASCII database file produced from this utility must be converted into a set of SIMSS-specific binary files before the database information can be used by SIMSS modules. Conversion of this ASCII file into binary is done by the ascii2bin utility, which is described in the ASCII2BIN chapter of the user's guide.

FF2AS-2.0 Run the Flat File to ASCII Database Utility

In the Utilities folder under the Executables directory, run the RunFlatfile2Ascii.bat batch file. It will bring up a window similar to the following.



FF2AS-2.1 Step 1: Enter Mission Specific Data Fields

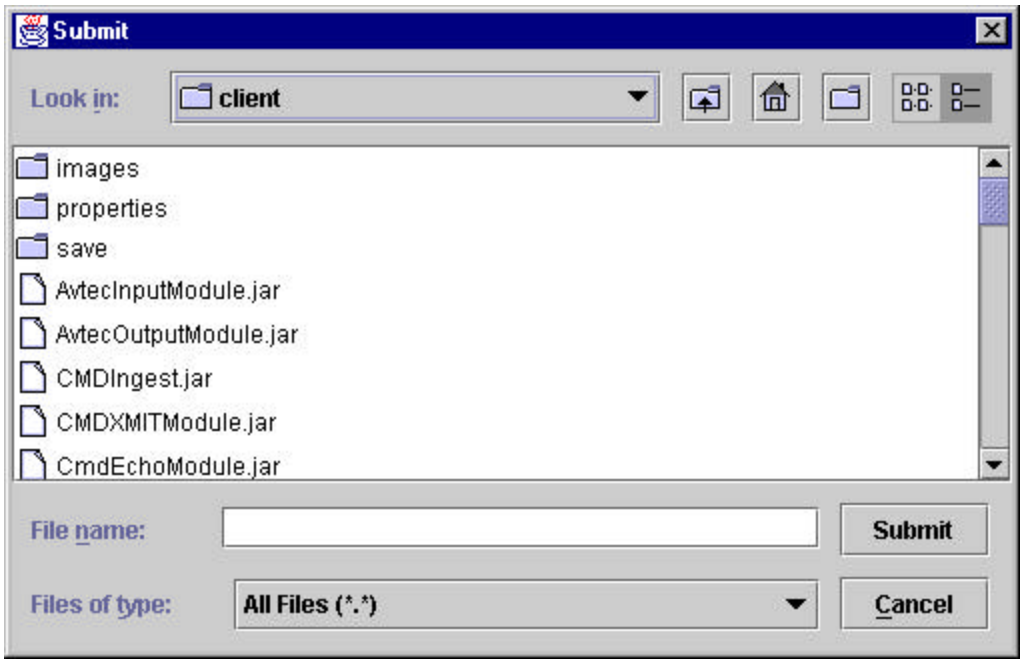
Enter the following mission specific data fields in the Input Control Window display. Often this information is provided in an Interface Control Document or other mission documentation and is not contained in the database flat files.

Input Control Data Field	Description
SCID	Spacecraft Identification in decimal
Byte Size	Length of VCDU in decimal bytes.
Whole VCDU	When set to true, the VCDU is transmitted when it is filled with data. When set to false, the VCDU is filled with fill packets and transmitted at a specific rate.
Cross Packets	When set to true, the telemetry packets may be broken across VCDU boundaries. When set to false, fill packets are

	used to fill gaps and data packets do not cross boundaries.

FF2AS-2.2 Step 2: Get An Input Database Flat File

The user can type in the name of an input file in the text field or use the file browser to select an input file. Clicking on the file browser will bring up a file selection window that looks like the following image.



FF2AS-2.3 Step 3: Enter Number of Fields and Field Delimiter

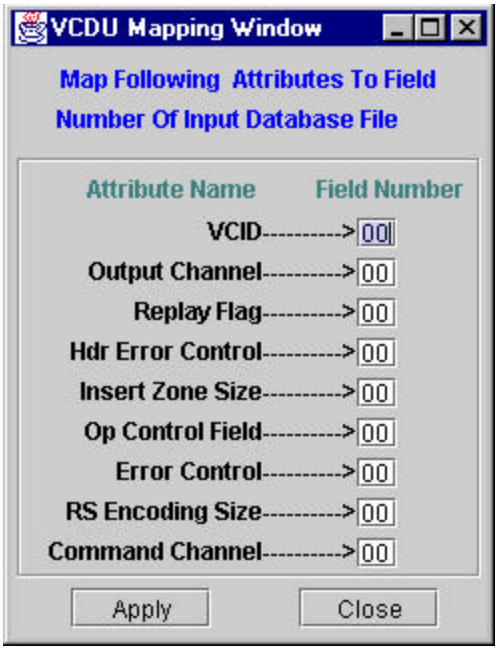
The user must enter the number of fields in the selected flat file's record format and the field delimiter character used.

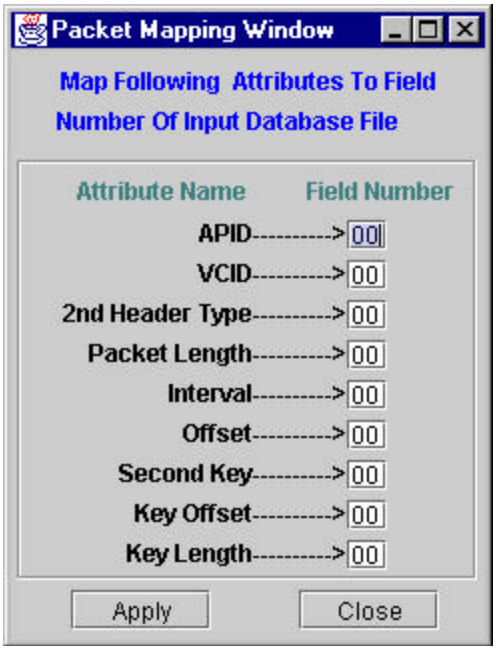
Limitation: This utility can ONLY process input files that have the same number of fields in all records and uses a single character field delimiter.

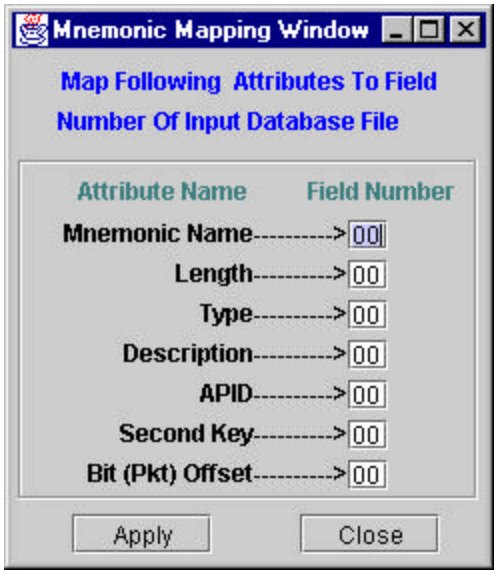
FF2AS-2.4 Step 4: Specify Mapping to SIMSS Data Fields

Next the user must select the type of information to be extracted from the currently selected flat file. There are three mapping types: VCDU, Packet, and Mnemonic.

One of the following three windows will be shown when a user clicks a mapping type. Next the user must specify the Field Numbers from the flat file record that contain each listed Attribute Name. For attributes that are not contained in the current flat file, the user should enter zero. Click **Apply** after all fields have been entered.







FF2AS-2.5 Step 5: Generate Internal Table

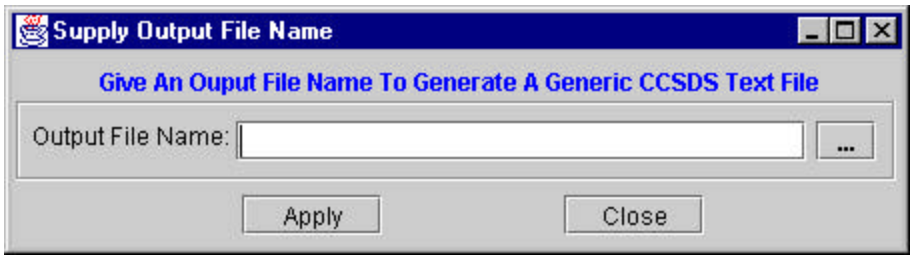
When the user clicks the **Generate an Internal Table** button, the utility will generate an internal table according to the user specified input data. The utility will generate one of the three types of internal tables (VCDU, Packet, Mnemonic) each time the user clicks this button. A corresponding internal table should be generated each time a flat file is mapped to its type. When multiple flat files are mapped to the same type, the internal table that is generated merges the information.

FF2AS-2.6 Step 6: Process Additional Flat Files

Repeat steps 2, 3, 4 and 5 until all of the flat file information has been processed. It is not necessary to have a one-to-one correspondence between the flat files and the map files. Enter the fields that are applicable from each flat file to each map file.

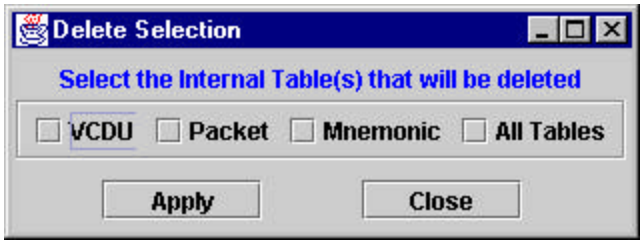
FF2AS-2.7 Step 7: Write Table(s) to Output File

The final step is to create a single ASCII output file. After a user generates all of the internal table(s), the user clicks the **Write Table(s) to Output File** button, which will bring up the window that is shown below. The user can either type the name of the output file on the text field or click the browser button to bring up the file selection window. After the output file name is shown on the text field, the user clicks the **Apply** button and the ASCII output file will be generated.



FF2AS-2.8 Step 8: Delete Internal Table(s)

A user can optionally delete any or all of the internal table(s). The user clicks the **Delete Internal Table(s)** button; which will bring up the following window. The user can select the table(s) to be deleted. If the user wants to delete all of the internal tables, select the All Tables choice. This step can be done any time after the user has generated internal table(s). Since data may be accumulated from multiple flat file mappings and merged into one of the internal tables, this step must be used to remove any improperly mapped data in order to start over. The user should be careful not to delete valid internal tables before they have been written to an output file.



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

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

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.

Parameter	Description
Pattern	Enter Sync. Patterns for the module to synchronize data
Size	Enter the sync size in bits
ID Location	Enter the byte location (start 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
Bit Rate	Enter data rate of the receiving data stream
Select Database	Select data base name for configuration parameters
Select PDF	Select pre-defined configuration parameters from a PDF name

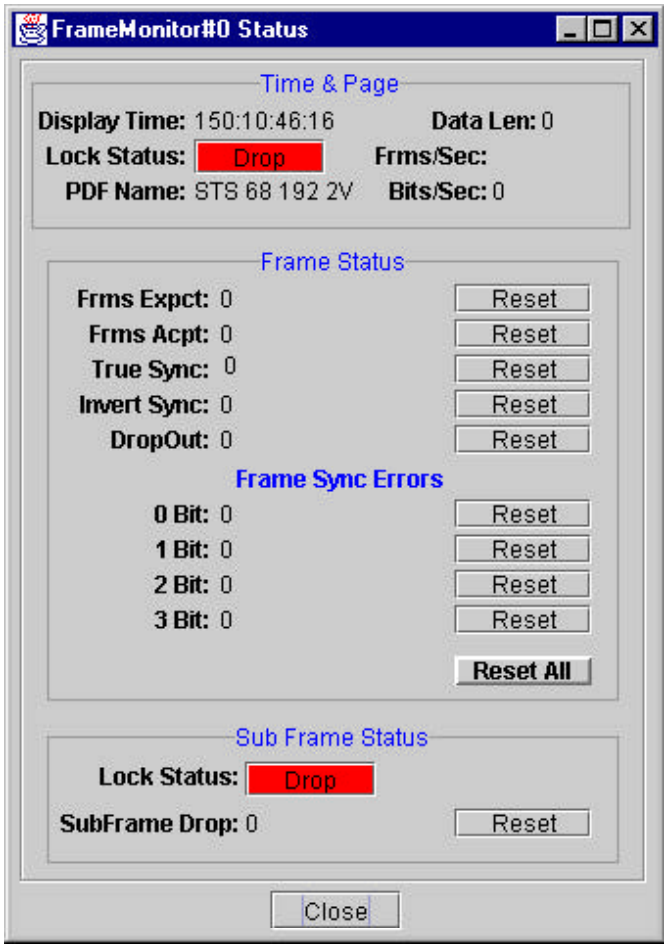
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

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 statistic.

- 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

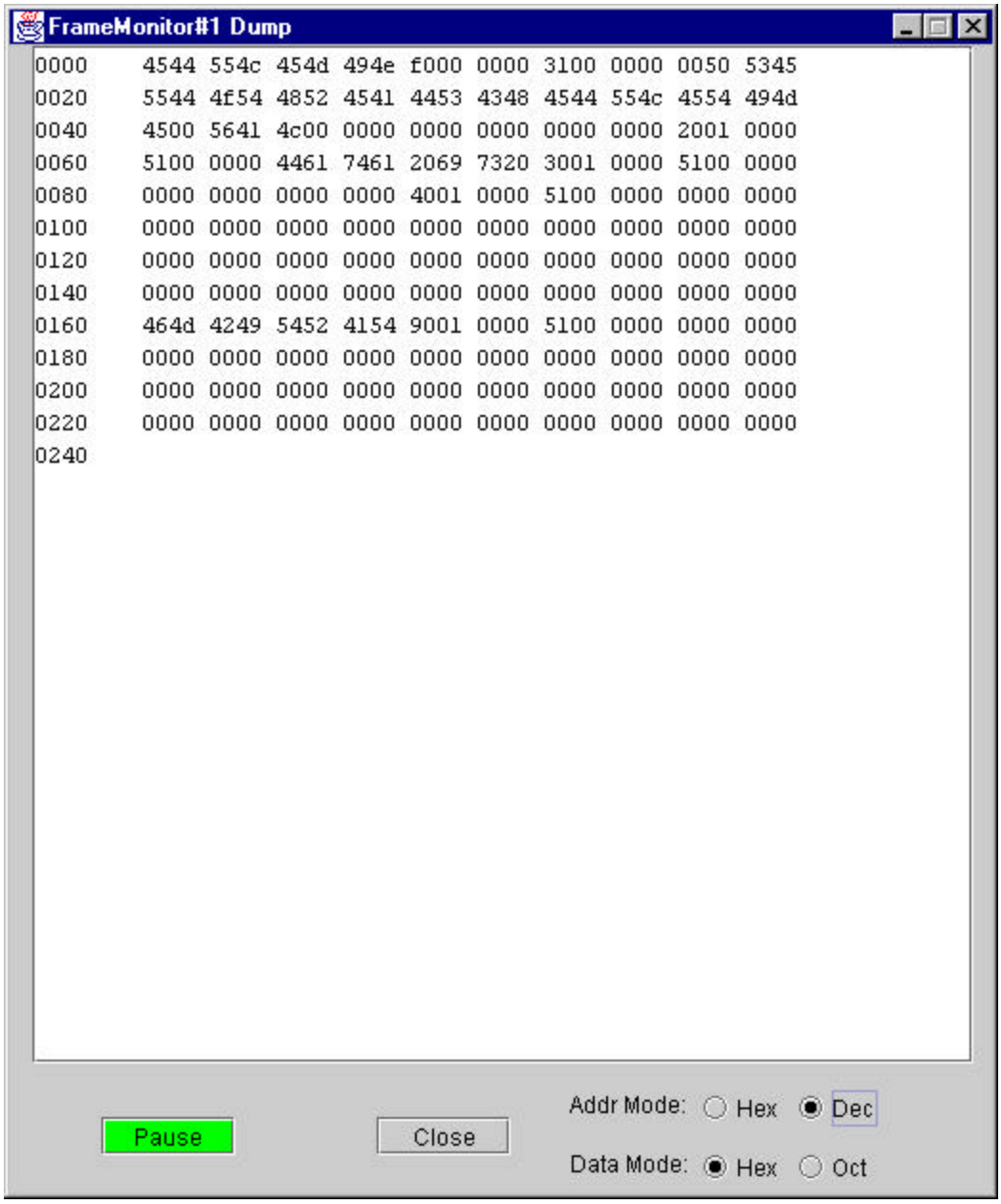
Select the "Show Output Block" option to view the available dump formats.

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. This display window has the following standard 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



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)

Frame Monitor for Shuttle Module

FrameMonitorShuttle-1.0 Overview

The Frame Monitor for Shuttle module provides the capability to receive asynchronous data from an input module, synchronize data based on sync patterns and frame size parameters and pass data to connected output modules.

FrameMonitorShuttle-2.0 Input

Channel	Description
1	Receive data from upper module through this channel

FrameMonitorShuttle-3.0 Output

Channel	Description
1	Synchronous data is output through this channel

FrameMonitorShuttle-4.0 Container Items

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

FrameMonitorShuttle-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

FrameMonitorShuttle-5.1 Configuration

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

Parameter	Description
Pattern	Enter Sync. Patterns for the module to synchronize data
Size	Enter the sync size in bits
ID Location	Enter the byte location (start 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
Bit Rate	Enter data rate of the receiving data stream
Select Database	Select data base name for configuration parameters
Select PDF	Select pre-defined configuration parameters from a PDF name

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

FrameMonitorShuttle-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

FrameMonitorShuttle-5.2.1 Display Status

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

- The "Reset All" button is to reset all counters
- The "Reset" button associated with each counter is to reset that counter
- Frames with sync patterns errors fall within number 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 "Search"
- If "NSP Selected" shows 1 or 2, then the "Unlock" or "Lock" status for "Frame Sync" and "Bit Sync" with associated with bit sync and frame sync 1 or 2.
- If "NSP Selected" shows 0, then the "Frame Sync" and "Bit Sync" always show "Unlock"
- 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

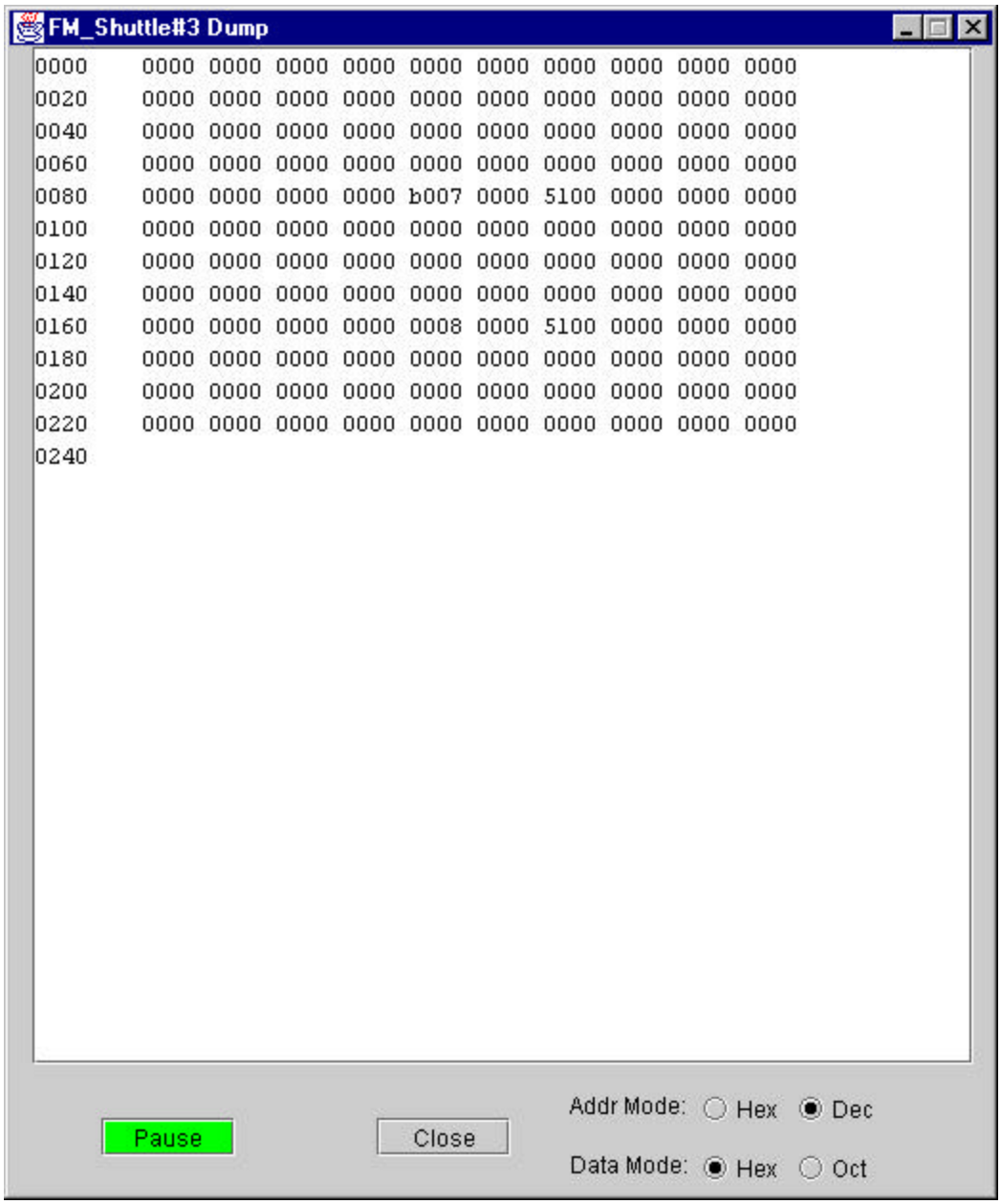
FrameMonitorShuttle-5.2.2 Display Dump Menu

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. This display window has the following standard 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



FrameMonitorShuttle-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).

FrameMonitorShuttle-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.

FrameMonitorShuttle-5.3 Remove

Select this option to remove the module from the project.

FrameMonitorShuttle-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.

FrameMonitorShuttle-6.0 Special Operating Instructions

The input channel of this module is commonly connected to the output channel 3 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).

Generic Command Ingest (GenericCmdIngest) Module

GenericCmdIngest-1.0 Overview

The CCSDS Generic Command Ingest receives command data in the form of command link transmission units (CLTUs), validates the CLTUs, generates a command link control word (CLCW) for each virtual channel and transmits these CLCWs to an external destination.

GenericCmdIngest-2.0 Inputs

The GenericCmdIngest module has 1 input channel.

Ch.	Data expected	Validation performed	Processing performed
1	CLTU stream	CCSDS Telecommand Standard	Commands parsed and validated per CCSDS standard. Event messages generated on valid or invalid commands.

GenericCmdIngest-3.0 Outputs

The GenericCmdIngest module has 1 output channel.

Ch.	Description
1	Command Link Control Words (CLCWs)

GenericCmdIngest-4.0 Container Items

This module accepts operator directives. 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. The container items for command processing fall into several groups: mission specific items, validation fields and processing buffers. The initial values for counts, flags and buffers are zero. Non-trivial initial values for other items are shown in parentheses following the description.

GenericCmdIngest-4.1 Mission Specific Container Items

These values should be set for the specific CCSDS spacecraft being simulated. Initial default values are shown in parentheses. Values may be changed using set or setBuffer directives from the directive line or a scenario file.

Name	Type	Description
GenericCmdCltuCodeblockSize	Fixed	Codeblock size in bytes (8)
GenericCmdCltuExpectedStartSequence	Buffer	Expected CLTU start sequence buffer (2 bytes) ($EB90_{16}$)
GenericCmdCltuExpectedTailSequence	Buffer	Expected CLTU tail sequence buffer(8 bytes)

		(C5C5 C5C5 C5C5 $C579_{16}$)
GenericCmdSCID	Fixed	Spacecraft Identifier (0)
GenericCmdSlidingWindowSize	Fixed	FARM-1 Sliding Window Size (180)

GenericCmdIngest-4.2 Command Validation Container Fields/Flags

Name	Type	Description
GenericCMDEnabled	Fixed	Command processing enabled flag. (0=disabled, 1=enabled)
GenericCmdCLTUValidation	Fixed	Command CLTU validation enabled flag. (0=disabled, 1=enabled)
GenericCmdCodeblockValidation	Fixed	Command codeblock parity validation enabled flag. (0=disabled, 1=enabled)
GenericCmd FrameValidation	Fixed	Command transfer frame header validation enabled flag. (0=disabled, 1=enabled)
GenericCmd FARMValidation	Fixed	Command Frame Acceptance and Reporting Mechanism (FARM) validation enabled flag. (0=disabled, 1=enabled)
GenericCmd PacketValidation	Fixed	Command packet validation enabled flag. (0=disabled, 1=enabled)
GenericCmdCLTUDebugEnabled	Fixed	CLTU debug flag (0=disabled, 1=enabled). When enabled, the current CLTU will be displayed in an event message even if global debug flag is disabled.
GenericCMDDebugEnabled	Fixed	Command subsystem debug messages flag. (0=disabled, 1=enabled) When enabled, additional event messages are generated throughout command ingest processing.
GenericCMDPacketDumpEnabled	Fixed	Packet Dump Enabled flag (0=disabled, 1=enabled). When enabled, the contents of valid type AD commands will be displayed in event messages
GenCmdClcw<##>CWT	Fixed	CLCW Control Word Type for VCID <##>
GenCmdClcw<##>Version	Fixed	CLCW Version for VCID <##>
GenCmdClcw<##>Status	Fixed	CLCW Status for VCID <##>
GenCmdClcw<##>COP	Fixed	CLCW COP In Effect for VCID <##>
GenCmdClcw<##>VCID	Fixed	CLCW VCID (0) for VCID <##>
GenCmdClcw<##>Spare1	Fixed	CLCW Spare field 1 for VCID <##>
GenCmdClcw<##>NoRFAvail	Fixed	CLCW No RF Avail Flag for VCID <##>
GenCmdClcw<##>NoBitLock	Fixed	CLCW No Bit Lock Flag for VCID <##>
GenCmdClcw<##>Lockout	Fixed	CLCW Lockout Flag for VCID <##>
GenCmdClcw<##>Wait	Fixed	CLCW Wait Flag for VCID <##>
GenCmdClcw<##>Retransmit	Fixed	CLCW Retransmit Flag for VCID <##>
GenCmdClcw<##>FarmCount	Fixed	CLCW Farm-B Counter for VCID <##>

GenCmdClcw<##>Spare2	Fixed	CLCW Spare field 2 for VCID <##>
GenCmdClcw<##>Report	Fixed	CLCW Report Value for VCID <##>

GenericCmdIngest -4.3 Command Container Buffers

Name	Type	Description
GenericCmdPolyRemainderTbl	Buffer	Polynomial remainder table for parity calculation. (256 bytes)
GenericCmd<##>CLCW	Buffer	CLCW buffer (4 bytes) for VCID <##>
GenericCmd<##>Pkt	Buffer	Command packet buffer (128 bytes) for VCID <##>
GenericCmd FrameBuffer	Buffer	Command transfer frame buffer (256 bytes)
GenericCmdCLTU	Buffer	Command link transmission unit buffer (6000 bytes)
GenericCmdCodeblock	Buffer	Compressed codeblock buffer (holds codeblock bytes without parity bytes) (6000 bytes)

GenericCmdIngest-4.4 Command Container Counters

Name	Type	Description
GenericCmdTotalCLTUs	Fixed	Count of all CLTUs received
GenericCmdValidCLTUs	Fixed	Count of valid CLTUs
GenericCmdRejectCLTUs	Fixed	Count of invalid CLTUs
GenericCmdTotalCodeblocks	Fixed	Count of all Codeblocks
GenericCmdValidCodeblocks	Fixed	Count of valid Codeblocks
GenericCmdRejectCodeblocks	Fixed	Count of invalid Codeblocks
GenericCmdTotalTransferFrames	Fixed	Count of all Transfer Frames
GenericCmdValidTransferFrames	Fixed	Count of valid Transfer Frames
GenericCmdErrorTransferFrames	Fixed	Count of invalid Transfer Frames
GenericCmdADFrames	Fixed	Count of Type AD Transfer Frames
GenericCmdACFrames	Fixed	Count of Type AC Transfer Frames
GenericCmdBCCMDS	Fixed	Count of Type BC Transfer Frames
GenericCmdBDCMDS	Fixed	Count of Type BD Transfer Frames
GenericCMDIgnoredCLTUs	Fixed	Number of CLTUs ignored while command processing is disabled.

GenericCmdIngest-5.0 Displays

To access the displays for this module, first click on the center of the GenericCmdIngest 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

GenericCmdIngest-5.1 Configuration Menu

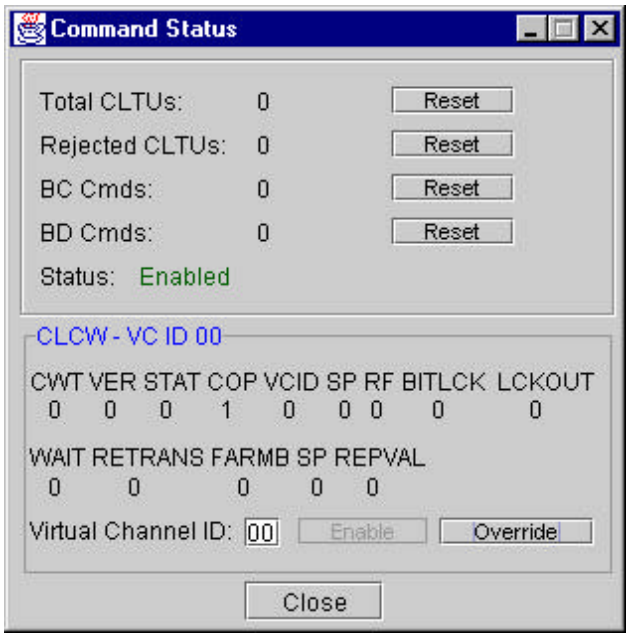
The configuration menu is not enabled for the GenericCmdIngest module.

GenericCmdIngest-5.2 Run-time Menu

The Run-time menu for the GenericCmdIngest module contains the following items.

Menu item	Description
Start	Enable command reception
Stop	Disable command reception
Display Status…	Enable/disable status, command counts, CLCW by virtual channel, edit fields of CLCW
Modify Validation Criteria…	Allows modification of validation options
Display Container Buffer	Display any command container buffer
Setup	Specify SCID, codeblock size, TF CRC

GenericCmdIngest-5.2.1 Display Command Status

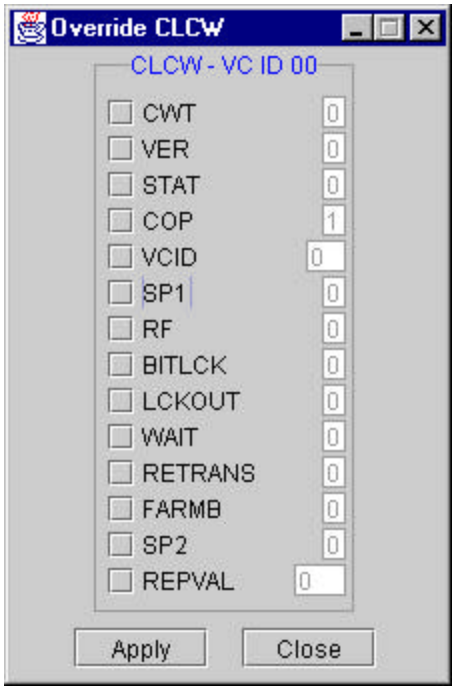


The **Command Status** display shows command counters, command enabled or disabled status and the Command Link Control Word (CLCW). It is necessary to enter the Virtual Channel ID (00 – 63) and click the **Enable** button to identify the CLCW to be displayed. This action will enable the **Override** button.

Click the **Override** button to display the **Override CLCW** panel.

Click the **Reset** button to set the associated counter to zero.

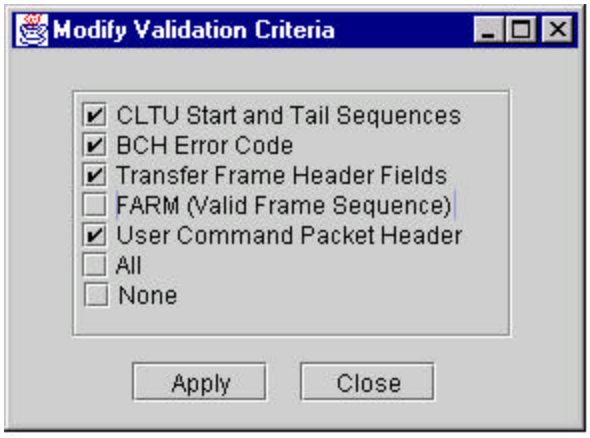
GenericCmdIngest-5.2.1.1 Override CLCW



This display shows the Command Link Control Word (CLCW) for the virtual channel entered on the Command Status panel, 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. Review changes on 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)

GenericCmdIngest-5.2.2 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.

GenericCmdIngest-5.2.2.1 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 must be large enough to contain a transfer frame header and a single byte of data or it will be rejected and an error event message will be generated.
- The virtual channel ID in the transfer frame header must fall within the range of 0 to 63, or the transfer frame will be rejected and an error event message will be generated.

GenericCmdIngest-5.2.2.2 Validation of CLTU Start and Tail Sequences

When CLTU start and tail sequence validation is enabled, the software will report appropriate event error messages if a valid instance of either sequence cannot be located, or if the start sequence is located but the tail sequence is not found.

GenericCmdIngest-5.2.2.3 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.

GenericCmdIngest-5.2.2.4 Transfer Frame Header Validation Option

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

- 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 must contain valid combination of mode flags (consisting of the bypass flag and the control command flag) or the transfer frame is rejected and an error event message is generated. Frame types AD, BC and BD are valid. Frame type AC is invalid .
- The transfer frame header SCID field must match the value in container item GenericCmdSCID or the frame is rejected and an error event message is generated.
- 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.

GenericCmdIngest-5.2.2.5 Farm (Valid Frame Sequence) Validation Option

When FARM validation is enabled, all fields related to the setting of the CLCW are checked. FARM validation consists of the following test:

- 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.

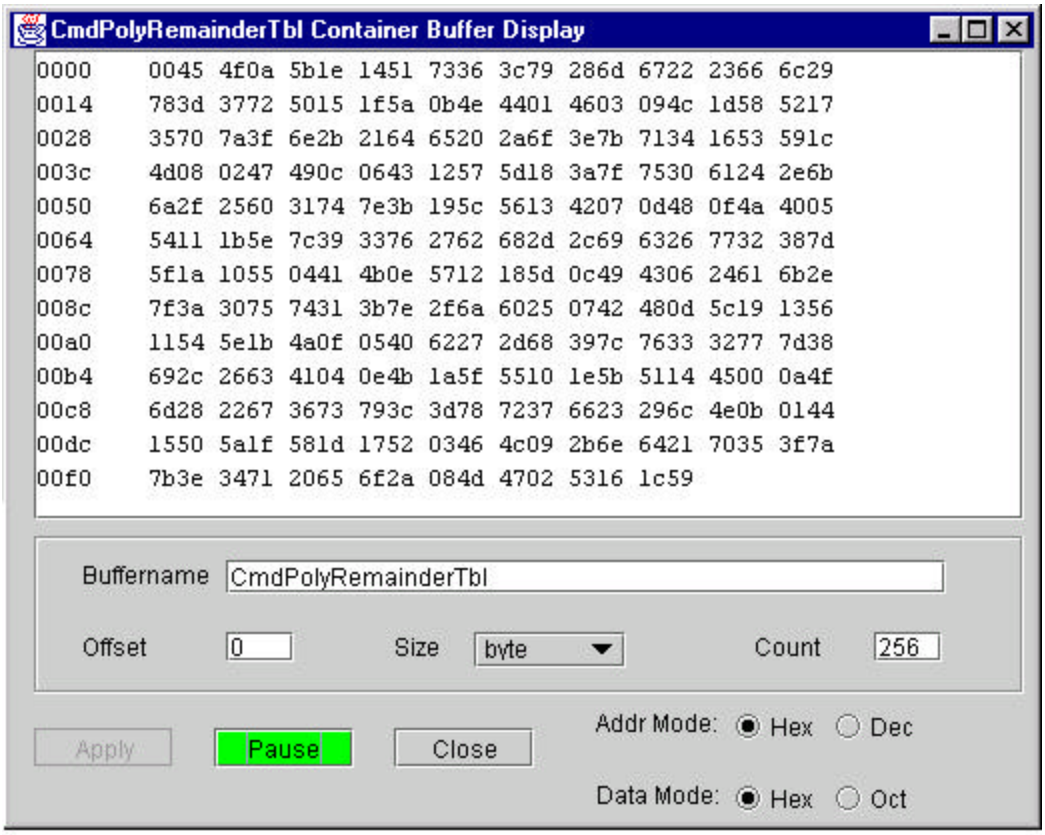
GenericCmdIngest-5.2.2.6 User Command Packet Header Validation Option

When packet validation is enabled, the version, type, sequence flag and packet length fields in the packet header will be validated as specified in the CCSDS Standard. Event messages are generated for every packet that fails the acceptance test.

GenericCmdIngest-5.2.3 Display Container Buffer

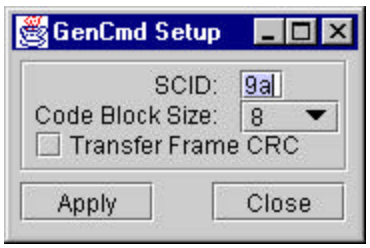
This display may be used to display container buffers that do not have a customized display available for them. Enter the buffer name, offset, size and count and click on the **Apply** button. If there are problems displaying a buffer, error messages are sent to the event message area of the screen.

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. 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 4.3, Command Container Buffers, 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

GenericCmdIngest-5.2.4 Setup



The Setup Display allows the user to enter the SCID in hex, to specify the number of bytes in a codeblock (default = 8) and to indicate the presence or absence of Transfer Frame CRC.

GenericCmdIngest-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.

GenericCmdIngest-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.

Generic Command Validation (GenericCmdValid) Module

GenericCmdValid-1.0 Overview

The CCSDS Generic Command Validation module receives command data in the form of Command Link Transmission Units (CLTUs), validates the CLTUs, then CodeBlock, then Frame generates a command link control word (CLCW) for each virtual channel and transmits these CLCWs to an external destination; also the valid frame (according to Frame Acceptance and Reporting Mechanism-1, FARM-1) is passed to packet processor to validate and then to identify each command with the Command Database.

GenericCmdValid-2.0 Inputs

The GenericCmdValid module has 1 input channel.

Ch.	Data expected	Validation performed	Processing performed
1	CLTU stream	CCSDS Telecommand Standard	Commands are parsed and validated with CCSDS standard. Event messages generated on valid or invalid commands.

GenericCmdValid-3.0 Outputs

The GenericCmdValid module has 1 output channel.

Ch.	Description
1	Command Link Control Words (CLCWs)

GenericCmdValid-4.0 Container Items

This module accepts operator directives. 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. The container items for command processing fall into several groups: mission specific items, validation fields and processing buffers. The initial values for counts, flags and buffers are zero. Non-trivial initial values for other items are shown in parentheses following the description.

GenericCmdValid-4.1 Mission Specific Container Items

The initial values for these items are based on the PM-1 spacecraft. Values may be changed using set or setBuffer directives from the directive line.

Name	Type	Description
GenericCmdValidCltuCodeblockSize	Fixed	Codeblock size in bytes (8)
GenericCmdValidCltuExpectedStartSequence	Buffer	Expected CLTU start sequence buffer (2 bytes) ($EB90_{16}$)

GenericCmdValidCltuExpectedTailSequence	Buffer	Expected CLTU tail sequence buffer (8 bytes) ($C5C5\ C5C5\ C5C5\ C579_{16}$)
GenericCmdValidSCID	Fixed	Spacecraft Identifier ($9A_{16}$)
GenericCmdValidSlidingWindowSize	Fixed	FARM-1 Sliding Window Size (180)

GenericCmdValid-4.2 Command Validation Container Fields/Flags

Name	Type	Description
GenericCmdValidEnabled	Fixed	Command processing enabled flag. (0=disabled, 1=enabled)
GenericCmdValidCLTUValidation	Fixed	Command CLTU validation enabled flag. (0=disabled, 1=enabled)
GenericCmdValidCodeblockValidation	Fixed	Command codeblock parity validation enabled flag. (0=disabled, 1=enabled)
GenericCmdValidFrameValidation	Fixed	Command transfer frame header validation enabled flag. (0=disabled, 1=enabled)
GenericCmdValidFARMValidation	Fixed	Command Frame Acceptance and Reporting Mechanism (FARM) validation enabled flag. (0=disabled, 1=enabled)
GenericCmdValidPacketValidation	Fixed	Command packet validation enabled flag. (0=disabled, 1=enabled)
GenericCmdValidDebugEnabled	Fixed	Global command subsystem debug messages flag. (0=disabled, 1=enabled) When enabled, additional event messages are generated throughout command ingest processing.
GenericCmdValidPacketDumpEnabled	Fixed	Packet Dump Enabled flag (0=disabled, 1=enabled). When enabled, the contents of valid type AD commands will be displayed in event messages
GenericCmdValidClcw<##>CWT	Fixed	CLCW Control Word Type for VCID <##>
GenericCmdValidClcw<##>Version	Fixed	CLCW Version for VCID <##>
GenericCmdValidClcw<##>Status	Fixed	CLCW Status for VCID <##>
GenericCmdValidClcw<##>COP	Fixed	CLCW COP In Effect for VCID <##>
GenericCmdValidClcw<##>VCID	Fixed	CLCW VCID (0) for VCID <##>
GenericCmdValidClcw<##>Spare1	Fixed	CLCW Spare field 1 for VCID <##>
GenericCmdValidClcw<##>NoRFAvail	Fixed	CLCW No RF Avail Flag for VCID <##>
GenericCmdValidClcw<##>NoBitLock	Fixed	CLCW No Bit Lock Flag for VCID <##>

GenericCmdValidClcw<##>Lockout	Fixed	CLCW Lockout Flag for VCID <##>
GenericCmdValidClcw<##>Wait	Fixed	CLCW Wait Flag for VCID <##>
GenericCmdValidClcw<##>Retransmit	Fixed	CLCW Retransmit Flag for VCID <##>
GenericCmdValidClcw<##>FarmCount	Fixed	CLCW Farm-B Counter for VCID <##>
GenericCmdValidClcw<##>Spare2	Fixed	CLCW Spare field 2 for VCID <##>
GenericCmdValidClcw<##>Report	Fixed	CLCW Report Value for VCID <##>

GenericCmdValid -4.3 Command Container Buffers

Name	Type	Description
GenericCmdValidPolyRemainderTbl	Buffer	Polynomial remainder table for parity calculation. (256 bytes)
GenericCmdValid<##>CLCW	Buffer	CLCW buffer (4 bytes) for VCID <##>
GenericCmdValid<##>CmdData	Buffer	Command Data buffer for VCID <##>
GenericCmdValid<##>Pkt	Buffer	Command packet buffer for VCID <##>
GenericCmdValid<##>FarmBuffer	Buffer	Command FARM buffer (1024 bytes) for VCID <##>
GenericCmdValidFrameBuffer	Buffer	Command transfer frame buffer (1024 bytes)
GenericCmdValidCLTUBuffer	Buffer	Command link transmission unit buffer (6000 bytes)
GenericCmdValidCodeblock	Buffer	Compressed codeblock buffer (holds codeblock bytes without parity bytes) (6000 bytes)

GenericCmdValid-4.4 Command Container Counters

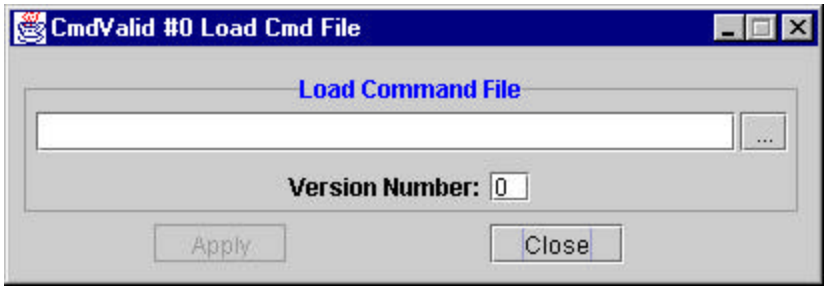
Name	Type	Description
GenericCmdValidTotalCLTUs	Fixed	Count of all CLTUs received
GenericCmdValidRejectCLTUs	Fixed	Count of invalid CLTUs
GenericCmdValidADFrames	Fixed	Count of Type AD Transfer Frames
GenericCmdValidACFrames	Fixed	Count of Type AC Transfer Frames
GenericCmdValidBCCMDS	Fixed	Count of Type BC Transfer Frames
GenericCmdValidBDCMDS	Fixed	Count of Type BD Transfer Frames

GenericCmdValid-5.0 Displays

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

Module Pop-Up Menu Item	Description
Load Cmd File	To load a command database flat file
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

GenericCmdValid-5.1 Load Cmd (Command Database) File Menu



A user can enter a command database file name by either type in or browse in, enter a version number if it is not 0.

