[illegible] W003273

1

ADST
DIS TEST TOOLS DOCUMENTATION

Part 1: ModSAF/IST-SAF Command Interface User Manual
Part 2: DIS Logger/Scanner User Manual

Loral Systems Company
12151-A Research Parkway
Orlando, Florida 32826

21 February 1994

Contract No. N61339-91-D-0001
Architecture and Standards Phase 2 Delivery Order (0035)
CDRL A003

Prepared for:

Naval Air Warfare Center
Training Systems Division
12350 Research Parkway
Orlando, FL 32826-3224

LORAL

REPORT DOCUMENTATION PAGE

Form approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 21 Februay 1994	3. REPORT TYPE AND DATES COVERED Version 1

4. TITLE AND SUBTITLE

DIS Test Tools Documentation

5. FUNDING NUMBERS

Contract No. N61339-91-D-0001
Architecture and Standards
Phase 2 Delivery Order

6. AUTHOR(S)

Alan Burke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Loral Systems Company
ADST Program Office
12151-A Research Parkway
Orlando, FL 32826

8. PERFORMING ORGANIZATION REPORT NUMBER

ADST/WDL/TR-93-W003273

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Air Warfare Center
Training Systems Division
12350 Research Parkway
Orlando, FL 32826-3224

10. SPONSORING ORGANIZATION REPORT NUMBER

A003

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The DIS Test Tools Documentation provide a basic users guide for the DIS Logger/Scanner and ModSAF/IST Test Tools.

14. SUBJECT TERMS

15. NUMBER OF PAGES

31

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	17. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	17. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

TABLE OF CONTENTS

Part 1: ModSAF/IST-SAF Command Interface User Manual

Part 2: DIS Logger/Scanner User Manual

LIST OF ILLUSTRATIONS

LIST OF TABLES

ModSAF/IST-SAF Command Interface User Manual

1. Introduction

This document provides user information for the IST-SAF command parser installed in ModSAF for test tool purposes. ModSAF itself is a substantial program with its own user manual; no attempt is made here to provide documentation of ModSAF beyond its text command parser and specifically the IST commands installed as part of that parser.

The goal of this development was to provide a command parser capable of reading script files prepared for the IST-SAF (which is based on an IBM PC) on a Unix workstation environment. Rather than develop a new simulation or attempt to port the PC code to Unix, ModSAF was chosen as a fully capable DIS simulation within which the IST commands could be executed.

2. ModSAF

2.1 Running ModSAF

Consult the ModSAF user manual for detailed information on running ModSAF.

The IST-SAF parser requires use of the text command parser. Most ModSAF users are unfamiliar with the text command parser, so it is introduced in the next section.

Assuming you have a runnable copy of ModSAF and its required data files and directories, go (cd) to the directory containing the executable.

Before running ModSAF be sure the textport is in a good location. Since ModSAF normally occupies the upper-left 95% of the workstation screen, position the text window in the lower right corner so it will not be completely covered by the ModSAF GUI window.

Start ModSAF normally. The IST-SAF quit command has not been emulated. To quit ModSAF either Pull down the "File" menu on the

GUI and select "Quit" (a confirm window will pop up), or type "quit" or "q!" at the ModSAF text command parser prompt.

2.2 The ModSAF text command parser

ModSAF incorporates a text command parser which runs in the parent textport from which the executable was started. This parser is reasonably user-friendly, including help and command completion. Parser commands consist of a series of space-separated keywords and values. At any point the user can type '?' and be rewarded with a description of legal responses given the partially completed command already typed. If a keyword is partially typed the escape key completes the keyword until an ambiguity is reached (for example, if the legal commands are foo1 and foo2, the user can type 'f', and hit the escape key to have the parser fill in the "oo"). Unambiguous abbreviations are also allowed (if the only legal command begins with 'f', only 'f' need be typed).

The ModSAF parser had 15 major commands. The IST command interface was established as a 16th command. Its keyword is "ist" and since no other commands begin with 'i' the command can be abbreviated to a single letter.

3. IST Commands

3.1 ModSAF-IST command interface

The ModSAF-IST command interface recognizes five keywords which control the new IST functionality. Two of these actually accept IST-format commands and the others provide related support. All five may be abbreviated as single characters.

The new ModSAF commands are listed in Table 3.1-1 below.

Table 3.1-1. New ModSAF Commands

COMMAND	ACTION DESCRIPTION
ist abort	abort execution of the current IST SAF script
ist continue	continue script (cancels current delay)
ist display center <x> <y>	move the display center point
ist display mpp <mpp>	set the display scale in meters-per-pixel
ist display print	print the current display center and scale
ist display scale <scale>	set the display map scale
ist file <file name>	execute commands from an IST SAF script file
ist single <command>	execute a single IST SAF command

Example: execute a single IST-format command:

```
ist single "i c t72 28000 21000 0 45"
```

The quoted string is exactly the same command the user would type into an IST SAF parser to perform the same actions. The example command shown will cause ModSAF to create a single T-72M at the given Cartesian location (x=28000, y=21000, z=0 but forced to local terrain elevation) and with a heading of 45 degrees (or northeast). Creating a vehicle is a multi-step process with ModSAF, so several lines of status information, including the new vehicle's IST-like entity number, its ModSAF vehicle number, and its ModSAF persistent object identifier as supplied. Internally, the IST command interface has created a data structure to store these and other data elements so it can translate IST references into appropriate ModSAF references.

