



**SIMDIS**  
ANALYSIS & DISPLAY

# Distributed Interactive Simulation Plug-in (DIS)

User Manual

September 2020

---

**U.S. Naval Research Laboratory**

ELECTRONIC WARFARE MODELING

AND

SIMULATION BRANCH

CODE 5770

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| 1.1      | Distributed Interactive Simulation (DIS) Plug-in . . . . . | 1         |
| <b>2</b> | <b>Configuration</b>                                       | <b>3</b>  |
| 2.1      | Requirements . . . . .                                     | 3         |
| 2.2      | Installation . . . . .                                     | 4         |
| 2.2.1    | DIS Plug-in Load Error . . . . .                           | 5         |
| 2.3      | DIS Plug-in Execution . . . . .                            | 6         |
| 2.4      | Network Pane . . . . .                                     | 8         |
| 2.5      | Statistics Pane . . . . .                                  | 9         |
| 2.6      | Application Pane . . . . .                                 | 10        |
| 2.7      | DIS Pane . . . . .   | 12        |
| 2.8      | Models Pane . . . . .                                      | 14        |
| 2.9      | About Pane . . . . .                                       | 15        |
| <b>3</b> | <b>Operation</b>   | <b>17</b> |
| 3.1      | Processing DIS Data . . . . .                              | 17        |
| <b>A</b> | <b>DIS Configuration File</b>                              | <b>25</b> |
| A.1      | Configuration File Format . . . . .                        | 25        |
| A.2      | Entity Type Model Mapping File Format . . . . .            | 31        |

# Chapter 1

## Introduction

### 1.1 Distributed Interactive Simulation (DIS) Plug-in

The DIS Plug-in for SIMDIS is written using the C++ SIMDIS Plug-in API distributed by the Electronic Warfare Modeling and Simulation Branch of the Tactical Electronic Warfare Division at the Naval Research Laboratory. The software combines version 1.22 of the SIMDIS Plug-in API library and version cpp-1.0 of Open-DIS, an open source implementation of the Distributed Interactive Simulation Protocol, developed mainly by the MOVES Institute at the Naval Postgraduate School. Together, they provide a means to subscribe and view data real time using the SIMDIS 3-D advanced visualization tool.

The Plug-in API defines a set of standard SIMDIS functions that are made available to a plug-in application. Included in the API are functions that provide SIMDIS-compatible cross-platform graphical user interface (GUI) development, data retrieval and injection functions, and functions to control the way the user may view SIMDIS. These functions allow the plug-in to seamlessly integrate with SIMDIS and provide specialized utility to the end user above and beyond the capabilities distributed with the base version of SIMDIS.

The DIS plug-in processes Entity State, Detonation, Weapon Fire and Emission Protocol Data Units (PDUs). The DIS plug-in does not perform dead reckoning, but does perform special processing for entities that are indicated to be static (by being given dead reckoning type 1). The DIS plug-in expects to receive (and therefore only processes) updates for such entities on the timeout interval. Updates that arrive for static entities between timeout intervals are processed only for non-positional changes such as appearance changes. The plug-in can restamp time based on time of arrival to achieve a monotonically increasing time stamp. If restamping is not used, then the PDU time stamp is expected to contain the number of time units since the start of an hour if an absolute time marker is received. Relative time or non-real time modes that required input from additional simulation management PDUs are not supported.

The SIMDIS DIS plug-in is multi-threaded, meaning the graphics visualization happens in the main application thread while the polling for available DIS data occurs in a background thread. The default background thread timer for polling the DIS socket is set to 5 Hz. If

the graphics frame rate drops below the update rate of participants, the callback delay, thread timer and various display graphics can be adjusted to improve performance. For exercises with a large number of slow updating participants, 10000 at 1 Hz, or exercises with a large number of fast updating participants, 1000 at 10 Hz, could present problems with the graphics redraw and subsequently the polling of available DIS data. Results will vary and depend largely upon the computer and video graphics hardware used.