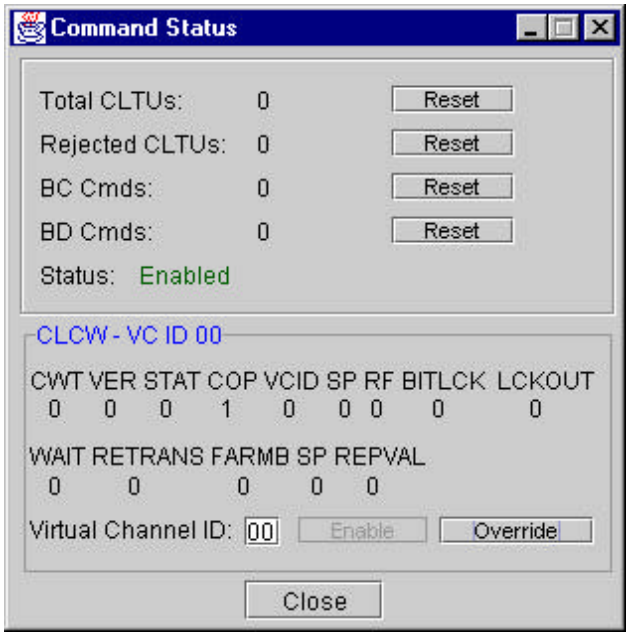
Note: GenericCmdValid module will act like Generic Command Ingest (command recognition will not be performed) if a user does not load a command database file.

GenericCmdValid-5.2 Run-time Menu

The Run-time menu for the GenericCmdValid module contains the following items.

Menu item	Description
Display Status	Enable/disable status, command counts, CLCW by virtual channel, edit fields of CLCW
Modify Validation Criteria	Allows modification of validation options
Setup	Specify SCID, codeblock size, has segment header flag

GenericCmdValid-5.2.1 Display Command Status

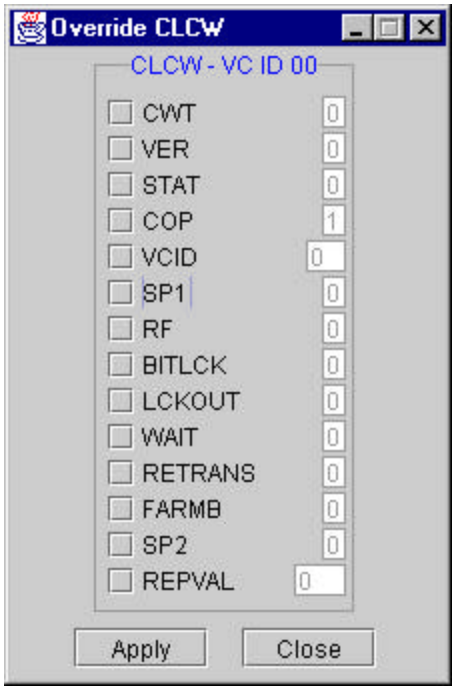


The Command Status display shows command counters, command enabled or disabled status and the Command Link Control Word (CLCW). It is necessary to enter the Virtual Channel ID (00 – 63) and click the **Enable** button to identify the CLCW to be displayed. This action will enable the **Override** button.

Click the **Override** button to display the Override CLCW panel.

Click the **Reset** button to set the associated counter to zero.

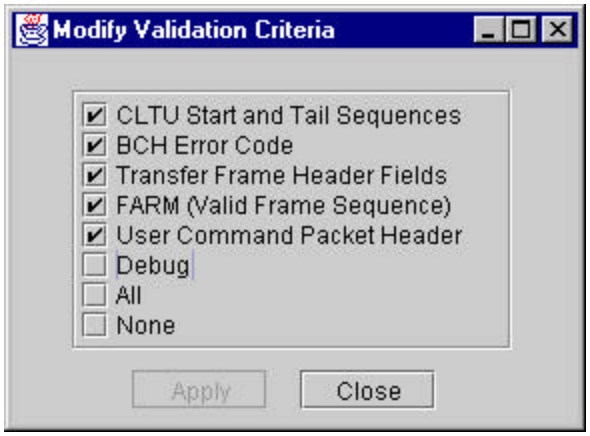
GenericCmdValid-5.2.1.1 Override CLCW



This display shows the Command Link Control Word (CLCW) for the virtual channel entered on the Command Status panel, 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. Review changes on 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)

GenericCmdValid-5.2.2 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.

GenericCmdValid-5.2.2.1 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 must be large enough to contain a transfer frame header and a single byte of data or it will be rejected and an error event message will be generated.
- The virtual channel ID in the transfer frame header must fall within the range of 0 to 63, or the transfer frame will be rejected and an error event message will be generated.

GenericCmdValid-5.2.2.2 Validation of CLTU Start and Tail Sequences

When CLTU start and tail sequence validation is enabled, the software will report appropriate event error messages if a valid instance of either sequence cannot be located, or if the start sequence is located but the tail sequence is not found.

GenericCmdValid-5.2.2.3 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.

GenericCmdValid-5.2.2.4 Transfer Frame Header Validation Option

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

- 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 must contain valid combination of mode flags (consisting of the bypass flag and the control command flag) or the transfer frame is rejected and an error event message is generated. Frame types AD, BC and BD are valid. Frame type AC is invalid .
- The transfer frame header SCID field must match the value in container item GenericCmdSCID or the frame is rejected and an error event message is generated.
- 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.

GenericCmdValid-5.2.2.5 Farm (Valid Frame Sequence) Validation Option

When FARM validation is enabled, all fields related to the setting of the CLCW are checked. FARM validation consists of the following test:

- 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.

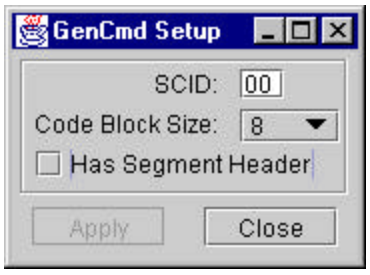
GenericCmdValid-5.2.2.6 User Command Packet Header Validation Option

When packet validation is enabled, the version, type, sequence flag and packet length fields in the packet header will be validated as specified in the CCSDS Standard. Event messages are generated for every packet that fails the acceptance test.

GenericCmdValid-5.2.2.7 Debug Option

When the debug flag is enabled, additional debug event messages will be generated during processing.

GenericCmdValid-5.2.3 Setup



The Setup Display allows the user to enter the SCID in hex, to specify the number of bytes in a codeblock (default = 8) and to indicate the presence or absence of Segment Header.

GenericCmdValid-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.

GenericCmdValid-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.

Generic Telemetry Module

GenTlm-1.0 Overview

The functions of the Generic Telemetry (GenTlm) module include generation and transmission of formatted telemetry. This module is also capable of displaying telemetry packets under operator direction. The GenTlm module uses database files generated by the ASCII to Binary (ascii2bin) Database Utility to define its CCSDS telemetry streams.

GenTlm-2.0 Inputs

Channel	Description
1	Updated CLCWs from the GenericCmdIngest module.

This module will use the command virtual channel id from the incoming CLCW to determine which telemetry virtual channel should receive the updated CLCW. Refer to Section 2.6, VCDU Record, in the Ascii2Bin User's Guide for a discussion of the Command Channel field and how it is used to associate the CLCW for a command virtual channel with a telemetry virtual channel. In addition, an input link to the GenTlm module's channel 1 may be connected from a Scenario module to support scenario directives. A set of database files must be configured prior to run-time.

GenTlm-3.0 Outputs

Channel	Description
1	CCSDS AOS formatted VCDUs containing telemetry packets
2	CCSDS AOS formatted VCDUs containing telemetry packets
3	CCSDS AOS formatted VCDUs containing telemetry packets

GenTlm-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.

GenTlm-4.1 Physical Channel Items

There 3 copies of the following items, one for each physical channel. From the following table substitute the output channel number (1, 2, or 3) for the "<#>" to get the item's name. For example the transmission count for channel 1 is "GenericTlmCh1XmitCount".

Name	Type	Description
GenericTlmCh<#>XmitCount	Fixed	Channel transmission count
GenericTlmEnabledCh<#>	Fixed	Channel enabled flag (0=disabled, 1=enabled)
GenericTlmInterval<#>	Fixed	IP interval in seconds for automatic sending

GenericTlmSCID<#>	Fixed	Spacecraft ID for channel
GenericTlmSerialInterval<#>	Fixed	Serial interval in seconds for automatic sending
GenericTlmVCDULength<#>	Fixed	Length of VCDU in bytes
GenericTlmVCDUType<#>	Fixed	Type of VCDUs to transmit (0=not full, 1=full)

GenTlm-4.2 Application Packet Items

For each APID in telemetry there is a container buffer to hold that APID's packet. For example, the container buffer for APID 397 is named "TlmPacket0397". Where "<APID>" appears in the following table, substitute the APID in 4 decimal digits.

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>Enabled	Fixed	Transmission flag (0=disabled, 1=enabled)
TlmPacket<APID>Interval	Fixed	Transmission interval in seconds

GenTlm-5.0 Displays

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

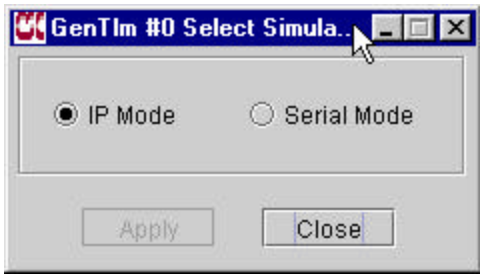
Module Pop-Up Menu Item	Description
Configure	Access the configuration menu
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

GenTlm-5.1 Configuration Menu

Clicking the "Configure" option from the module pop-up menu in the project window

Configuration Menu	Description
Select Simulation Mode	Configure IP or Serial mode
Configure DB	Specify set of database files

GenTlm-5.1.1 Select Simulation Mode

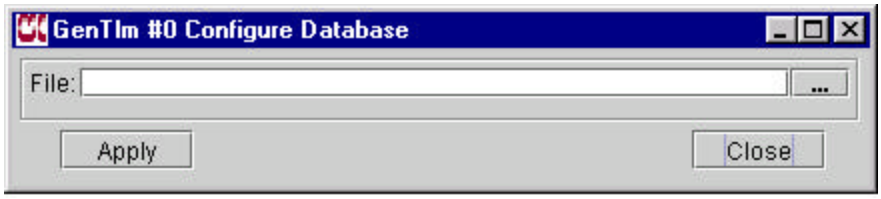


Select "Select Simulation Mode" from the configuration menu to access this window:

The user may select telemetry transmission via IP or serial mode. IP mode transmits packets. Serial mode transmits VCDUs.

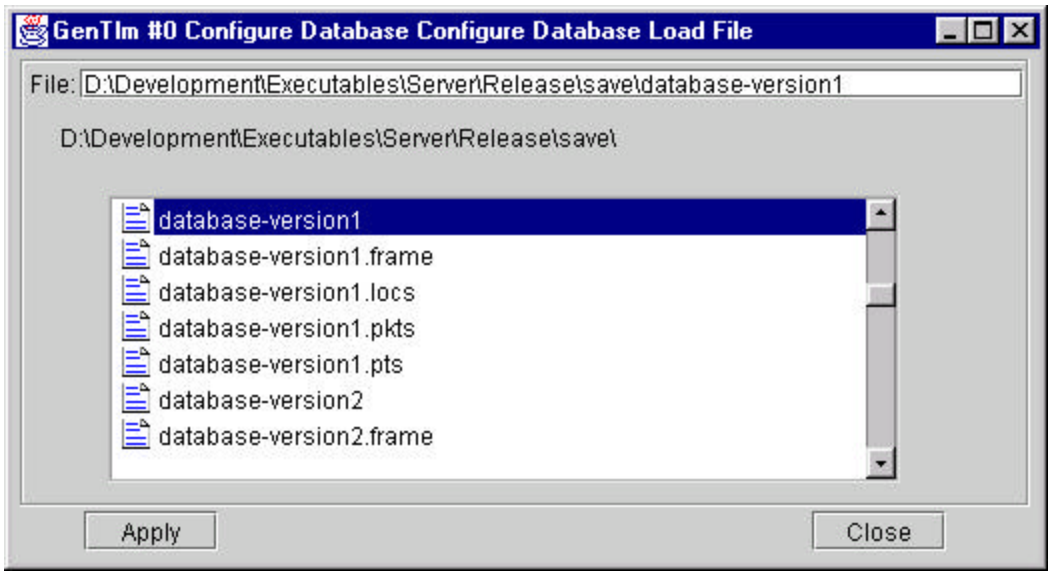
GenTlm-5.1.2 Configure Database

Select "Configure DB" from the configuration menu to access the following window:



The user must now specify the telemetry database to be used. Typically this file name does not end with ".frame", ".pkts", ".pts", ".locs" or ".txt". The file name without these extensions specifies this group of files as a database.

The button to the right of the File field is the file-browsing button. Clicking the file-browsing button produces a file selection screen such as the following:



This screen may be used to navigate to the directory where the database files are stored. In this example, the "database-version1" set of database files is being selected. The **Apply** button must be used to complete the database selection process.

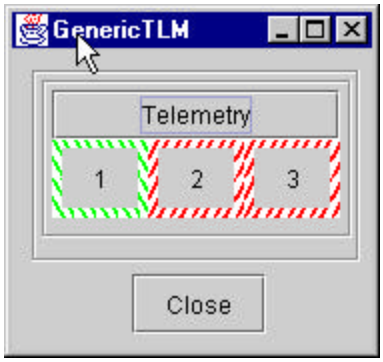
If there are any problems accessing the requested database, or if the "Configure DB" step was skipped, error event messages will be sent to the event message region of the project window.

GenTlm-5.2 Run-time Menu

Run-time Menu Item	Description
Control	Request the main display
DB Pkt Display	Request the Database Packet display
DB Summary Display	Request the Database Summary display
Stop	Stop the module
Restart	Restart the module

GenTlm-5.2.1 Main Display

Clicking the "Control" option of the run-time menu causes the main display to appear.



The main display for the GenTlm module provides access to run-time operations and information about the enabled or disabled status of telemetry transmission. The latter is indicated by colored diagonal stripes around the appropriate boxes. The telemetry channel boxes are numbered for each telemetry output channel. Red stripes angling up and to the right indicate disabled, while green stripes angling down and to the right indicate enabled.

There are four menu buttons on the main display: **Telemetry** and channel buttons **1, 2,** and **3**.

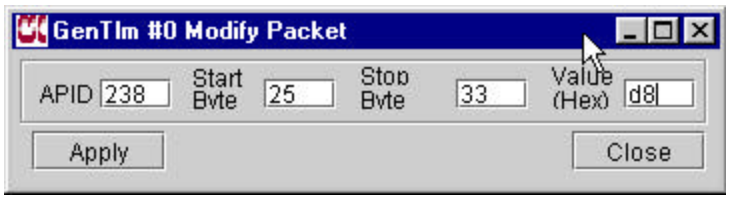
GenTlm-5.2.1.1 Main Display Telemetry Menu

Clicking the **Telemetry** button on the main display accesses the following menu.

Telemetry Menu Item	Description
Modify Packet…	Modify any byte in any telemetry packet
Display Packet…	Display packet header and contents
Display Status	Display Telemetry Status
Control Packet…	Enable/disable packet transmission
APID Status	Display Telemetry Status by APID
Container Items	Display/Set Container Items
Container Buffer	Display Container Buffer

GenTlm-5.2.1.1.1 Modify Packet Display

The modify packet display allows the user to select an APID and to modify a range of bytes. Bytes in the packet header may be overwritten by automatic formatting updates.

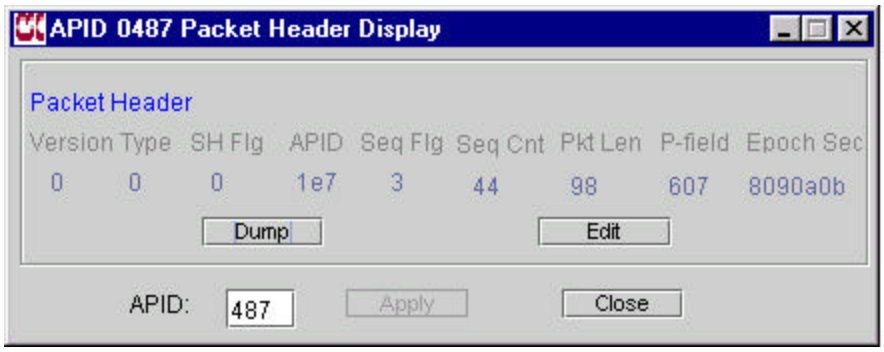


The user must enter a valid APID value in the APID field and a valid range then click the **Apply** button before any values will be changed.

Data Entry Field	Description
APID	Application identifier in decimal (packet number)
Start Byte	Decimal byte location in packet to start modification, 0 indicates first byte of packet header
Stop Byte	Decimal byte location in packet to stop modification, inclusive
Value	Hexadecimal value to store in the specified bytes

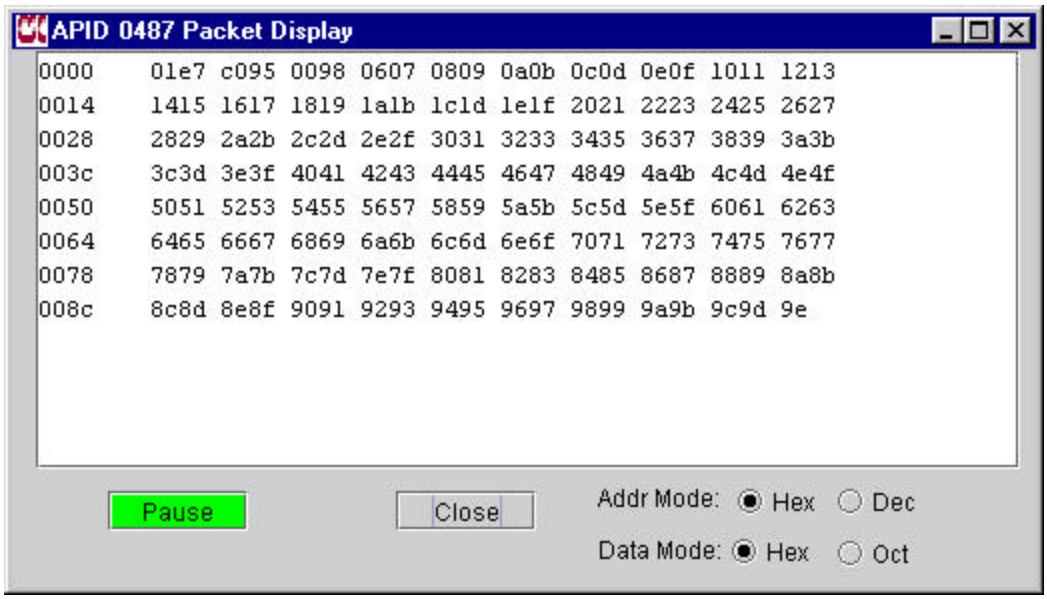
GenTlm-5.2.1.1.2 Telemetry Packet Display

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.



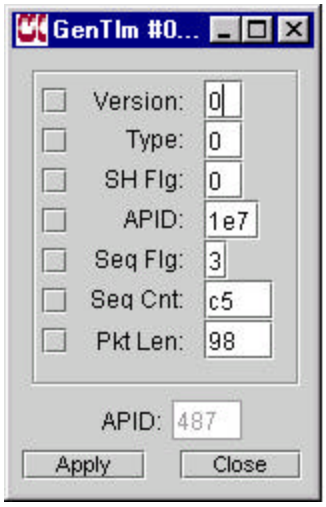
The user must enter a valid APID value in the APID field and then click the **Apply** button before any other values will be reported.

GenTlm-5.2.1.1.2.1 Telemetry Packet Display Dump Button



This is a standard dynamically updating SIMSS module dump display.

GenTlm-5.2.1.1.2.2 Telemetry Packet Display Edit Button



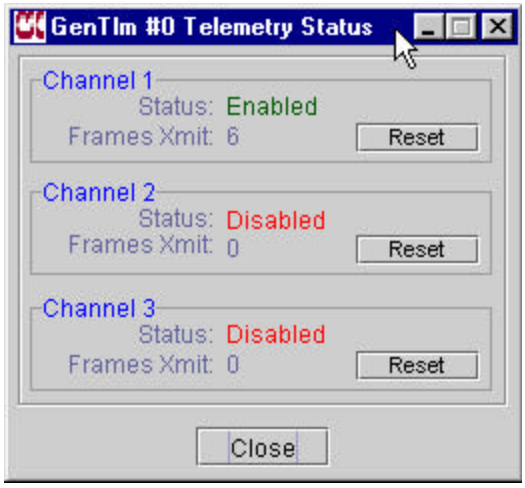
Clicking the **Edit** button on the Packet Header Display will bring up the Modify Tlm Packet Header Display. This display allows the user to modify fields in the packet header (see the following table for a description of the fields). Click on an enable box to the left of the field to enable modification of a specific header field.

Clicking the **Apply** button puts the settings into effect.

Clicking the **Close** button exits the display.

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)

GenTlm-5.2.1.1.3 Telemetry Status Display

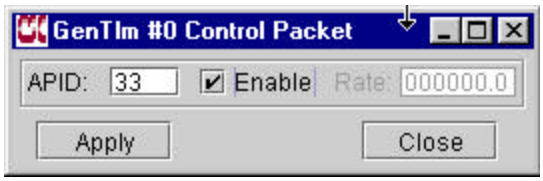


The telemetry status window allows the user to monitor the number of VCDUs or Frames that have been transmitted per channel. The user may reset the number of VCDUs or Frames per channel

Status Field	Description
Status	Enabled or disabled
Frames Xmit	Total number of VCDUs or packets transmitted on this output channel
Reset	Set total number of VCDUs or packets transmitted back to zero

GenTlm-5.2.1.1.4 Control Packet Display

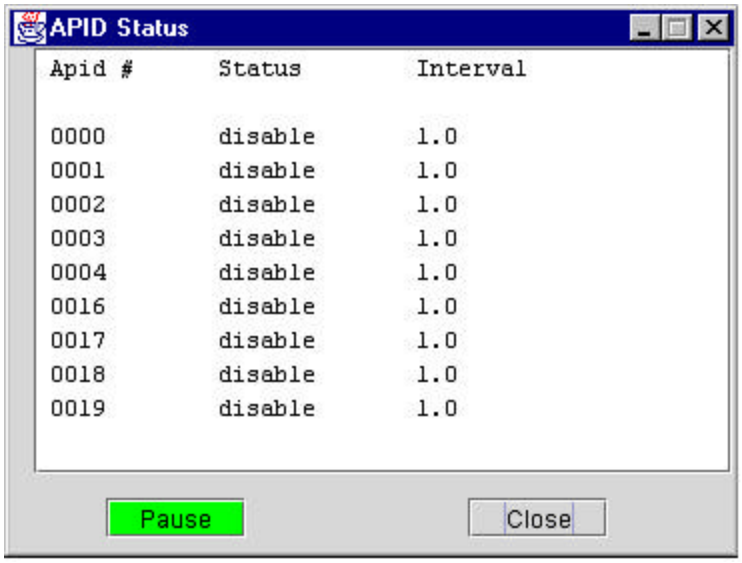
The control packet window allows the user to enable or disable a packet for transmission.



Enter a valid APID and then check the Enable box to enable a packet's transmission. Clear this box to disable transmission. The **Apply** button must be used for the change to take effect.

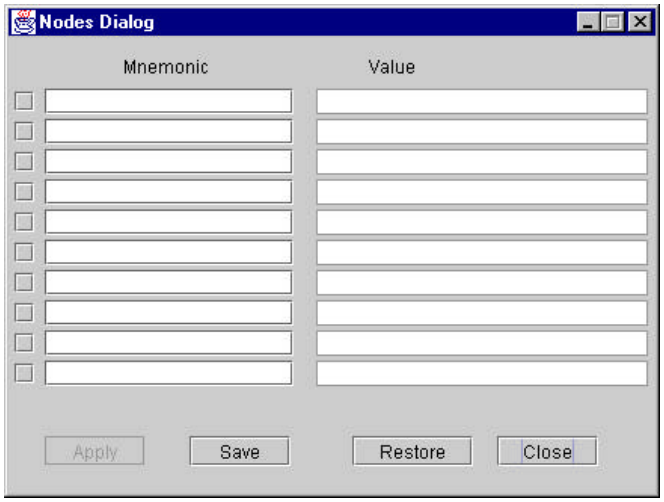
Data Entry Field	Description
APID	Application identifier (packet number)
Enable	Selected means to transmit the packet, unselected means to not transmit the packet

GenTlm-5.2.1.1.5 APID Packet Status Display



This dynamically updating display provides a scrollable list of the current status of all telemetry packets created from the project database. The APID number is shown in decimal. The interval between packet transmissions is in seconds. Click on the **Pause** button to interrupt the display's updates. Click on the same button to resume status updates. Use the **Close** button to dismiss the display.

GenTlm-5.2.1.1.6 Display/Set Container Items

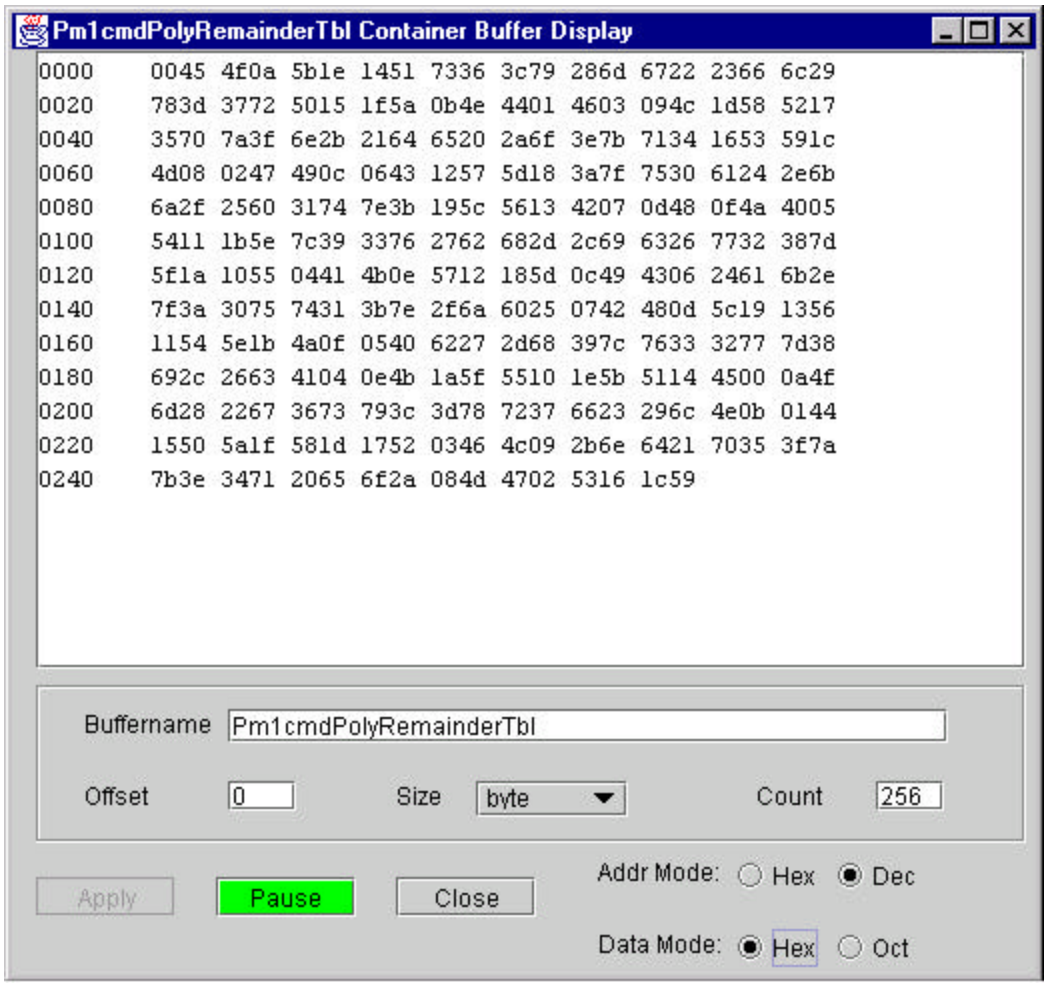


This display allows the user to view and modify individual items from the module's container. Since telemetry points are stored as container items, telemetry points may be monitored and overwritten with this display. The first two items in the sample screen are telemetry points. To request an item for 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.

Field	Description
Checkbox	Click on this box to modify an item's value.
Mnemonic	This field identifies a container item for display. 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.
Save Button	The list of container item names may be saved to a file. A file selection screen will appear when this button is used.
Restore Button	The list of container item names may be loaded from a previously saved file. A file selection screen will appear when this button is used.

GenTlm-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. 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 off screen.) The button's name is changed to **Cont**. To resume updates of the buffer, press the **Cont** button. 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.

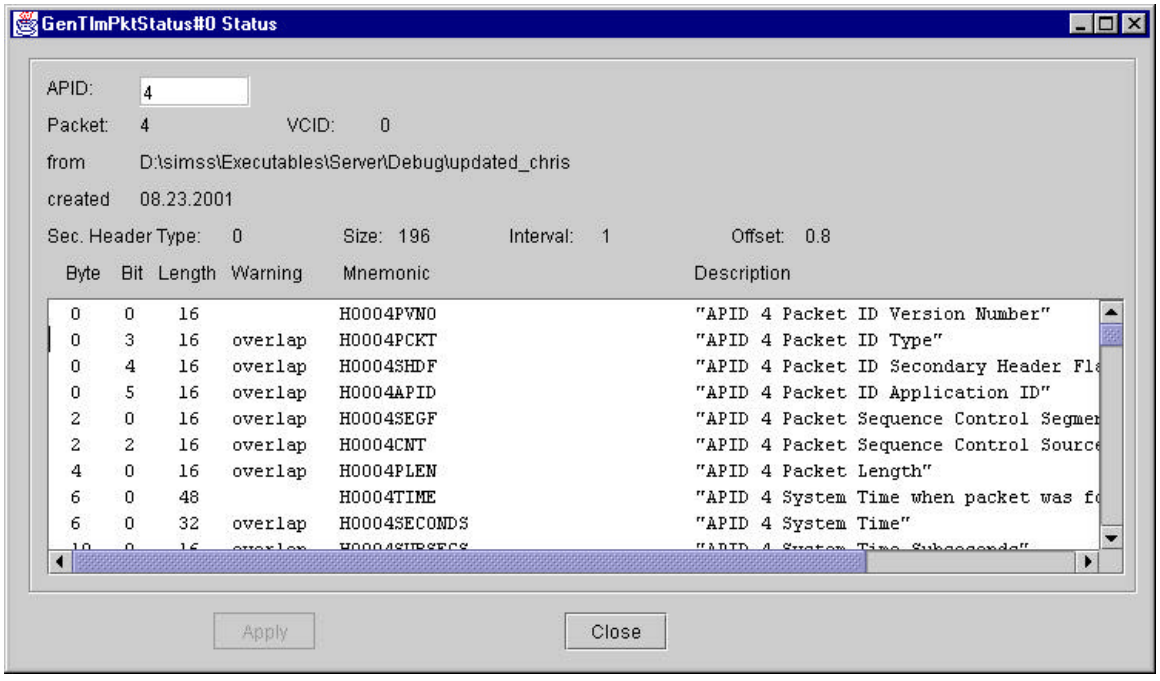
Field	Description
Buffername	This field identifies the container buffer for display.
Offset	This field specifies the first byte to be displayed. This display is limited to 1400 data bytes at a time. This field is used with the count field to specify the portion of the buffer to display.
Size	Size of 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

GenTlm-5.2.1.2 Main Display Channel Menus

Click on a numbered button for the following choices for the associated physical channel.

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

GenTlm-5.2.2 DB Pkt Display

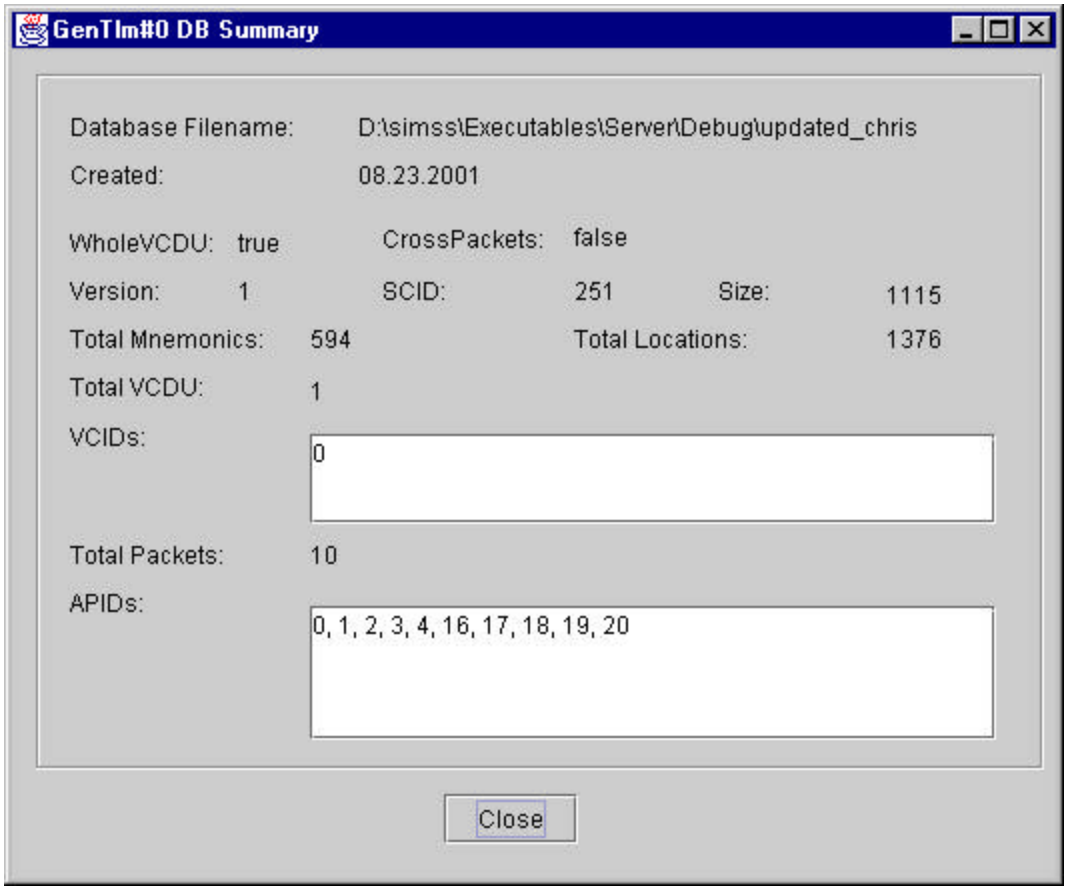


DB Pkt Display Item	Description
APID	The APID to display.
Packet	The APID currently displayed.
VCID	The VCID that the APID is in.
From	The current database name.
Created	The date the current database was created/modified.
Sec. Header Type	Not currently implemented
Size	The APID size (bytes).
Interval	The APID interval (secs). This is the amount of time between the transmissions of the packet.
Offset	The APID offset (secs). This is the amount of time waited before transmitting the current APID after starting the data.

Each packet has a bit breakdown explaining what is in the packet:

Bit Breakdown Item	Description
Byte	Starting byte for the breakdown line (counting from 0).
Bit	The first bit for the breakdown line (counting from 0).
Length	The length (in bits) of the breakdown line.
Warning	If more than one mnemonic stakes claim to any bit, a warning message will appear.
Mnemonic	The mnemonic name. A value of "---" means no mnemonics were assigned to these bits.
Description	The description of the mnemonic (taken from the database).

GenTlm-5.2.3 DB Summary Display



DB Summary Item	Description
Database Filename	The file name for the current database.
Created	The date the current database was created/modified.
WholeVCDU	If this Whole VCDU flag is set to true, the module will not send out a non-fill VCDU unless it is full of data.
CrossPackets	If this Cross Packets flag is set to true any packet will be allowed to cross the VCDU boundary (be split between VCDUs).
Version	The version that will appear in the VCDU primary header. Currently, only version 1 (VCDU) is supported.
SCID	The spacecraft ID that will appear in the primary header.
Size	The size of a frame.
Total Mnemonics	The number of mnemonics defined by the current database.
Total Locations	The number of locations defined by the current database.
Total VCDU	The number of VCDUs defined by the current database.
VCIDs	A list of the VCIDs defined by the current database.
Total Packets	The number of packets defined by the current database.
APIDs	A list of the APIDs defined by the current database.

GenTlm-6.0 Special Operating Instructions

Please read the chapter about the ASCII to Binary (ascii2bin) Database Utility for a complete description of the ASCII database fields. Not all of the database fields are fully implemented in the Generic Telemetry module for this release.

The value in the VERSION record must be specified as 1 to indicate VCDU data.

In the VCDU record, the following fields are not implemented:

Header Error Control
Insert Size
Op Control

When the VCDU record's command channel flag is true, the resulting command link control word (CLCW) is static. It contains an indication of COP-1 and the command virtual channel will match the VCDU virtual channel. The user cannot modify the CLCW field.

In the PACKET record, the Secondary Header Type determines whether the packets will contain secondary headers, plus the value of the Secondary Header Flag in the primary packet header. The user may also set the Secondary Header Flag using the Modify Packet Header Display. Currently, only one secondary header format is supported (Secondary Header Type = 1). This format is an 8 byte CUC with 1 byte pfield, 1 byte pfield extension, 4 bytes coarse time and 2 bytes fine time. The epoch is 1/1/1958. If the Secondary Header Type = 0, the packet does not contain secondary headers and the Secondary Header Flag in the primary packet header is set to 0.

The SKEY record has not been implemented at this time.

GmsecInput Module

GmsecInput -1.0 Overview

The Goddard Mission Services Evolution Center (GMSEC) Input (GmsecInput) module subscribes to data from a *SmartSockets* server.

GmsecInput -2.0 Outputs

The GmsecInput module has an output channel, which is listed below.

Channel	Data sent	Validation performed	Processing performed
1	Data Buffer	None	Data is received from an upstream module, processed and sent to a downstream module.

GmsecInput -3.0 Inputs

The GmsecInput module does not take input from any SIMSS modules. The input for GmsecInput must come from a *SmartSockets* server running locally.

GmsecInput -4.0 Container Items

GmsecInput does not have any containers that effect the operation of this module.

GmsecInput -5.0 Displays

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

GmsecInput -5.1 Configuration

The GmsecInput module currently does not allow any configuration.

GmsecInput -5.2 Run-time

When running, GmsecInput will subscribe to a *SmartSockets* server. The server must be running on the local machine. The directory subscribed to is "/spacecraft/telemetry".

GmsecOutput Module

GmsecOutput -1.0 Overview

The Goddard Mission Services Evolution Center (GMSEC) Output (GmsecOutput) module sends data to a *SmartSockets* server.

GmsecOutput -2.0 Inputs

The GmsecOutput module has an input channel, which is listed below.

Channel	Data expected	Validation performed	Processing performed
1	Data Buffer	None	Data is received from an upstream module, processed and sent to a downstream module.

GmsecOutput -3.0 Outputs

The GmsecOutput module does not send any data to other SIMSS modules. It does send data to a *SmartSockets* server, which must be running locally.

GmsecOutput -4.0 Container Items

GmsecOutput does not have any container items that effect the operation of this module.

GmsecOutput -5.0 Displays

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

GmsecOutput -5.1 Configuration

The GmsecOutput module currently does not allow any configuration.

GmsecOutput -5.2 Run-time

When running, GmsecOutput will publish all data it receives to a *SmartSockets* server. The server must be running on the local machine. All data will be published to the "/spacecraft/telemetry" directory.

GOES-N Checksum Module

GOES-N Checksum-1.0 Overview

The GOES-N Checksum module receives a GOES-N telemetry stream. A checksum is performed and command verifiers are updated.

This module was designed to meet GOES-N requirements; however, it should be usable to support similar spacecraft if necessary.

GOES-N Checksum-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	GOES-N Telemetry Stream	GOES-N specific	Checksum computed. Command counters updated. Telemetry verification of commands updated.
1	9002 message from GOES-N Command Ingest Module	None	Telemetry is modified based on the commands received from GOES-N Command Ingest.

GOES-N Checksum-3.0 Outputs

Channel	Description
1	Modified GOES-N telemetry stream.

GOES-N Checksum-4.0 Container Items

Name	Type	Description
GOESNChecksumPerform	Int	Flag to determine if checksum should be calculated
GOESNChecksumFrameSize	Int	Frame size
GOESNChecksumBcRt	Int	Dealing with BC or RT
GOESNChecksumParam	Int	Which parameter to change
GOESNChecksumValue	Int	What the new value is
GOESNChecksumPersistant	Int	Persistency of user overwrite
GOESNChecksumOutBuffer	Int	Output buffer

GOES-N Checksum-5.0 Displays

To access displays for this module, click in the center of the command ingest module in the project window. The following pop-up menu choices will appear. The “Remove” option can be used during project design to remove this 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.

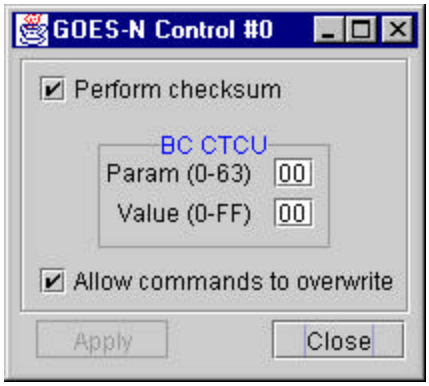
Module Pop-Up Menu Item	Description
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display Information about the module

GOES-N Checksum-5.1 RunTime Displays

Clicking the "Run-time" option of the GOES-N Checksum module pop-up menu produces a Run-time menu. There are three items in the menu list.