The single-command command uses the same IST command parser as the file interpreter, but there is one important command difference. The IST command to query for information about an entity is '?' which is a special reserved character (help request) in the ModSAF parser. As a result it is impossible to enter the command "ist single 0?" (which would cause a status printout for IST entity 0). Instead ModSAF will try to explain the current legal options the instant the '?' key is hit. This is not a problem if the command occurs inside an IST-format script since the ModSAF parser does not actually read the script file. For single command purposes, the command "ist single 0q" will elicit the desired response ('q' is not a defined entity command in IST-SAF; this is just a work-around).

Another example: execute an IST-SAF script file: ist file 4113air.scr

The script file named is a standard compliance test script which creates an A-10 and puts it through a series of maneuvers to verify transmission of entity state information. Exactly the same file is read and it causes (as nearly as ModSAF allows) the same sequence of events to occur. IST-format scripts are executed as an independently scheduled series of events under the ModSAF scheduler. Each scripted operation produces "warm fuzzy" output to the text port describing the current scripted operations.

While an IST-format script is being executed, the command interface will not accept another "ist file" command. However, all the other parser (and ModSAF GUI) controls and commands remain available. Of course, the IST script-format command to execute another script (terminating the current script) is not subject to this prohibition.

Two other commands are available which impact the execution of IST-format command scripts. The command "ist abort" will stop the execution of a script which is already in progress. The command "ist continue" will force the script interpreter to continue with the next scripted command immediately, ignoring the current scripted delay between commands.

The last new command duplicates the IST-SAF display manager functions without actually using the IST command parser. This is a remnant from development but seemed a reasonable addition to the ModSAF command parser so it was not removed.

3.2 IST-SAF script format

The script files are ASCII text files, line-oriented. The line format is controlled by the contents of the first character of the line, as shown in Table 3.2-1 below.

Table 3.2-1. Script File Line Format

CHAR	LINE INTERPRETATION
[0-9]	Line is read as a float, white space, and an IST-SAF command. The float value is a delay (seconds) before the command is executed.
' '	Comment; rest of line ignored.
'*'	Echo comment; the rest of the line is sent to standard output.
'_'	Rest of the line is interpreted as a script processing directive. Only one such directive is currently defined: 'k', followed by a digit. The digit begins a float field, the float value being the delay (seconds) per character during script command entry.

Since IST-SAF is hosted on an IBM-compatible PC the script files use the DOS standard end-of-line (which is a two-character sequence) instead of the unix standard end-of-line (which is only one character). The ModSAF-IST script parser will accept scripts using either convention. This makes it easier to use IST-SAF scripts since the extra end-of-line characters don't have to be stripped.

Only two modifications to IST-SAF scripts are normally required to use them with ModSAF. The coordinates for entity creation and other activity generally need to be changed to ensure that test entities are placed at reasonable places in the database (this is a common change whenever IST scripts are used with different systems under test). Second, ModSAF is more stubborn about real-world time delays required for actions to occur. As a result the delays for certain actions (especially firing) usually need to be extended to allow ModSAF to model the required actions. Neither of these changes obstructs the purpose of providing test entities with test behaviors using ModSAF.

3.3 Implemented IST commands

Table 3.3-1 below presents a list of the IST commands implemented in the test tool, along with a very short description of what each command does. For more detailed information, see the IST-SAF user manual.

The IST-SAF command parser requires spaces between elements of a command. However, the actual commands are always single character-unique so that the spaces are not actually needed (they do improve the readability of a script file). The ModSAF parser does not require the spaces but does accept them. Except where noted, these commands are not case sensitive.

Table 3.3-1. IST Commands Implemented in the Test Tool

COMMAND	ACTION DESCRIPTION
<entity> Z <alt_agl>	(case sensitive) Set altitude of (aircraft) entity.
<entity> @ <target>	Entity fire at target, bypassing checks.
<entity> l <target>	Entity fire at target, bypassing checks.
<entity> t <rate>	Set entity turn rate (deg/sec, CW).
<entity>] 2	Go on current heading.
<entity>] 1	Entity Suicide (set damage to catastrophic).
<entity>] 3	Entity halt (just like "<entity> h).
<entity> ad	Set speed to "double", used to mean 2/3 of max.
<entity> an	Set speed to "normal", used to mean 1/3 of max.
<entity> ar	Set speed to "rush", used to mean 100% of max.
<entity> as	Set speed to "stop", meaning 0.
<entity> h	Stop entity.
<entity> p g <x> <y>	All these cause immediate "go" (vmove)
<entity> p g <x> <y> a	absolute x, y
<entity> p g <x> <y> r	relative to current position
<entity> p g <hdg> <junk> nsew	hdg is compass heading (degrees E of N).
<entity> g	Continue on to the next waypoint...
<entity> g f	Use route planned with "p" or "pa"
<entity> g w	Use route planned with "p w"
<entity> p <x> <y>	These are simple 1-step plans (here to there).
<entity> p <x> <y> a	absolute x,y
<entity> p <x> <y> r	relative to current position

Table 3.3-1. IST Commands Implemented in the Test Tool (continued)