# Chapter 2

## Configuration

### 2.1 Requirements

The plug-in is meant to be run with the latest version of SIMDIS, however it is compatible with versions of SIMDIS that support the Plug-in API version 1.22 and newer. Refer to the Plug-in API compatibility matrix at <https://simdis.nrl.navy.mil/PluginApiCompatibility.aspx> for which releases of SIMDIS support which Plug-in API versions. Comments, issues or feature requests regarding the DIS plug-in can be sent via the SIMDIS help desk at <https://simdis.nrl.navy.mil/jira>.

Refer to the SIMDIS User Manual for the minimum and recommended system configurations.

## 2.2 Installation

Extract the DIS plug-in from the Optional Plug-ins distribution to the SIMDIS installation folder found on your system. The DIS plug-in requires version 10 SR1 (or newer) to be installed first.

The following files will be needed on your system:

1. DIS plug-in configuration file: **sampleDIS.cfg**

Placed in **SIMDIS/config/DIS** folder

2. DIS entity type to SIMDIS model mapping file: **model.dis**

Placed in **SIMDIS/config/DIS** folder

3. SIMDIS DIS plug-in: **piDIS\_win64\_vc-14.0.dll** (as an example)

Placed in **SIMDIS/config/DIS** folder

Once the DIS plug-in has been installed, SIMDIS will automatically detect it. This will be indicated by the New Plug-in Found dialog. To load the plug-in SIMDIS select the **Continue** button.

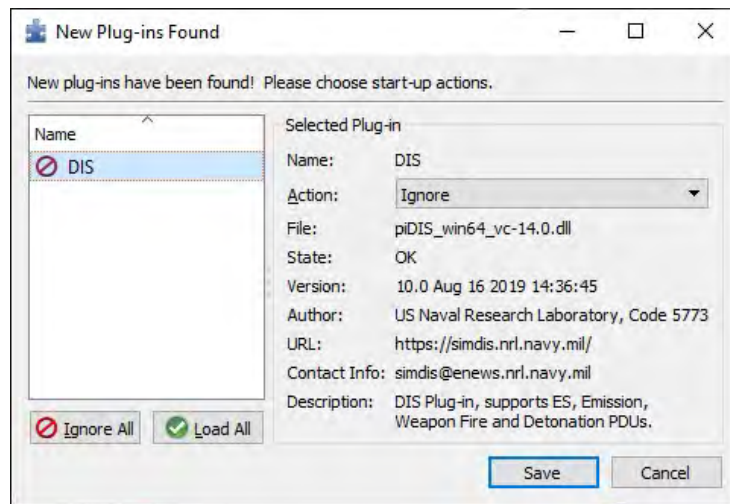


Figure 2.1: New Plug-in Found Dialog

Once successfully loaded, the plug-in can be accessed via the Plug-in Tools pull down menu or via the **[F]** hot key. Either press **[F]** or select the **DIS** option and the DIS plug-in will appear.



Figure 2.2: Plug-in Tools with Loaded DIS Plug-in

### 2.2.1 DIS Plug-in Load Error

If the plug-in fails to load or start, then either an error dialog will appear or console errors will be reported.