Run-time Menu Item	Description
Control	Show control information
Display Output Buffer	Dump incoming command stream
Resume	Resume the GOES-N Checksum module after a pause
Pause	Pause the GOES-N Checksum module

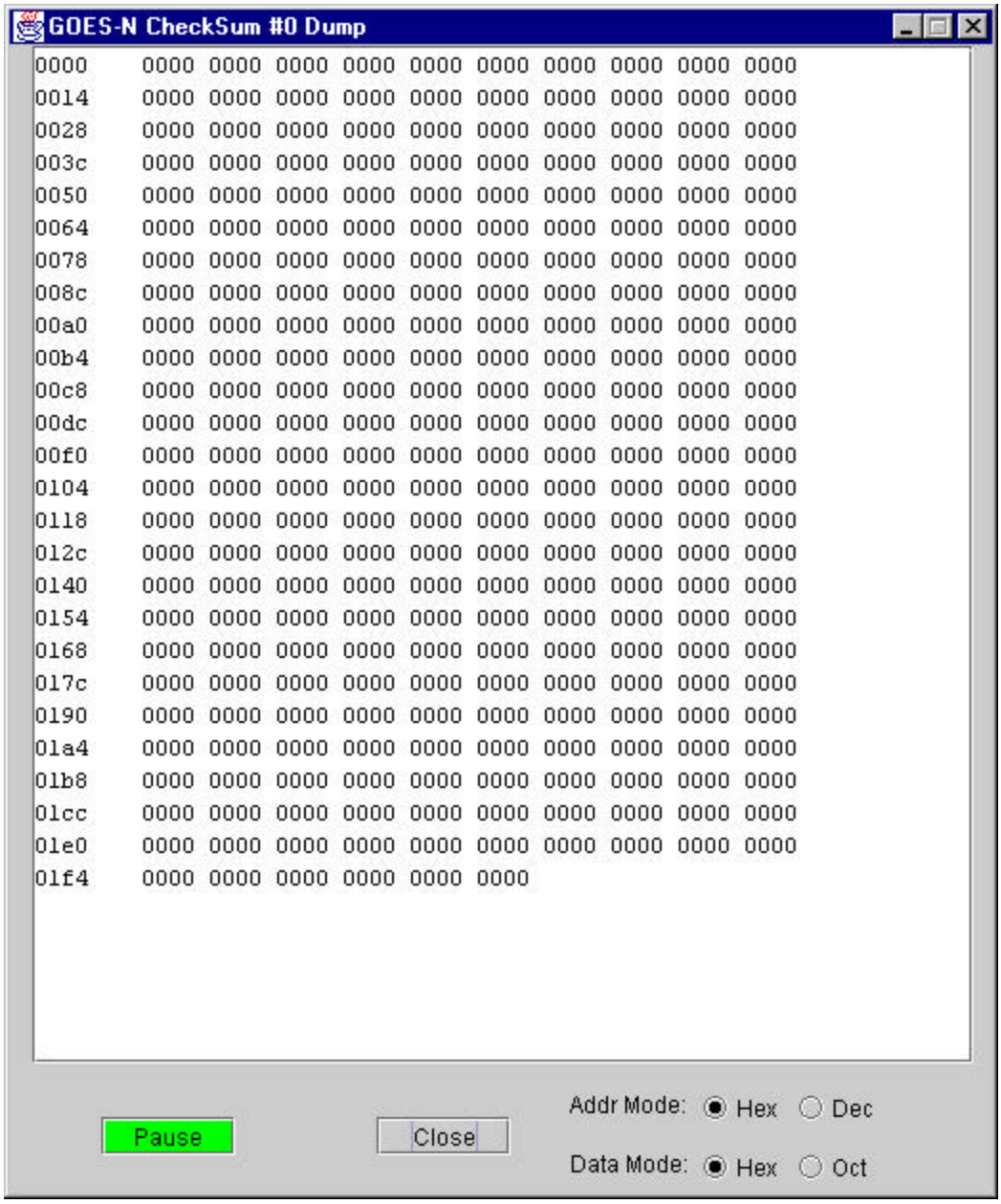
GOES-N Checksum-5.1.1 Status Display



Field	Description
Perform Checksum	Perform minor frame checksum on telemetry.
BC CTCU	Change parameters by hand.
Allow commands to overwrite	Clicking this will allow incoming commands to overwrite the parameters changed by hand.

GOES-N Checksum-5.1.2 Buffer Display

The command stream buffer display shows the contents of the outgoing telemetry stream.



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 format.

GOES-N Command Ingest Module

GOES-N Command Ingest-1.0 Overview

The GOES-N Command Ingest module receives a stream commands then performs the following validation checks on this stream:

- Command word format, including barker (or sync) code.
- Spacecraft identifier (or address)
- Hamming (or Polynomial) code

This module was designed to meet GOES-N requirements; however, it should be usable to support similar spacecraft if necessary.

GOES-N Command Ingest-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Command word	GOES-N specific	Commands parsed and validated. Event messages generated on valid or invalid commands.

GOES-N Command Ingest-3.0 Outputs

Channel	Description
1	9002 message describing which bytes in telemetry should be changed (based on command). The output of GOES-N Command Ingest module should be one of the inputs for the GOES-N Checksum module.

GOES-N Command Ingest-4.0 Container Items

Name	Type	Description
GOESNCMDIngestValidate	UnsByte	Flag to valid a command word
GOESNCMDIngestConvert	UnsByte	Flag to convert from NRZM to NRZL. This functionality is no available.
GOESNCMDIngestUpdate	UnsByte	Flag to update command counter
GOESNCMDIngestDebug	UnsByte	Flag to debug
GOESNCMDIngestSCID	UnsByte	Valid spacecraft identifier
GOESNCMDIngestReceived	UnsDword	Count number of total command words that are received
GOESNCMDIngestValid	UnsDword	Count number of total valid command words
GOESNCMDIngestErrors	UnsDword	Count number of total error command words
GOESNCMDIngestRejected	UnsDword	Count number of total command words that are rejected
GOESNCMDIngestInBuffer	Container Buffer	Buffer that holds incoming command stream

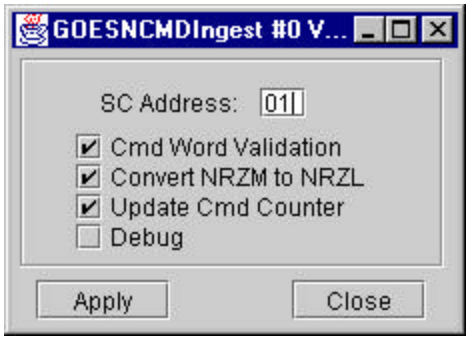
GOES-N Command Ingest-5.0 Displays

To access displays for this module, click in the center of the command ingest module in the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove this 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 Information about the module

GOES-N Command Ingest-5.1 Configuration Displays

Select the "Configure" pop-up menu option for a display similar to the following screen.



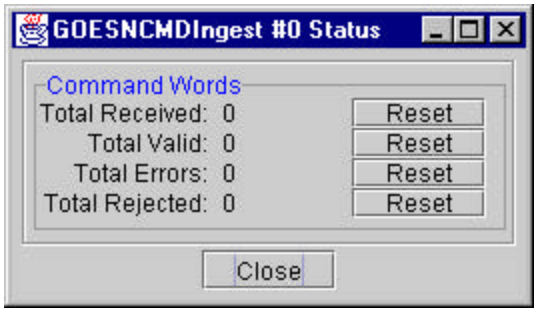
Field	Description
SC Address	Enter the valid spacecraft address
Cmd Word Validation	Check to enable/Clear to disable command word validation
Convert NRZM to NRZL	This is disabled (will be removed in future)
Update Cmd Counter	Enable/Disable the command counter updates
Debug	Enabled/Disabled show debug messages on event log window

GOES-N Command Ingest-5.2 Run-Time Displays

Click the "Run-time" option of the module pop-up menu to see the Run-time menu.

Run-time Menu Item	Description
Display Status	Show status information
Display Input Buffer	Dump incoming command stream
Resume	Resume the GOES-N Command Ingest module after a pause
Pause	Pause the GOES-N Command Ingest module

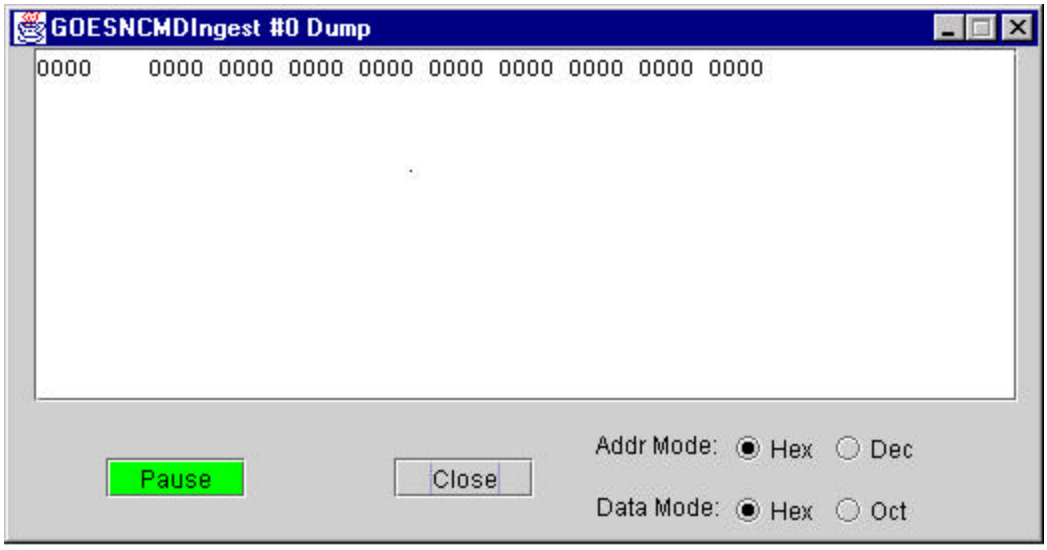
GOES-N Command Ingest-5.2.1 Status Display



Field	Description
Total Received	Total number of command words that are received
Total Valid	Total number of valid command words
Total Errors	Total number of error command words
Total Rejected	Total number of command words are rejected

GOES-N Command Ingest-5.2.2 Buffer Display

The command stream buffer display shows the contents of the received command stream.



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 format.

HST Command Ingest (HSTCmdIngest) Module

HSTCmdIngest-1.0 Overview

The HST Command Ingest module receives a stream commands then performs the following validation checks on this stream:

- Command block format, it is including (1) barker (or sync) code (2) preamble (3)
- postamble.
- Spacecraft identifier (or address)
- Hamming (or Polynomial) code

This module was designed to meet HST requirements; however, it should be usable to support similar spacecraft if necessary.

HSTCmdIngest-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Command stream	HST specific	Commands parsed and validated. Event messages generated on valid or invalid commands.
None	Command database flat file	File formats and max. length of mnemonic check	Build an internal command lookup table

HSTCmdIngest-3.0 Outputs

Channel	Description
1	Command Counter and Mnemonic/Values pairs

HSTCmdIngest-4.0 Container Items

Name	Type	Description
HSTCmdBlockValidationFlag	UnsByte	Flag to valid a command block
HSTCmdWordValidationFlag	UnsByte	Flag to valid a command word
HSTCmdConvertNrzMtoLFlag	UnsByte	Flag to convert from NRZM to L
HSTCmdDebugFlag	UnsByte	Flag to debug
HSTCmdUpdateCmdCounterFlag	UnsByte	Flag to update command counter
HSTCmdTotalBlockCount	UnsDword	Count number of total command blocks that are received
HSTCmdValidBlockCount	UnsDword	Count number of total valid command blocks
HSTCmdErrorBlockCount	UnsDword	Count number of total error command blocks

HSTCmdTotalWordCount	UnsDword	Count number of total command words that are received
HSTCmdValidWordCount	UnsDword	Count number of total valid command words
HSTCmdErrorWordCount	UnsDword	Count number of total error command words
HSTCmdRejectedCmdWordCount	UnsDword	Count number of total rejected command words
HSTCmdBlockBuffer	Container Buffer	Buffer that holds incoming command stream
HSTCmdPreamble	Word	Expected preamble byte pattern
HSTCmdPreambleLength	UnsByte	Expected preamble sequence length
HSTCmdPostamble	Word	Expected postamble byte pattern
HSTCmdPostambleLength	UnsByte	Expected postamble sequence length
HSTCmdBarker	Word	Expected barker (or sync) code
HSTCmdSCAddress	Word	Expected spacecraft address (or Id) code
HSTCmdCounter	UnsByte	Internal command counter
HSTCmdEntry1Byte	UnsWord	Byte location
HSTCmdEntry1Frame	UnsWord	Minor frame number
HSTCmdEntry1Subcom	UnsWord	Subcom depth
HSTCmdEntry1SubSel	UnsWord	Subcom type selection (0 for All, 1 for None, 2 for Subcom)
HSTCmdEntry2Byte	UnsWord	Byte location
HSTCmdEntry2Frame	UnsWord	Minor frame number
HSTCmdEntry2Subcom	UnsWord	Subcom depth
HSTCmdEntry2SubSel	UnsWord	Subcom type selection (0 for All, 1 for None, 2 for Subcom)
HSTCmdSpared	Word	Expected spare code
HSTCmdFixed	Word	Expected fixed code
HSTCmdFileName	SimString	File name of command database
HSTCmdSerialMode	UnsByte	Serial transferred mode

HSTCmdIngest-5.0 Displays

To access displays for this module, click in the center of the command ingest module in the project window. The following pop-up menu choices will appear. The “Remove” option can be used during project design to remove this 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. Shown as following:

Module Pop-Up Menu Item	Description
Configure	Access the configuration menu
Run-time	Access the Run-time menu for the module

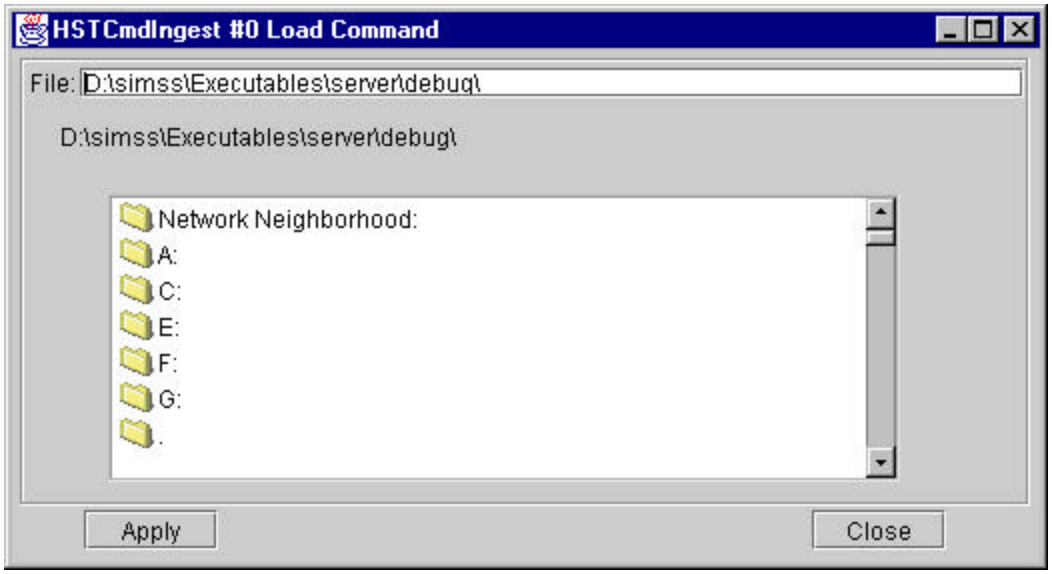
Remove	Remove the module from the project
About	Display Information about the module

HSTCmdIngest-5.1 Configuration

Selecting the “Configure” option produces a Configure menu with the following choices.

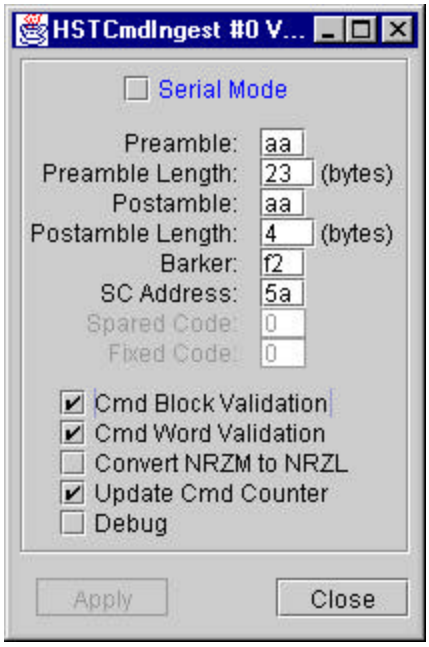
Configure Menu Item	Description
Load Command File	Configure the file name of command database
Validation	Configure the validation flags, pre and post ambles’ byte pattern, their lengths, barker code and spacecraft address
Cmd Counter Location	Configure the command counter location

HSTCmdIngest-5.1.1 Load Command File



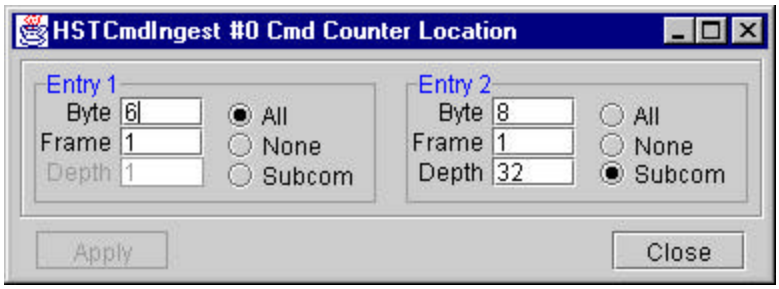
The user can enter a file name by either typing or browsing.

HSTCmdIngest-5.1.2 Validation



Field	Description
Serial Mode	Serial/Block mode selection (checked for serial mode; unchecked for block mode)
Preamble	The byte pattern of the preamble
Preamble Length	The length of preamble in byte
Postamble	The byte pattern of the postamble
Postamble Length	The length of postamble in byte
Barker	Checking code of Barker/Sync
SC Address	Checking sequence of the Spacecraft Address
Spared Code	Checking code of spare when debug is enabled
Fixed Code	Checking code of fixed when debug is enabled
Cmd Block Validation	Enabled/Disabled command block validation
Cmd Word Validation	Enabled/Disabled command word validation
Convert NRZM to NRZL	Enabled/Disabled conversion of NRZM to NRZL
Update Cmd Counter	Enabled/Disabled update the command counter
Debug	Enabled/Disabled show debug messages on event log window

HSTCmdIngest-5.1.3 Cmd Counter Location



Field	Description
Byte	Byte location
Frame	Minor frame number per major frame
Depth	Subcom depth when “Subcom” is selected; otherwise it is ignored
All	All minor frames
None	It is just the one minor frame
Subcom	It is subcommed with depth Depth

The following table lists the command counter locations for some of the common formats.

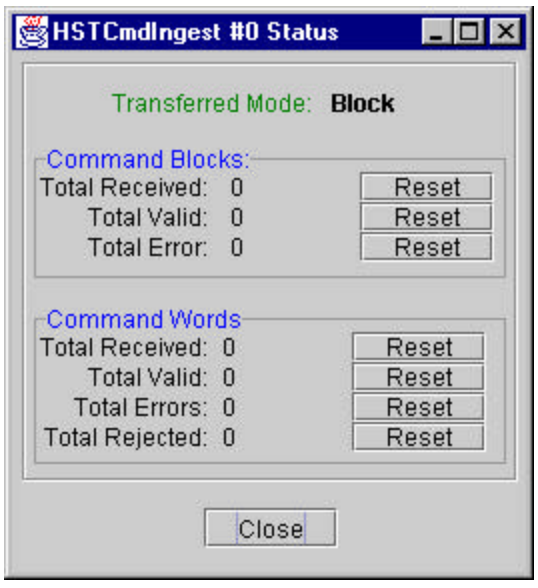
Format	CDIA	CDIB	Subcom
A	90	91	All frames
D,E	18	17	All frames
F	171	172	10
H	107	108	All frames
P	183	184	All frames
T	87	88	All frames
X,Y,Z	199	200	20

HSTCmdIngest-5.2 Run Time

Selecting the “Run-time” option produces a Run-time menu. There are four items in the menu list shown as following:

Run-time Menu Item	Description
Display Status	Show status information
Display Buffer	Dump incoming command stream
Resume	Resume the Command Ingest module after a pause
Pause	Pause the Command Ingest module (enabled to configure)

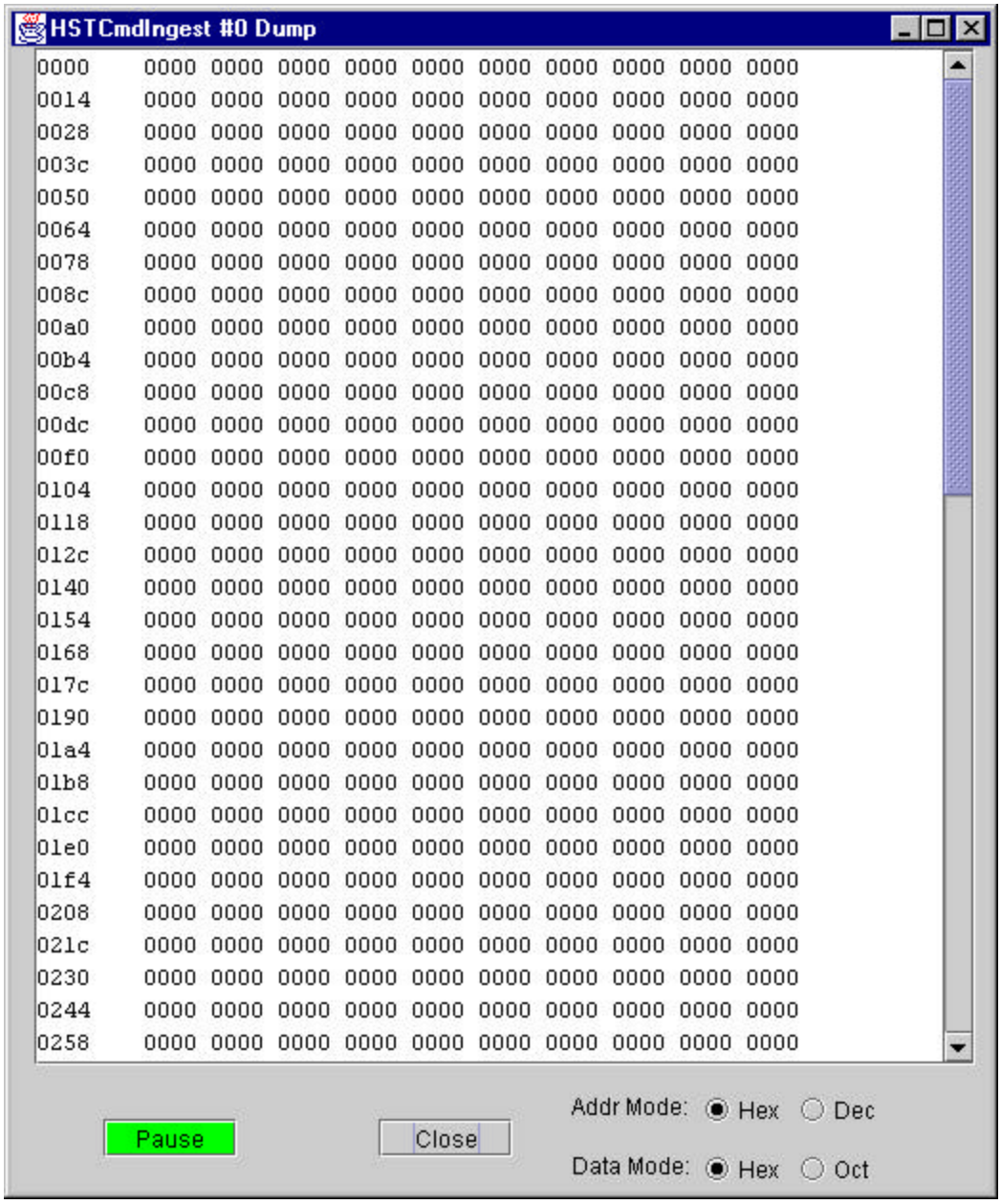
HSTCmdIngest-5.2.1 Status



Field	Description
Transferred Mode	Block/Serial mode
Command Blocks:	Only for block mode; whole box will gray out if serial mode (in other words, it is meaningless if serial mode)
Total Received	Total number of command blocks that are received, it equals: Total Valid + Total Errors
Total Valid	Total number of valid command blocks
Total Errors	Total number of error command blocks
Command Words:	
Total Received	Total number of command words that are received, it equals: Total Valid + Total Errors + Total Rejected
Total Valid	Total number of valid command words
Total Errors	Total number of error command words
Total Rejected	Total number of command words are rejected

HSTCmdIngest-5.2.2 Buffer

The command stream buffer display shows the contents of the received command stream.



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 format.

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 a module and can pass that data to another module and send it to an external destination. Both the Input and Output IP modules are discussed in this one section 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

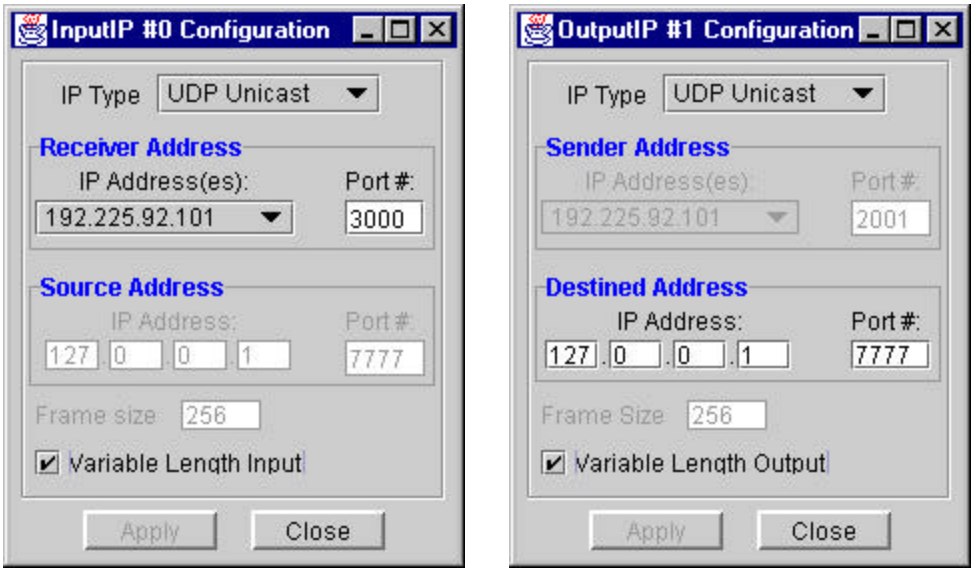
IP module container items are not accessible via operator directives, so they are not listed here.

InternetProtocol-5.0 Displays

Click in the center of the module in the project window to access displays. The following 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



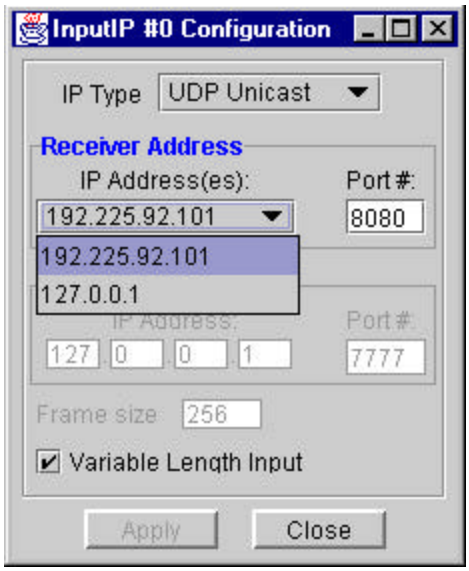
Clicking on the "Configure" item of the Input (or Output) IP module pop-up menu produces a display similar those shown above.

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.

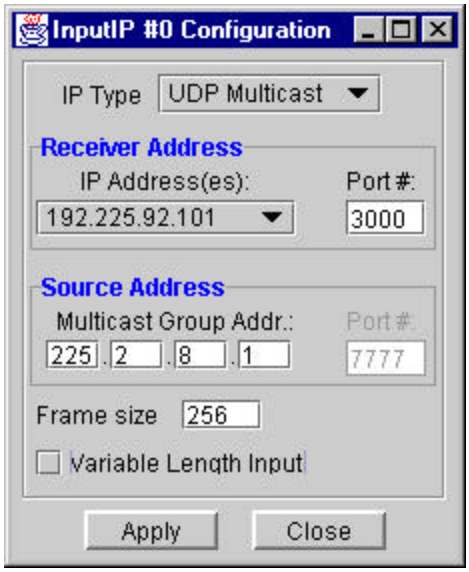
Note: The valid range of port numbers is 1000 to 65535. The valid range of multicast group IP addresses is 224.0.1.0 through 239.255.255.255.

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.

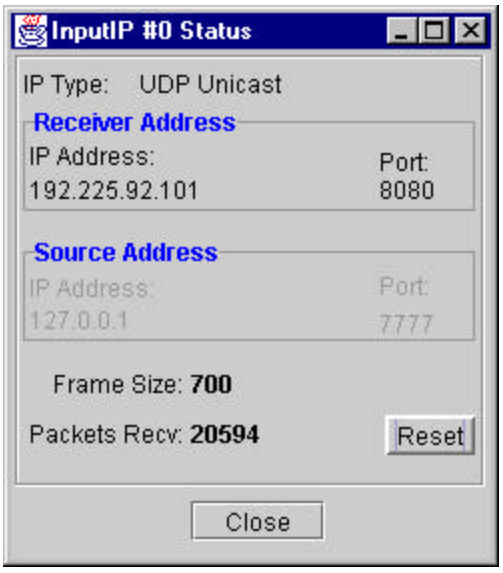
InternetProtocol-5.2 Run-time Menu

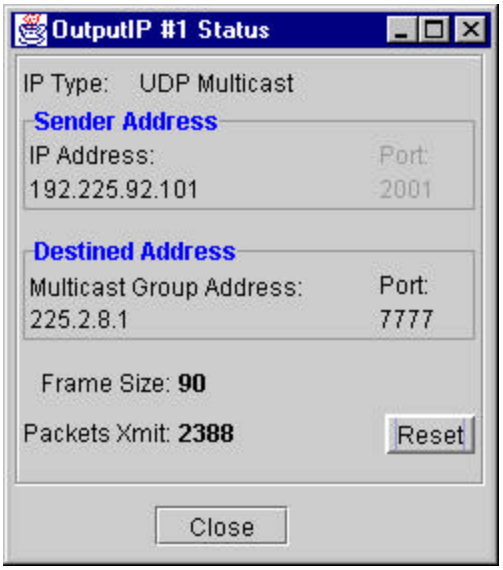
The Run-time menu for the IP modules contains the following four items.

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.

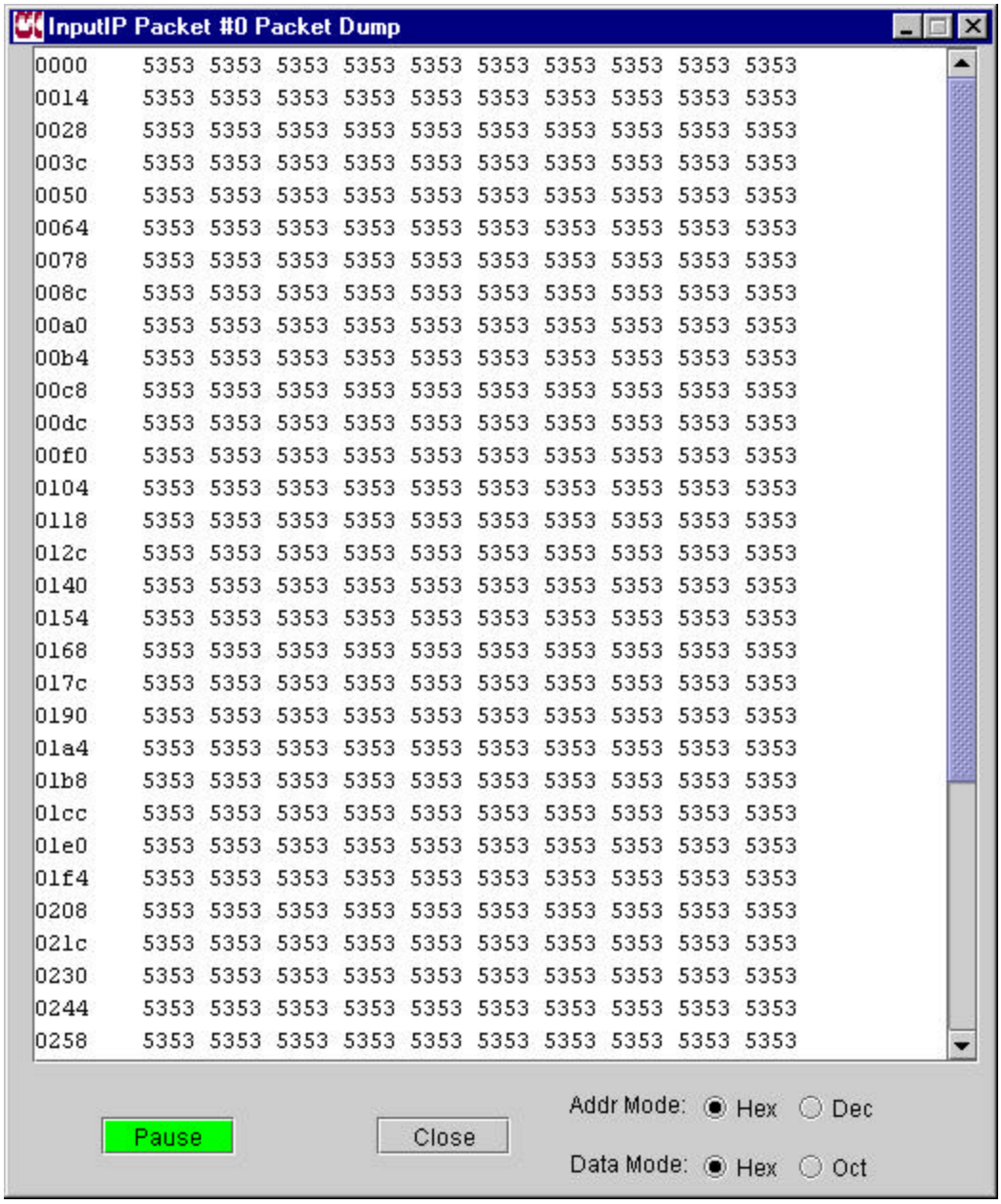




This display shows the Input (or Output) IP module's IP Type, Receiver (Sender) Address (including the port number, if appropriate for the mode), Source (Destination) Address (IP address, or multicast group IP address and port number), Frame Size, and number of packets received (or transmitted). If variable 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 can not 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 color around the module’s border will change from green-striped (indicates run state) to red-striped (indicates non-run state). This option's text will be changed to “Resume”. When an Input IP module is paused, a limited amount of data received will be queued up and sent on when the module is Resumed. When an Output IP module is paused, all data passed to it is dropped.

InternetProtocol-5.2.4 Resume

After an individual module has been paused, select “Resume” from the Run-time Menu to resume the module's execution. The color around the module’s border will change from red to green. This option's text will be changed 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 the IP module’s processing. The color around the module’s border will change from green to red. This option's text will be changed to “Restart”. Once the module has been stopped, the “Configure” option of the module pop-up menu is available again and the module may be reconfigured. Any data passed to stopped modules is discarded.

InternetProtocol-5.2.6 Restart

After an individual module has been stopped, select “Restart” from the Run-time Menu to restart the module. The color around the module’s border will change from red to green. This option's text will be changed back to “Stop”. Since their configurations may have changed, the IP modules reset all counters and only process new data.

InternetProtocol-5.3 About

Selecting the “About” option from an IP module pop-up menu produces a display that lists the module’s number of inputs and outputs, whether directives are allowed, names of authors and the version number.

InternetProtocol-6.0 Special Operating Instructions

There are no special operating instructions for this release.

JSCSim Command Encoding (JSCCmdEncode) Module

JSCCmdEncode-1.0 Overview

The JSCSim Command Encoding module adds shuttle BCH encoding and time authentication to raw 48-bit shuttle commands and command sequences. Both the BCH encoding and the time authentication can be enabled or disabled independently. Encoded blocks output by the module are 128-bit blocks. If all encoding is disabled, the 48-bit block will be passed through unchanged.

JSCCmdEncode-2.0 Inputs

Channel	Description
1	Receives the raw 48-bit command blocks from another module.

JSCCmdEncode-3.0 Outputs

Channel	Description
1	Sends the 128-bit encoded command blocks to another module, or passes through the raw 48-bit command block if encoding is disabled.

JSCCmdEncode-4.0 Container Items

Modification of the JSCSim Command Encoding module's container items via operator directives is not recommended, so they are not listed here.

JSCCmdEncode-5.0 Displays

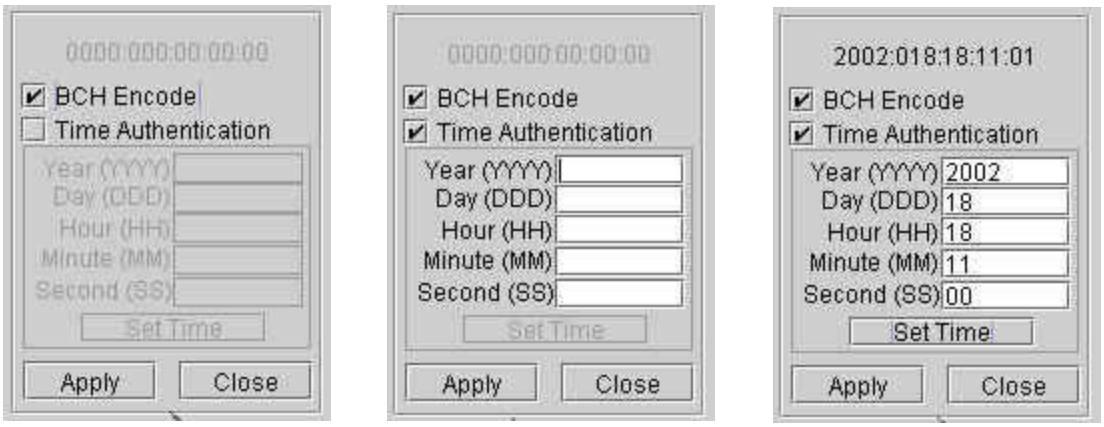
To access displays for a module, click in the center of the module in the project window. The module pop-up menu will appear. The "Remove" option can be used during project design to remove the module. Note that there is no configuration screen for this module. All configuration is done at run time through the run-time displays. The run-time menu is accessible once the project is set to run.

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

JSCCmdEncode-5.1 Run

The Run-time menu for JSCCmdEncode contains the following items:

Run-time Menu Item	Description
Control	Request the main display
Resume	Resume the paused module
Pause	Pause the running module



This is the main display for the JSCCmdEncode module, which allows BCH encoding and time authentication of incoming command sequences. To enable the BCH encoding, check the box next to that item. Once BCH encoding is selected, the user can also select time authentication. When time authentication is selected, the user is then able to enter and set the current time, which will then update and be used to add the time authentication to the commands. If time authentication is enabled but the current time is not set, the time authentication cannot be added to the commands.

Use the **Set Time** and the **Apply** buttons after making selections.

JSCSim Voice Pattern (JSCVoicePattern) Module

JSCVoicePattern-1.0 Overview

The JSCSim Voice Pattern module sends voice pattern formatting configurations to TDMGen for insertion into shuttle uplink frames.

JSCVoicePattern-2.0 Inputs

The JSCSim Voice Pattern module has no input channels.

JSCVoicePattern-3.0 Outputs

The JSCSim Voice Pattern module has a single output channel.

Channel	Description
1	Sends the voice pattern formatting to TDMGen for insertion into shuttle uplink frames.

JSCVoicePattern-4.0 Container Items

Modification of the JSCSim Voice Pattern module's container items via operator directives is not recommended, so they are not listed here.

JSCVoicePattern-5.0 Displays

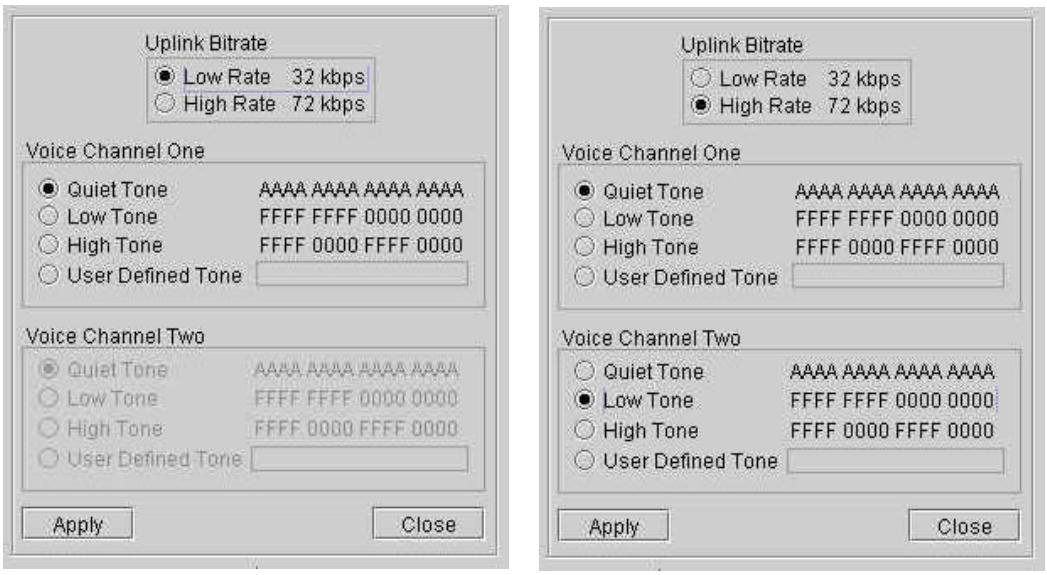
To access displays for a module, click in the center of the module in the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove the module. Note that there is no configuration screen for this module. All configuration is done at run time through the run-time displays. The run-time menu is accessible once the project is set to run.

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

JSCVoicePattern-5.1 Run

The Run-time menu for JSCVoicePattern contains the following items:

Run-time Menu Item	Description
Control	Request the main display
Resume	Resume the paused module
Pause	Pause the running module



This is the main display for the JSCVoicePattern module, which allows the user to select the uplink bitrate that is being generated from TDMGen, and the voice pattern or patterns to insert into the module. Make sure that the uplink rate selected matches the shuttle frame configuration set in TDMGen. The frame size for 32kbps is 80 bytes, and the frame size for 72kbps is 180 bytes.

Use the **Apply** button after making selections. Changes to the uplink stream will not take effect until the **Apply** button is pressed.