COMMAND	ACTION DESCRIPTION
<entity> p <hdg> <junk> nsew	hdg is compass heading (degrees E of N).
<entity> p a <x> <y>	Append this new route to current route
<entity> p a <x> <y> a	absolute x, y
<entity> p a <x> <y> r	relative to current position
<entity> p a <hdg> <junk> nsew	hdg is compass heading (degrees E of N).
<entity> p w <x> <y> a	Append this point to current route
<entity> p w <x> <y> c	Clear the current current route
<entity> p w <x> <y> p	prepend this point to current route
<entity> (<rng>	Open fire at range.
<entity>)	Cease fire.
<entity> z <gunAz>	(case sensitive) Set turret azimuth
<entity> e <gunEl>	Set gun elevation
<entity> B <x> <y>	(case sensitive) Set position (maintain AGL alt).
<entity> f <hdg>	Set entity heading (degrees East of North). Does not work for moving entities.
<entity> ?	Dump of entity status information.
<entity> Q	Not IST! just like '?' but ModSAF Parser eats '?' on single commands (scripts are okay).
,	Toggle between local and global entity ID mode.
%	List all entities (IST and ModSAF vehicle ID's).
i ;	Prints local site and host values.
i r <entity>	Remove an entity from the simulation.
i c <type> <x> <y> <z> <Hdg	Create entity of given type. Type must be given, others default to 0.
do <x> <y>	Set display center to map coordinates (default 0 0).
dm <x> <y>	Move display center by coordinates (default 0 0).

Table 3.3-1. IST Commands Implemented in the Test Tool (continued)

COMMAND	ACTION DESCRIPTION
ds <ppm>	Set display scale in pixels-per-meter (default 1).
d	Set display scale to 1 and redraw screen.
ke <rest_of_line>	Echo print <rest_of_line>.
ks <scriptFileName>	Open scriptFileName and read commands from it. Current script (if any) is closed.

In many cases the IST-SAF documentation is ambiguous about the convention used for command data such as headings and angles. Where the actual test scripts make the intent clear the same conventions were used. In other cases the value was assumed to be consistent with similar values used with other commands. In only one case was an actual conflict encountered: The IST-SAF command to set an entity's turn rate "<entity> t <rate> is used in the scripts as if it represents degrees per second (having values like "9" expected to produce two complete circles in 80 seconds), yet the same command for a tank is obviously interpreted by the IST-SAF program as radians per second (values like 0.2 producing complete turns in about half a minute). Since the other references to heading were in degrees, the ModSAF command interprets the value as degrees per second in all cases.

DIS Logger/Scanner User Manual

1. Introduction

This document provides user information for the initial version of the DIS logger/scanner program. This version of the program is intended to provide the functionality of the IST logger and scanner programs on a Unix workstation, using X11/Motif for the graphical user interface.

The tool provides three basic functions: (1) Create a verbatim binary log file of the protocol data units (PDU's) transmitted during operation of DIS 2.0.3 simulations. (2) Replay those packets with the same order and timing as they were logged. (3) Provide means for users to identify PDU's of interest and to examine the contents of those PDU's in human-readable (text) format.

The program is controlled entirely through the GUI; there are no text port commands. However, some functionality intended for later versions of the program is currently implemented in crude form using the text port for output. Even in this crude form, these tools have proven useful and are documented here as currently implemented. In addition there are several warning messages which will eventually migrate to notifier pop-up windows but are currently simple text messages on standard output.

2. Getting Started

2.1 Starting and Stopping

Two files are required in order to use the tool. The first is the executable program, "dislgr". The second is the X-windows resource file "Dislgr", which defines default settings for various attributes of the program). Both files should reside in the same directory. By default, the program will create and search for log files in the current directory of the parent shell (i.e., the shell from which you execute the program. Once per login session and before running the executable, type the following command: "xrdb -merge Dislgr"

This will establish the default values for the program. Next execute the program. If your environment's PATH includes the current directory (a common situation for normal users), type "dislgr". If not (common for super-users), type "./dislgr". After a few seconds of initialization, the program's main interface window will appear.

There are five optional command-line arguments. These are listed below in Table 2.1-1. Their default values are established in the X resource file "Dislgr" and may be changed there.

Table 2.1-1. Optional Command-Line Arguments

COMMAND	DEFAULT VALUE
-netdev <device name>	: Network device name (et0, ec0, etc.)
-sync <True I False>	: Use synchronous UDP?
-port <nnnn>	: UDP port to connect
-exercise <nnn>	: DIS exercise to record/play
-logdir <path>	: directory for log files

The correct value for -netdev is a function of the hardware configuration of the workstation running the executable. For Silicon Graphics Indigo, Indigo 2 and Indy workstations, the usual device name is "ec0". For Silicon Graphics Crimson machines, the usual name is "et0". The values of sync, port, and exercise need to be compatible with the interacting DIS applications. The log file directory is simply used to assure that the named directory exists (if not, it will be created). Only the exercise value can be changed during execution; if the others are initialized to the wrong values, quit and restart the program.

To quit the program at any point, use the mouse to select (left mouse button) the "File" menu from the menu bar at the top of the main window. On the File menu, select "Quit". Alternatively, the program will cleanly exit by typing <ctrl>C in the parent text port. DO NOT use the X-window frame's "Close" or "Quit" options. These are not yet handled properly and will either shut down the GUI (leaving the program running) or will cause a segmentation fault and core dump.

2.2 The Main Window

The main interface window (Figure 2.2-1) contains a menu bar, a mode control area, a player control panel (which resembles the buttons on a VCR), a traffic monitor, a scanner list window (not visible in record mode), and a scanner text window (also not visible in record mode). Most of these will be discussed in more detail in later sections.

```
        Main Window:
+-----------------------------------+
| menu bar                                    |
+-----------------------------------+
| mode control area                           |
+-----------------------------------+
| player control panel                        |
+--------------------------------[ ]-+
|                                             |
| traffic monitor                             |
|                                             |
+--------------------------------[ ]-+
| scanner list (playback only)                |
+--------------------------------[ ]-+
| scanner text (playback only)                |
+-----------------------------------+
```

Figure 2.2-1. Main Interface Window Display

The menu bar and mode control area are fixed in vertical height. The subwindows below the mode control area are adjustable in height. The borders between adjustable regions are shown as thin horizontal lines with tiny square buttons near their right ends. Use the mouse to resize these windows at any time as follows: Place the mouse pointer over the button, press and hold the left mouse button, and slide the mouse up or down. The window border will follow the mouse. Release the left mouse button when the border is in the desired location.

Two menus are currently installed on the menu bar (Figure 2.2-2). The "File" menu should be familiar to most window-application users: its two choices are "Open..." and "Quit". The "Filter" menu offers three pop-up windows controlling the three PDU filters which are explained later.

```
Menu Bar:
+----------------------------------------------------------+
| +"File"-----+   +"Filter"-----+                           |
+-|          |----|             |-----------------------------+
  |"Open..." |    |"Entity ID"  |
  |"Quit"    |    |"PDU Type"   |
  +----------+    |"Time"       |
                  +-----------+
```

Figure 2.2-2. Menu Bar Display

The mode control area (Figure 2.2-3) shows the current program mode, the port being used for network communication, and the DIS exercise number. The mode may be changed by c.ı ing on the diamond beside the desired mode. The port number cannot be changed without restarting the program. The exercise number may be changed by selecting (left mouse, as usual) the box and entering a new number. Whenever the user enters information into a box like this, the return key must be pressed to complete the change. The new data is then checked and applied. Exercise ID numbers are limited to the range
{0 .. 255}. Exercise 0 has special meaning (courtesy of the ModSAF packet valve): record all exercises on the indicated port.

```
Mode Control Area:
+----------------------------------------------------------+
| Mode:   <> Record   <> Playback     [nnnn]    [nnn ]      |
|                                      Port     Exercise    |
+----------------------------------------------------------+
```

Figure 2.2-3. Mode Control Area Display

There are two fundamental operating modes. Record mode uses the network as a source of PDL's which can be monitored and/or recorded. This is the initial mode of the program. Playback mode uses a binary log file as the PDU source and can either play PDU's back to the network or examine the data in detail.

Below the mode control area is the player control panel (Figure 2.2-4). At startup all the buttons on this panel will be inactive and "faded". This is because there is as yet no file to record or play back, and therefore none of the player controls is allowed at this point. As the available options change during use of the logger, these buttons will be active or inactive as appropriate to the current state of the program.

```
Player Control Panel:
+------------------------------------------------+
|   "||"    "[]"    ">"    "()"    "(*)"    "|<<"  |
+------------------------------------------------+
  (pause)  (stop)  (play)  (loop)  (record) (rewind)
```

Figure 2.2-4. Player Control Panel Display

2.3 Recording Packets

When entering record mode the program begins monitoring PDU traffic on the network. The traffic monitor (explained in a later section) displays the traffic statistics as they are collected. To actually record the traffic the user must select a file to receive the data and start the recording.

To open a file, pull down the "File" menu and select "Open..." A standard Motif file selection box will appear (Figure 2.3-1):

```
File Selection Box:
+-----------------------------------------------+
| Filter                                              |
| [ filter entry box ]                                |
| Directories               Files                     |
| +--------------------+    +-------------------+     |
| |                    |    |                   |     |
| | (list of directories)  |  | (list of files)  |  |
| |                    |    |                   |     |
| +--------------------+    +-------------------+     |
| Selection                                           |
| [ file name entry box ]                             |
+-----------------------------------------------+
|           "OK"        "Filter"     "Cancel"         |
+-----------------------------------------------+
```

Figure 2.3-1. Standard Motif Selection Box Display

The Filter entry box defines the search path and wildcards used to build the list of files. To change the search directory to a directory shown in the directory list, double-click on that directory name. Since the program is in record mode the file selection box will not allow selection of any existing file names. The user must type a name into the file name entry box. If the named file exists with a non-zero size the open will fail.

After opening a new file to receive the PDUs, the program is ready to record but remains in monitor mode. To actually start recording, select the player control panel's record button (the red dot-and-circle, second from the right) which is no longer shaded to indicate that recording is now permitted. Recording begins immediately. At this time, the record button becomes inactive and the pause and stop (leftmost two) buttons become active. To stop the recording, select stop. As soon as stop is selected, the file which was just created is no longer acceptable for recording, so the playback buttons become active. Automatically this file is rewound, closed, and reopened as an input file. To record more data, a new file must be opened.

During record operation, the pause key is also active. When selected the program will enter the pause state. PDU's received during pause are not recorded. Selecting either pause or record ends the pause state and recording resumes. The log file timestamps (but NOT the PDU timestamps) are adjusted by the length of time spent paused.