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	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

Name	Type	Description
LOGFILENAME	SimString	Logging file name
LOGFIXBOOL	UnsByte	1 for fixed length logging, 0 for variable length logging
LOGPSIZE	UnsWord	Packet size for fixed length logging only
LOGHEADERBOOL	UnsByte	1 for logging with header, 0 for logging without header
LOGMSIZE	UnsDword	Maximum size limit in bytes for the log file
LOGBLOGGED	UnsDword	Number of bytes that are written into the log file
LOGNPACKET	UnsDword	Number of packets that are written into the log file

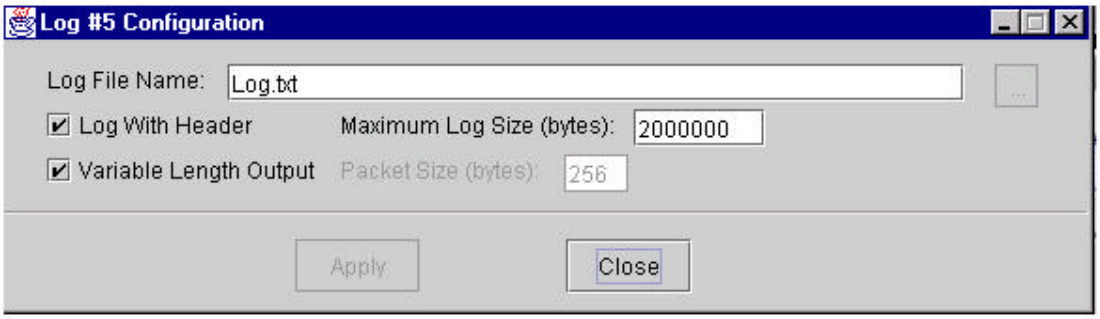
Log-5.0 Displays

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

Log-5.1 Configuration

Selecting the "Configure" pop-up menu option produces a display similar to the following screen.



Log-5.1.1 Log File Name

The Log File Name specifies where the log data is written. If this file already exists, its data will be overwritten. If there is more than one Log module in a project, they cannot write to the same log file.

Log-5.1.2 Maximum Log Size (bytes)

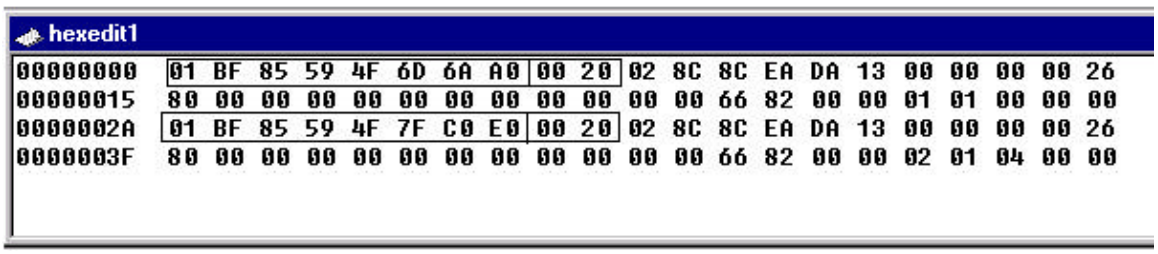
The Maximum Log Size in bytes limits the amount of storage used for logging. When this maximum is reached, the log file is closed and an event message is generated.

Log-5.1.3 Packet Size (bytes)

The Packet Size defines the size of the data buffer to be written to the log file. A packet size that is smaller than the actual data buffer received will result in truncation of the data that is logged. A packet size that is larger than the actual data buffer received will be zero filled to the packet size and then written to the log file.

Log-5.1.4 Log With Header

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 unchecked, all records will be written without a header.



Log-5.1.5 Variable Length Output

When the Variable Length Output 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 packet size field is desensitized.

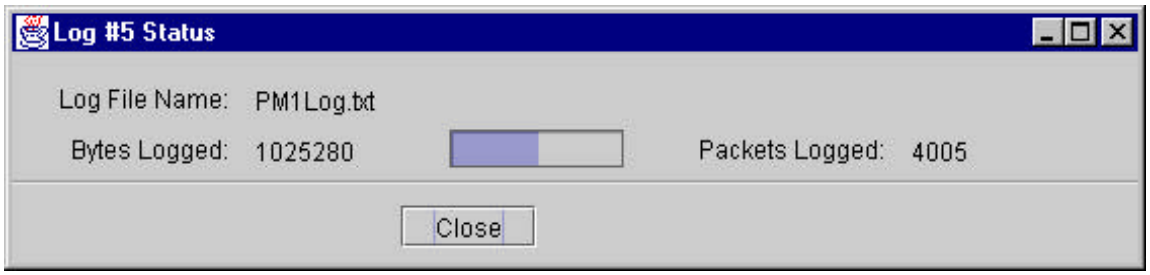
Log-5.2 Run-time

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

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

Log-5.2.1 Show Status

When the "Show Status" option is selected from the Run-time Menu, a screen like the following is displayed. The bar in the center of the screen provides and indication of the percentage of the log area in use. The blue portion of the bar represents the logged data.



Log-5.2.2 Pause/Resume

Select "Pause" from the Run-time Menu to pause the Log module's processing. The color around the module's border will change from green (indicates run state) to red (indicates stopped state). The text of this menu option will change 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. The color around the module's border will change from red to green. The text of this menu option will change to "Pause" and the module pop-up menu will have the normal Run-Time selections.

Log-5.2.3 Stop/Restart

Select "Stop" from the Run-time Menu to stop the Log module's processing. The color around the module's border will change from green to red. The text of this menu option will change to "Restart". Once the module has been stopped, the "Configure" option of the module pop-up menu is available again and the module may be reconfigured.

After a Log module has been stopped, select “Restart” from the Run-time Menu to restart the module's execution. The color around the module’s border will change from red to green. The text of this option will change to “Stop" and the module pop-up menu will have the normal Run-Time selections.

Log-5.3 About

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

Log-6.0 Special Operating Instructions

There are no special operating instructions for this release.

Model Generator Interface (ModGenIF) Module

ModGenIF-1.0 Overview

The ModGenIF module receives a packet from the Model Generator through an IP input module. The four bytes of sync are optionally checked on this packet. Next the mnemonic name from the packet is compared against a list in the input file. If the name is found, the current value from the packet is stored in an output packet and transmitted on the output channel.

This module was designed to transmit data to the TDMGen module to meet TDM requirements; however, it should be usable to support other project configurations.

ModGenIF-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	stream packets	Packet fields are compared to configured values	Packet sync and mnemonic fields are parsed and validated. Event messages generated for valid and invalid packets.

ModGenIF-2.1 Input Packet Format

The Model Generator generates the input packet, the IP input module with the TCP/IP socket receives it, then it is passed down to the ModelGenIF module by the IP input module. The input packet consists of three parts; the first part is sync code, the second part is the message length, and the third part is the content of message.

The input packet looks like the following:

Sync Code (4 Bytes)	Message Size (4 Bytes in hex)	Message
RBF0	000E	6,URADNTMP,15<NULL>

Message is comma-delimited text, include the trailing null character. A message would consist of the following:

<command>, <mnemonic>, <value> <NULL>

command – 6 is the only command currently implemented

mnemonic – any telemetry mnemonics that is up to 8 characters long

value – in PCM counts

ModGenIF-3.0 Container Items

Name	Type	Description
MGIFPacketReceived	UnsDword	Total packets received
MGIFPacketTransmitted	UnsDword	Total packets transmitted

ModGenIF-4.0 Displays

To access displays for this module, click in the center of the ModelGenIF module in the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove this 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 Item	Description
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display Information about the module

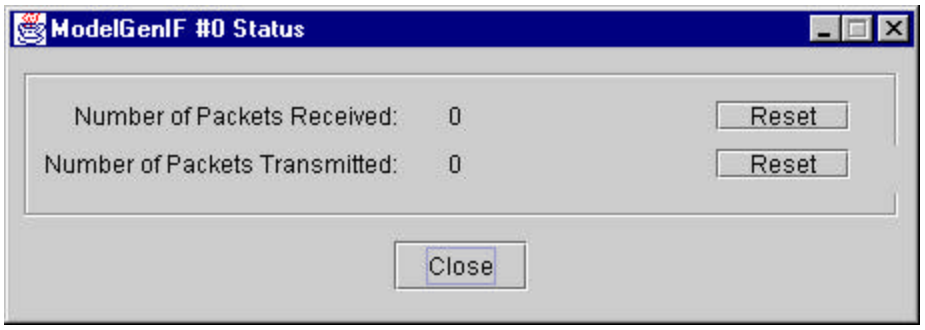
ModGenIF-4.2 Run Time

Selecting the "Run-time" option from the module pop-up menu produces a Run-time menu. There are three items in the Run-time menu list as shown:

Run-time Menu Item	Description
Show Status	Show status information
Resume	Resume the module after a pause
Pause	Pause the module

ModGenIF-4.2.1 Status

When the "Show Status" option is selected from the run-time menu, a display similar to the following is shown:



Status Field	Description
Number of Packets Received	Total number of packets received from the Model Generator
Number of Packets Transmitted	Total number of packets transmitted on output channel 1.

ModGenIF-4.2.2 Pause

Selecting this run-time menu option causes the module to temporarily stop execution.

ModGenIF-4.2.3 Resume

Selecting this run-time option causes the module to continue execution after pausing.

ModGenIF-4.3 About

Selecting the “About” option from the module pop-up menu produces a display that lists the module’s number of input and output channels, whether operator directives are allowed, names of authors, and software version information.

ModGenIF-5.0 Special Operating Instructions

ModGenIF should be connected directly to TlmMod.

Monitor Module

Monitor-1.0 Overview

The function of the Monitor module is to be able to see data (usually telemetry) in different formats such as bit shifted or octal or nrz-m.

Monitor-2.0 Inputs

The Monitor module has one input channel.

Channel	Description
1	Data (usually telemetry) to be displayed.

Monitor-3.0 Outputs

The Monitor module has no output channels.

Monitor-4.0 Container Items

Monitor-4.1 Container Items

Name	Type	Description
MonitorTransmitBuffer	Buffer	Contains the data to be displayed

Monitor-5.0 Displays

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

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

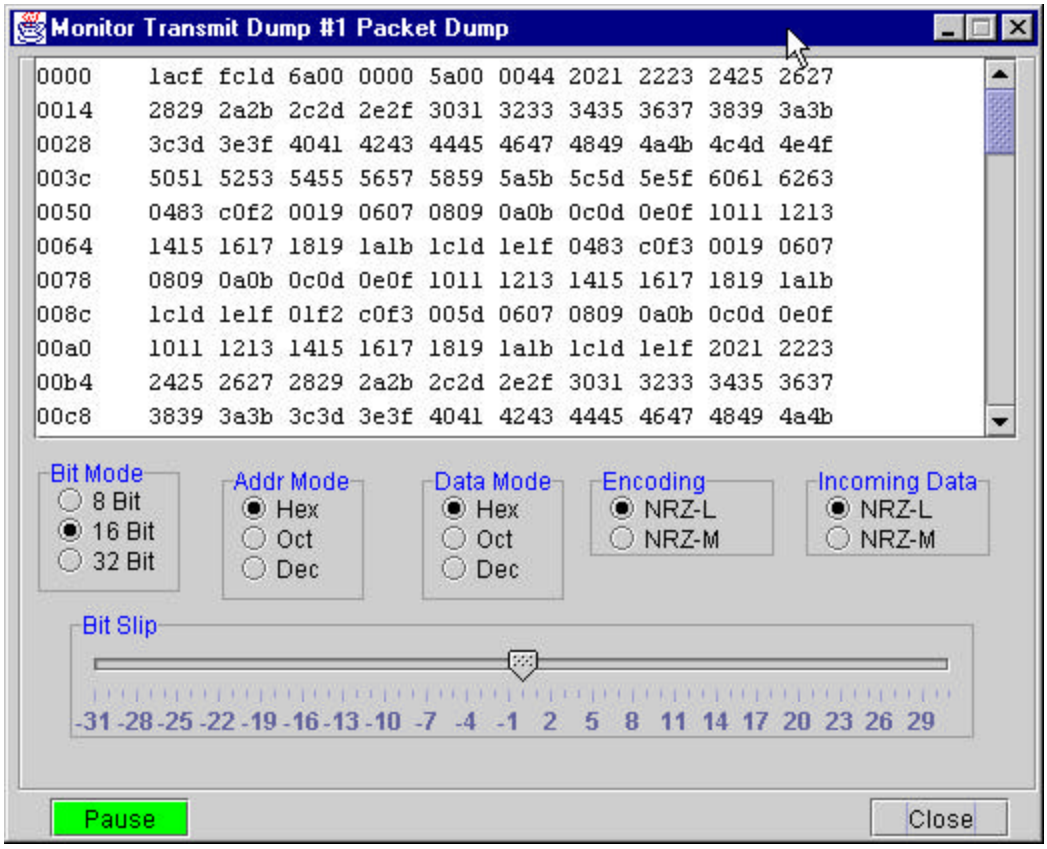
Monitor-5.1 Run-time

The Run-time menu for Monitor contains the following item.

Run-time Menu Item	Description
Control	Display Transmit Buffer

Monitor-5.1.1 Display Transmit Buffer

Click the “Display Transmit Buffer” option of the run-time menu to request this display.



This is the only display for the Monitor module. The user can select any of the display options. In Bit Mode, the user can see the data in 8 bit, 16 bit and 32 bit forms. In Addr Mode, the user can select which number base the address that appears on the left will appear in. In Data Mode, the user can select which number base the data is to appear in. In Encoding, the data will be converted from the format in which it was received into the selected format. In Incoming Data, the user tells the display how the data is coming in. Bit Slip allows the user the shift bits up to 31 bits to the left and up to 31 bits to the right. Pause allows the user the keep the display from displaying another block of data while the user exams the current block.

Packet Processor Module

Packet Processor-1.0 Overview

The Generic CCSDS Packet Processor is used to monitor and verify packets and 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 stream.

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

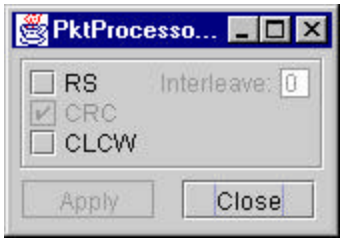
Click in the center of the module within the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove the module. The "Configure" option should be used prior to running the project and is unavailable at run-time. The "Run-time" option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

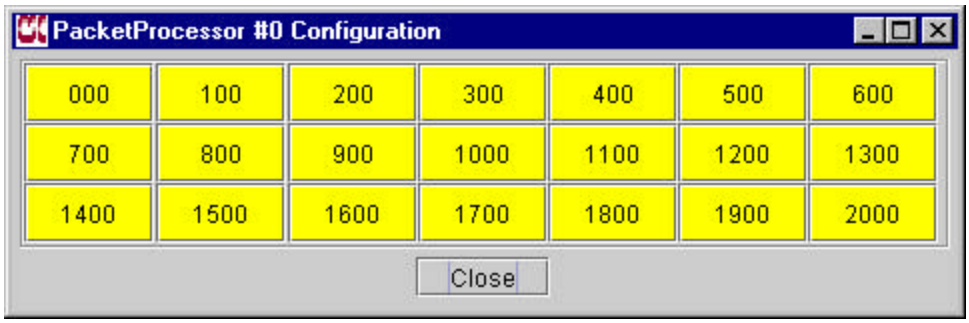
Packet Processor -5.1 Configure

There are two static configuration windows for this module. Module must be configured to CCSDS encoding method. This will ensure correct packet location and its length. Before proceed to configure, user must know specific spacecraft CCSDS format. The module can be re-configured in runtime (see 5.2 below). There are three different re-configuration windows: Main Configuration, Set Channel and Set Offset.

Packet Processor-5.1.1 VC configuration Window



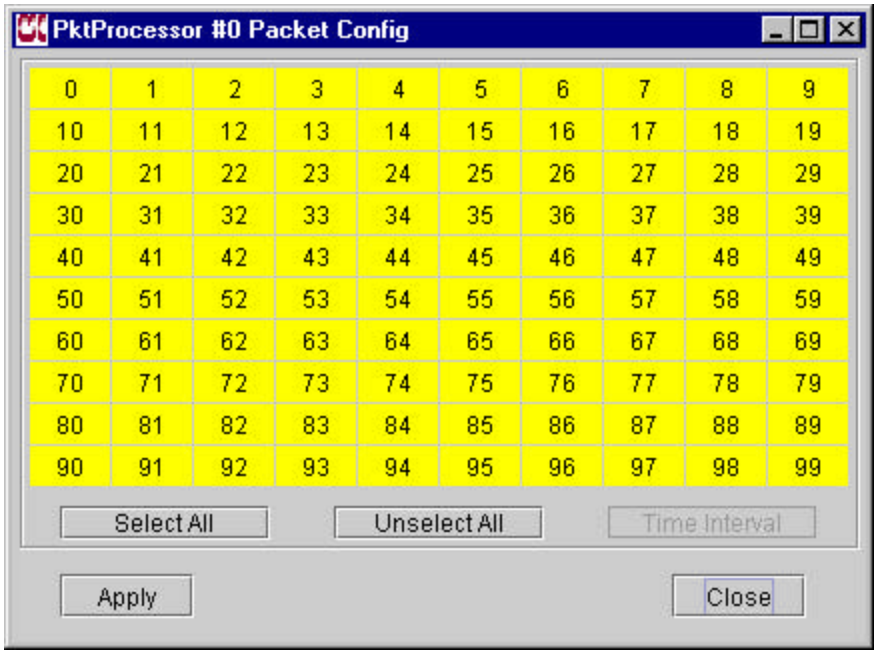
Packet Processor-5.1.2 Packet Configuration Window



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

This is the main configuration window. By selecting one of the configuration buttons, a user will be provided with APID selection window.

Packet Processor-5.1.3 Select APID



This window is the APID filtering configuration window. Any number of APIDs displayed above, can be transmitted to the module's output channel. The user should select the desired APIDs.

Packet Processor-5.1.4 Display Offset

(Not implemented yet)

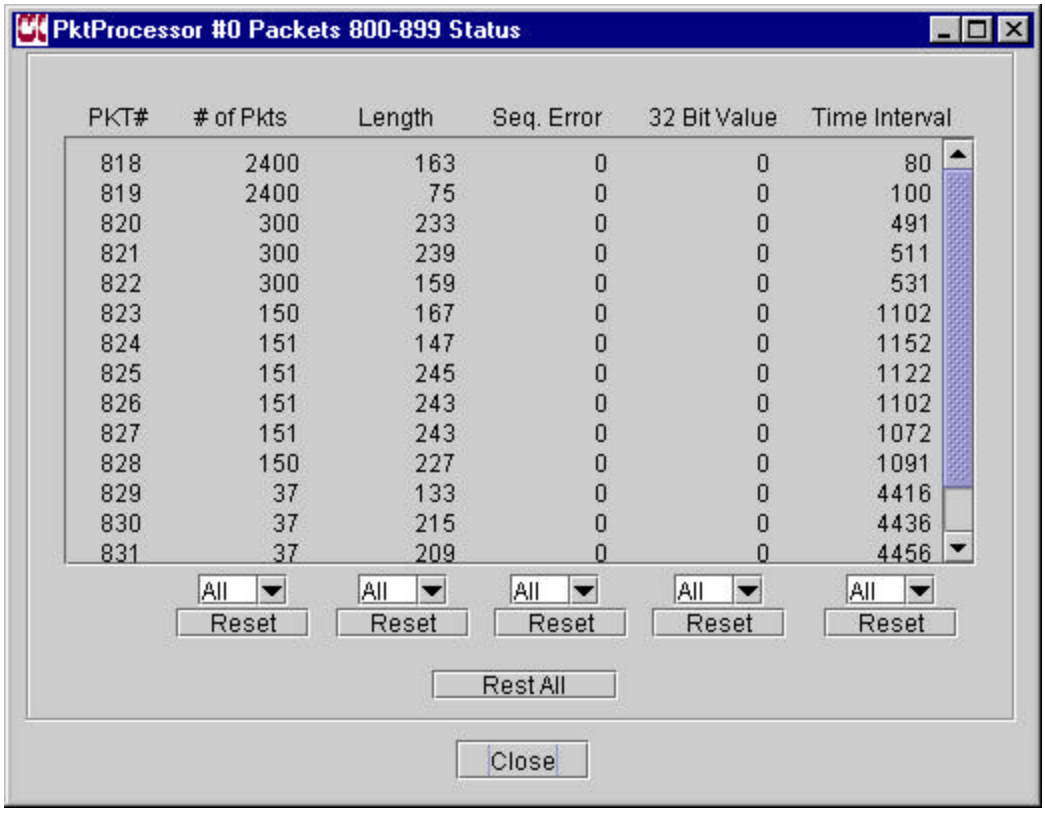
Packet Processor-5.2 Run

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.

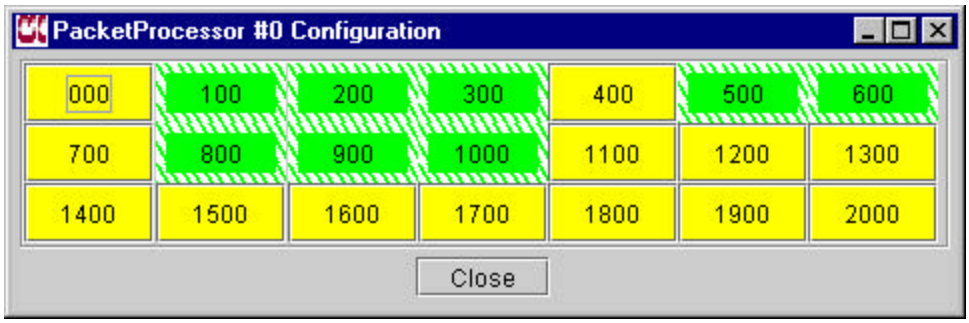
Packet Processor-5.2.1 Show Packet Status



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-5.2.2 Display APID status



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

Yellow Inactive APIDs

Green	Active APIDs with no errors
Red	Active APIDs with sequence errors

All APIDs start out as inactive. When a Packet APID is detected in the data stream, the corresponding box turns green or red, depending on quality of data. The user can click on any boxes and see the status of the APID indicated in range.

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 to do the reconfigure.

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 down-linked 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

When the project containing a scenario module is started, the server reads a property.txt file in the server's properties directory to configure several project parameters. One of these parameters, ***ConcurrentScenario***, defines how scenario modules will execute nested scenario files and module triggered scenario files.

The default setting for *ConcurrentScenario* is one. When set to one, the scenario module executes nested and external module requested scenario files concurrently. Additional system resources are dynamically allocated and released as these scenarios start and stop execution.

If the setting for *ConcurrentScenario* is zero, both types of non-GUI controlled scenario files are collected in a queue and executed in first-in-first-out (FIFO) order. This method is less taxing on system resources but may not execute scenario files quickly enough in some cases. This mode should be used if there are system resource problems over time or if the execution of scenario files causes degradation of other processing.

Scenario-2.2 Module Triggered Scenarios

The Scenario 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 a way to pass names of scenario files to be executed. Processing specific to the down-linked module may trigger the passing of scenario file names back 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 scenario directory, which is below the server executable directory, unless a full pathname is specified. Relative pathnames are assumed to start from the scenario directory.

Scenario-2.3.1 Scenario File Contents

The format of ASCII text scenario files is very simple. Each line is limited to 260 characters. Each line contains one of the following items. Tab characters and blanks may be used to indent

scenario lines for readability. Although Scenario-module specific key words are shown below in uppercase, they are not case sensitive.

Line	Description
comment line	A semicolon in the first column defines a comment.
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. For a GUI controlled scenario, a sleep forever is equivalent to the pause button being used. For non-GUI controlled scenarios, the SLEEP forever instruction is ignored.
START SCENARIO name	The scenario module starts execution of a nested scenario file. This nested file is not available for operator intervention.
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 directives 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 following this line up to the next ELSE IF, ELSE or ENDIF statement to be executed.
ELSE IF (expression)	Specifies a conditional expression that must be true in order for the directives following this line up to 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	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 following this line up to the ENDWHILE statement to be executed. One of the directives within the loop body should be a SLEEP directive.
ENDWHILE	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 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. 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.

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 directives to be conditionally executed 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
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 necessary to prevent CPU-intensive wait loops from depleting system resources.

```
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 should be linked to input channel 1 of the module 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 items with a fixed 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.

There are 6 container items for each of the five GUI-controlled scenarios. To modify or view most of these fields, it is easier to use the Run-time Control display. To access the items for the first controlled scenario, enter the name without the "<number>" field. To access the items for the 2nd, 3rd, 4th, and 5th controlled scenarios, replace the "<number>" field with "1", "2", "3", or "4". For example, to turn on echoing of the directive lines to the event log for the 5th controlled scenario, enter the following on the directive line with the Scenario module selected to receive it.

```
set Scenario4DirMsg 1
```

To disable the echoing of directive lines to the log for the first controlled scenario, enter

```
set ScenarioDirMsg 0
```

Name	Type	Description
Scenario<number>Currentline	Buffer	Current directive line in scenario file
Scenario<number>DirMsg	Fixed	Flag to list lines read from scenario file (0=no, 1=yes)
Scenario<number>Execution	Fixed	Run flag for scenario execution (0=pause, 1=run)
Scenario<number>Filename	Fixed	Scenario's Filename
Scenario< number >LineNumber	Fixed	Current line number in scenario file
Scenario< number >Status	Fixed	Scenario file's execution status (0=Finished, 1=Running, 99=Paused)

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 Confguration menu for this module
Run-time	Access Run-time menu for this module
Remove	Remove module from the project
About	Display generic module information

Scenario-5.1 Configure

There are no configuration displays for this module. Refer to section Scenario-2.1 for information on configuring the scenario execution mode, *ConcurrentScenario*, from the Server/Properties/Property.txt file.

Scenario-5.2 Run-time

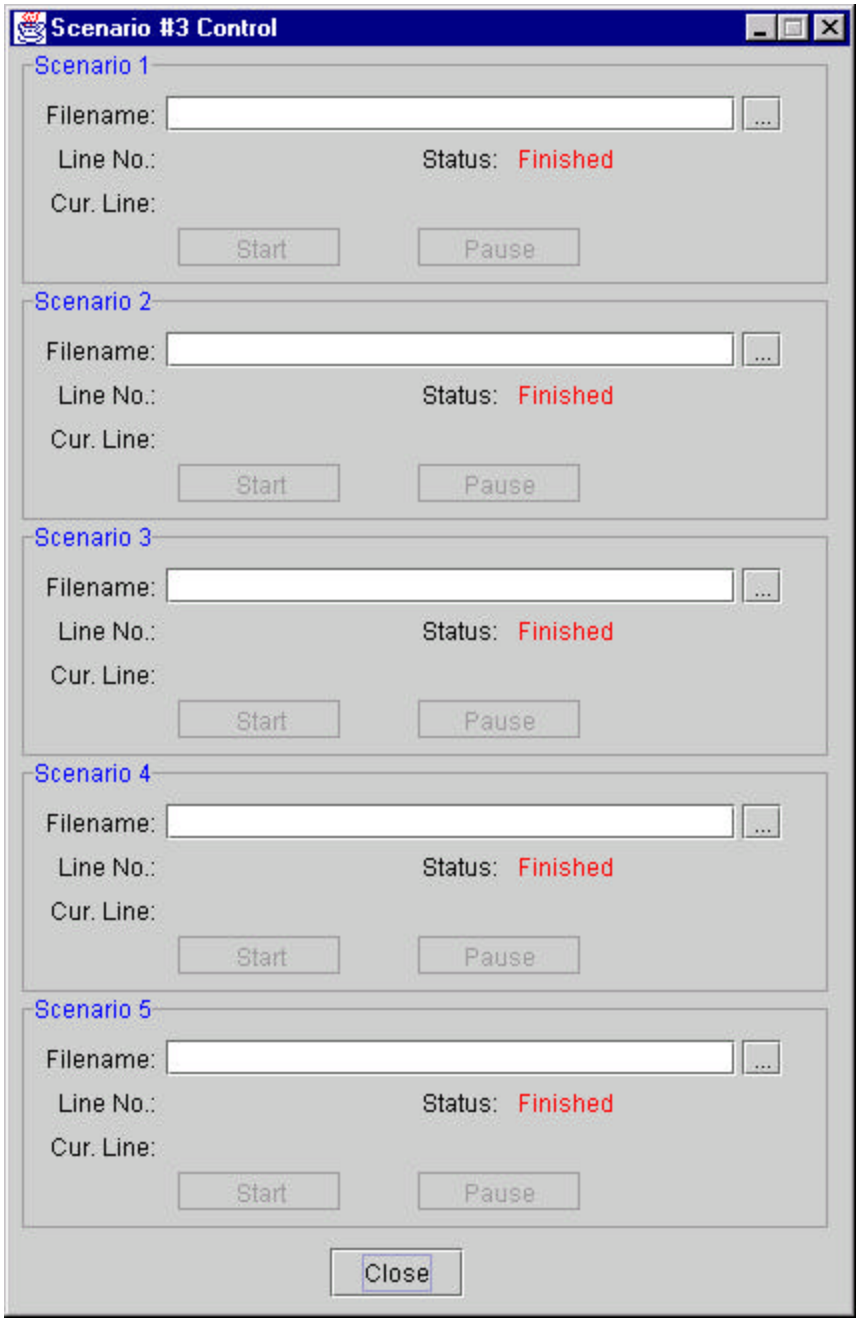
Click on the module pop-up menu "Run-time" option to request the Run-time menu.

Run-time Menu Item	Description
Control	Request Control Scenario 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.

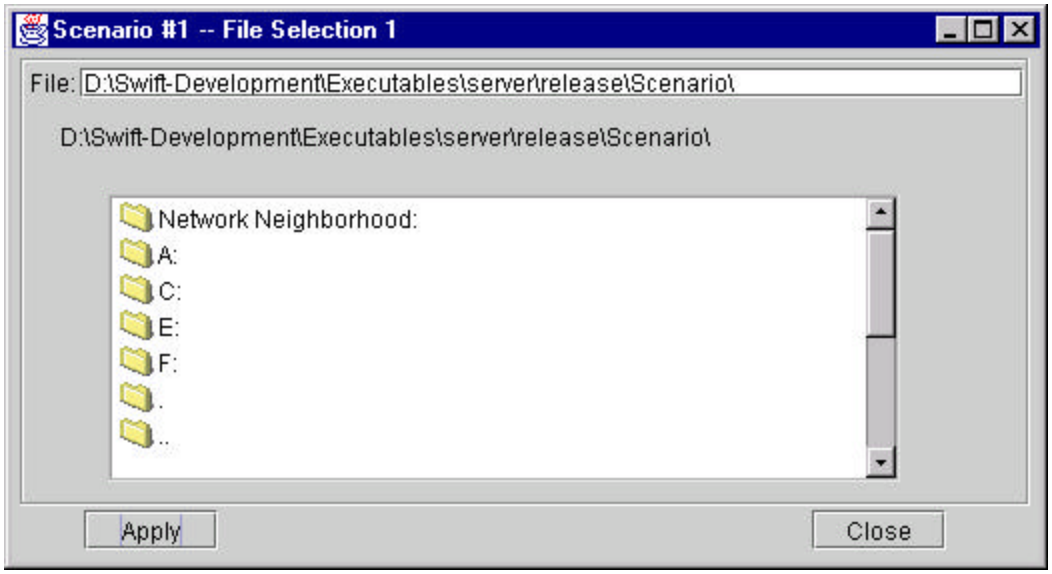
Field	Description
Filename	Enter the full or relative pathname for the scenario file in this field.
File browser 	Click on this button to the right of the filename field to invoke file browsing to select a file.
Start/Stop Button	This button's title changes to Start or Stop as appropriate. To start processing 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

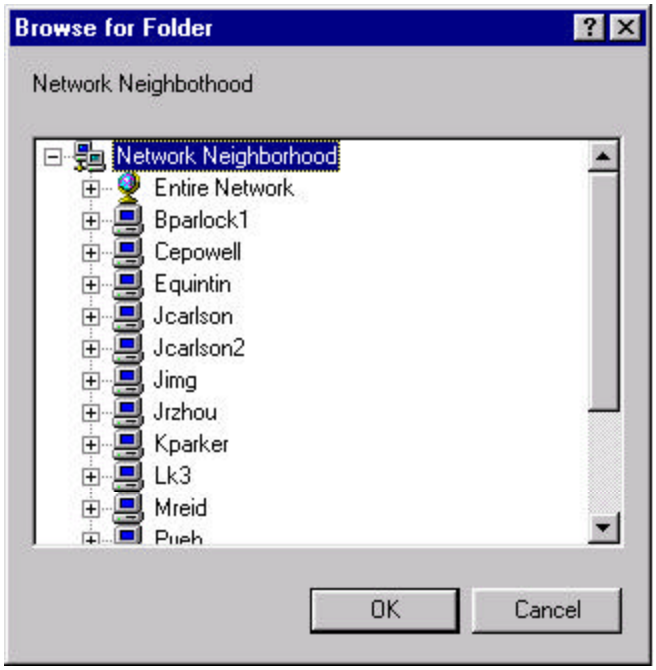
When any of the file browser buttons is selected from the Scenario Control Display, a file selection display similar to the next screen will be displayed. The current search directory will be shown in the File field. Single click on the appropriate folder icons to navigate to the desired directory. Click on the ".." folder icon to go up the directory structure. Click on a named folder to display its contents. Single 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 and dismiss the file selection window. 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: This file browse window allows navigation to any folder within the network neighborhood. If the Network Neighborhood folder is clicked, another "Browse For Folder" window will be opened (it may be hidden under the first browse window).

Navigate this browser to the desired folder and click the **OK** button or use the **Cancel** button to dismiss this browser.



If a network folder was selected, the files from that folder are now available for selection from the original file browse window. Scroll down to view these file names. Single click to select a 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 and dismiss the file selection window.

Use the **Close** 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 the module’s number of input and output channels, whether operator directives are allowed, names of authors, and the version number.

Scenario-6.0 Special Operating Instructions

Scenario-6.1 Built-in Delays in 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. A proper remedy is under investigation. In the meantime it is suggested 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 out from the event window, up to 50 directives may be executed each second.