2.4 Playback

To play back a file, first change the program mode to playback by selecting the playback button in the mode control area. If recording was just done, the newly created file is already loaded for playback. If not, or if another file is desired for playback, pull down the "File" menu and select "Open". This time the selected file must exist and must have a non-zero size.

With a file loaded the player control buttons are again reset: The play and loop buttons are active along with rewind (the rightmost button). Begin the replay by selecting the play button. The stop button will stop the replay, or replay will automatically end when the end of file is reached. To replay the file again, select the rewind button and then select the play button again (rewind, unlike a VCR, takes only an instant).

Frequently we want to replay a logged sequence of PDU's from start to finish, rewind, and play again, indefinitely. The loop button causes this to occur. Loop operates exactly like play, except that instead of stopping at the end of file, loop automatically rewinds and restarts the playback. Loop can be selected while play is occuring, and play can be selected while loop is running since the only difference is in how end-of-file is handled. However, if loop is desired AFTER play has hit the end-of-file, the file must first be rewound.

The pause button is active during playback. When selected, the logger stops replaying and waits for the pause to end. Pause is ended when pause, play, loop, or stop are selected. Unless the pause is ended with a stop the playback simply picks up where it left off.

The scanner operates in conjunction with playback mode. When playback mode is selected the scanner adds two sub-windows to the bottom of the main window. Ignore these for now unless scanner operation is desired. See the sections on filters and the scanning for information on the scanner.

3. The Traffic Monitor

The traffic monitor (Figure 3-1) provides at-a-glance information on the DIS PDU rates and cumulative totals being received or transmitted. The monitor also is used during file scanning to show the total number of PDU's passing the current filter set (see the section on file scanning). The monitor consists of an array of labeled boxes, most of which are immediately recognizable as DIS PDU types. These boxes indicate the total number of PDU's of that type accumulated since the last reset of the monitor. The monitor is reset when the program mode changes, a file is loaded, or the record, play, or stop states are entered.

```
Traffic Monitor:
+-----------------------------------------------------------------+
| Other   Collis  ResCan  RemEnt  ActReq  Data    Laser   total      |
| [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]        |
| EntSta  SerReq  RepCom  StaRes  ActRes  Event   Transm  Inst. Rate |
| [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]        |
| Fire    ResOff  RepRes  StoFre  DatQue  Messag  Signal  Avg. Rate  |
| [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]        |
| Detona  ResRec  CreEnt  Acknow  SetDat  Emissi  Receiv             |
| [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]     [ ]        |
+-----------------------------------------------------------------+
```

Figure 3-1. Traffic Monitor Display

Four boxes are not DIS PDU types. The box in the upper left corner counts all the PDU's which were of types not known to the program. These are logged, are transmitted during playback, and can be examined using the the scanner, although they are simply treated as unstructured blocks of binary data.

The column of boxes on the right side of the monitor gives summary data covering all PDU types (including the "Other" types). The top item is a grand total PDU count. Below that is an "instantaneous" rate, which is actually updated approximately once per second. The last box is a time-weighted-average PDU rate. The time weighting scheme used is an exponential decay with a time constant of 30 seconds.

When the program is in record mode but not actually recording the traffic monitor accumulates data from the network. It can be easily reset by switching to playback mode and then back to record mode.

When recording begins the monitor resets and displays the traffic being recorded. When recording stops the monitor resets and again shows current network traffic.

When the program is in playback mode but not actually playing the monitor is reset. If a file is loaded the scanner can be used. In this case the monitor will show counts of which PDU's passed the most recent filtering action, but the two rate outputs are meaningless. During actual playback the monitor will show the traffic being placed on the network from the file as well as the current and averaged PDU transmission rate.

If the traffic monitor is not visible during playback mode, move the subwindow boundaries (see section 2, "Getting Started") to expose it.

4. Filters

The scanner portion of the program provides three PDU filters. However, these filters have significant (although only partially implemented) value during record and playback operations as well.

A PDU filter is a function which yields a Boolean (True or False) result when applied to a particular PDU. By combining such filters using Boolean algebra the user can define an arbitrary subset of PDU's for consideration. In later implementations this will include the ability to log or replay only some of the PDU's instead of all of them. In this initial implementation however, the filters are combined only as a logical "and" of all three filters. That is, a PDU must pass all three tests in order to pass the filter set. Initally the filters are set to pass all PDU's and there is a button to restore that state for each filter type.

All PDU filter windows have "Apply" and "Done" buttons at the bottom. If a readable log file is loaded the "Apply" button will cause the scanner to apply all three filters to the file and generate a list of PDU's which pass the filter. See the section on scanning for more on this. If a filter has been changed the "Done" button will also cause a scanner pass on the file. In any event, the "Done" button will close the filter window.

Access the PDU filters by pulling down the "Filter" menu and selecting the desired filter. Each filter has a pop-up window from which the filter's parameters are controlled. These are "sticky" windows: They hang around until specifically dismissed via the "Done" button. The filter windows can be moved, resized and iconified just like any other window.

4.1 PDU Type Filter

The simplest PDU filter is the PDU type filter (Figure 4.1-1). Pull down the "Filter" menu and select "PDU Type". Put the new window in a convienent location. This window contains an array of toggle buttons, each labeled for a PDU type. Initially all the buttons are "on" or depressed. Clicking on a particular button toggles its state. Frequently only a single PDU type is of interest at a particular moment. This is easily accomplished by clicking on the "All Off" button at the bottom of the window, then clicking on the desired PDU type(s). The "All On" button will reset this filter to the initial pass-everything state. Only the scanner currently uses this filter.

```
PDU Filter:
+------------------------------------------------------------+
|  [] Other    [] ResRec   [] StoFre   [] Event    |
|  [] EntSta   [] ResCan   [] Acknow   [] Messag   |
|  [] Fire     [] RepCom   [] ActReq   [] Emissi   |
|  [] Detona   [] RepRes   [] ActRes   [] Laser    |
|  [] Collis   [] CreEnt   [] DatQue   [] Transm   |
|  [] SerReq   [] RemEnt   [] SetDat   [] Signal   |
|  [] ResOff   [] StaRes   [] Data     [] Receiv   |
+------------------------------------------------------------+
|  "All Off"   "All On"    "Apply"     "Done"      |
+------------------------------------------------------------+
```

Figure 4.1-1. PDU Filter Display

4.2 Time Filter

Pull down the "Filter" menu and select "Time" (Figure 4.2-1). Put the new window in a convenient location. The time filter displays the program's current time value and allows specification of a start and stop time. PDU's which were logged after the specified start time and before the specified end time will pass this filter. These times are specified as absolute or relative to the current time according to which box is checked below the current time display box. Internally, the logger maintains time values in milliseconds; therefore time values are always displayed in seconds with three decimal places. However, the logger file format duplicates the IST logger file format so the PDU's are only logged to the nearest 10 milliseconds. The first PDU in a log file is logged at "absolute" time. The first PDU in a log file is logged at "absolute" time zero and all other zero and successive PDU's are logged according to how much later they were received.

Time filter:

```
+-------------------------------------------------------------------+
|  [ nnnn.nnn ]        [ entry box ]        [ entry box ]           |
|  Current Time        Start Time           Stop Time               |
|  <> Absolute         "-Infinity"          "+Infinity"             |
|  <> Relative                                                      |
+-------------------------------------------------------------------+
|  "Forever"           "Apply"              "Done"                  |
+-------------------------------------------------------------------+
```

Figure 4.2-1. Time Filter Display

The initial filter condition is absolute with the start time value set to "-Infinity" and end time set to "+Infinity". This state causes all logged PDU's to pass the filter and may be restored by selecting the "Forever" button at the bottom of the window. The "-Infinity" button will restore the start time to the earliest possible time (effectively the beginning-of-file time). The "+Infinity" button restores the stop time to the latest possible time (effectively the end-of-file time).

During scanner operation the current time can be modified by selecting that box, editing the time value as desired, and pressing the return key. In absolute mode this has no actual effect, but by checking the relative mode box the start and stop times are converted to offsets from the value of current time. For example, set the current time to 10.0 (don't forget the carriage return when finished--this causes the value to be read, checked and applied internally). Next select the checkbox labeled "Relative". Select the start time entry box and enter "-1", then select the stop time box and enter "+1". The time filter will now pass PDU's from -1.000 to +1.000 seconds from the current time of 10.000, or between 9.000 and 11.000 seconds absolute time. Enter a new current time of 20 and the filter will pass PDU's from 19.000 to 21.000 seconds absolute time. Finally, select absolute mode and observe that the start and stop times hav been converted from relative to absolute so that the filter continues to pass the same PDU set.

During record and playback operations the current time value is automatically updated to show the current log time being recorded or played back. This provides useful feedback as operation continues. More importantly this feature can aid scanner operation: pause the playback when an interesting event occurs and use relative time mode to examine PDU's logged near that point in time.

The time filter offers one more useful feature: If the time mode is absolute during playback, a "rewind" will go to the specified start time if it is later than the beginning of the file. Playback will stop (and loop playback will rewind and restart) when time reaches the specified stop time. This permits the playback of an arbitrary time segment within the file.

4.3 Entity ID Filter

The logger/scanner maintains two lists which together contain all the DIS entity Identifiers (site/application/entity) known to the program. The lists are shown in the entity ID filter window (Figure 4.3-1), accessed by pulling down the "Filter" menu and selecting "Entity ID". The "show" list (on the left side of the window) contains entity ID's which will pass this filter. The "hide" list contains all the entity ID's which will fail this filter. The user directly controls which entities are on which list using the controls provided in the filter window.

Entity Filter:

```
+------------------------------------------------------------------------+
| "Show:"                                  "Hide:"                        |
|+--------------------------------+        +----------------------------+ |
||                                |        |                            | |
||  (list of nnn/nnn/nnnn)        |        |  (list of nnn/nnn/nnnn)    | |
||                                |        |                            | |
|+--------------------------------+        +----------------------------+ |
| [entity ID entry box]                    <> Show New Entities           |
| "Show"       "Hide"     "Flush Lists"    <> Hide New Entities           |
| -----------------------------------------------------------------------|
|      "Hide All"       "Show All"    "Apply"          "Done"            |
+------------------------------------------------------------------------+
```

Figure 4.3-1. Entity Filter Display

The initial state of the entity filter places every known entity on the show list. This state may be restored by selecting the "Show All" button at the bottom of the window. Whenever a new entity ID is first encountered, the program automatically adds it to the appropriate list as directed by the "Show New Entities"/"Hide New Entities" check boxes. Note that the "Show All" button also sets this selection to "Show New Entities".