```
; Sample scenario file
Set MOD_SS_DRNADSTEP 1
SET CEF_NR_ETRQ_1 2
SET AS2_TS_DETECTOR 3
SLEEP 1000
SET AMR_NR_MWASPEED 65
SET AMR_SS_BITPKT1 1
SET AMR_SS_BITPKT4 1
SLEEP 125
SET EPS_SS_MAINBUSVR 0XFF
sleep 5000
SETBUFFER TLMPACKET0140 20 BYTE 0XA 10 012
```

Scenario-6.2 Stopping Nested or Triggered Scenarios

Even though the user does not have control over the execution of scenarios started internally, it is possible to stop them without shutting down the simulator. To stop all ongoing scenario files of all types, click on the Project's Run menu and select the "Stop" option. The user should then immediately click on the "Run" option to restart the entire project. Since all of the modules will be stopped briefly, there could be dropouts in data and other consequences. The scenario module does not automatically restart any scenario files when restarted after a project stop.

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.

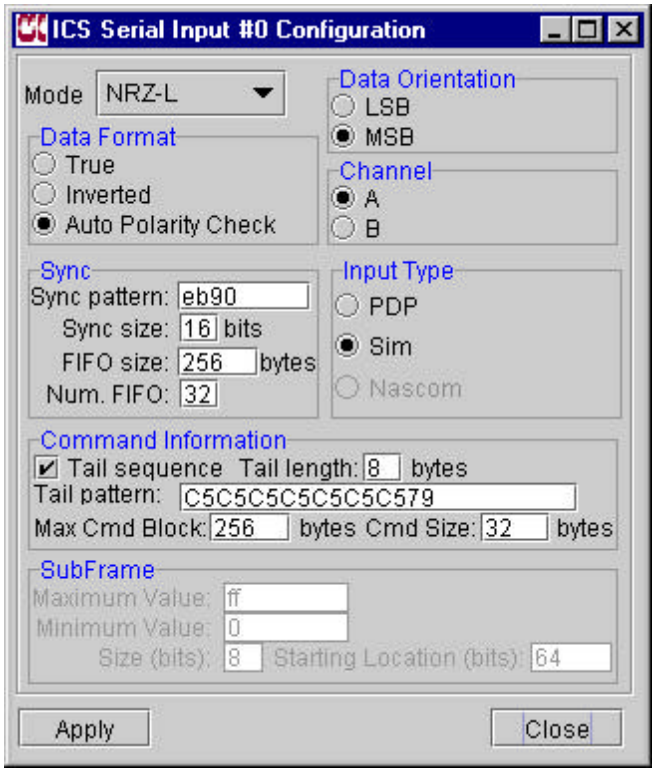
SerialInput-5.0 Displays

To access displays for this module, click in the center of the 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.

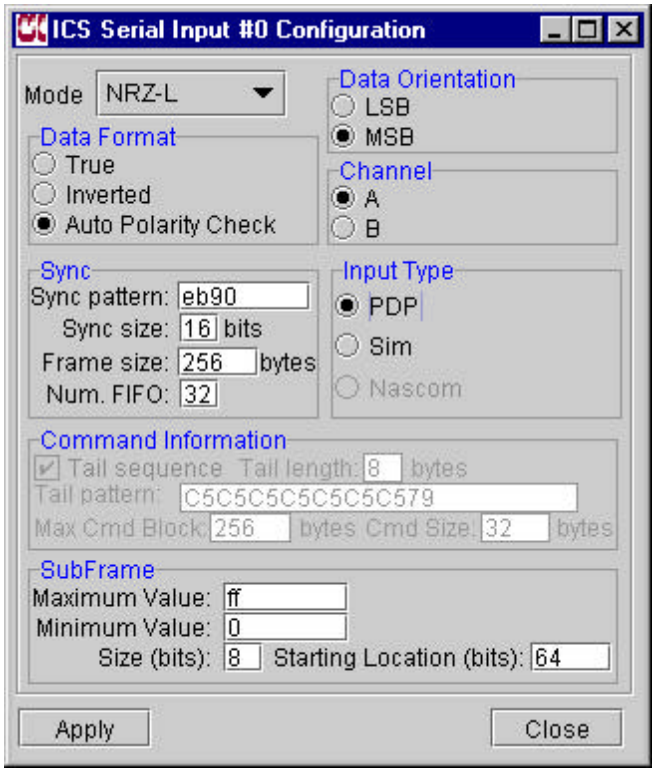
Module Pop-up Menu Item	Description
Configure	Access the module's configuration window
Run-time	Access the module's run-time menu
Remove	Remove the module from the project
About	Display generic module information

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.

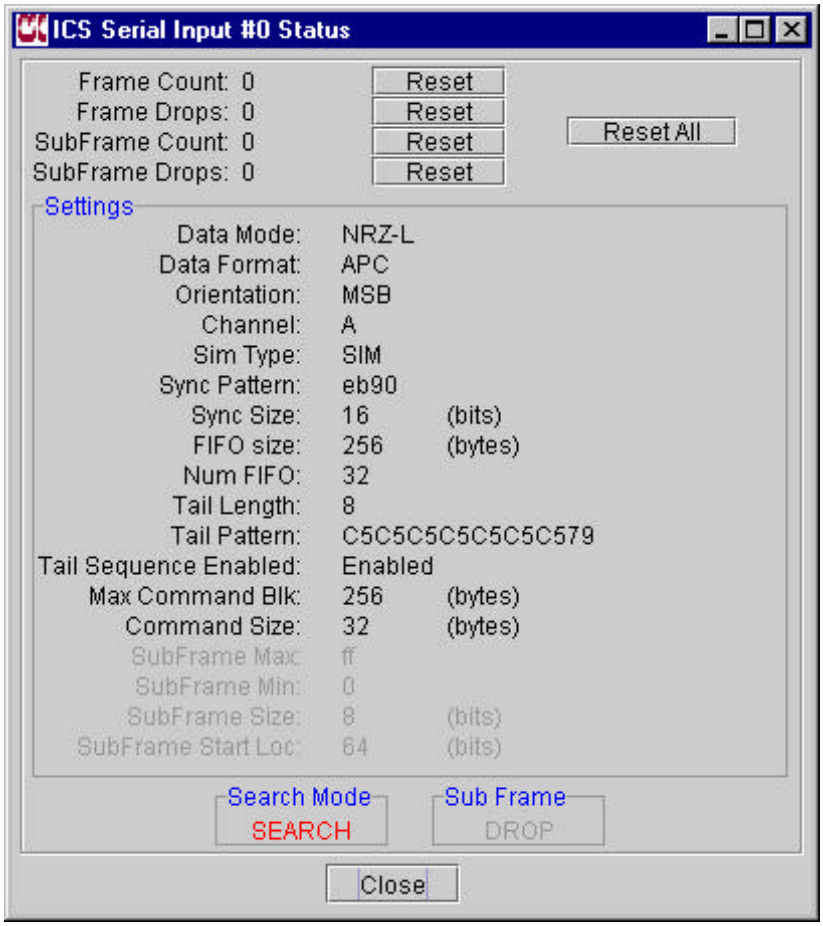
SerialInput-5.2 Run-time

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.

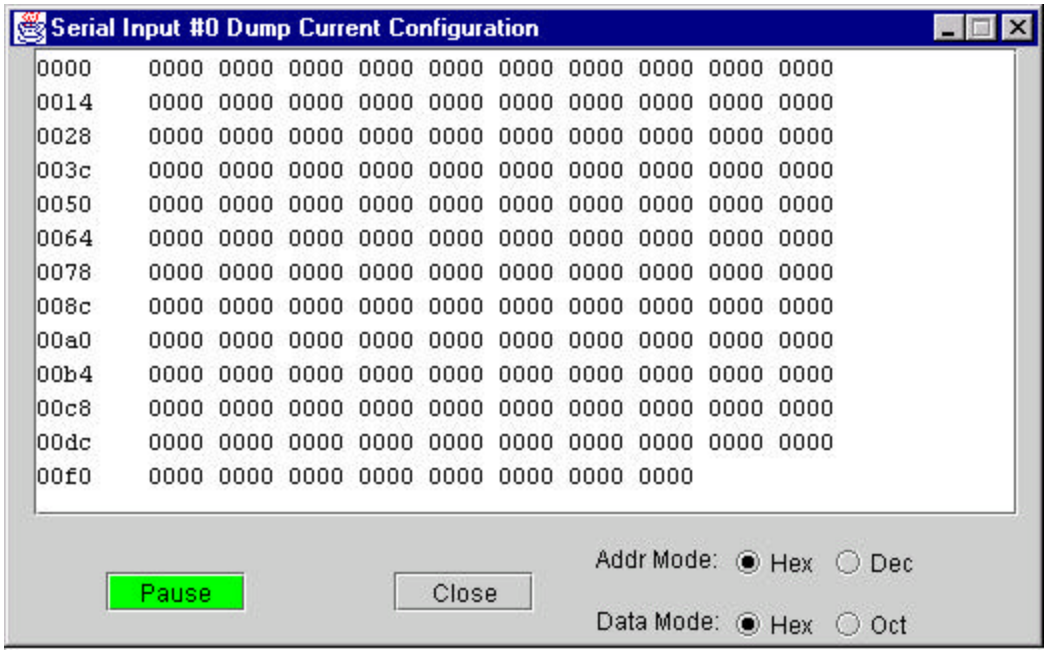
Search Mode and Sub Frame status fields are only applicable when receiving telemetry data.

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, Serial Input module has two different ways to pass received command data to next module, depending on how “Max Cmc 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.

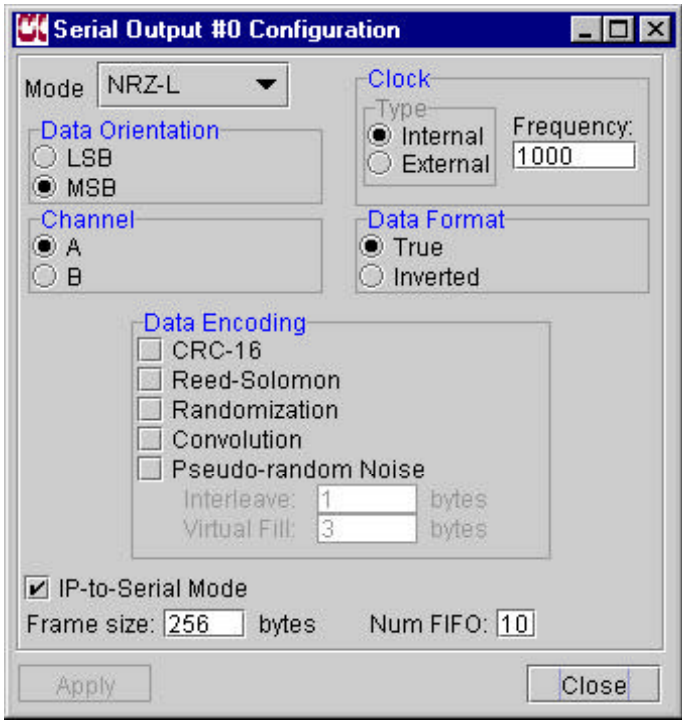
SerialOutput-5.0 Displays

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

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)

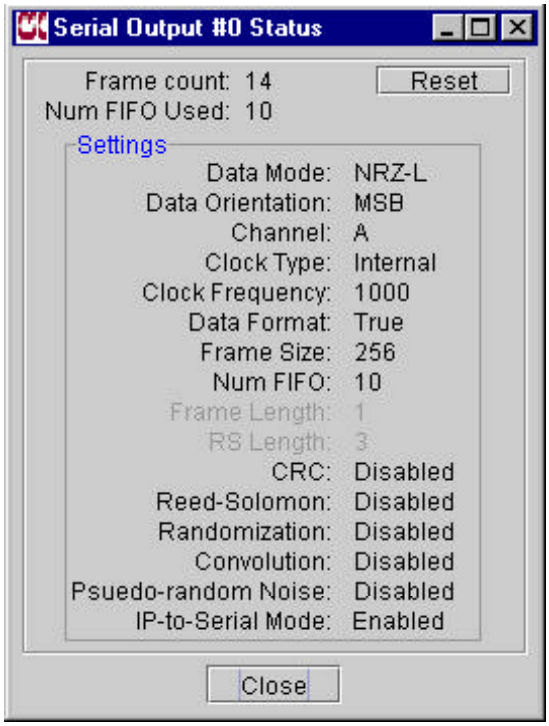
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

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.

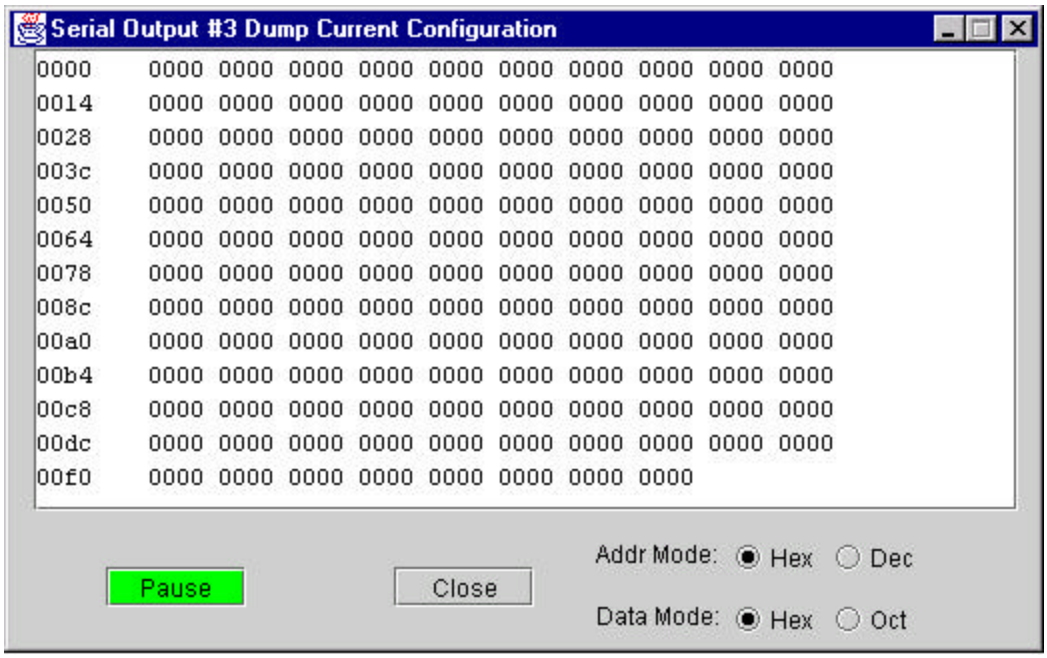
Note: if the Serial Output module is configured to receive data from an Input IP module, the "Num FIFO Used" value varies from 0 to "Num FIFO" value, defending on the difference of data rate between Input IP module and Serial Output module. In any other configuration the "Num FIFO Used" is always at "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 color around 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 module's processing. The color around the module's border will change from green (indicates run state) to red (indicates stopped state). While paused, the 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, names of authors and the module's version number.

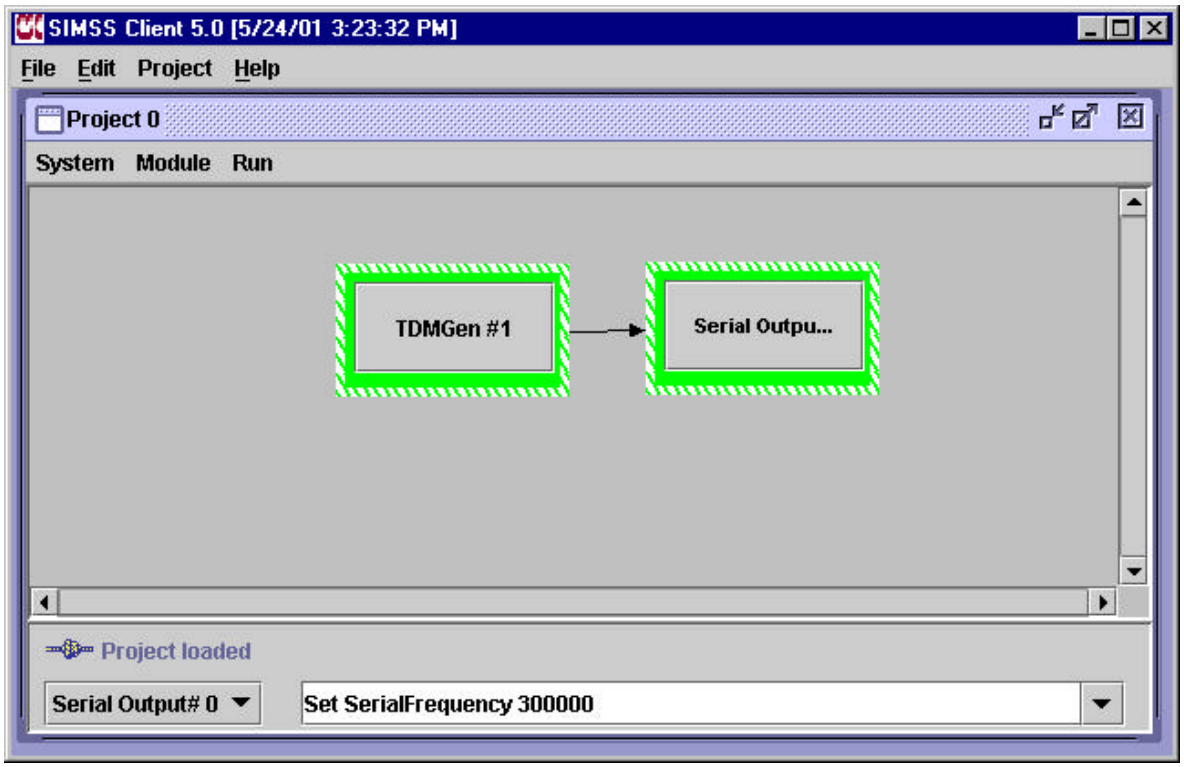
SerialOutput-6.0 Special Operating Instructions

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), 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.

SoftSyncSearch Module

SoftSyncSearch-1.0 Overview

The SoftSyncSearch module syncs data.

SoftSyncSearch-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Data stream	Syncs data	Searches for sync pattern. When patter is found, the following frame is sent to the next module.

SoftSyncSearch-3.0 Outputs

Channel	Description
1	Syncronous data

SoftSyncSearch-4.0 Container Items

Name	Type	Description
SerialDropCount	Int	Count number frames dropped
SerialSearchMode	Int	Sync Status
SerialFRAMESIZE	Int	Select frame length
SerialSYNC	Int	Select sync. patterns
SerialCOUNT	Int	Number of Frames Transmitted/Received
SerialFrameDump	Int	Data Container
SerialNumBitsSync	Int	Bits of Sync Pattern
SerialSubFrameMax	Int	Sub Frame Max
SerialSubFrameMin	Int	Sub Frame Min
SerialSubFrameCnt	Int	Sub Frame Counter
SerialSubFrameDrp	Int	Sub Frame Status
SerialSubFrameStartLocation	Int	MSB of sub frame counter
SerialSubFrameCount	Int	sub frame counter
SerialSubFrameDropCount	Int	sub frame drop counter
SerialOutputDisplayType	Int	Select Data Display Type

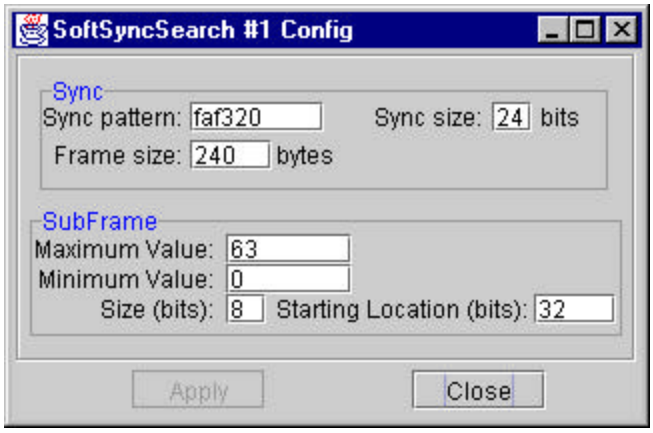
SoftSyncSearch-5.0 Displays

To access displays for this module, click in the center of the command ingest module in the project window. The following pop-up menu choices will appear. The “Remove” option can be used during project design to remove this 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 Information about the module

SoftSyncSearch-5.1 Configuration

The configuration display is used to configure the module. The sync pattern must occur at the beginning of the frame.



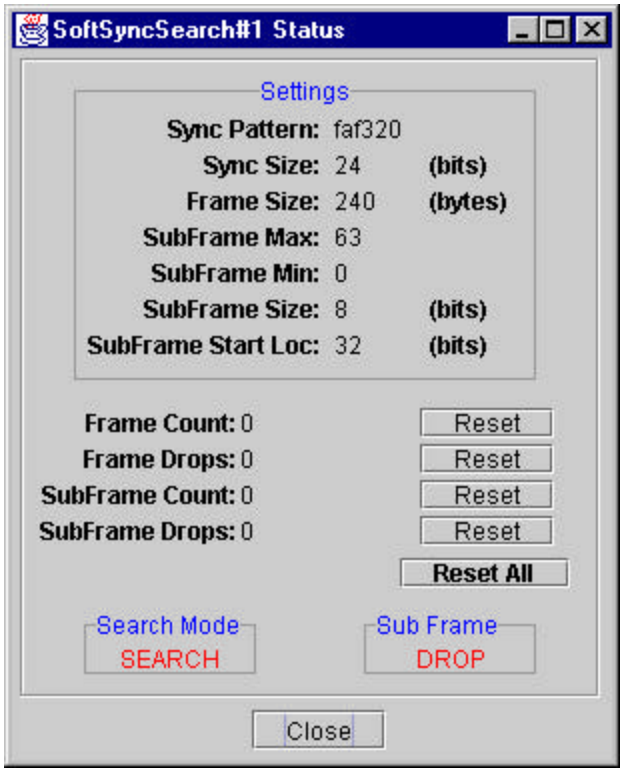
SoftSyncSearch-5.1 Run-Time Displays

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

Run-time Menu Item	Description
Display Status	Show status
Display Dump	Dump the display
Resume	Resume the module after a pause
Pause	Pause the module

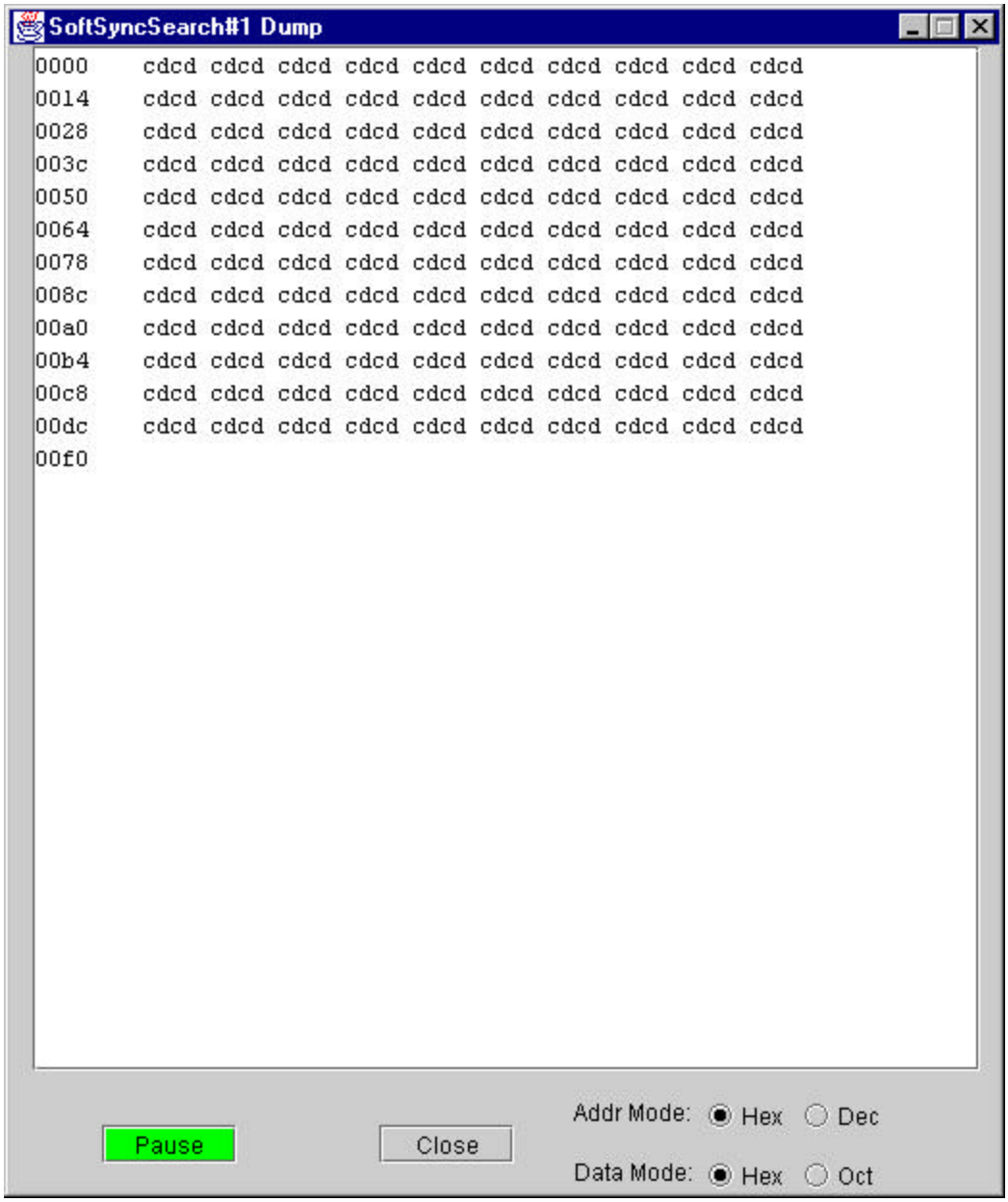
SoftSyncSearch-5.1.1 Status Display

The following display is used to check the status of the SoftSyncSearch module.



SoftSyncSearch-5.1.2 Buffer Display

The command stream buffer display shows the contents of the outgoing telemetry stream.



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 format.

Stripper Module

Stripper-1.0 Overview

The Stripper module receives data from an upstream module, processes and passes the data on to a downstream module. The processing of the data includes deleting the header, footer, and removing the fill-data in the incoming data field from specific formats defined in a configuration file. Typical usage is the retrieving of data field from a NASCOM data block.

Stripper-2.0 Inputs

The Stripper module has an input channel, which is listed below.

Channel	Data expected	Validation performed	Processing performed
1	Data Buffer	None	Data is received from an upstream module, processed and sent to a downstream module.

Stripper-3.0 Outputs

The Stripper module has a single output channel.

Channel	Description
1	Stripper module sends the received and processed data to another module.

Stripper-4.0 Container Items

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

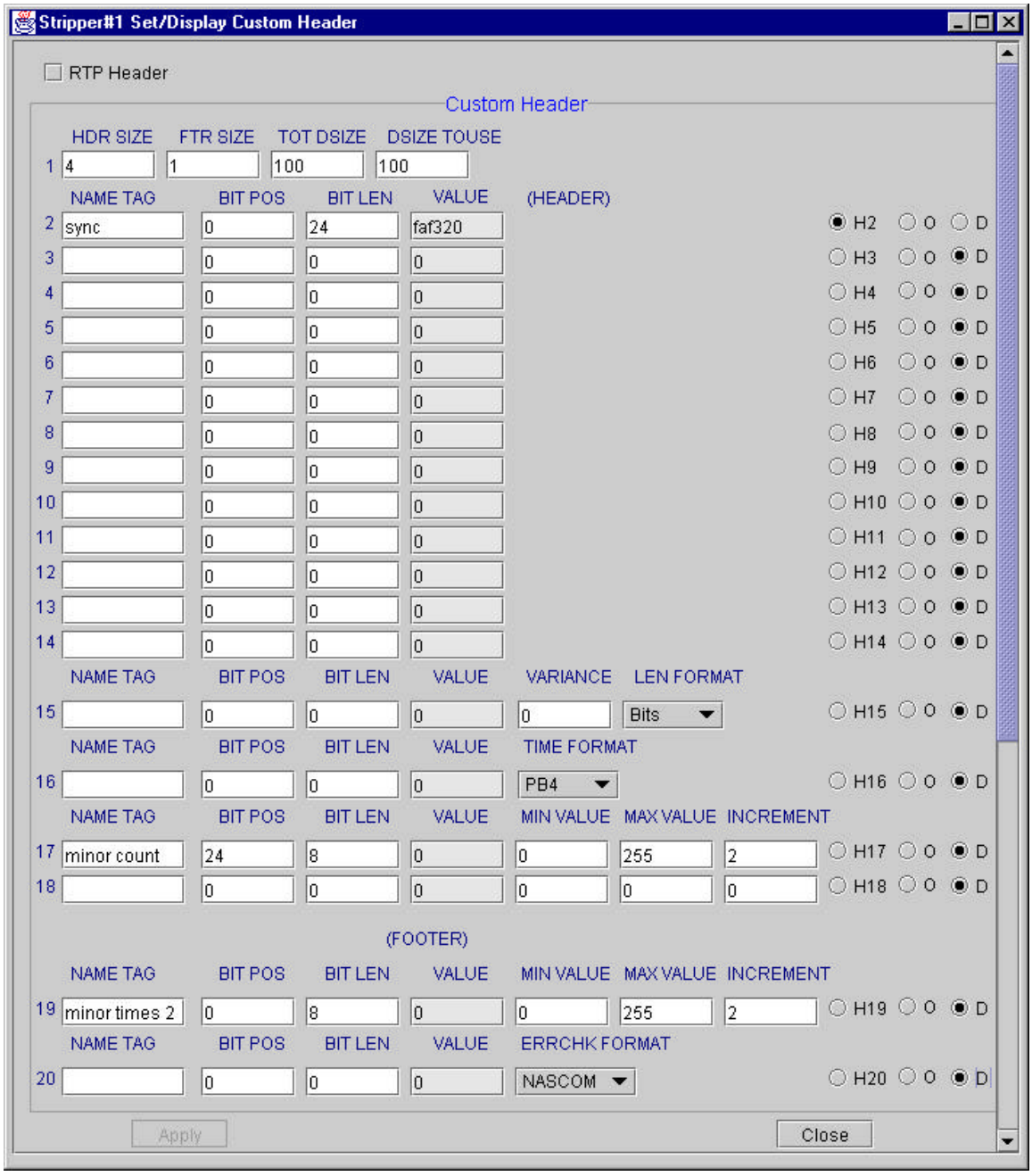
Stripper-5.0 Displays

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

Stripper-5.1 Configuration

Clicking on the module pop-up menu "GUI Configure" item produces the following display.



Click the **Apply** button to save your selected filename. Click the **Close** button to dismiss the Configuration display without making any changes.

Stripper-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
Resume	Restart the module

Pause	Pause the module
Show Output Block	Request Output Data Block Display
Show Status	Request Status Display

Stripper-5.2.1 Resume

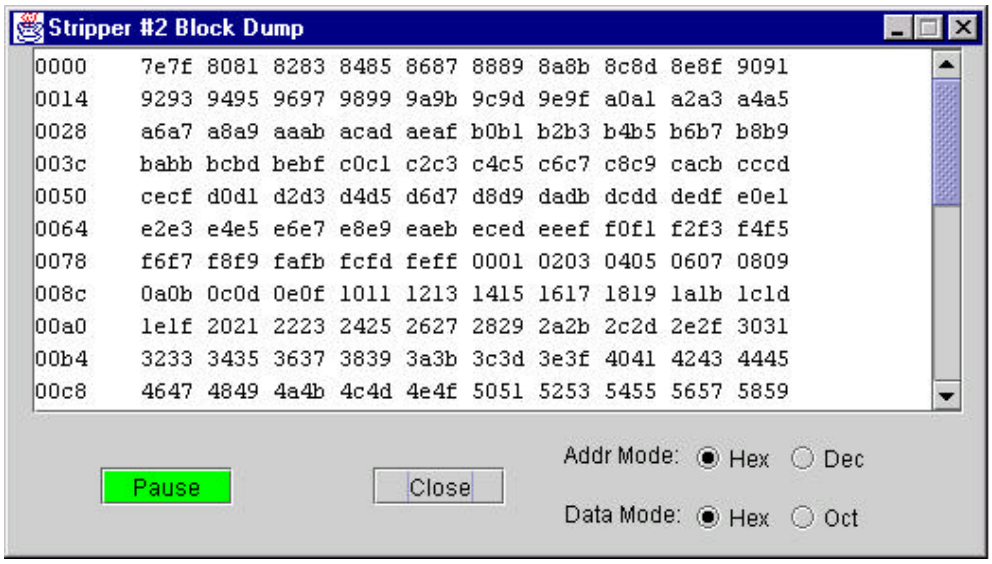
After the Stripper module has been stopped/paused, select "Resume" from the Run-time Menu to restart the module. The module's border changes from red (indicates stopped/paused state) to green (indicates run state).

Stripper-5.2.2 Pause

Select "Pause" from the Run-time Menu to temporarily stop the module's processing. The module's border changes from green (indicates run state) to red (indicates stopped/paused state). Once the module has been stopped, the "Configure" option of the module pop-up menu is available again and the module may be reconfigured. The Resume option must then be used to return to run mode.

Stripper-5.2.3 Show Output Block

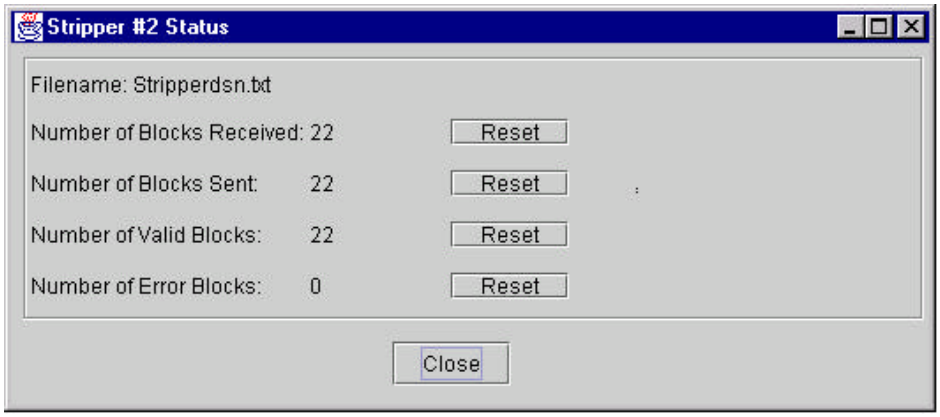
Select "Show Output Block" from the Run-time Menu to show data received and processed by the module. This produces a display similar to that shown below.



The address region (which is not used in this case) 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 last buffer processed. Buffers not shown while the display was frozen cannot be displayed. Click the **Close** button to dismiss the display.

Stripper-5.2.4 Show Status

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



This display shows the Stripper module's configuration filename, number of data blocks received, number of data blocks sent, number of valid blocks, and number of error blocks. The **Reset** button may be clicked to reset the number of blocks respectively. Click the **Close** button to dismiss the display.

Stripper-5.3 About

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

Stripper-6.0 Special Operating Instructions

The configuration file for this module is no longer being used.

TDM Command Generation (TDMCmdgen) Module

TDMCmdgen-1.0 Overview

The Generic Time Division Multiplexed (TDM) Command Generation module generates and transmits formatted TDM command sequences. Transmissions can be on a one-shot basis or repeated with a specified number of repeat transmissions and interval for repeating. It can also continuously send the sequence out until stopped by the operator.

TDMCmdgen-2.0 Inputs

The Generic TDM Command Generation module has no input channels.

TDMCmdgen-3.0 Outputs

The Generic TDM Telemetry module has a single output channel.

Output Channel	Description
1	Sends the generated TDM command sequences to another module.

TDMCmdgen-4.0 Container Items

Modification of the Generic TDM Command Generation module's container items via operator directives is not recommended, so they are not listed here.

TDMCmdgen-5.0 Displays

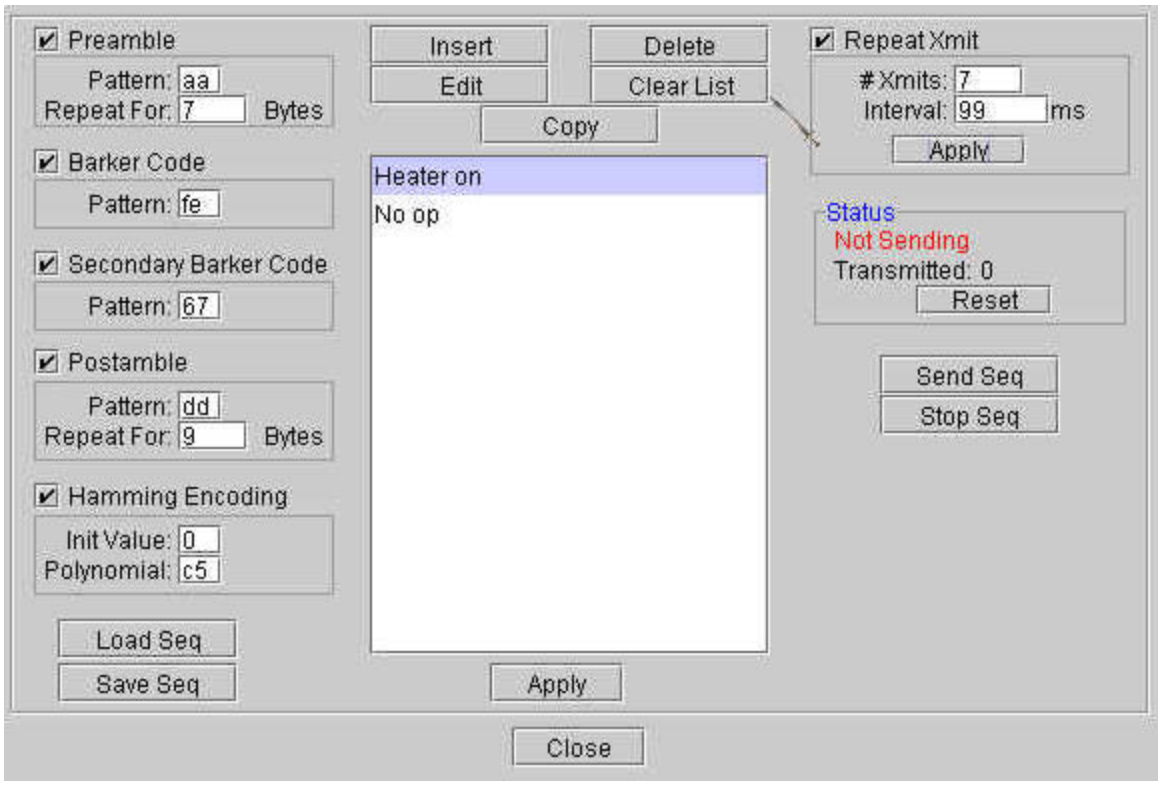
To access displays for a module, click in the center of the module in the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove the module. Note that there is no configuration screen for this module. All configuration is done at run time through the run-time displays. The run-time menu is accessible once the project is set to run.

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

TDMCmdgen-5.1 Run

The Run-time menu for TDMCmdgen contains the following items:

Run-time Menu Item	Description
Control	Request the main display
Resume	Resume the paused module
Pause	Pause the running module



This is the main display for the TDMCmdgen module, which allows building and sending of TDM command sequences. Sequences can be loaded and saved using the **Load Seq** and **Save Seq** buttons. Two files will be saved with the **Save Seq** button. The first is the sequence configuration and uses the name specified in the file chooser. The second contains the hex buffer that is sent out when **Send Seq** is pressed, and is named with the same name specified plus ".bin". Use the **Apply** button under the command list before attempting to transmit the defined sequence.

TDMCmdgen-5.1.1 Preamble

The preamble is a hexadecimal pattern that is transmitted before the command sequence. It typically consists of one byte repeated a given number of times. If a preamble is desired, the "Preamble" checkbox should be checked, and the values for the pattern and repeat count should be filled in. The pattern should be specified in hexadecimal. A repeat count of 0 is an error if "Preamble" is selected.

TDMCmdgen-5.1.2 Barker Code

The barker code signals the beginning of the incoming command sequence and sets it off from the preamble. Typically a barker code is one byte long. If a barker code is required, the "Barker Code" checkbox should be checked, and the hexadecimal value for the pattern filled in.

TDMCmdgen-5.1.3 Secondary Barker Code

The secondary barker code is provided for those spacecraft that require a 16 bit barker code. To utilize, fill in the “Barker Code” with the first byte of the barker code, then check the “Secondary Barker Code” checkbox and fill in the second byte of the barker code. This is also to be specified in hexadecimal.

TDMCmdgen-5.1.3 Postamble

The postamble is a way to signify that the command sequence is finished. If a postamble is desired, check the “Postamble” checkbox and fill in the pattern and repeat count. The pattern is specified in hexadecimal. A repeat count of 0 is an error if “Postamble” is selected.

TDMCmdgen-5.1.4 Hamming Encoding

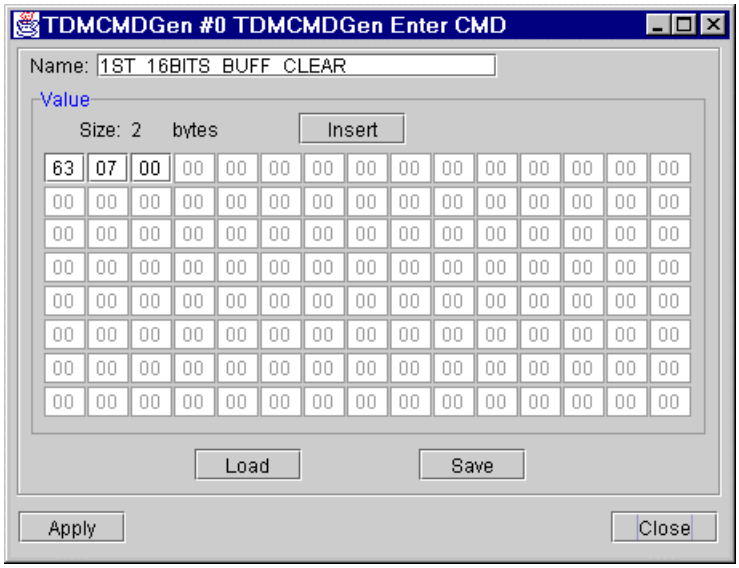
Hamming encoding can be added to each command in the command sequence by checking the “Hamming Encoding” checkbox. Default initial value is 0 and default polynomial is C5, which are the typical values. If some other initial value or polynomial is desired, it may be entered in hexadecimal.

TDMCmdgen-5.1.5 Command Sequence List

This is where the individual commands in the sequence may be specified. There are five buttons here that manipulate the list.

“Insert” inserts a command at the end of the list. Once inserted, this command may be edited by either double-clicking on its entry in the list or by selecting the command and pressing the “Edit” button. Initiating either action displays a command editing window. “Delete” will remove the selected command from the list, and “Clear” will remove everything from the list. “Copy” will add a duplicate of the currently selected command to the end of the list. It may be edited as usual after it is created in this manner.

From the command editing window mentioned above, the name of the command may be changed, and the command data may be entered in hexadecimal. The “Insert” button in this window inserts a hexadecimal command byte at the current cell location, and shifts the following hexadecimal values one cell position to the right. There are buttons here to both save and load the individual command. Both a configuration file and a straight hexadecimal file will be saved, with the hexadecimal file containing only the hexadecimal data specified for the command and named the same as the configuration file plus “.bin”. Always remember to press the **Apply** button to implement your changes.



TDMCmdgen-5.1.6 Repeat Xmit

When sending, this module will only send out the current sequence once unless "Repeat Xmit" is checked. If a repeated transmit is desired, check this checkbox, then fill in the desired repeat count and repeat interval. The repeat count determines how many times the entire sequence is transmitted. A repeat count of 0 indicates that the sequence should be repeated indefinitely until the **Stop** button is pressed. The repeat interval specifies how many milliseconds are to elapse between repeat transmissions. Note that the parameters for "Repeat Xmit" have their own **Apply** button. Use the **Apply** button before attempting to transmit.

TDMCmdgen-5.1.7 Status

This area displays whether the module is currently sending and how many sequences have been transmitted. The counter may be reset to 0 by pressing **Reset**.

TDMCmdgen-5.1.7 Send and Stop

Once a valid sequence has been defined, and transmit repeat parameters, if any, have been set, the "Send" button will send the sequence out. If "Repeat Xmit" is not checked, the sequence will be sent out once. If "Repeat Xmit" is checked and the repeat count parameter is greater than 0, the sequence will be transmitted the specified number of times. If "Repeat Xmit" is checked and the repeat count parameter is 0, the sequence will be sent out continuously until the **Stop** button is pressed. **Stop** may be pressed at any time to stop the transmission. However pressing **Stop** on a sequence that is to be sent out only once will not catch the transmission fast enough to prevent it from being sent.

TDM Command Ingest (TDMCMDIngest) Module

TDMCMDIngest-1.0 Overview

The TDM Command Ingest module receives a stream commands then performs the following validation checks on this stream:

- Command block format, it is including (1) barker (or sync) code (2) preamble (3)
- postamble.
- Spacecraft identifier (or address)
- Hamming (or Polynomial) code

This module was designed to meet TDM requirements; however, it should be usable to support similar spacecraft if necessary.

TDMCMDIngest-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Command stream	TDM specific	Commands parsed and validated. Event messages generated on valid or invalid commands.
None	Command database flat file	File formats and max. length of mnemonic check	Build an internal command lookup table

TDMCMDIngest-3.0 Outputs

Channel	Description
1	Command Counter and Mnemonic/Values pairs

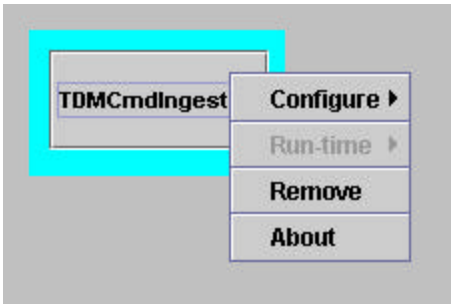
TDMCMDIngest-4.0 Container Items

Name	Type	Description
TDMCMDBlockValidationFlag	UnsByte	Flag to valid a command block
TDMCMDWordValidationFlag	UnsByte	Flag to valid a command word
TDMCMDConvertNrzMtoLFlag	UnsByte	Flag to convert from NRZM to L
TDMCMDDebugFlag	UnsByte	Flag to debug
TDMCMDUpdateCmdCounterFlag	UnsByte	Flag to update command counter
TDMCMDTotalBlockCount	UnsDword	Count number of total command blocks that are received
TDMCMDValidBlockCount	UnsDword	Count number of total valid command blocks
TDMCMDErrorBlockCount	UnsDword	Count number of total error command blocks
TDMCMDTotalWordCount	UnsDword	Count number of total command words that are received
TDMCMDValidWordCount	UnsDword	Count number of total valid

		command words
TDMCMDErrorWordCount	UnsDword	Count number of total error command words
TDMCMDRejectedCmdWordCount	UnsDword	Count number of total rejected command words
TDMCMDBlockBuffer	Container Buffer	Buffer that holds incoming command stream
TDMCMDPreamble	Word	Expected preamble byte pattern
TDMCMDPreambleLength	UnsByte	Expected preamble sequence length
TDMCMDPostamble	Word	Expected postamble byte pattern
TDMCMDPostambleLength	UnsByte	Expected postamble sequence length
TDMCMDBarker	Word	Expected barker (or sync) code
TDMCMDSCAddress	Word	Expected spacecraft address (or Id) code
TDMCMDCounter	UnsByte	Internal command counter
TDMCMDEntry1Byte	UnsWord	Byte location
TDMCMDEntry1Frame	UnsWord	Minor frame number
TDMCMDEntry1Subcom	UnsWord	Subcom depth
TDMCMDEntry1SubSel	UnsWord	Subcom type selection (0 for All, 1 for None, 2 for Subcom)
TDMCMDEntry2Byte	UnsWord	Byte location
TDMCMDEntry2Frame	UnsWord	Minor frame number
TDMCMDEntry2Subcom	UnsWord	Subcom depth
TDMCMDEntry2SubSel	UnsWord	Subcom type selection (0 for All, 1 for None, 2 for Subcom)
TDMCMDSpared	Word	Expected spare code
TDMCMDFixed	Word	Expected fixed code
TDMCMDFileName	SimString	File name of command database
TDMCMDSerialMode	UnsByte	Serial transferred mode

TDMCMDIngest-5.0 Displays

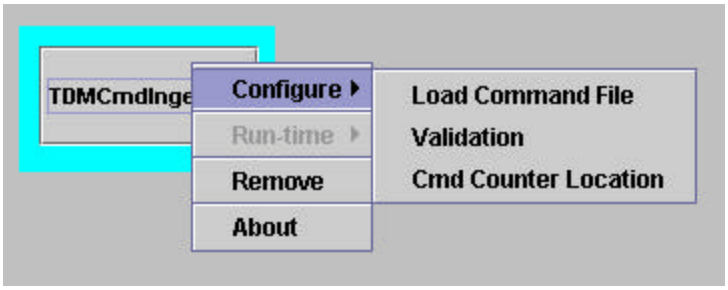
To access displays, click in the center of the module in the project window. The “Remove” option can be used during project design to remove this module. The “Configure” option should be used prior to running the project and is unavailable at run-time. The Run-time option is available only when the project is running.



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

TDMCMDIngest-5.1 Configuration

Selecting the "Configure" option produces a Configure menu. There are two items in the menu list shown as following:

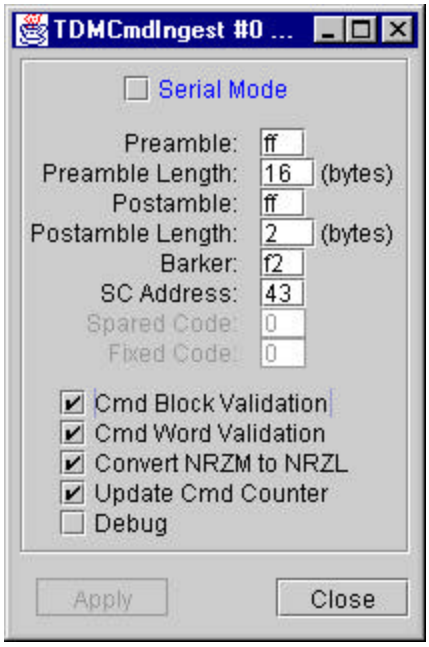


Configure Menu Item	Description
Load Command File	Configure the file name of command database
Validation	Configure the validation flags, pre and post ambles' byte pattern, their lengths, barker code and spacecraft address
Cmd Counter Location	Configure the command counter location

TDMCMDIngest-5.1.1 Load Command File

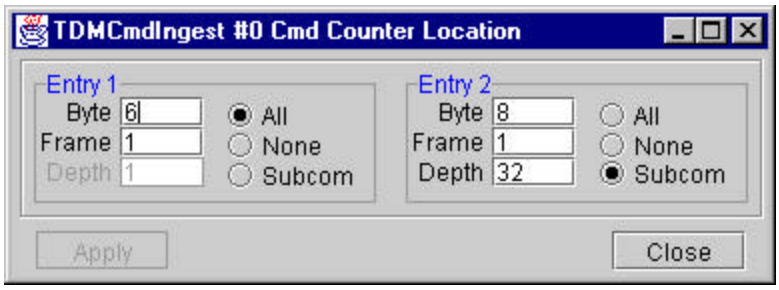
A user can enter a file name by typing it or browsing for a file.

TDMCMDIngest-5.1.2 Validation



Field	Description
Serial Mode	Serial/Block mode selection (checked for serial mode; unchecked for block mode)
Preamble	The byte pattern of the preamble
Preamble Length	The length of preamble in byte
Postamble	The byte pattern of the postamble
Postamble Length	The length of postamble in byte
Barker	Checking code of Barker/Sync
SC Address	Checking sequence of the Spacecraft Address
Spared Code	Checking code of spare when debug is enabled
Fixed Code	Checking code of fixed when debug is enabled
Cmd Block Validation	Enabled/Disabled command block validation
Cmd Word Validation	Enabled/Disabled command word validation
Convert NRZM to NRZL	Enabled/Disabled conversion of NRZM to NRZL
Update Cmd Counter	Enabled/Disabled update the command counter
Debug	Enabled/Disabled show debug messages on event log window

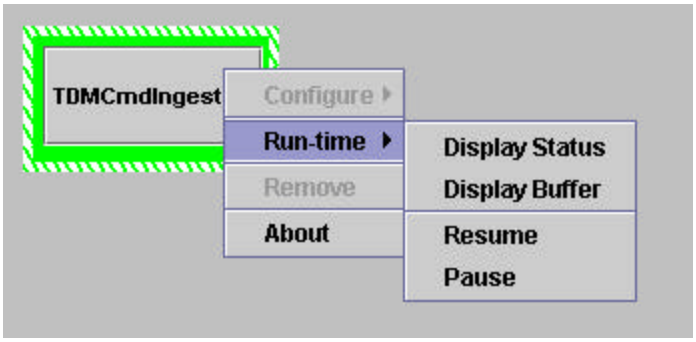
TDMCMDIngest-5.1.3 Cmd Counter Location



Field	Description
Byte	Byte location
Frame	Minor frame number per major frame
Depth	Subcom depth when “Subcom” is selected; otherwise it is ignored
All	All minor frames
None	It is just the one minor frame
Subcom	It is subcommed with depth Depth

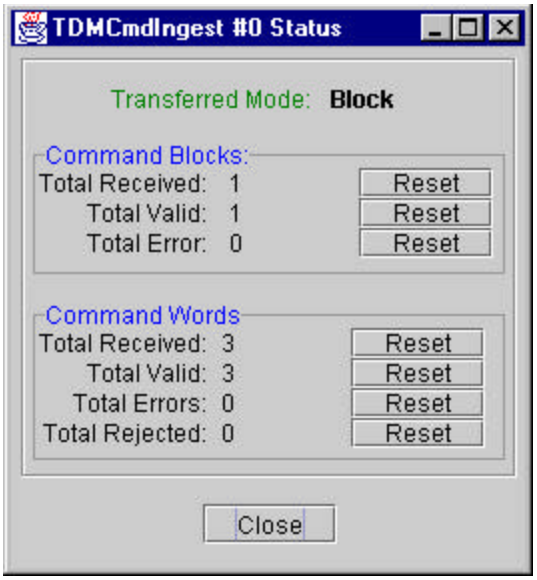
TDMCMDIngest-5.2 Run Time

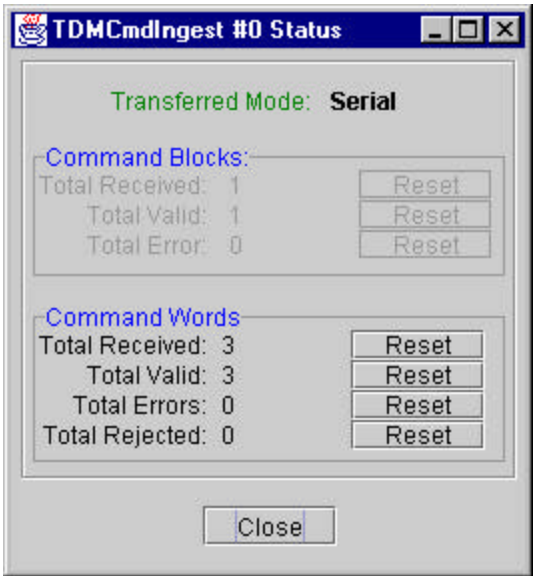
Selecting the “Run-time” option produces a Run-time menu. There are four items in the menu list shown as following:



Run-time Menu Item	Description
Display Status	Show status information
Display Buffer	Dump incoming command stream
Resume	Resume the Command Ingest module after a pause
Pause	Pause the Command Ingest module (enabled to configure)

TDMCMDIngest-5.2.1 Status

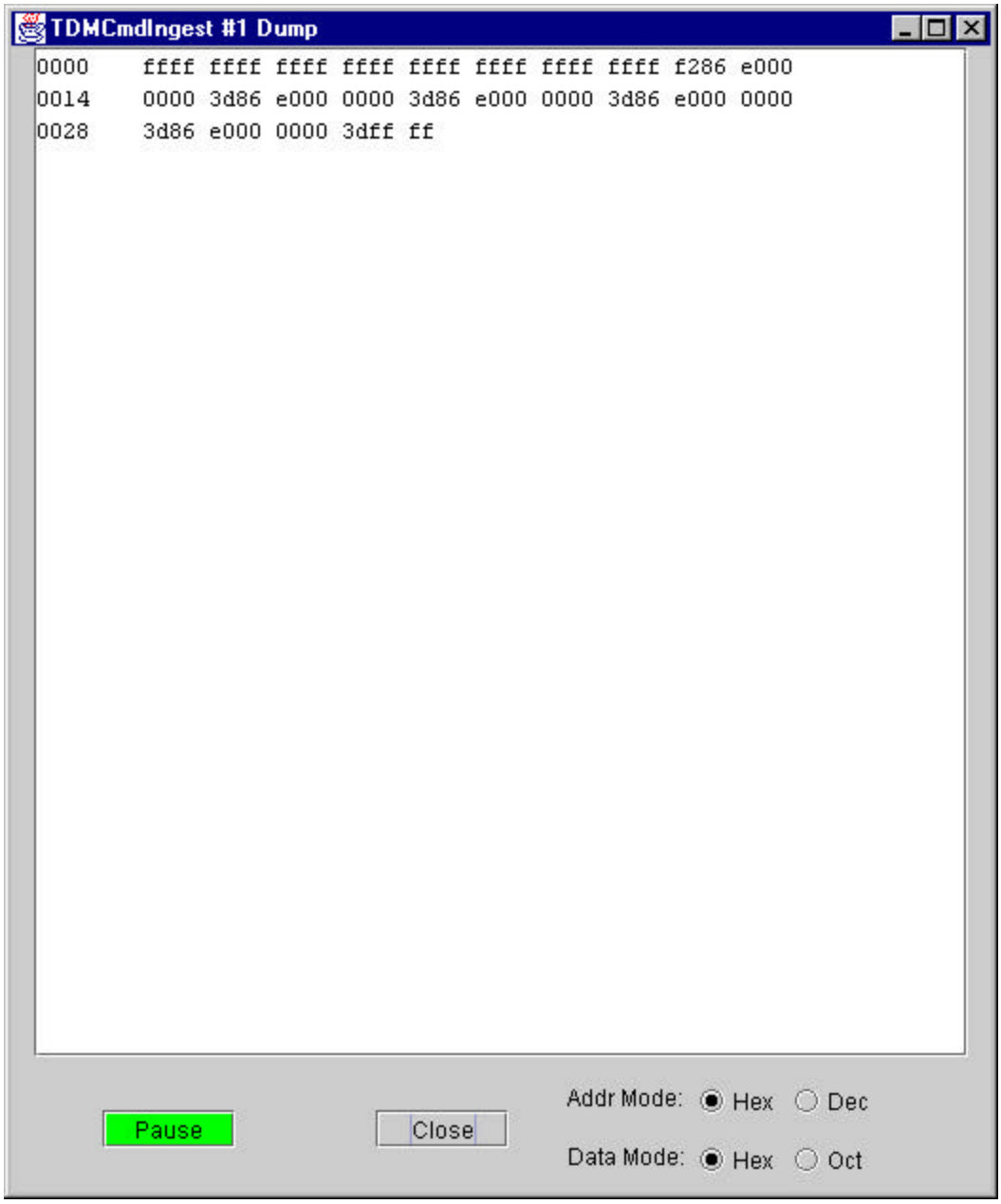




Field	Description
Transferred Mode	Block/Serial mode
Command Blocks:	Only for block mode; whole box will gray out if serial mode (in other words, it is meaningless if serial mode)
Total Received	Total number of command blocks that are received, it equals: Total Valid + Total Errors
Total Valid	Total number of valid command blocks
Total Errors	Total number of error command blocks
Command Words:	
Total Received	Total number of command words that are received, it equals: Total Valid + Total Errors + Total Rejected
Total Valid	Total number of valid command words
Total Errors	Total number of error command words
Total Rejected	Total number of command words are rejected

TDMCMDIngest-5.2.2 Buffer

The command stream buffer display shows the contents of the received command stream.



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 format.

TDM Demux (TDMDemux) Module

TDMDemux-1.0 Overview

The Generic Time Division Multiplexed (TDM) Demultiplexer module extracts up 6 different data streams from the input datastream.

TDMDemux-2.0 Inputs

The Generic TDM Demux module has one input channel.

Channel	Description
1	The input buffers to be demuxed.

TDMDemux-3.0 Outputs

The Generic TDM Demux module has six output channels.

Channel	Description
1-6	Outputs for each of up to six extracted data streams..

TDMDemux-4.0 Container Items

Modification of TDMDemux's container items via operator directives is not recommended, so they are not listed here.

TDMDemux-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 "Remove" option can be used during project design to remove the module. Note that there is no configuration screen for this module, and this choice is grayed out. All configuration is done at run time through the run-time displays. The run-time menu is accessible once the project is set to run.

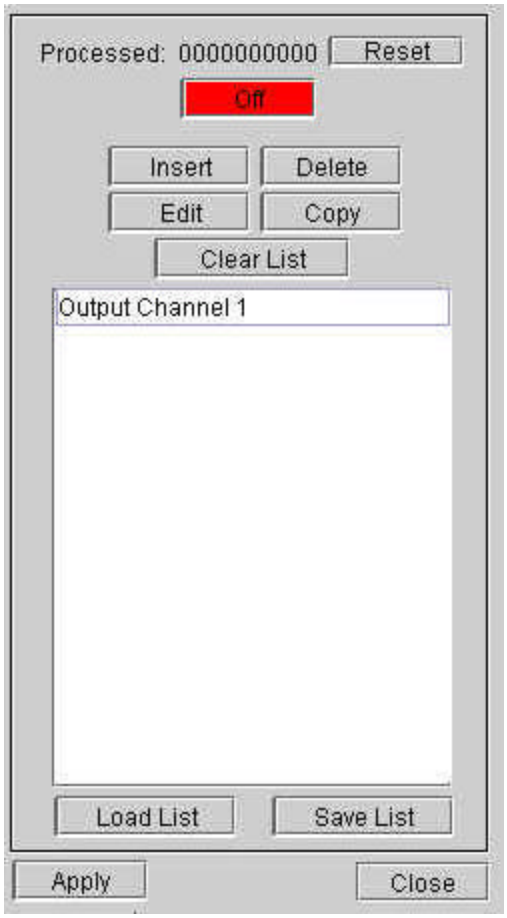
Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run menu for the module
Remove	Remove the module from the project
About	Display generic module information

TDMDemux-5.1 Run

The Run-time menu for TDMDemux contains the following items:

Run-time Menu Item	Description
Control	Request the main display
Resume	Resume the paused module

Pause	Pause the running module



This is the main display for the TDMDemux module, which allows definition of different data streams to be extracted from the input data. Data demux definitions can be loaded and saved using the **Load List** and **Save List** buttons.

Press the **Apply** button under the command list before attempting to transmit the defined sequence.

TDMDemux-5.1.1 Definition list

This is where the individual data streams to be extracted may be specified. There are five buttons here that manipulate the list.

Insert inserts a data stream at the end of the list. Once inserted, this data stream may be edited by either double-clicking on its entry in the list or by selecting the data stream and pressing the **Edit** button. Doing either will bring up the data stream editing window. From this window, the name of the data stream may be changed, the output channel specified, and the starting byte and length of the desired data stream may be defined. Always remember to press **Apply** to implement your changes. **Delete** will remove the selected command from the list, and **Clear List** will remove everything from the list. **Copy** will add a duplicate of the currently selected command to the end of the list. It may be edited as usual after it is created in this manner.

TDMDemux-5.1.2 On/Off

When idle, the button shown will be red, and when running, the button shown will be green. Pressing the button will start or stop the demuxing.

TDMDemux-5.1.3 Status

The module displays the number of input frames processed. There is a reset button that allows the resetting of this counter.

TDMDemux-6.0 Special Operating Instructions

Note that when defining data streams to be extracted, several items may be sent to the same channel. When this is done, they will be packaged into that channel's output frame in the order that they are defined in the list. The definitions for a single channel may be mixed with definitions for another channel; the order for the output frame is determined only on a channel-by-channel basis.

TDM DQM Module

TDM DQM-1.0 Overview

The Generic Time Division Multiplexed Data Quality Monitor (TDM DQM) module validates and de-muxes TDM data.

TDM DQM-2.0 Inputs

The Generic TDM DQM module has a single input channel.

Channel	Description
1	Accepts external data stream. Data must be synchronous and begin with the first byte of a minor frame.

TDM DQM-3.0 Outputs

The Generic TDM DQM 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 patterns to other modules.

TDM DQM-4.0 Container Items

Minor Frame Configuration Containers	Description
MinorFrameSize	The size in bytes of a complete minor frame.
MinorFrameMinValue	The minimum value of the minor frame counter.
MinorFrameMaxValue	The maximum value of the minor frame counter.
MinorFrameCounterSize	The size in bits of the minor frame counter.
MinorFrameCounterByteLocation	The byte location of the minor frame counter.
MinorFrameCounterBitLocation	The bit offset (from byte location) of the minor frame counter.
Major Frame Counter Configuration Containers	**Description**
MajorFrameCounterSize	The size in bits of the major frame counter.
MajorFrameCounterByteLocation	The byte location of the major frame counter.
MajorFrameCOunterBitLocation	The bit offset (from byte location) of the major frame counter.
Sync Configuration Containers	**Description**
SyncPattern	The sync pattern.
MaskPattern	The mask pattern.
SyncByte	The byte location of the sync pattern.
SyncBit	The bit offset (from byte location) of the sync pattern.

Telemetry Display Configuration Containers, note:*XX* is a decimal number representing a specific display (00-19)	**Description**
Display*XX*StartFrame	The first frame to begin extraction
Display*XX*Subcom	The subcom depth for extraction
Display*XX*StartByte	The byte to begin extraction with.
Display*XX*StartBit	The bit to begin extraction with (offset from byte)
Display*XX*BitLength	The number of bits to extract (max of 32)
Command Counter Display Configuration Containers, note:*X* is a decimal number representing a specific display (0-1)	**Description**
Counter*X*StartFrame	The first frame to begin extraction
Counter*X*Subcom	The subcom depth for extraction
Counter*X*StartByte	The byte to begin extraction with.
Counter*X*StartBit	The bit to begin extraction with (offset from byte)
Counter*X*BitLength	The number of bits to extract (max of 32)

The user should have no valid reason for accessing the other containers, so they are not listed here.

TDM DQM-5.0 Displays

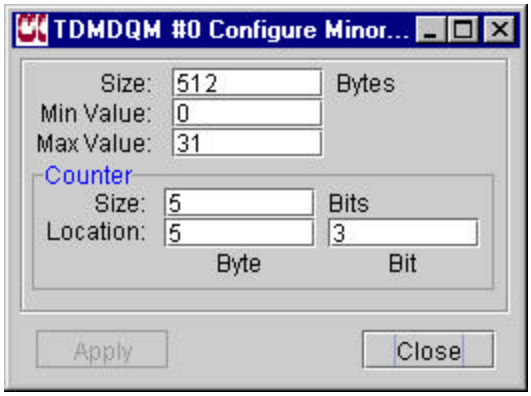
To access displays, click in the center of the module in the project window. The module pop-up menu will appear. The "Remove" option can be used during project design to remove the module. The "Configure" option should be used prior to running the project and is unavailable at run-time. The Run-time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

TDM DQM -5.1 Configure

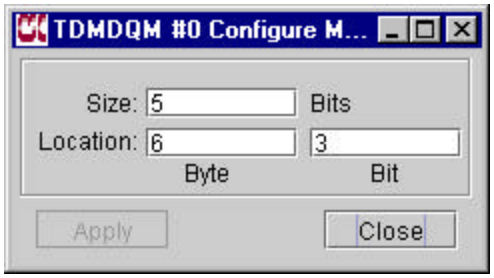
There are seven different configuration screens: Minor Frame, Major Frame Counter, Sync, Outputs, Displays, Load, and Save. A description of each screen follows.

TDM DQM-5.1.1 Configure Minor Frame



Field Name	Description
Size	The size of a complete minor frame (in bytes). This must be configured for TDM DQM to operate. If this is not configured, TDM DQM will discard all data sent to it.
Min Value	The minimum value of the minor frame counter.
Max Value	The maximum value of the minor frame counter.
Counter Size	The size of the minor frame counter (in bits). TDM DQM module can handle a maximum of 32 bit minor frame counter.
Counter Location (byte)	The byte location of the minor frame counter. The first byte in a minor frame is byte 0.
Counter Location (bit)	The bit location (offset from byte location) of the minor frame counter. The most significant bit is bit 0, the least significant bit is 7.

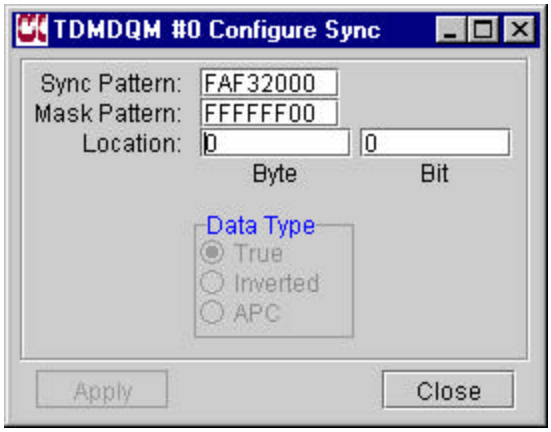
TDM DQM-5.1.2 Configure Major Frame Counter



Field Name	Description
Size	The size (in bits) of the major frame counter. TDM DQM module can handle a maximum of 32 bit major frame counter.
Location (byte)	The byte location of the major frame counter. The first byte in a major frame is byte 0.
Location (bit)	The bit location (offset from byte location) of the major frame counter. The most significant bit is 0, the least significant is 7.

To disable the major frame counter validation, set the size equal to zero.

TDM DQM-5.1.3 Configure Sync



Field Name	Description
Sync Pattern	The sync pattern used to validate the data.
Mask Pattern	The mask pattern to use on the data before comparison with the sync pattern.
Location (byte)	The byte location of the sync pattern. The first byte in a major frame is byte 0.
Location (bit)	The bit location (offset from byte location) of the sync pattern. The most significant bit is 0, the least significant is 7.

The sync pattern is used as the first step in validating the data. If the sync pattern is not present in the location specified, the data is discarded and the frame drop count is incremented (see Runtime: Display Status 5.2.1).

To disable the sync pattern validation, enter a mask value of zero. This will allow all data to go through to the other validation processes.

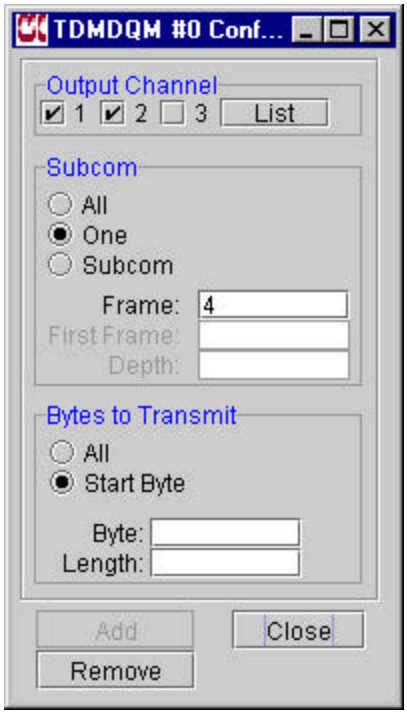
The sync pattern and mask pattern are 32-bit values. In order to search for a sync at the beginning of the minor frame, the sync and mask values must be padded with zeros. For example, a sync pattern of FAF320 is not the same as a sync pattern of FAF32000. The first one (FAF320) is interpreted as 00FAF320 which looks for the first F at the second byte after the specified location. The second value (FAF32000) looks for the first F at the first byte of the location.

The sync pattern must complement the mask pattern. This means that for each bit in the sync pattern that equals one, the corresponding bit in the mask pattern must also equal one. The module will check this when the Apply button is pressed. If there is a problem, the sync pattern will be changed so that for each bit in the mask pattern that equals zero, the corresponding bit in the sync pattern will also equal zero.

An example of the above situation:

User inputs:	sync pattern:	FAF32000
	Mask pattern:	0FFFFF00
Sync is changed to:		0AF32000

TDM DQM-5.1.4 Configure Outputs



Field Name	Description
Output Channel	Select which channel(s) to configure.
Subcom	Which frame(s) to output. See table below for details.
Bytes to Transmit	What bytes in the frame to transmit. See table below for more information.

Subcom Fields	Description
All	Transmit every frame.
One	Transmit a single minor frame per major frame. When the minor frame counter equals the value in the Frame field, the frame gets transmitted.
Subcom	Transmit multiple, non-continuous minor frames. When the minor frame counter equals the value in the First Frame field, or the First Frame field plus a multiple of the Depth field, the frame gets transmitted. For example, to transmit the frames with an even minor frame counter select Subcom, with a First Frame zero, and a Depth of two.

Bytes To Transmit Fields	Description
All	Transmit the entire minor frame.
Start Byte	Byte is the byte to start transmission (first byte of a minor frame is zero), Length is the length in bytes to transmit. The length of data transmitted will be exactly the value in the

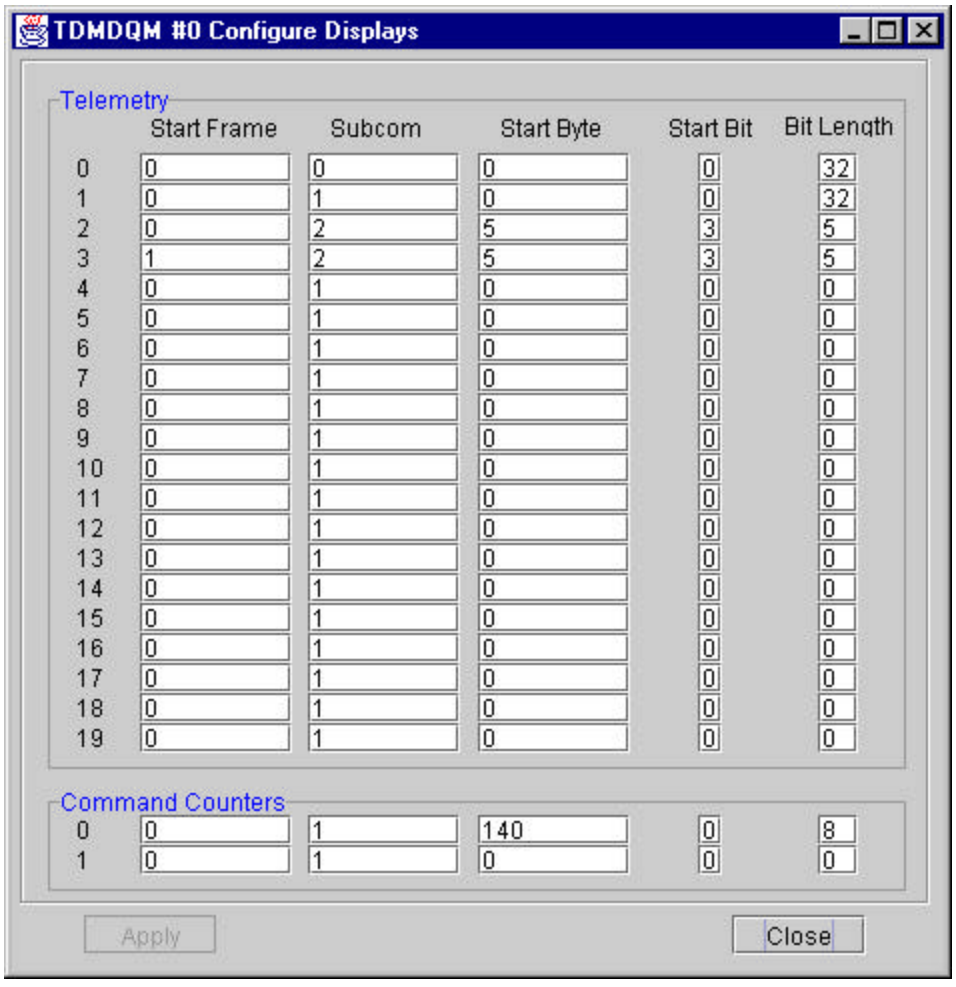
	Length field. If this goes past the end of a minor frame, then the data will be zero-filled at the end.

To configure the outputs for the TDM DQM follow the following steps:

- Select which output channel(s) to use. To see what is already configured click the List button and look at the event log. A new entry to an output channel will erase any previous configuration on that channel.
- Choose a subcom depth.
- Choose what bytes to transmit.
- Click the Add button. This will add (perhaps overwrite) the new configuration to all output channels selected.

To remove configurations, select the channel(s) to clear and click the remove button.

TDM DQM-5.1.5 Configure Displays

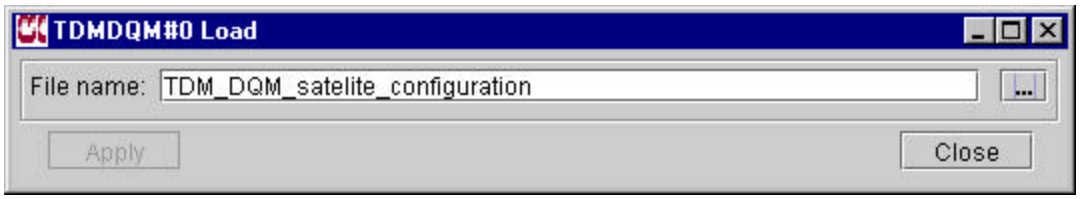


Field Name	Description
Start Frame	The frame to start extracting from.
Subcom	The interval between extracted frames. A subcom of zero will extract data from a single minor frame (the

	Start Frame). A subcom of one will extract data from every minor frame starting with the Start Frame.
Start Byte	The first byte to start extracting data from. The first byte of a minor frame is byte zero.
Start Bit	The first bit to start extracting from. The most significant bit is 0, the least significant bit is 7.
Bit Length	The number of bits to extract. The maximum value allowed is 32 bits.

This area is used to display areas of the telemetry (max size of 32 bits) in real time. The only difference between the telemetry and command counter extraction is the name.

TDM DQM-5.1.6 Configuration Load

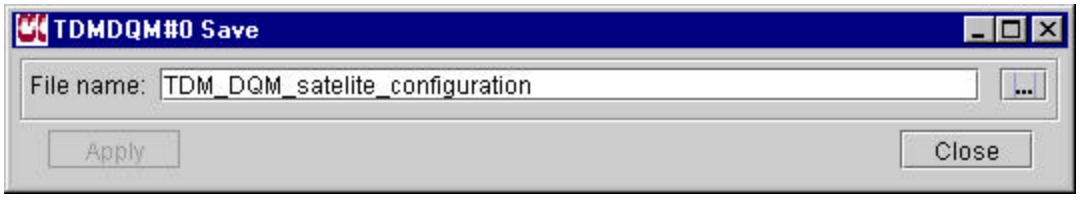


Field Name	Description
File Name	The name of the file to use for configuring the TDM DQM module

TDM DQM module can load and save configurations independent from the project load/save. The module specific load/save will save the entire configuration, while the project load/save will not save the Display information (due to internal memory restrictions).

Using the module load/save also allows the user to switch TDM DQM configurations without having to close the existing project and bring up a new project.

TDM DQM-5.1.7 Configuration Save



Field Name	Description
File Name	The name of the file to use for saving the TDM DQM module configuration.

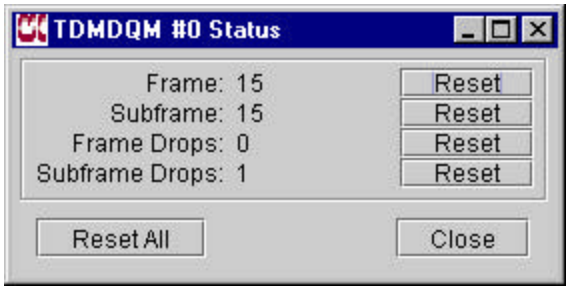
See notes above (Load 5.1.6) for description and benefits of using the TDM DQM module specific load/save over the project load/save.

TDM DQM-5.2 Run

The Run-time menu for TDM DQM contains the following items.

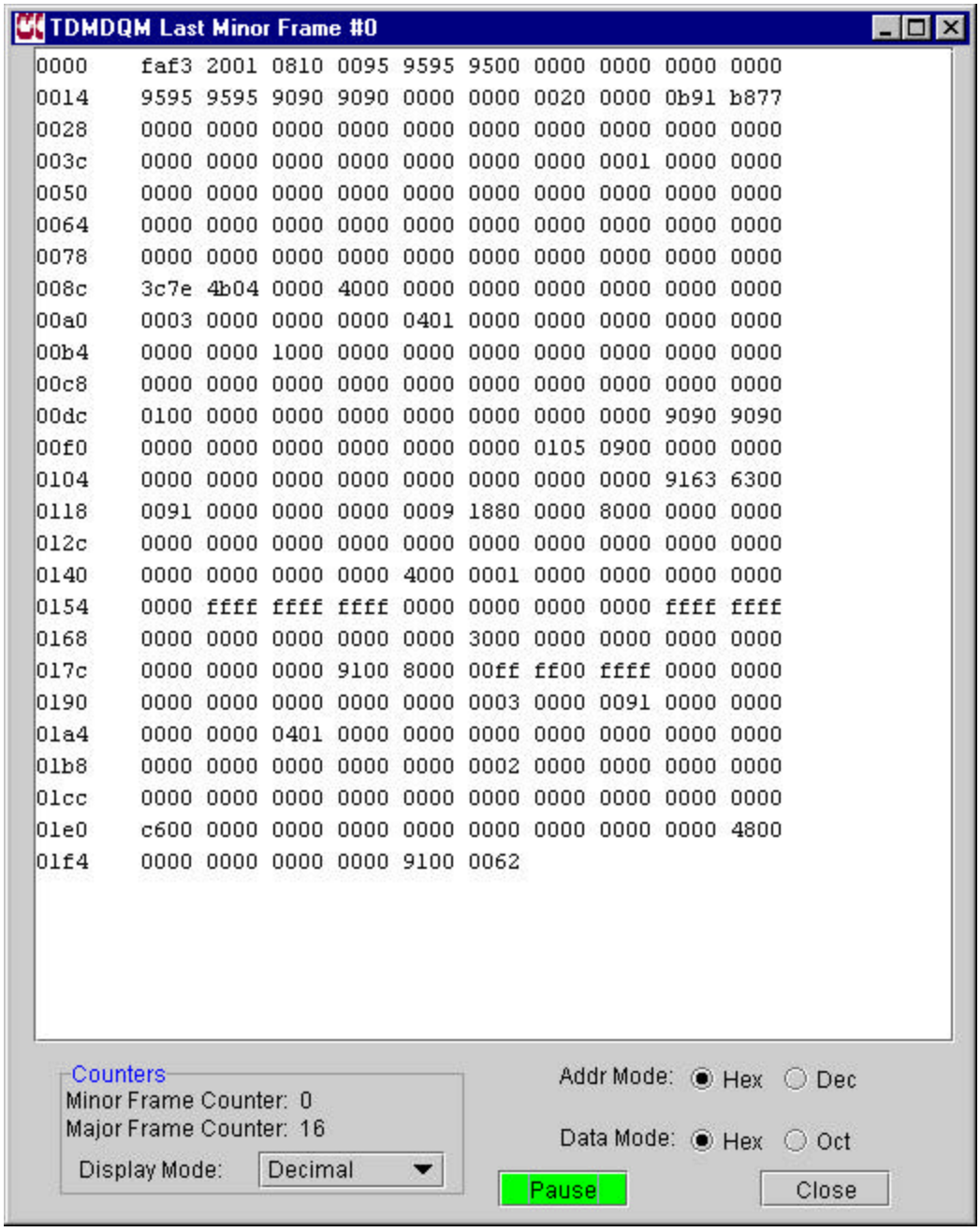
Run-time Menu Item	Description
Display Status	Display various counts.
Display Last Minor Frame Seen	A dump of the last complete minor frame seen (sync must match).
Displays	Display of user-configured telemetry extraction.
Resume	Resumes module.
Pause	Pauses module.

TDM DQM-5.2.1 Display Status



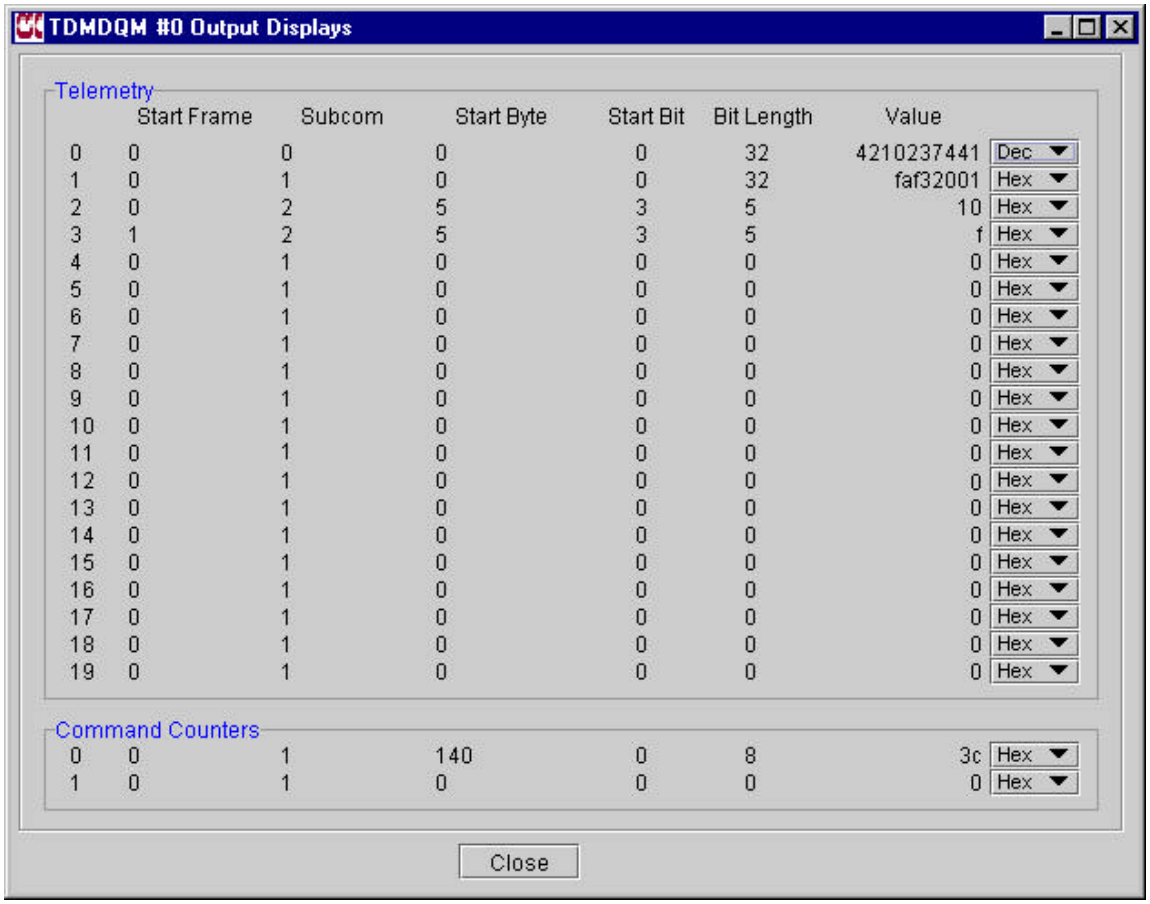
Field Name	Description
Frame	The number of valid frames received (sync pattern is valid).
Subframe	The number of subframes (minor frames) received.
Frame Drops	The number of invalid blocks of data (sync pattern does not match).
Subframe Drops	The number of times there is a sequence error in the subframe counter.

TDM DQM-5.2.2 Display Last Minor Frame



This is a dump of the last complete minor frame seen. Both minor and major frame counters are displayed in the bottom left corner of the display. The user can select to display the counters as decimal, hexadecimal, or octal.

TDM DQM-5.2.3 Displays



This is the display of extracted telemetry. The user can display the value as decimal, hexadecimal, or octal. All other values on this display are decimal values. Currently the size and location of the data to be extracted is not configurable while the module is running.

TDM DQM-6.0 Special Operating Instruction

Currently the TDM DQM module depends on being given synchronous data. Also, the first byte that TDM DQM sees must be the first bite of a minor frame (TDM DQM can not start by receiving half of a minor frame).

During runtime, if the data gets off track (extra/missing byte(s)) the TDM DQM will discard all data. This is because the sync will not match up to the location in configuration.

If this happens, there is a back door for reseting the module. When the module is reset it will assume the next byte received is the first byte of a minor frame. This reset can be done in run-time. To execute the reset, on the TDM DQM command line type:

set reset 1

TDM Telemetry Generator (TDMGen) Module

TDMGen-1.0 Overview

The Generic Time Division Multiplexed (TDM) Telemetry module generates and transmits formatted TDM telemetry frames, and can take up to four external data streams and multiplex (mux) them together into the outgoing data stream.

TDMGen-2.0 Inputs

The Generic TDM Telemetry module has four input channels.

Channel	Description
1-4	Any of these can take external data points from other modules. They may be persistent or not. Any of these can be configured as an external data stream that is to be muxed into the telemetry stream. Section 5 provides details about configuring this module.

TDMGen-3.0 Outputs

The Generic TDM Telemetry module has a single output channel.

Channel	Description
1	Sends the generated TDM frames to another module.

TDMGen-4.0 Container Items

Modification of the Generic TDM Telemetry module's container items via operator directives is not recommended, so they are not listed here.

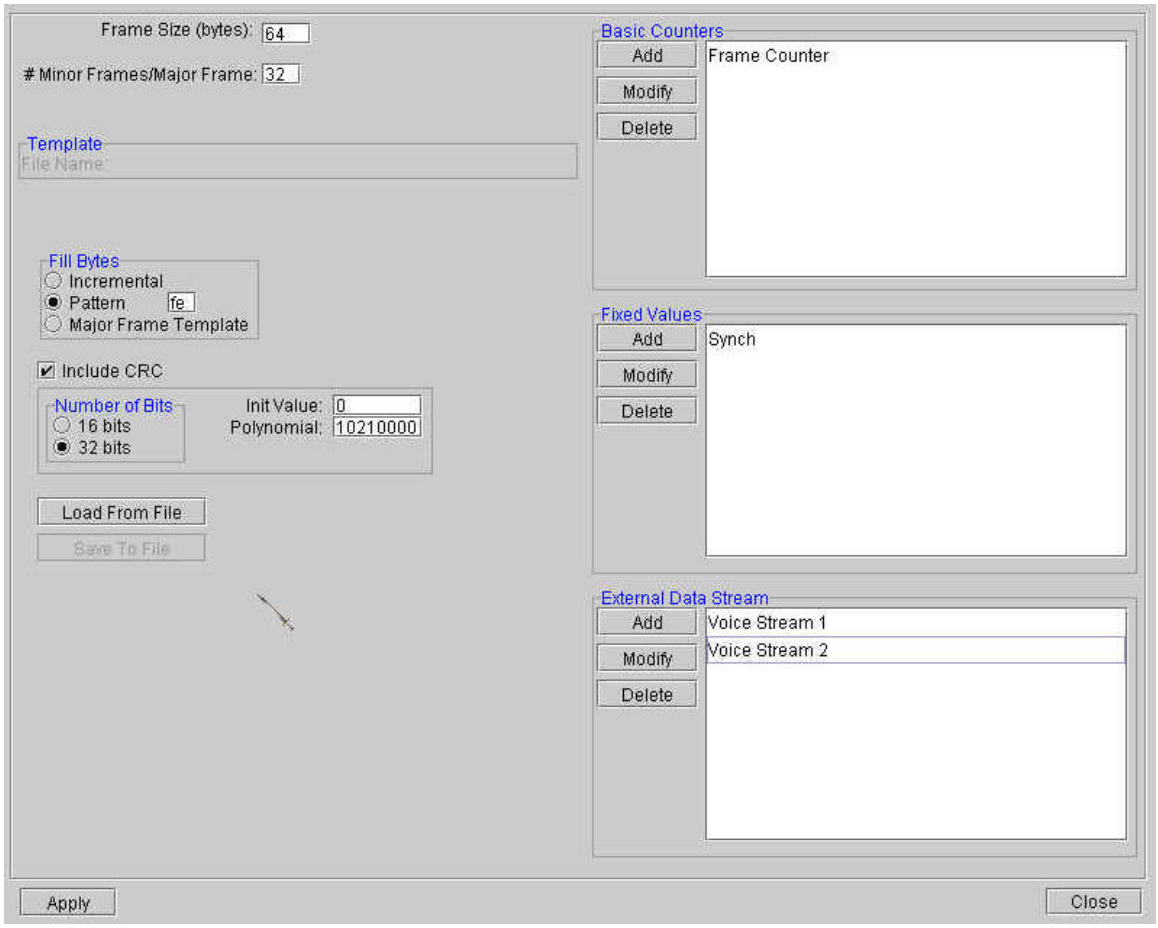
TDMGen-5.0 Displays

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

TDMGen -5.1 Configure

Clicking the “Configure” option causes the main configuration display to appear.



“Frame Size (bytes)” is the size of a minor frame in bytes, entered as a decimal integer, and “# Minor Frames/Major Frame” is the number of minor frames in a major frame, also entered as a decimal integer.

Click **Apply** to apply the changes to all values and perform some error checking on the supplied parameters, and **Close** to exit the window. It is important to remember to **Apply** any changes before closing the window.

TDMGen-5.1.1 Fill Bytes

The area labeled “Fill Bytes” displays three choices: “Incremental”, “Pattern”, and “Major Frame Template”. This is where to specify the pattern to be used in the otherwise unfilled portions of the frame.

“Incremental” will place a 0x00 in byte 0 of the frame, 0x01 in byte 1, etc.

“Pattern” allows a one byte hexadecimal pattern to be specified in the accompanying text entry field. This pattern must be entered in hexadecimal.

"Major Frame Template" allows the selection of a file to use as a template for one major frame of data. Selecting this will cause a standard SIMSS file selection box to appear. Once the file has been chosen, it will be reflected in the "Template" area just above the "Fill Bytes" area. The file will be interpreted as raw hexadecimal, and must be long enough to contain the specified number of minor frames of the specified size. If the **Apply** button is pressed from the main configuration window, and an error or warning message about the processing of the file is displayed, the module should not be started until a valid configuration is entered.

TDMGen-5.1.2 CRC

The checkbox labeled "Include CRC" is used to indicate whether a CRC should be calculated and inserted into each frame's final bytes. When this box is checked, the values in the box directly below it will be enabled for user modification.

"Number of Bits" contains two choices to specify either a 16-bit or 32-bit CRC. The CRC will be placed in the final bytes of each minor frame; the last two bytes when 16 bits is chosen and the final four bytes of each frame; when 32 bit is chosen.

"Init Value" is the initial value of the shift register used for calculating the CRC. The default is 0x00000000, but any hexadecimal pattern may be specified.

"Polynomial" is the polynomial used for the calculation, specified as a hexadecimal pattern. The hexadecimal pattern is determined based on the coefficients of the desired polynomial. The initial term of the polynomial, which is only used to determine the polynomial's order (16 or 32), should be left off before determining this hexadecimal value.

For example, the default polynomial is $x^{16} + x^{12} + x^5 + 1$, a commonly used polynomial for CRC calculation. The polynomial is of order 16 and can be written as

$$1x^{16} + 0x^{15} + 0x^{14} + 0x^{13} + 1x^{12} + 0x^{11} + 0x^{10} + 0x^9 + 0x^8 + 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0.$$

The hexadecimal pattern for the CRC algorithm is obtained by dropping the first term, and taking the coefficients of the remaining terms:

0001 0000 0010 0001 = 0x1021.

This is the value entered in the "Polynomial" text box. Note that for 16 bit CRCs the value is left shifted. In the case of the default value, this means that the number appears as 0x10210000.

TDMGen-5.1.3 Load From File / Save To File

Once the desired configuration has been entered, applied, and validated, it may be saved to a file. **Pressing Save to File** will invoke the standard SIMSS file selection dialog. Pressing **Load from File** will invoke the same file selection dialog, and allows a TDMGen configuration file to be chosen. Loading a configuration will replace whatever configuration already exists.

TDMGen-5.1.4 Basic Counters

The list box labeled “Basic Counters” allows the setting of basic telemetry counters that appear in every frame or in a subset of subcommed minor frames. This will most commonly be used for setting the major/minor frame counter, but multiple counters may be set and maintained; simple ramping functions may be set in this manner, for example.

Clicking the **Add** button will add a counter to the list. To modify, select the new counter and press **Modify**, or double-click in the list on the new counter. This will cause the Basic Counter configuration window to appear.

The “Name” field can be used to give the counter a meaningful name. The “#Minor/Major” field will default to the value set in the “# Minor Frames/Major Frame” field in the main configuration window, but may be changed here. This will not change the setting for number of minor frames per major frame, only the value for the counter currently being configured.

To include the major counter in the telemetry stream, check “Include Major Counter”; to include the minor counter, check “Include Minor Counter”. This will enable the three entry fields below the selection. In “Start Byte”, enter the number for the starting byte within the telemetry stream. Note that the bytes are numbered from 0 to <size of major frame> - 1. In the “Size” entry field, enter the number of bytes the telemetry value will occupy. In the increment field, if an increment value other than 1 is desired, it may be specified. It is an error to set the increment to 0. The “Start Value” is the value with which the counter will begin counting. The default start value is 0.

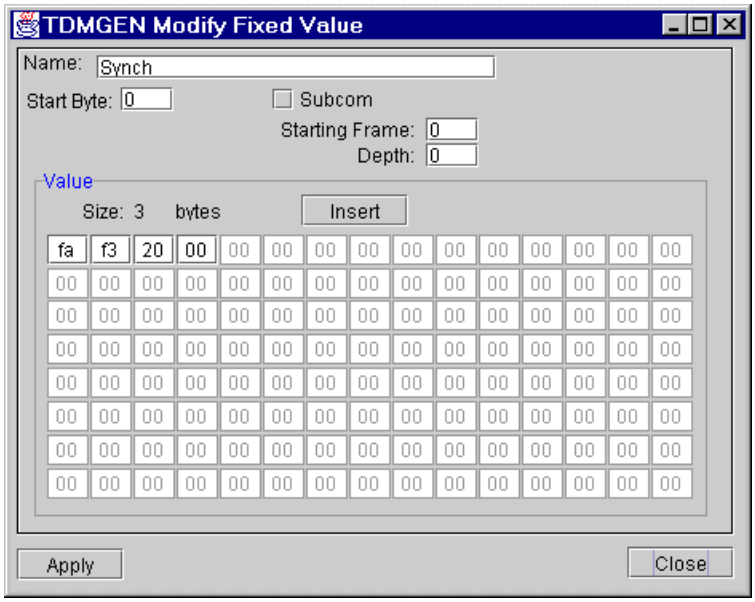
Please see section 5.1.7 for a description of “Subcom”, which is common to counters, fixed values (section 5.1.5), and external data streams (section 5.1.6).

Apply will apply the settings, and **Close** will exit the window.

TDMGen-5.1.5 Fixed Values

The list box labeled “Fixed Values” allows the setting of basic telemetry fixed values that appear in every frame or in a subset of subcommed minor frames. This will most commonly be used for setting the synch pattern, but other fixed bit patterns may also be specified.

Clicking the **Add** button will add a fixed value to the list. To modify, select the new item and press **Modify**, or double-click in the list on the new item. This will cause the Fixed Value configuration window to appear.



The “Name” field can be used to give the fixed value a meaningful name. In “Start Byte”, enter the number for the starting byte within the telemetry stream. Note that the bytes are numbered from 0 to <size of major frame> - 1. In the “Value” array, enter the telemetry point’s value. Note that the “Size” field will automatically update to match the size of the entered value in bytes. The “Insert” button in this window inserts a hexadecimal command byte at the current cell location, and shifts the following hexadecimal values one cell position to the right.

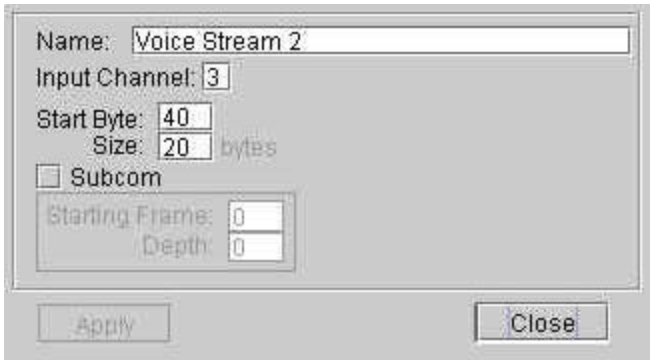
Please see section 5.1.7 for a description of “Subcom”, which is common to counters (section 5.1.4), fixed values, and external data streams (section 5.1.6).

Apply will apply the settings, and **Close** will dismiss the window.

TDMGen-5.1.6 External Data Streams

The list box labeled “External Data Streams” allows the setting of locations for external data streams that are fed into the module. This can be used to allow multiplexing of several data streams together.

Clicking the **Add** button will add a single external data configuration to the list. To modify, select the new item and press **Modify**, or double-click in the list on the new item. This will cause the External Data Stream configuration window to appear.



The "Name" field can be used to give the data stream location a meaningful name. In "Start Byte", enter the number for the starting byte within the telemetry stream. Note that the bytes are numbered from 0 to <size of major frame> - 1. In the "Size" field, enter the number of bytes to be inserted into the frame from the stream.

Please see section 5.1.7 for a description of "Subcom", which is common to counters (section 5.1.4), fixed values (section 5.1.5), and external data streams.

Apply will apply the settings, and **Close** will dismiss the window.

Several External Data Stream location items may be defined for the same channel to put sequential data chunks from the incoming stream into different areas of the minor frame.

Note that the order in which they are defined in the list is the order in which they will be inserted into the frame, and that the input data is read sequentially from the incoming channel.

Also note that it is up to the operator to make sure the data rate is compatible with the configuration entered into the TDM Telemetry Generation module.

TDMGen-5.1.7 Subcom

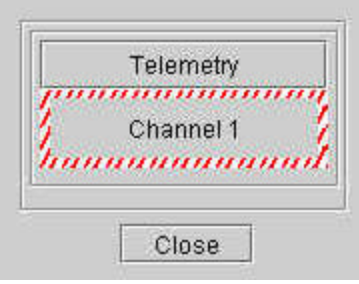
Each of the three list items, counters (section 5.1.4), fixed values (section 5.1.5) and external data streams (section 5.1.6), have a subcom option. Checking the subcom checkbox in any of these configuration screens will enable the two related fields "Starting Frame" and "Depth". In "Starting Frame", enter the number of the first minor frame of the major frame that the value is to appear. In "Depth", enter the number of frames before the value is to be repeated. For example, 2 indicates that starting with the "Starting Frame", the value will appear every other frame. A value of 1 indicates the value will show up in every frame following the "Starting Frame". A value of 0 is reserved to mean that the value shows up only once per major frame in the minor frame indicated in "Starting Frame".

TDMGen-5.2 Run

The Run-time menu for TDMGen contains the following three items.

Run-time Menu Item	Description
Control	Request the main display

Stop	Stop the module
Restart	Restart the module



This is the topmost display for the TDMGen module, allowing access to basic information about its current state, and the enabled/disabled status of telemetry transmission. The latter is indicated by colored diagonal stripes around the appropriate box: 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. Left clicking on the "Channel" box will display a menu that allows enable/disable of the channel.

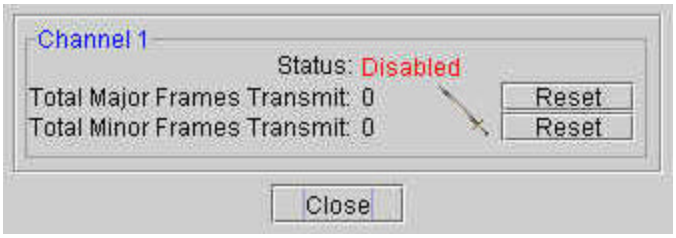
Channel Menu Item	Description
Start	Enables the channel and begins transmission of frames.
Stop	Disables the channel and halts transmission of frames.

Left clicking on the **Telemetry** button in the main display allows access to subordinate displays. The details are described in the following subsection.

TDMGen-5.2.1 Telemetry

Clicking on the **Telemetry** button displays the telemetry control menu.

Telemetry Menu Item	Description
Display Status	Request the status display
Reset Counters	Reset all counters in the telemetry to zero.



This is the main display for the TDMGen module, allowing access to basic information about its current state. The counts are for the total number of frames transmitted so far. The **Reset** buttons displayed will only reset the total displayed to zero; actual counters within the telemetry are unaffected by these buttons.

TDMGen-6.0 Special Operating Instructions

Currently, TDMGen only runs in serial mode, responding to timed interrupts obtained from the Serial Output module. It is important to remember to set the frame size in the Serial Output module to the same value as is set in the TDMGen module. If this is not done, Serial Output will be waiting for a buffer of a different size than is being sent, and will not send additional interrupts. An alternative to using the Serial Output module is to use the Test module, which now will produce the same interrupts.

TDRSS Checksum Module

TDRSS Checksum-1.0 Overview

The TDRSS Checksum module receives a TDRSS telemetry stream. A checksum is performed.

This module was designed to meet TDRSS requirements; however, it should be usable to support similar spacecraft if necessary.

TDRSS Checksum-2.0 Inputs

Ch.	Data expected	Processing performed
1	TDRSS Telemetry Stream	Checksum computed. Command counters updated. Telemetry verification of commands updated.

TDRSS Checksum-3.0 Outputs

Channel	Description
1	TDRSS telemetry stream.

TDRSS Checksum-4.0 Container Items

Name	Type	Description
TDRSSCheckSumPerform	Int	Flag to determine if checksum should be calculated
TDRSSCheckSumFrameSize	Int	Frame size
TDRSSCheckSumBcRt	Int	Dealing with BC or RT
TDRSSCheckSumParam	Int	Which parameter to change
TDRSSCheckSumValue	Int	What the new value is
TDRSSCheckSumPersistant	Int	Persistancy of user overwrite
TDRSSCheckSumOutBuffer	Int	Output buffer
frames	Int	Number of frames received
errors	Int	Number of errors received
scid	Int	Spacecraft ID
minorFrame	Int	Current Minor Frame
crcCalculated	Int	Calculated CRC on last error
crcReceived	Int	Received CRC on last error

TDRSS Checksum-5.0 Displays

To access displays, click in the center of the command ingest 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 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.

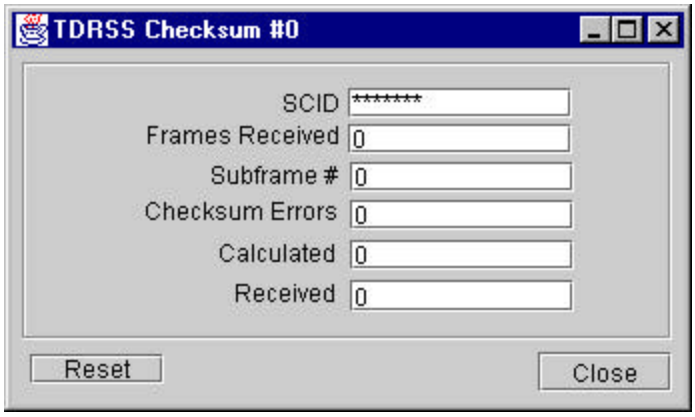
Module Pop-Up Menu Item	Description
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display Information about the module

TDRSS Checksum-5.1 RunTime Displays

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

Run-time Menu Item	Description
Checksum	Show checksum information
Display Output Buffer	Dump incoming command stream
Resume	Resume the module after a pause
Pause	Pause the module

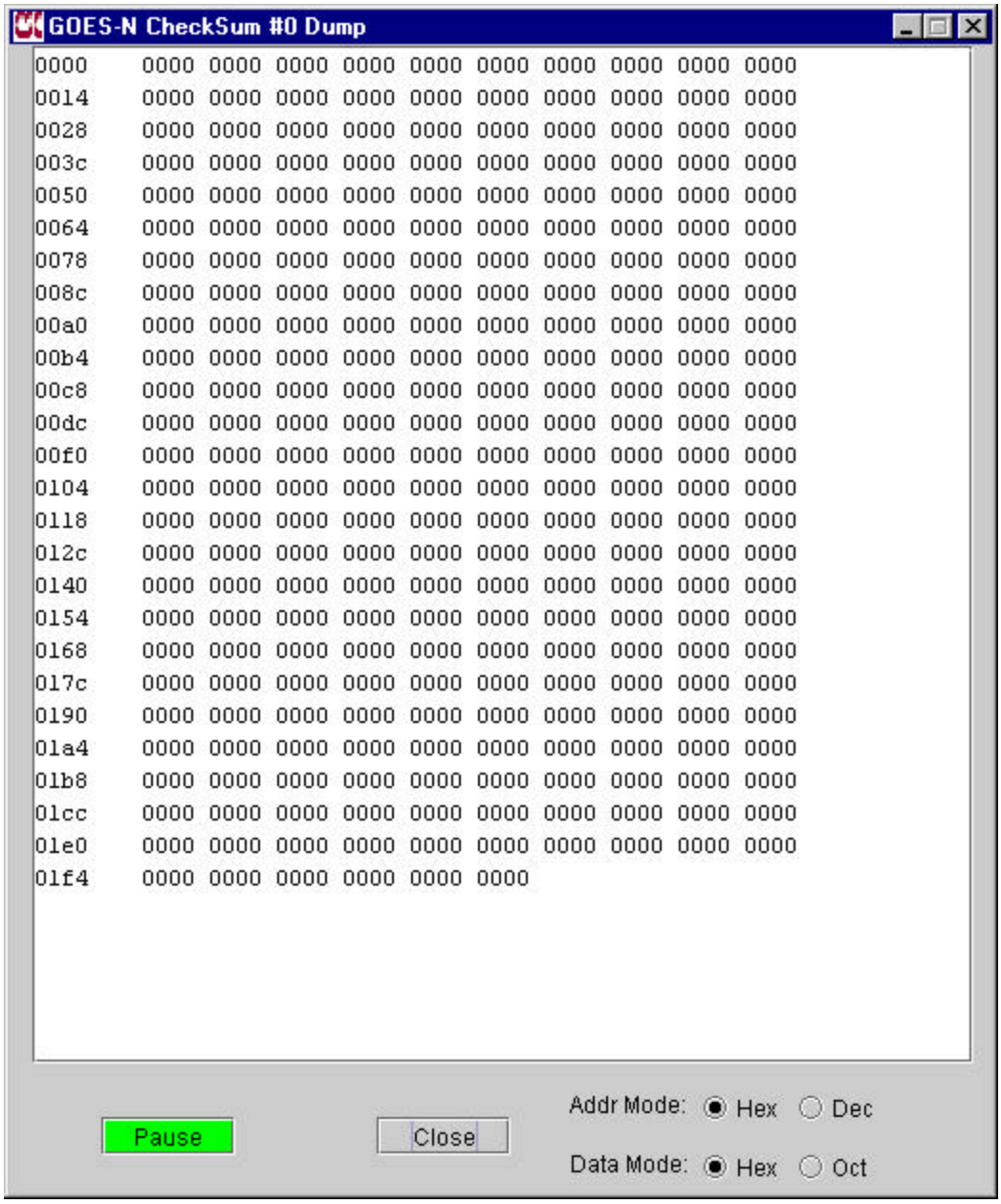
TDRSS Checksum-5.1.1 Status Display



Field	Description
SCID	Spacecraft ID
Frames Received	Number of frames received
Subframe number	Then number of the current subframe
Checksum Errors	The number of checksum errors
Calculated	The calculated checksum for the current frame
Received	The received checksum for the current frame

TDRSS Checksum-5.1.2 Buffer Display

The command stream buffer display shows the contents of the outgoing telemetry stream.



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 format.

Test Module

TestModule-1.0 Overview

The Test module serves as a source or destination for SIMSS module data. It is intended to be used for testing only and has no operational functionality. It can be used during testing as a substitute for the serial output module and will generate data request signals as the serial output module does if the TestSignalFlag is set.

TestModule-2.0 Inputs

Channel	Description
1	Data received and stored in internal input buffer

TestModule-3.0 Outputs

Channel	Description
1	Blocks of data sent from internal output buffer

TestModule-4.0 Container Items

Name	Type	Description
TestModuleRun	Fixed	Run/pause flag
TestReceiveBuffer	Buffer	Buffer of data received
TestReceiveCount	Fixed	Count of blocks received
TestSend	Fixed	Flag set to send a block
TestLength	Fixed	Number of bytes to send in a block
TestTransmitBuffer	Buffer	Buffer of data to send
TestTransmitCount	Fixed	Count of blocks sent
TestSignalFlag	Fixed	Set to enable serial-style signaling (default is 0)
TestTick	Fixed	Milliseconds between signals (serial mode only)

TestModule-5.0 Displays

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

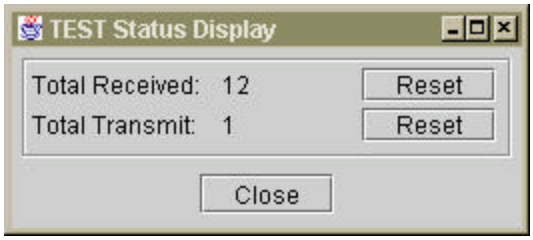
Module Pop-Up Menu Item	Description
Configure	Access the configuration display (not implemented)
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

TestModule-5.1 Run-time

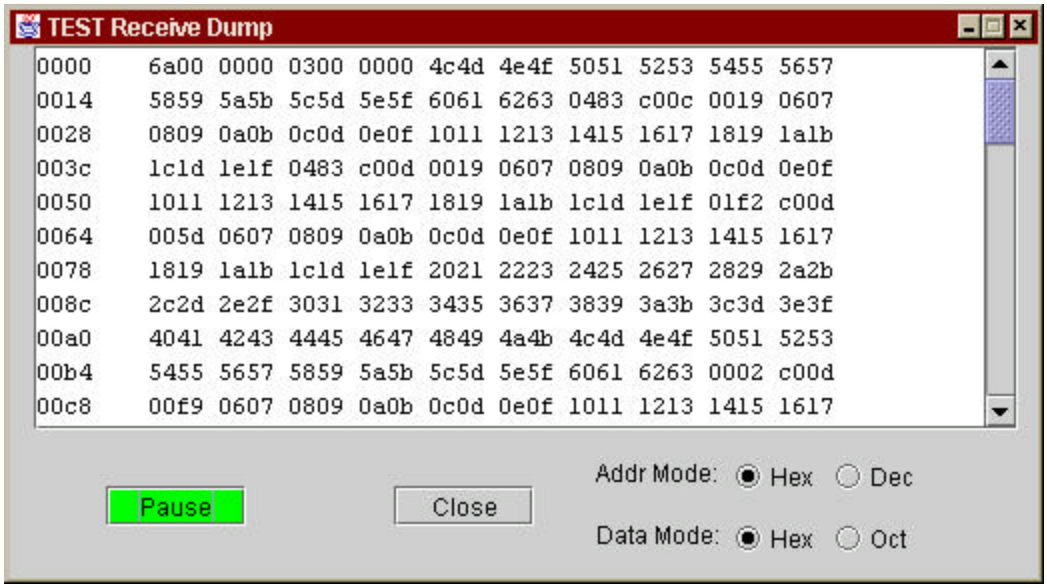
The Run-time menu for Test module contains the following items.

Run-time Menu Item	Description
Display status	Show receive and transmit block counts
Display receive buffer	Show current contents of receive buffer
Display transmit buffer	Show current contents of transmit buffer
Edit transmit buffer	Edit/set data values in transmit buffer
Send transmit buffer	Send current contents of transmit buffer

TestModule-5.1.1 Status Display



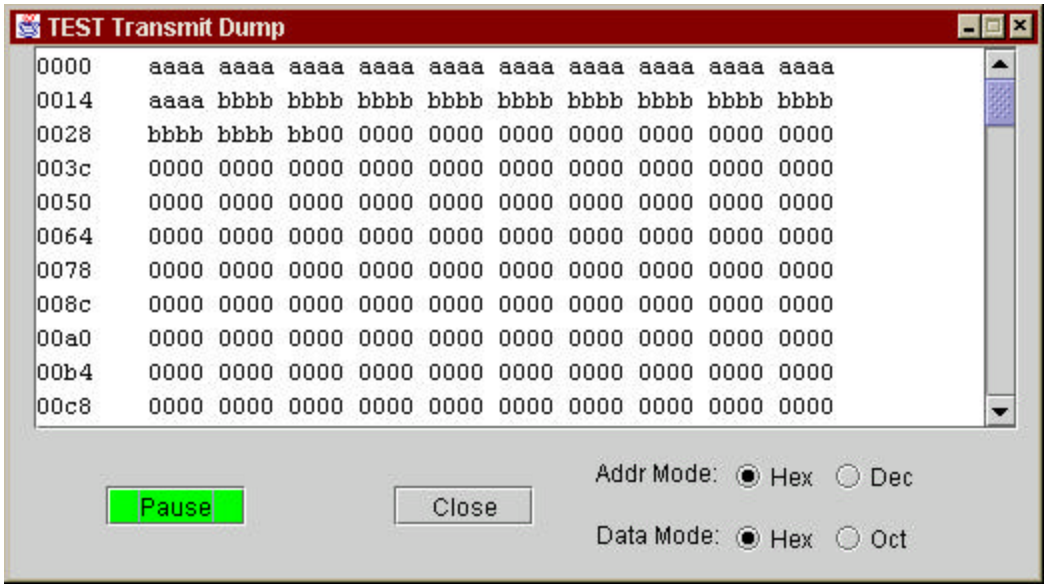
This shows the number of data blocks received and transmitted. Clicking the reset button to the right of a counter resets it to zero. Pressing the close button closes the display.



TestModule-5.1.2 Display Receive Buffer

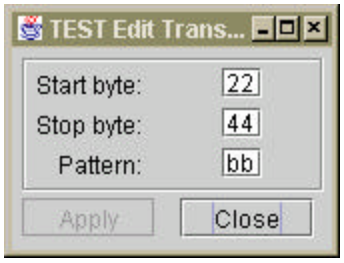
The address field of the Receive dump display may be toggled between hexadecimal and decimal display formats. The data portion may be toggled between hexadecimal and octal formats. Pressing the **Pause** button will stop the display from being updated until it is pressed again. Pressing the **Close** button will dismiss the display.

TestModule-5.2.3 Display Transmit Buffer



The address field of the Transmit dump display may be toggled between hexadecimal and decimal display formats. The data portion may be toggled between hexadecimal and octal formats. Pressing the **Pause** button will stop the display from being updated until it is pressed again. Pressing the **Close** button will dismiss the display.

TestModule-5.2.4 Edit Transmit Buffer



Telemetry Modification (TlmMod) Module

TlmMod-1.0 Overview

The Generic TDM Telemetry Modification Module is used to modify telemetry during runtime.

TlmMod-2.0 Inputs

The TlmMod module has a single data input channel. TlmMod also accepts 9002 messages from other modules.

Channel	Description
1	Accepts external data stream of TDM data. Data must be synchronous.

TlmMod-3.0 Outputs

The TlmMod module has a single output channel.

Channel	Description
1	The modified telemetry.

TlmMod-4.0 Container Items

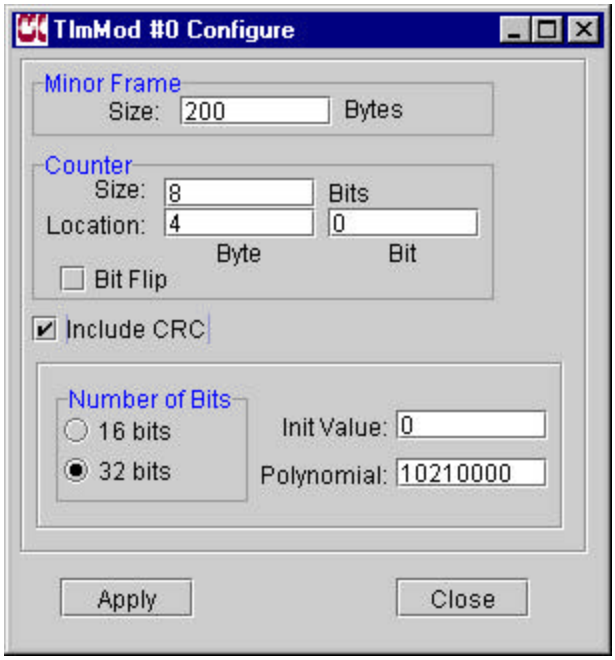
Container Item Name	Description
CNTRSIZE	The minor frame counter size (in bits).
BYTE	The byte offset for the minor frame counter location. First byte in a minor frame is byte zero.
BIT	The bit offset for the minor frame counter location. Valid numbers are zero through seven. The MSB is bit zero.
COUNTER	The current minor frame counter. This is extracted from the telemetry.
FILENAME	The name of the file to load from or save to.
SIZE	The minor frame byte size.
FILELOAD	Save and Load configuration files. Setting this mnemonic to zero causes the current configuration to be saved to the file named by the FILENAME mnemonic. Setting FILELOAD to one is a clear load. The clear load will not clear the memory by itself. Use the GUI or DELETE mnemonic. Setting FILELOAD to two is an append load.
VERSION	Targets a modification line. If this line does not exist, it is created.
DELETE	Deletes the modification line pointed at by the VERSION mnemonic.

TlmMod-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 “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

TlmMod -5.1 Configure



Field Name	Description
Size	The size (in bytes) of the minor frame.
Counter size	The size (in bits) of the minor frame counter.
Counter Byte Location	The byte location of the minor frame counter. The first byte of a minor frame is byte zero.
Counter Bit Location	The bit location of the minor frame counter. The MSB is bit zero.
Bit Flip	This allows for bit flipped minor frame counter. A bit flipped counter is where the bits are backwards.
Include CRC	This allows the user to add either 16 bit or 32 bit CRC to the end of the minor frame. See below for more details on this option.

TlmMod-5.1.12 CRC

The checkbox labeled “Include CRC” is used to indicate whether a CRC should be calculated and inserted into each frame’s final bytes. When this box is checked, the values in the box directly below it will be enabled for user modification.

“Number of Bits” contains two choices to specify either a 16-bit or 32-bit CRC. The CRC will be placed in the final bytes of each minor frame; the last two bytes when 16 bits is chosen and the final four bytes of each frame; when 32 bit is chosen.

"Init Value" is the initial value of the shift register used for calculating the CRC. The default is 0x00000000, but any hexadecimal pattern may be specified.

"Polynomial" is the polynomial used for the calculation, specified as a hexadecimal pattern. The hexadecimal pattern is determined based on the coefficients of the desired polynomial. The initial term of the polynomial, which is only used to determine the polynomial's order (16 or 32), should be left off before determining this hexadecimal value.

For example, the default polynomial is $x^{16} + x^{12} + x^5 + 1$, a commonly used polynomial for CRC calculation. The polynomial is of order 16 and can be written as

$1x^{16} + 0x^{15} + 0x^{14} + 0x^{13} + 1x^{12} + 0x^{11} + 0x^{10} + 0x^9 + 0x^8 + 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$.

The hexadecimal pattern for the CRC algorithm is obtained by dropping the first term, and taking the coefficients of the remaining terms:

0001 0000 0010 0001 = 0x1021.

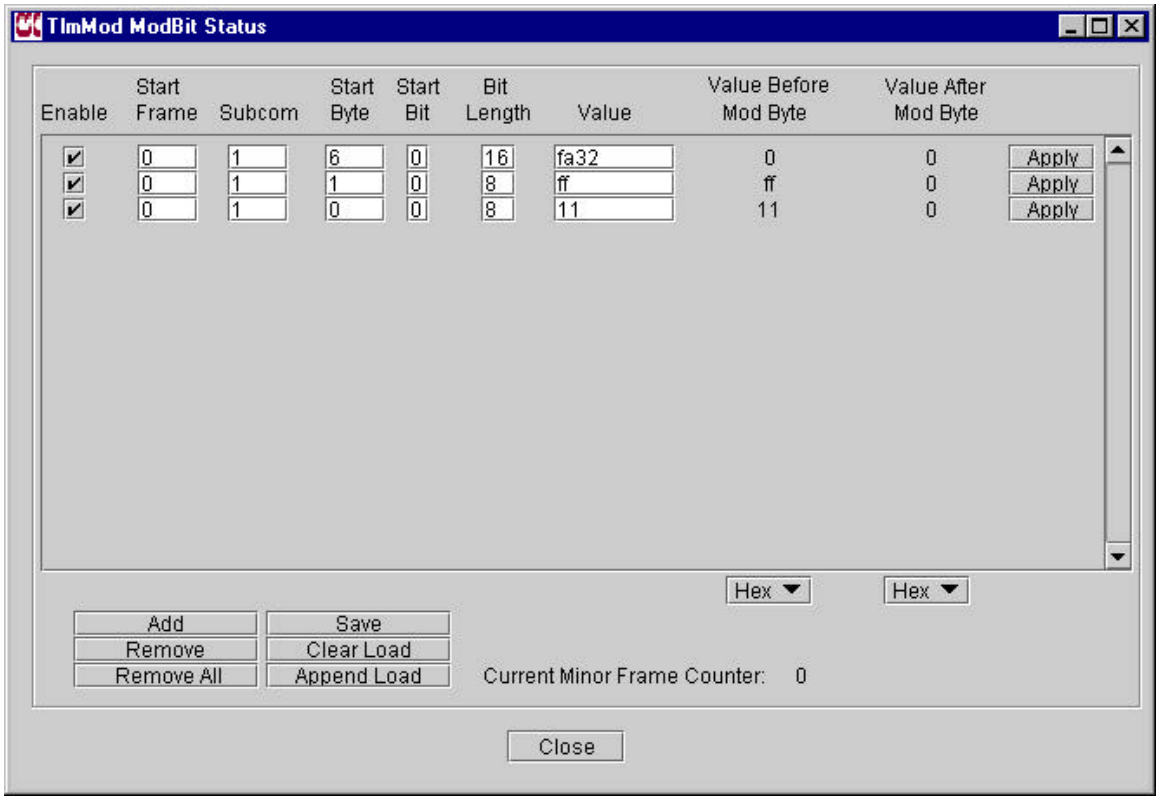
This is the value entered in the "Polynomial" text box. Note that for 16 bit CRCs the value is left shifted. In the case of the default value, this means that the number appears as 0x10210000.

TlmMod-5.2 Run

During runtime the TlmMod module will perform user defined modifications on the TDM data stream being sent through it. The Run-time menu for VC Processor contains the following four items.

Run-time Menu Item	Description
Modbit	Display the current bit level modifications.
Database	Display the current database status.
Resume	Resumes module.
Pause	Pauses module.

TlmMod-5.2.1 Modbit



The main part of this window is the modification lines display. This display starts out blank, you need to add some modification lines. This is done by clicking the Add button or one of the Load buttons in the bottom left hand corner of this window (explanation of buttons below). The following chart explains the parts of a modification line.

Item	Description
Enable	When checked, the modification takes place as the telemetry goes through the module. When unchecked, the modification does no occur.
Start Frame	The first frame to be effected by the modification line.
Subcom	The subcom depth for the modification. A value of zero means a single frame.
Start Byte	The first byte to be effected. The minor frame starts with byte zero.
Start Bit	The first bit to be effected. The MSB is bit zero.
Bit Length	The bit length of the modification. Do not worry about going over byte boundaries.
Value	The new value to be placed into telemetry.
Value Before Mod Byte	For a new modification line, this value will be the value extracted from telemetry. After a modification line is changed, this value will be the value that was previously in the “Value” field.
Value After Mod Byte	The value in telemetry after being modified.
Apply	When changes are made to a modification line, this button must be

	pushed for the changes to go into effect.

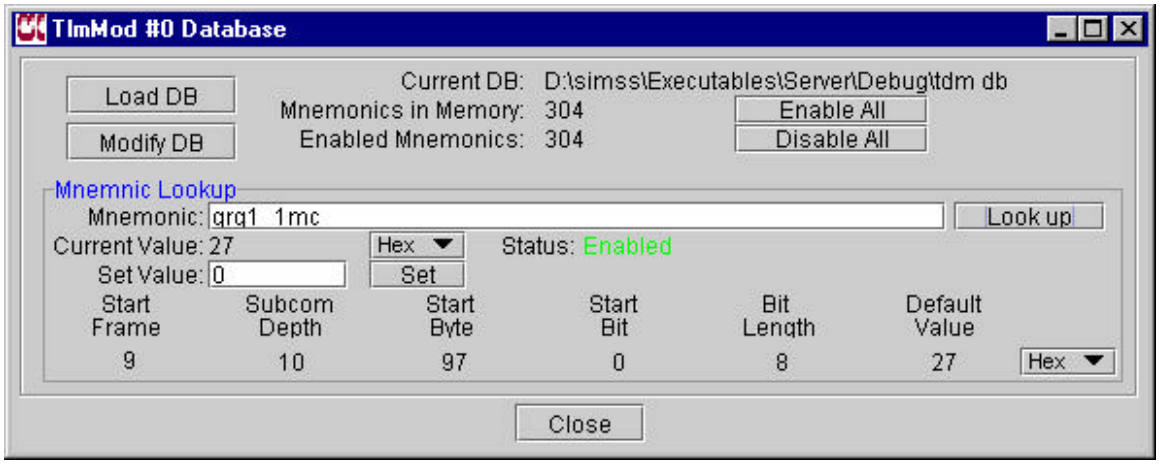
Each modification line has ten mnemonics. A description of these follows. In the mnemonic column, replace the number sign (#) with the number that corresponds with the modification line to be manipulated.

Mnemonic	Description
Enable#	Enable/disable the modification line. Zero is disabled, other is enabled.
StartFrame#	The starting frame for the modification. The first frame in a major frame is zero.
Subcom#	The subcom depth of the modification line. A value of zero means a single frame.
StartByte#	The first byte to be modified. The first byte in a minor frame is byte zero.
StartBit#	The first bit to be modified. The MSB is bit zero.
BitLength#	The number of bits to be modified. The max value is 32.
PreValue#	The value before modification. This value is set internally, so there is no reason for the user to set it.
PostValue#	The value in telemetry after the modification. This value is set internally, so there is no reason for the user to set it.
NewValue#	The value that is placed into the telemetry. When this mnemonic is set, some error checking on frame boundaries and overlapping occurs for this modification line.
Apply#	This is used byt the GUI. There is no reason for the user to set this mnemonic.

There are six button grouped together in the bottom left hand corner of this window. The three buttons on the left are for adding and removing modifications by hand. The three buttons on the right are used for saving and loading modification configurations. The following chart explains the use of each button.

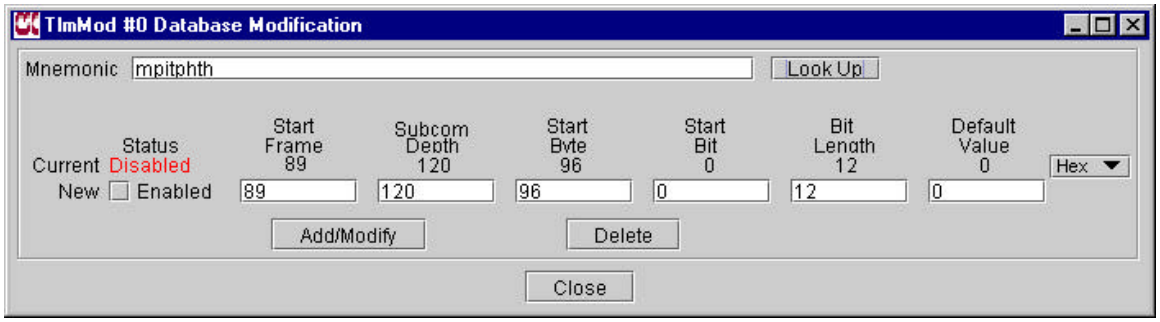
Button	Description
Add	This will add an un-configured modification line to the window. After filling in all the fields, click the apply button, and the modification will be in effect. Make sure the enable check box is checked if you want the modification to occur.
Remove	This will remove all selected modification lines.
Remove All	Remove all modification lines.
Save	Save the current modification lines to a file.
Clear Load	Load modification lines from a file. Doing this will remove all current modification lines before loading the file.
Append Load	Load modification lines from a file. This will not remove any current modification lines.

TlmMod-5.2.2 Database



TlmMod allows the use of a database to control mnemonics. A database is loaded into the TlmMod module through the Database window (click on the **Load B** button). When setting the value of a mnemonic, remember that the value entered is always a hex value.

TlmMod-5.2.3 Database Modification



The user is allowed to modify the database on the fly. This is done using the Database Modification window. To modify the database type in the mnemonic to be modified, enter the new values and add/delete/modify the mnemonic. When setting new values for a mnemonic, remember that the values entered are always hex values.

TlmMod-5.2.4.1 Scenario File for modbit function

Use of this module with scenario file is not straight forward. It is also not fully worked out, so expect major changes in a future release. In other words, don't put a lot of effort into creating scenario files for this module right now.

It is highly recommended that the user **NOT** use a single TlmMod module for both user defined (including loaded files) and scenario files. If the user needs both, use two separate modules.

The first step for using scenario files is to create the modification lines in memory. This is done by setting the VERSION mnemonic. Each modification line needs to have a unique VERSION number. **WARNING:** If running more than one scenario file, the modification line versions of one scenario file should be unique from the modification line versions.

After the modification line is created in memory, the user can manipulate it at will by manipulating the modification lines mnemonics (listed above).

Here are some example scenario files:

TlmMod-5.2.4.1.1 Sample Modbit Scenario File 1