Below the show list is an entity ID entry box. Like the other entry boxes this one is used by selecting it, entering the desired text, and pressing return. In this case the text is expected to be and entity identifier in the form of three integers seperated by two non-integers ('/' is preferred but the program attempts to handle any non-digit as a separator). In playback mode the parsed entity ID is rejected if the entity is not known in the currently loaded file. In record mode such checking is impossible since the requested entity may not yet be active. Once an entity ID is entered it can be placed on either list by selecting the "Show" or "Hide" button as desired. If the entity is already on the selected list nothing happens. If the entity is currently on the other list, it is moved from there to the selected list.

If an entity is already on one of the lists, select it and it will be placed in the entry box (which saves typing and avoids possible errors. An even more direct way to move an entity from one list to another is simply to double-click (left mouse, rapidly) on the entity. The entity will be moved directly to the other list.

The "Hide All" button moves all known entities to the hide list and selects "Hide New Entities", making it the complete inverse of "Show All". This is most useful as a first step in selecting only a few entities for the show list.

During record mode the logger maintains a list of entity identifiers found in PDU's received. Although no actual filtering occurs, the list itself is useful because it shows all the players on the selected port and exercise. Since the list is cumulative, entities may be listed even though considerable time has elapsed since the last PDU referencing them. The "Flush Lists" button clears the list--active entities will reappear within a few seconds.

In many cases a DIS PDU will actually contain two entity ID's. An example is the collision PDU which has an ID for the entity which detected the collision as well as the ID of the entity collided-with. In all such cases the PDU is considered to "belong" to the entity which initiates the PDU and the PDU will pass the filter only if that entity is on the show list. Unless the collided-with entity also transmits PDU's (collision, entity state, or whatever) it will not even appear on the entity lists. So far this has not been a problem since an entity which does not transmit PDU's generally does occur.

5. Scanning

The scanner only operates in playback mode. When playback mode is selected the scanner adds two subwindows to the bottom of the main window but does not resize the main window. Eventually the scanner will open its own window, but for now the user must resize the main window to use the scanner. The suggested size is to make the main window as tall as the entire screen but leave the width alone. Then grab the lowest adjustable subwindow border and move it to a position about 1/3 of the main window height from the bottom. Grab the second-lowest adjustable border and move it to a position roughly 1/3 of the main window height from the top. This should expose the whole traffic monitor as well as the scanner list window and scanner text window. Both are scrolled windows although the scroll bars are not visible unless the windows are too small to display all the information provided by the scanner.

5.1 Applying Filters

To use the scanner, select playback mode, resize the window as described above, and load an existing log file. If there are no filter windows up pull down the "Filter" window and select a filter (see the section on filters for details). Select "Apply" in any filter window. This will cause the scanner to run through the log file index, applying the three filters to the PDU's and constructing a list of PDU's that pass all three filters.

If the filters were in their default state and the file is large, the scanner may take several seconds to complete its scan pass. During this time periodic messages will appear on standard output (the parent textport) to give some evidence of continued program life. The final list of PDU's will be displayed in the scanner list window (Figure 5.1-1). If more than 500 PDU's make the list, only the first 500 will be displayed and a warning will appear on standard output.

List Window:

```
+-----------------------------------------------------------+--+
| 1     : Logged time  : PDU type   : primary entity     |/\|
| 2     : Logged time  : PDU type   : primary entity     |__|
|                                                         |__|
|        . . .                                            |  |
|                                                         |  |
| nnn   : Logged time  : PDU type   : primary entity     |\/|
+- ---------------------------------------------------------+--+
```

Figure 5.1-1. Scanner List Window Display

The true total number of PDU's in the scan list will be shown on the traffic monitor, including the number by PDU type. If the list is large, this will be helpful in determining which PDU types might be excluded (using the PDU type filter) to bring the list down to a managable size. This also provides a quick way to verify that particular PDU's exist in the scan list (without scrolling through 500 entries and reading them). The scan may be repeated with different filter settings by going to the desired filter window, making the desired change, and selecting "Apply".

The most frequent situation is to filter for one or two PDU types from one or two entities during a short time interval of interest. Another use is to check the file for any occurances of a PDU type. For example, if the log file is of a collision detection test, select "All Off" on the PDU type filter and select the collision PDU type to turn it back on. Select the "Forever" button on the time filter and "Show All" on the entity ID filter, then select "Apply" (on any filter). The scanner will produce a list of every collision PDU in the log file, giving the time of receipt and the entity sending the PDU.

Knowing the time and entity involved in the collision permits a different filter definition to examine entity behaviour during collisions (to continue the example). Set the start and stop times to a few seconds before and a few seconds after the collision, select "Hide All" and then double-click on the entity listed for the collision PDU, select "All On" on the PDU type filter, and again select "Apply". The new scanner list will show the sequence of PDU's from that entity about the time of the collision.

5.2 Examining PDU contents

Having reduced the scanner list to a reasonable size, select one of the PDU's on the list. Immmediately the scanner will produce a formatted text version of the contents of the PDU in the text window (Figure 5.2-1). The PDU data is generally shown in both hexadecimal format and an a format more appropriate to the actual data type for each element.

Text Window:

```
+------------------------------------------------------------+--+
|              (formatted text dump of DIS PDU contents)      |/\|
|                                                             |__|
|                                                             |__|
|                                                             |\/|
+------------------------------------------------------------+--+
|<  [_________________________]                             >|  |
+------------------------------------------------------------+--+
```

Figure 5.2-1. Text Window Display

Figure 5.2-2 details an example of an Entity State PDU text dump.

```
--------------- DIS PDU ----------------
Version:              (0x0003 =    3)
Exercise:             (0x0001 =    1)
Kind:                 (0x0001 =      1) ENTITY_STATE
Time Stamp:           (0x2C565CA4 = 743857316)
Length:               (0x0090 =  144)
Entity ID:            (0x0004:0x00FD:0x03F1 = 4:253:1009)
Force ID:             (0x01 =    1 = FORCE_FRIENDLY)
Entity Type:          (0x01,02,00E1,02,04,00,00 =  1,  2,  225,  2,  4,  0,  0)
Alt. Ent. Type:       (0x01,02,00E1,02,04,00,00 =  1,  2,  225,  2,  4,  0,  0)
Location      X: -2676521.8826907 Y: -4423237.3484584 Z: 3728908.8564447
Velocity      X: -212.1642914       Y: 81.6250534        Z: -54.8714409
Orientation   Psi: 2.7743227        Theta: 0.2368490 Phi: 3.5228243 RADIANS
D.R. algorithm:       (0x02 =  2)
Accel.        X: 0.0000000          Y: 0.0000000         Z: 0.0000000
Ang. vel.     X: 0x00000000         Y: 0x00000000        Z: 0x00000000
Character Set:        (0x01 =   1)
Marking:      "IST_0"
Capabilities:
Repair:       False
Recovery:     False
Fuel:         False
Ammo:         False
Num Parts:    (0x00 =  0)
```

Figure 5.2-2. Entity State PDU Text Dump Example

If the PDU type is not understood by the program, the listing will show it as "UNKNOWN_PDU KIND" These PDU's should still have valid DIS PDU headers, but the data fields cannot be interpreted. In such cases the PDU can still be selected for text dump, but the dump will be in a generic hexidecimal format. Figure 5.2-3 details an example of an unknown PDU text dump.

```
--------------- DIS PDU ---------------
Version:     (0x0003 =    3)
Exercise:    (0x0001 =    1)
Kind:        (0x00C8 =      200) UNKNOWN PDU KIND
Time Stamp:  (0x03447E84 =  54820484)
Length:               (0x0024 =   36)
Hexidecimal dump of 36 bytes:
03030303     03030303      00000000      00000000
01010101     00000000      01010101      00000000
C0C0C0C0
```

Figure 5.2-3. Unknown PDU Text Dump Example

5.3 Dead Reckoning Analysis

A very crude dead reckoning analysis tool is already in place with this version of the scanner. A more comprehensive tool is planned, but an ongoing project needed a simple tool and so this preliminary version was created. The tool operates whenever the file is scanned with just one entity ID on the show list and with entity state PDU's selected to pass the filter test. The tool projects each PDU's velocity forward to the time of the next state PDU and computes the residual (the difference between the projected position and the new PDU position). This is computed two ways: The first "correct" way uses the PDU timestamp, while the second method uses the time of receipt (Logged time) of the PDU and thus incurs errors due to PDU latency jitter. The results of this analysis are currently sent to standard output. Figure 5.3-1 details a sample dead reckoning analysis output.

```
PDU dt 0.135: -0.02 0.18 0.54 0.57 | LOG dt 0.141: 1.26 -0.25 0.94 1.59
PDU dt 0.135:  0.01 0.18 0.54 0.57 | LOG dt 0.141: 1.28 -0.27 0.92 1.60
PDU dt 0.135:  0.01 0.15 0.49 0.51 | LOG dt 0.070: 3.57 -5.18 3.25 4.88
PDU dt 0.136:  0.01 0.15 0.49 0.51 | LOG dt 0.141: 1.07 -0.26 0.76 1.34
PDU dt 0.135:  0.04 0.18 0.57 0.60 | LOG dt 0.141: 1.31 -0.33 0.88 1.62
PDU dt 0.135:  0.05 0.16 0.52 0.55 | LOG dt 0.141: 1.32 -0.37 0.80 1.59
PDU dt 0.135:  0.05 0.13 0.47 0.49 | LOG dt 0.141: 1.32 -0.42 0.73 1.56
```

Figure 5.3-1. Sample Dead Reckoning Analysis Output

The output is in two major columns: the first uses the PDU time stamp and the second uses the logged receipt time. The fields of each column are the time difference between two consecutive PDUs, the Geocentric position residual's X, Y, and Z components, and the distance between the dead reckoned position and the new PDU position.

So far at least one DIS system has been found transmitting corrupted PDU time stamps, an error which became obvious when this tool was applied. Another system (both were under development when tested) was using the time of receipt instead of the PDU time. In the example above the position residuals are small (about half a meter) when the PDU time stamps are used, but using the time of receipt introduces substantially larger and more erratic errors (as expected).

6. Known Bugs

Entity ID Filter problem: Sometimes when an entity is selected on one of the filter lists (which puts it into the entry box at the bottom of the panel) and then the "Show" or "Hide" button is clicked, the entity ID gets mangled when it is added to the appropriate list. The work around is to always double-click entities to move them from one list to another.