```
;-------------------------------------------------------------------------------------------------------------
; This scenario file will create a modification line (enabled), have the modification
; take place for ten seconds, then disable the modification.
;-------------------------------------------------------------------------------------------------------------
; Set up a modification line
set version 0
; Make sure this modification line is enabled
set enable0 1
; Set the modification configuration.  This will change the fourth byte of every frame to
; a value of 0xFF
set startframe0 0
set subcom0 1
set startbyte0 4
set startbit0 0
set bitlength0 8
set newvalue0 255
; Wait for ten seconds
sleep 10000
; Disable the modification
set enable0 0
; Note: the modification is still in memory and can be reconfigured or re enabled.
; To delete the modification use the following lines:
; set version 0
; set delete 1
```

TlmMod-5.2.4.1.2 Sample Modbit Scenario File 2

```
;-------------------------------------------------------------------------------------------------------------
; This scenario file will create a modification line (enabled), have the modification
; start at zero and go to seven.
;-------------------------------------------------------------------------------------------------------------
; Set up a modification line
set version 0
; Make sure this modification line is enabled
set enable0 1
; Set the modification configuration.  This will change the last three bits of the sixth byte
; of every even frame to a new value.  The value will increment by one every five
; seconds.
set startframe0 0
set subcom0 2
set startbyte0 6
```

```
set startbit0 5
set bitlength0 3
set newvalue0 0
; Wait for five seconds
sleep 5000
; Change value and wait
set newvalue0 1
sleep 5000
; Change value and wait
set newvalue0 2
sleep 5000
; Change value and wait
set newvalue0 3
sleep 5000
; Change value and wait
set newvalue0 4
sleep 5000
; Change value and wait
set newvalue0 5
sleep 5000
; Change value and wait
set newvalue0 6
sleep 5000
; Change value and wait
set newvalue0 7
sleep 5000
```

TlmMod-5.2.4.2 Scenario File for database function

Using scenario files to change mnemonic values is pretty straight forward. Type:

```
set newvalueMNEMONIC VALUE
        MNEMONIC  is the name of the mnemonic to be changed.
        VALUE     is the new value for the mnemonic.
;-------------------------------------------------------------------------------
; Here is a sample scenario file that changes the value of four mnemonics.
;-------------------------------------------------------------------------------
set newvalueMBUSA 0
set newvalueMBUSB 0
set newvalueSMBSMSA 0
set newvalueSMBSMSB 1
sleep 60000
set newvalueMBUSA 1
set newvalueMBUSB 1
set newvalueSMBSMSA 1
set newvalueSMBSMSB 0
```

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

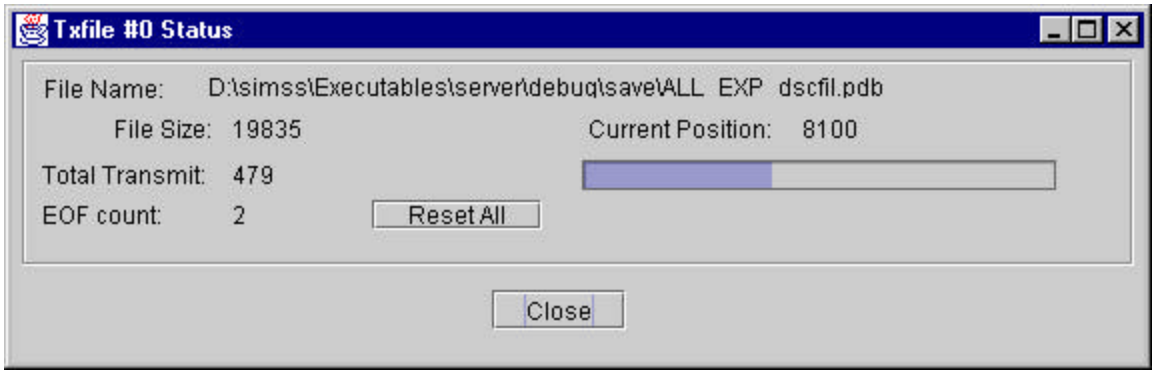
There are no configuration displays for this module. However, if during run-time TxFile receives a request for more data signal from the down-linked module, the transmission mode will be automatically reconfigured to serial and an event message will be generated to inform the operator.

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

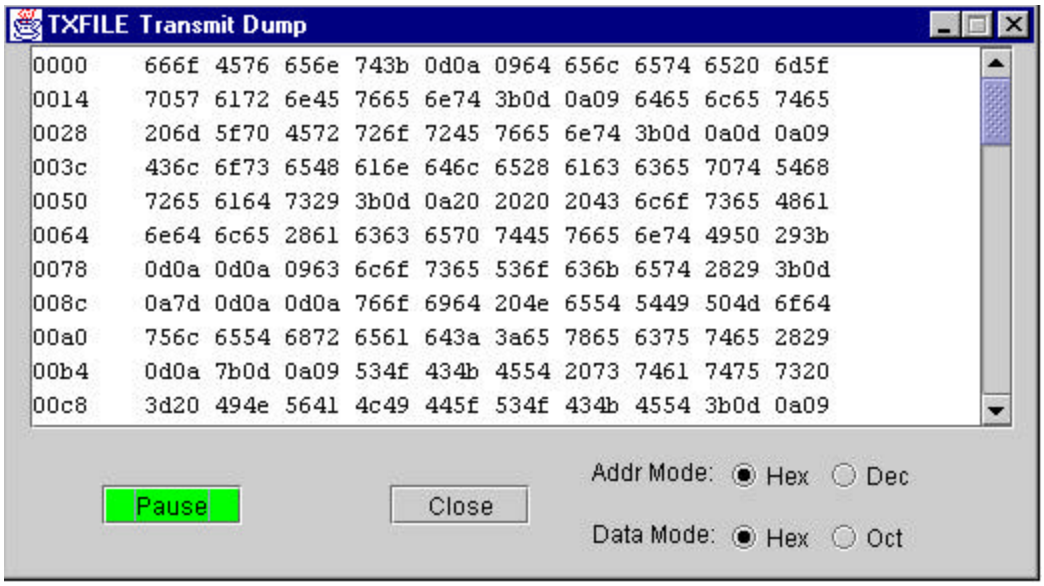
The status display shows the current filename, the size of the file, current file pointer position, and the total number of blocks transmitted.



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

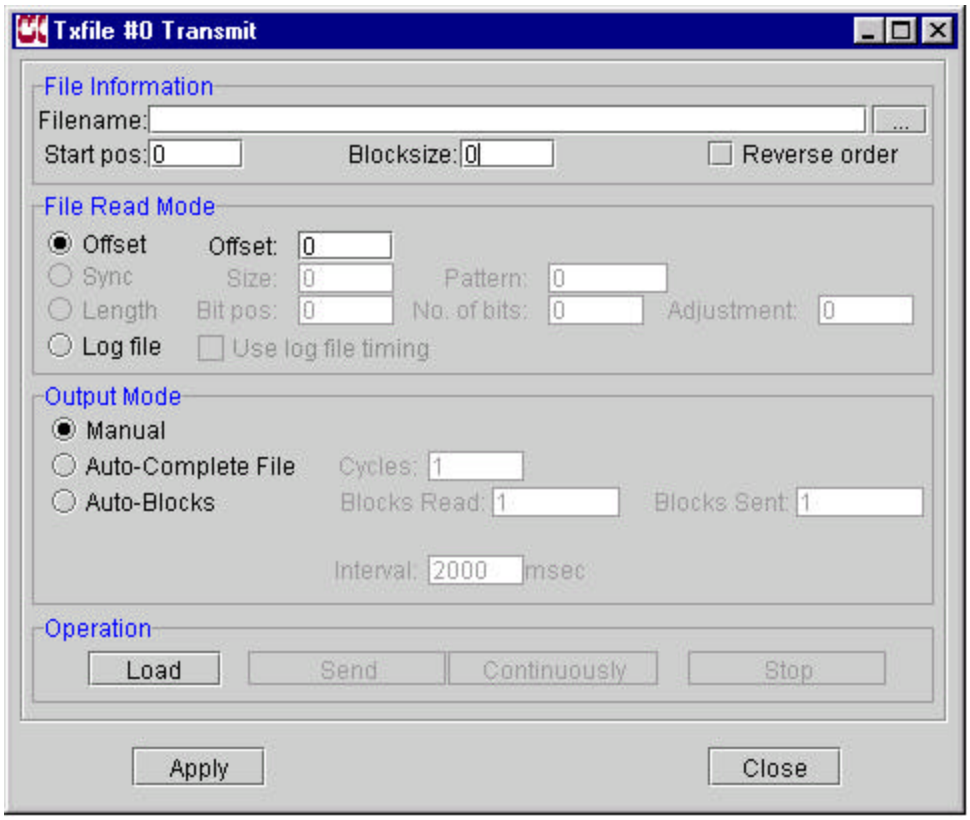
The transmit buffer display shows the contents of the currently loaded block.



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.



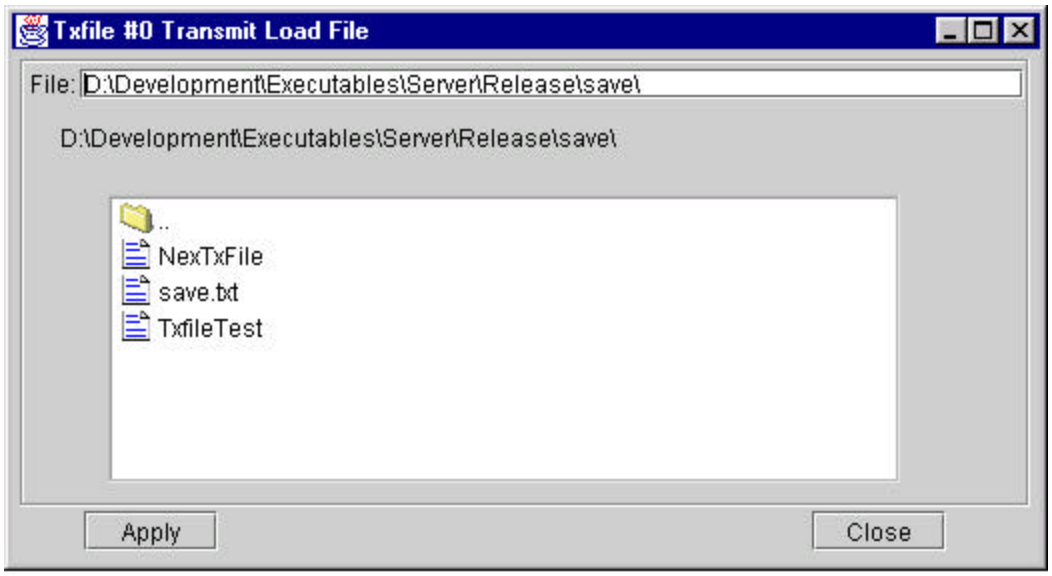
This is the initial Send from File Display before any parameters have been set. The default interval (for automatic transmission) is 2000 milliseconds (2 seconds) between blocks. The default read mode is Offset. The default output mode is Manual.

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

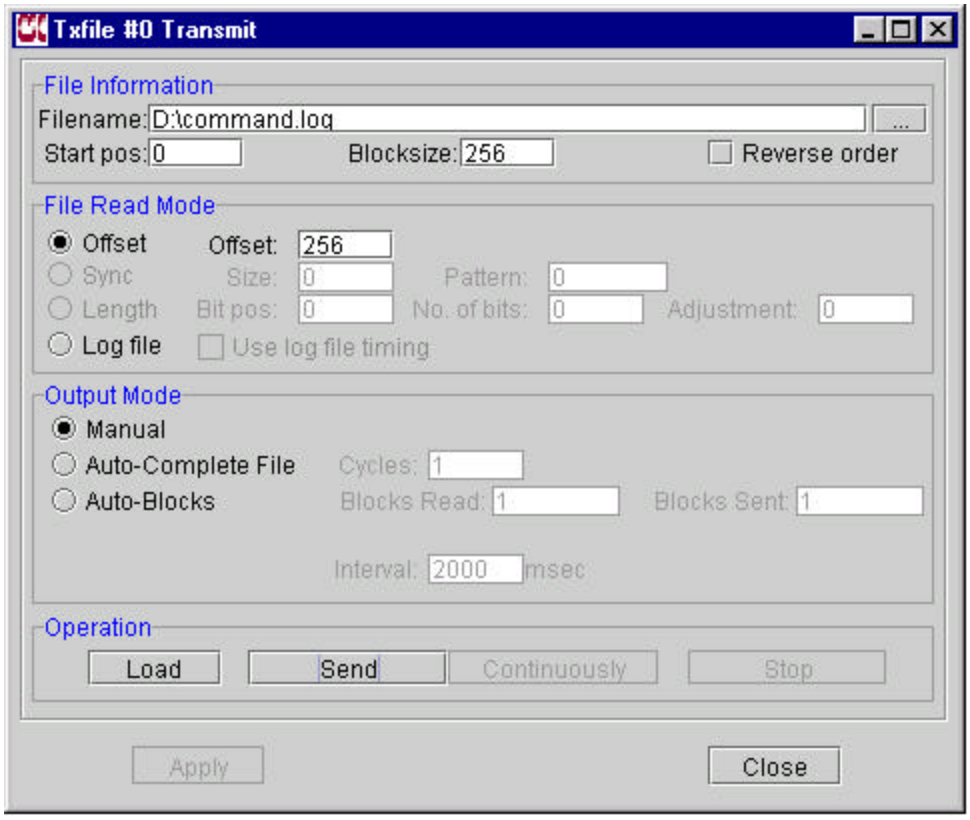
Note Since the Sync and Length read modes are not yet implemented, TxFile can not properly process raw files with varying size data. Every data byte can be read and transmitted using modes valid for fixed raw files, but the timing and grouping of the data may not be valid for the receiving system. These 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.

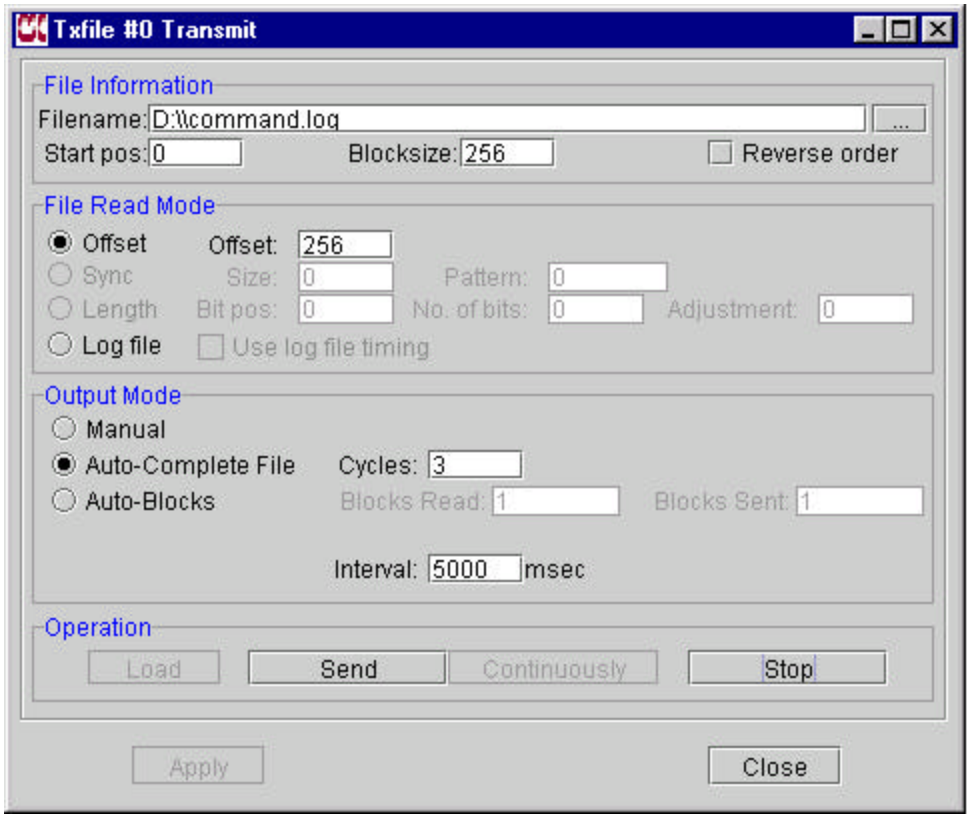
The following screen shows TxFile configured for manual output mode using a raw data file with entries of 256 bytes. Steps 1 through 4 (Apply) have been done. Steps 5 and 6 must be executed to transmit data.



TxFile-5.2.3.4 Example 2 Auto-Complete File Transmission

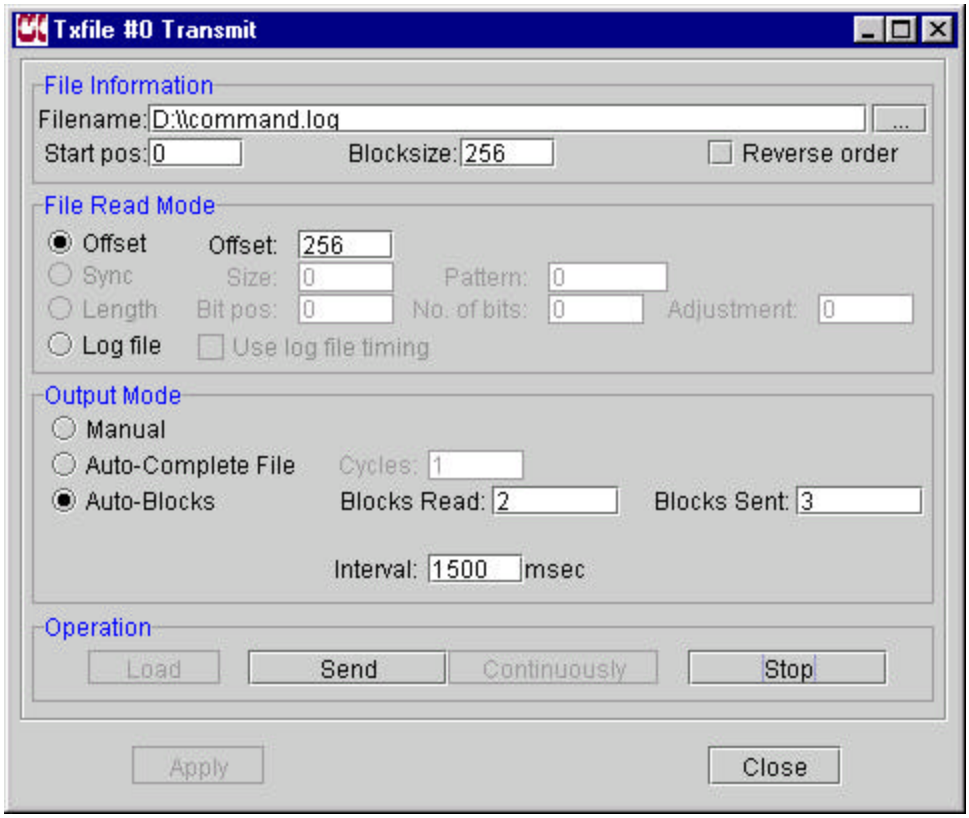
Step	Auto-Complete File Transmission Description
1	Enter the file name in the Filename field.
2	Select Auto-Complete File in Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, select Offset File Read Mode and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

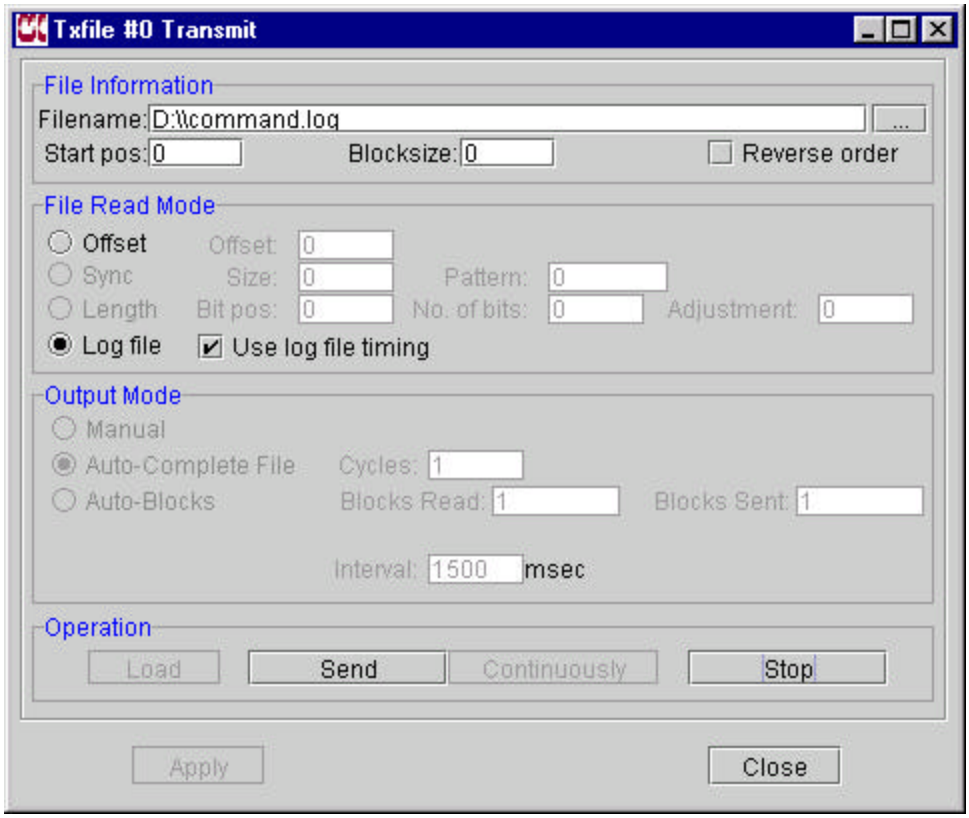


The above screen shows TxFile configured for Auto-Blocks output mode using a raw data file with entries of 256 bytes. The first 2 blocks of the file will be transmitted 3 times with 1500 milliseconds between blocks. Steps 1 through 7 (Apply) have been done. A single click on the Send button will start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

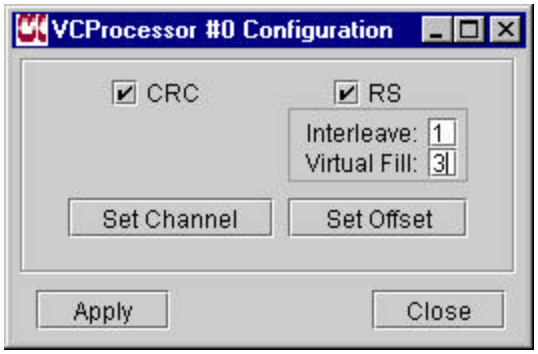
To access displays for a module, click in the center of the module in the project window. The following pop-up menu choices will appear. The "Remove" option can be used during project design to remove the module. The "Configure" option should be used prior to running the project and is unavailable at 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, Set Channel and Set Offset. This module can also be re-configured during runtime.

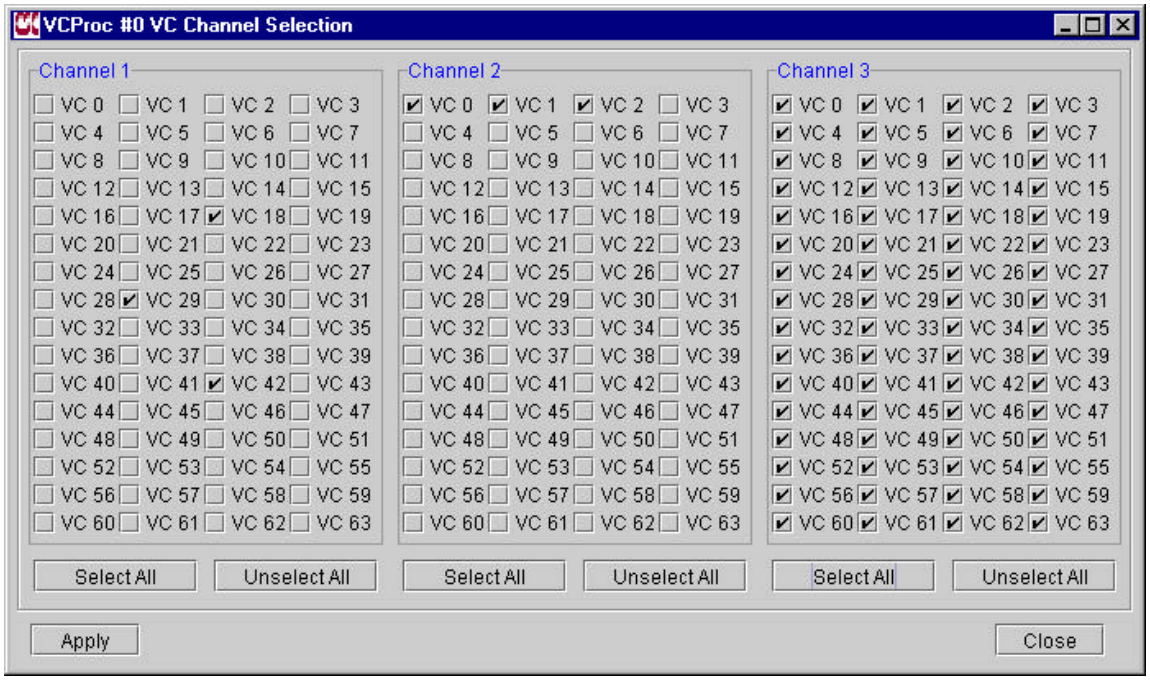
VC Processor-5.1.1 Main Configuration Window



Field Name	Description
CRC	Enable/disable CRC check.
RS	Enable/disable RS encoding check.
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 Channel	Set virtual channels to transmit on each of the three output channels.
Set Offset	Set the start byte location for display value (one per virtual channel).

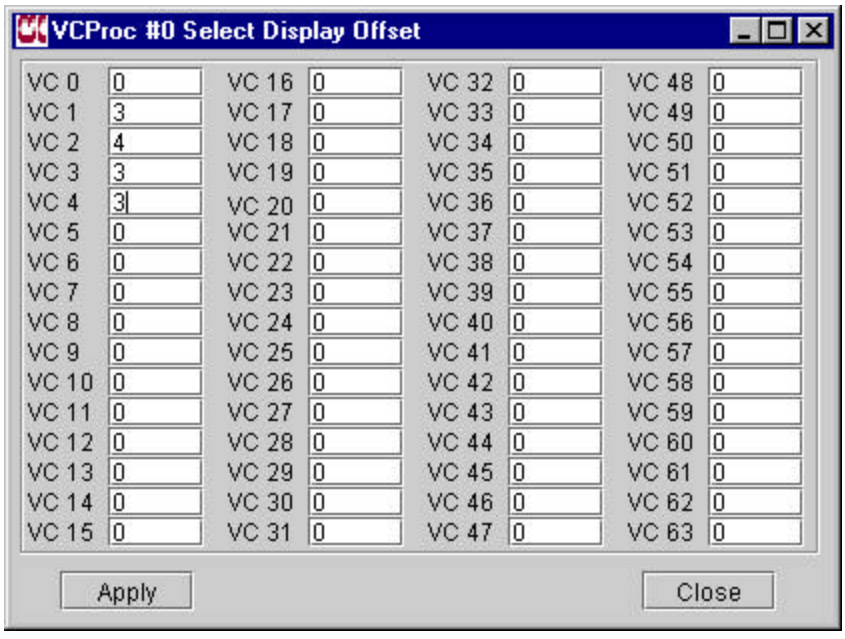
This is the main configuration window. CRC and RS checking is enabled/disabled from this window. The two other configuration windows are accessible from this window.

VC Processor-5.1.2 Set Channel



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 Display Offset



Each VC will 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.

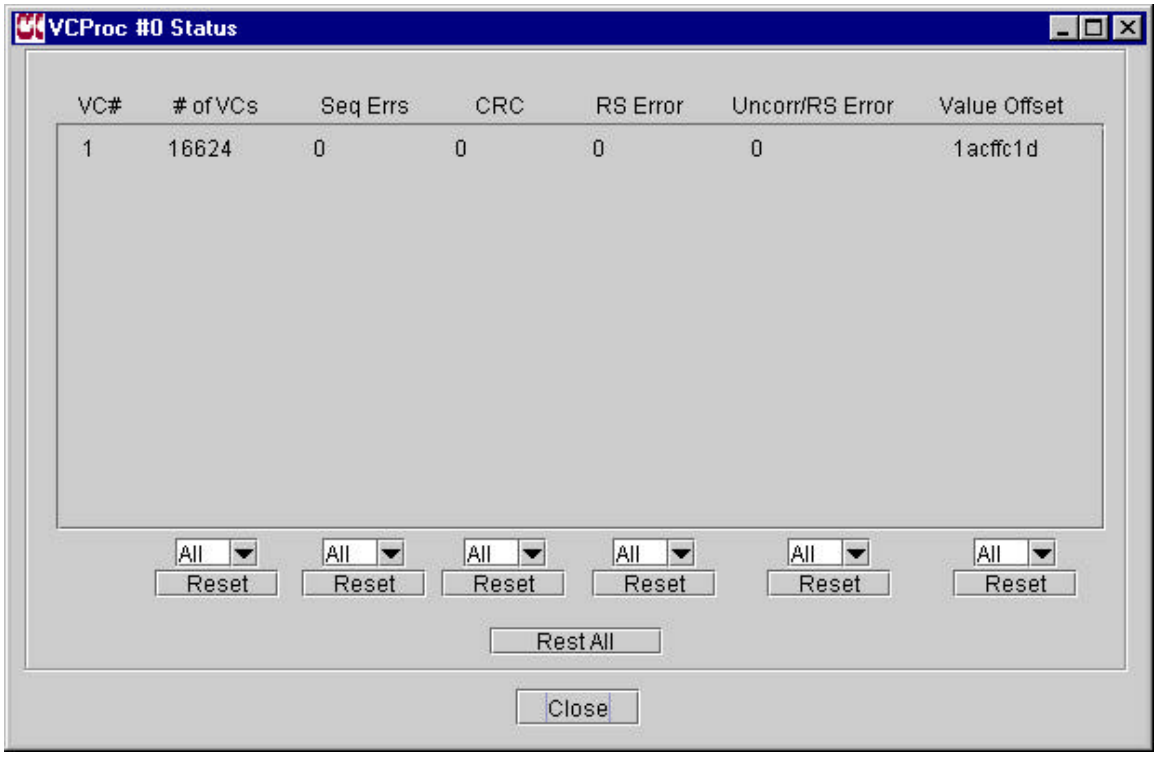
VC Processor-5.2 Run

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

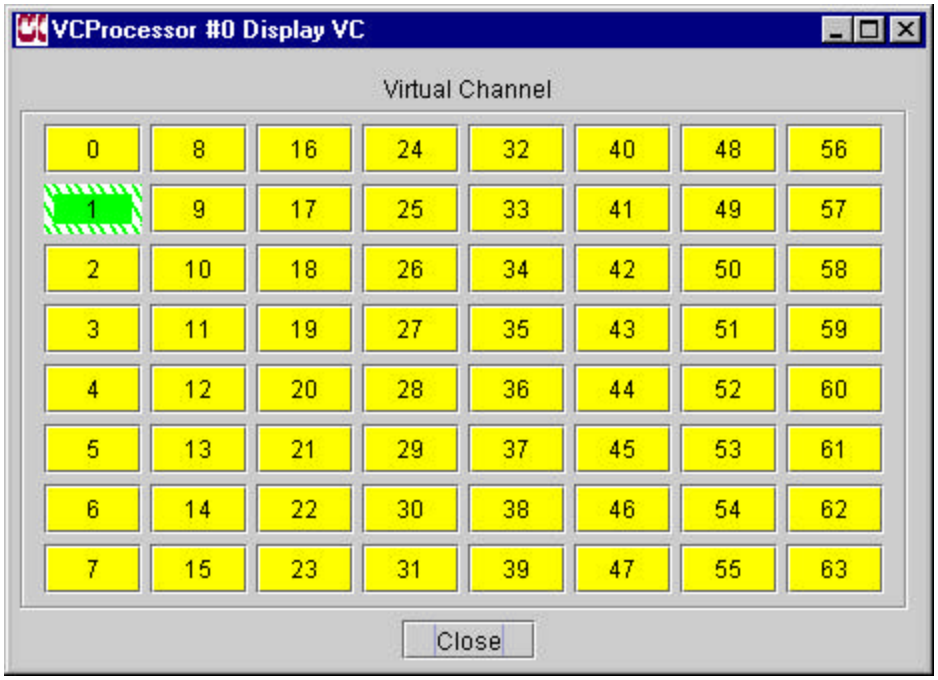


This window shows the status of each active VC.

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 (5.1) for details.

Wrapper Module

Wrapper-1.0 Overview

The Wrapper module receives data from an upstream module, processes and passes the data to a downstream module. Processing of the data includes adding a header, appending a footer, and packaging the incoming data into a configured specific size. Typical usage is the construction of a NASCOM data block.

Wrapper-2.0 Inputs

Channel	Data expected	Validation performed	Processing performed
1	Data Buffer	None	Data is received from an upstream module, wrapped, and processed.

Wrapper-3.0 Outputs

Channel	Description
1	Wrapper module sends the received and processed data to another module.

Wrapper-4.0 Container Items

This module accepts operator directives. Use the Set and Get directives to access items with a fixed, integer, float, or string types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

Name	Type	Description
Wrapper-BypassSignal	Integer	Flag to bypass block mode when data input size is less than totalsize field specified in the configuration txt file (0 = blocked; 1 = bypass). Default is 0.
Wrapper-RelayTick	Integer	Flag to relay tick signal to the upstream module (0 = not relay; 1 = relay). Default is 0.

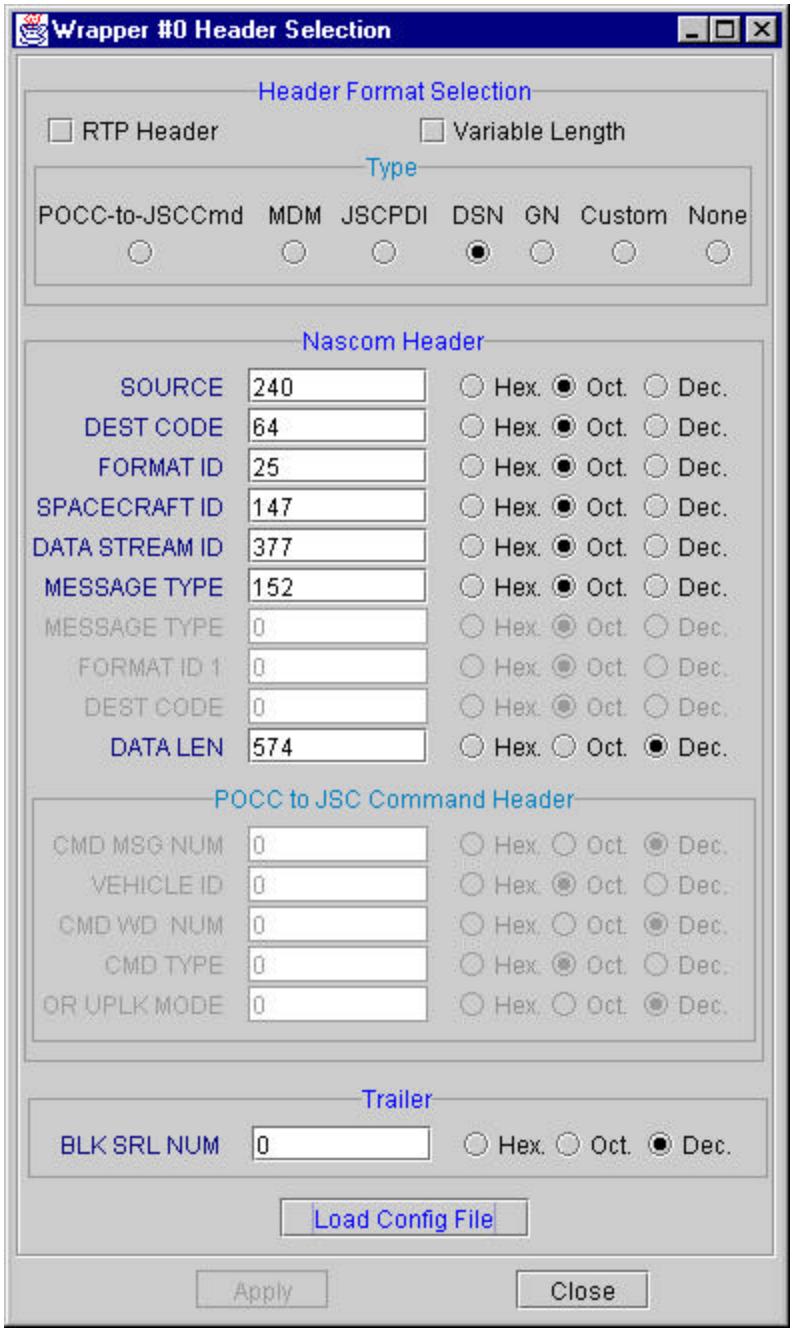
Wrapper-5.0 Displays

To access displays, click in the center of the module in the project window. The module pop-up menu will appear. The "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 used when the project is running.

Module Pop-Up Menu Item	Description
GUI Configure	Access the configuration display. This may be accessed during configuration and runtime.
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display generic module information

Wrapper-5.1 Configuration

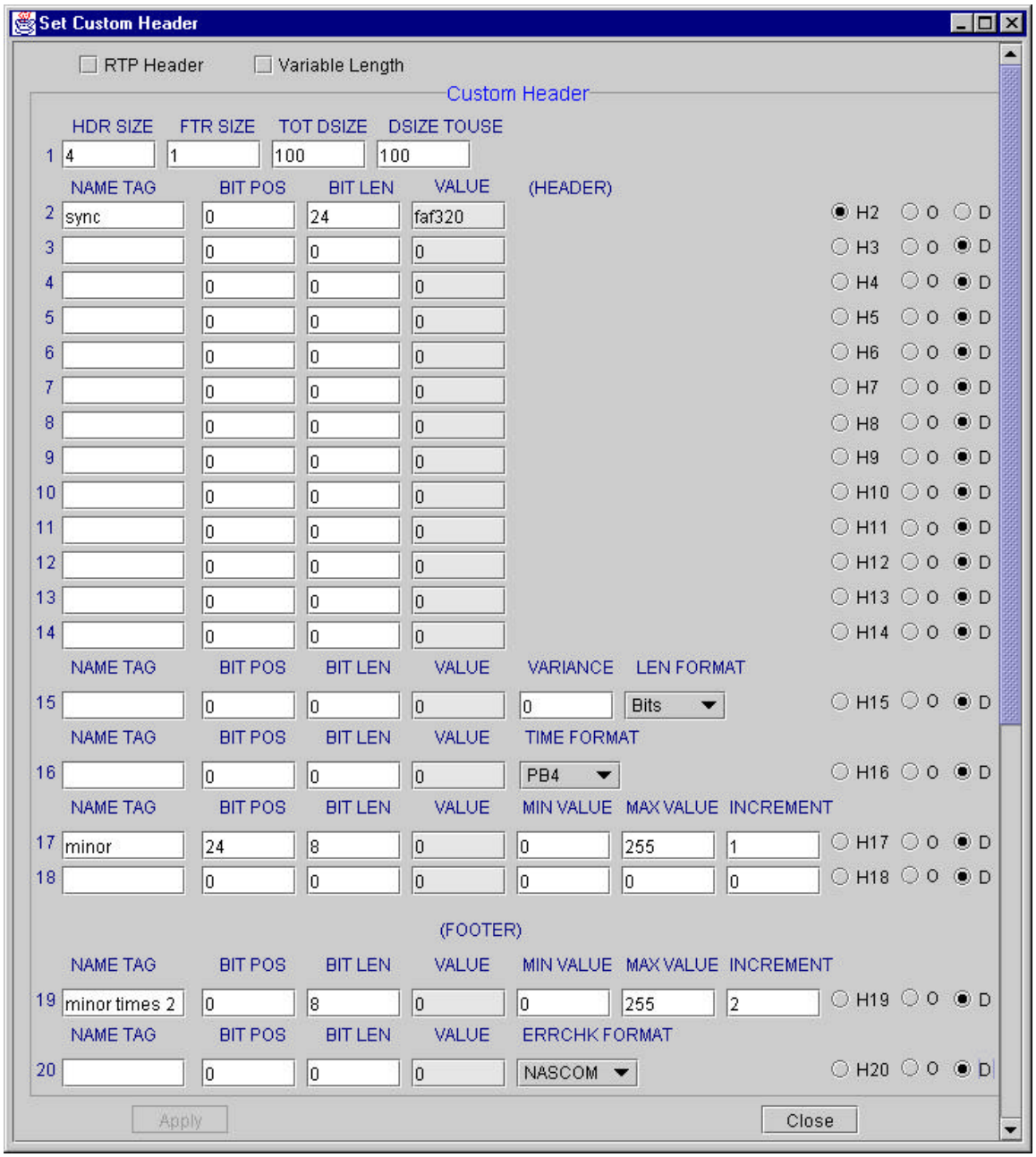
Clicking on the module pop-up menu “Configure” item produces the following display.



The checkboxes at the top of the window is for adding an RTP header and selecting variable length data. When the Variable Length box is checked the wrapper module will wrap (using a fill pattern if necessary) and send incoming data immediately when it is received. When the Variable Length box is unchecked the wrapper module will wait until it receives enough data to fill a full block before wrapping and sending the data.

The Type radio buttons are used to select the format being used by the module. Only NASCOM blocks are currently supported.

Click the **Apply** button to save your selected filename. Click the **Close** button to dismiss the Configuration display without making any changes.



The wrapper module allows the user to customize the configuration. Click the "Custom" radio button on the main configuration window to bring up this custom configure window. The following table identifies what each line is used for in the custom configuration window.

Line(s)	Description
1	Header size, footer size, total size, data size to use.
2-14	Header fixed values.
15	Header length.
16	Header time format.
17-18	Header counters.
19	Footer counter.
20	Footer error check.

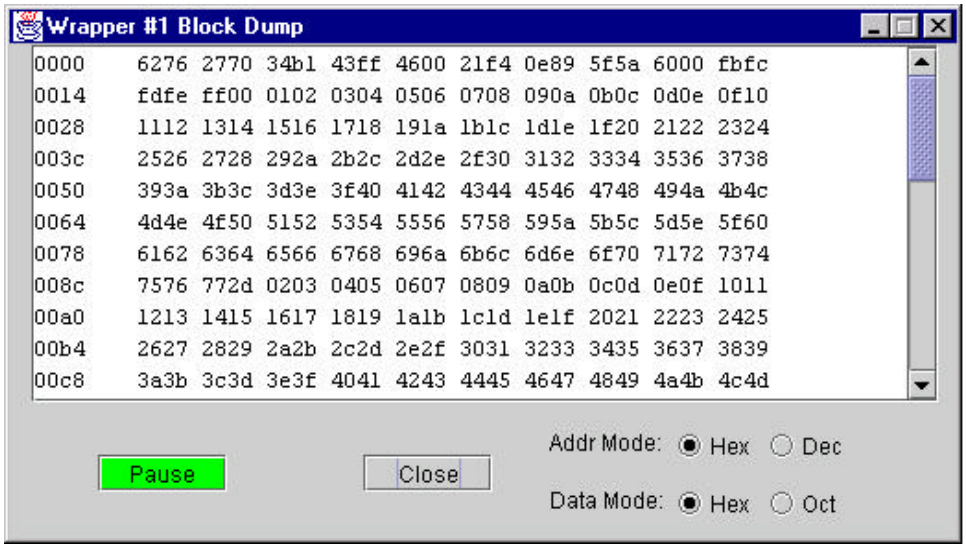
Wrapper-5.2 Run-time

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

Run-time Menu Item	Description
Show Output Block	Request Output Data Block Display
Show Status	Request Status Display
Resume	Restart the Wrapper module
Pause	Pause the Wrapper module

Wrapper-5.2.1 Show Output Block

Select "Show Output Block" from the Run-time Menu to show data received and processed by the module.

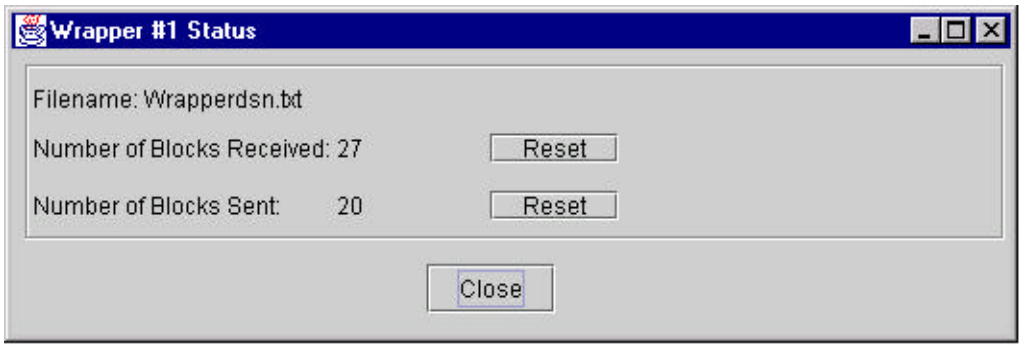


The address region (which is not used in this case) may be displayed in decimal or hexadecimal by clicking on the respective radio button. Likewise, the data region may be displayed in hexadecimal or octal. Since this display updates as the buffer contents are changed; the **Pause** button may be used 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 last buffer processed. Buffers not shown while the display was frozen can not be displayed. Click the **Close** button to dismiss the display.

Wrapper-5.2.2 Show Status

Select the "Show Status" item to request the module's status.



This display shows the module's configuration filename, number of data blocks received, and number of data blocks sent. Note that, the number of data blocks received and sent may not be the same if the length of input buffer is different from the desired NASCOM block size. The **Reset** button may be clicked to reset the number of blocks received/sent count. Click the **Close** button to dismiss the display.

Wrapper-5.2.3 Resume

After the module has been stopped/paused, select "Resume" from the Run-time Menu to restart the module. The module's border will change from red (indicates stopped/paused state) to green (indicates run state).

Wrapper-5.2.4 Pause

Select "Pause" from the Run-time Menu to temporarily stop the module's processing. The module's border will change from green (indicates run state) to red (indicates stopped/paused state). Once the module has been stopped, the "Configure" option of the module pop-up menu is available again and the module may be reconfigured. The Resume option must then be used to return to run mode.

Wrapper-5.3 About

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

Wrapper-6.0 Special Operating Instructions

Previous versions of the wrapper module supported custom configurations via a configuration file. These are no longer supported.

Acronyms

APID	Application Identifier
ASCII	American Standard Code for Information Interchange
BCH	Bose-Chaudhuri-Hocquenghem
C&DHS	Command and Data Handling System
CADU	Channel Access Data Unit
CCSDS	Consultative Committee on Space Data Systems
CD	Compact Disc
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
COP	Command Operations Procedure
COTS	Commercial, off-the-shelf
CUC	CCSDS Unsegmented Time Code
CSC	Computer Sciences Corporation
EDOS	EOS Data and Operations System
EDU	EDOS Data Unit
EGS	EOS Ground System
EMOS	EOS Mission Operations System
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information Systems
EPGS	EOS Polar Ground Stations
ETS	EOSDIS Test System
FARM	Frame Acceptance and Reporting Mechanism
FTP	File Transfer Protocol
GMSEC	Goddard Mission Services Evolution Center

GMT	Greenwich Mean Time
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
IP	Internet Protocol
I/O	Input/Output
JDK	Java Development Kit
NT	New Technology
PC	Personal Computer
PDB	Project Database
SC	SIMSS Spacecraft Module
SIMSS	Scalable, Integrated, Multimission Simulation Suite
TCP	Transmission Control Protocol
TDM	Time Division Multiplexed
UDP	User Datagram Protocol
VCDU	Virtual Channel Data Unit
VCID	Virtual Channel Identification