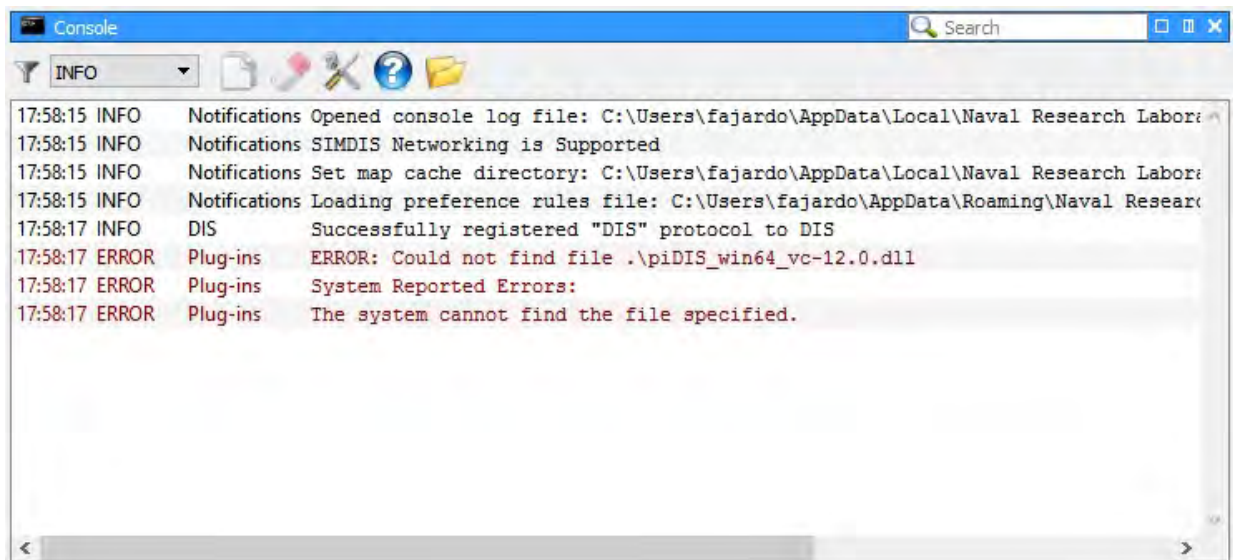


Figure 2.3: Console Plug-in Load Error

The failure to load the plug-in can be attributed to several factors:

1. The plug-in is loaded in an older version of SIMDIS. The DIS plug-in is based on version 1.22 of the SIMDIS Plug-in API, thus only SIMDIS version 10 SR1 or newer will load the plug-in.
2. A compiler mismatch between the plug-in and SIMDIS. The compiler version of the plug-in, vc-10.0, vc-12.0, vc-14.0, vc-14.1, or vc-14.2 must match the version used for SIMDIS. The SIMDIS compiler version can be determined from the About dialog found in the SIMDIS Help menu.

## 2.3 DIS Plug-in Execution

Once the DIS plug-in dialog is visible, a user can connect to an existing DIS broadcast by pressing the Connect button. Once connected, the text field options for the DIS IP address and port will be disabled. These options are “grayed out” as evident in the picture of the plug-in on the right.

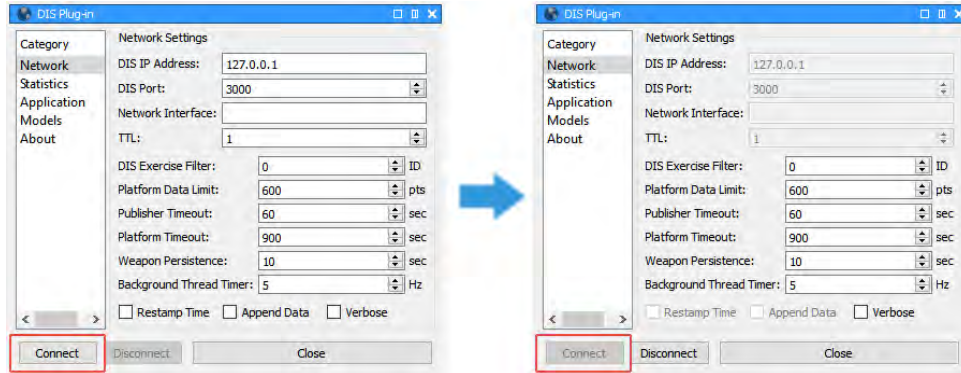


Figure 2.4: DIS Plug-in

The **Disconnect** button will allow the DIS plug-in to disconnect from the DIS broadcast feed and subsequently place SIMDIS in file mode. A user can connect and disconnect as often as they like, however previously recorded data will be lost upon a reconnection. Once the plug-in has disconnected from the DIS feed, previous options that were disable are now enabled. The **Close** button on the bottom right of the dialog hides the dialog. This button has no effect on the reception of data and simply hides the dialog.

If either the DIS IP address or port is misconfigured the dialog shown in [Figure 2.5](#) will appear. The dialog will continue to appear until the proper settings are entered.

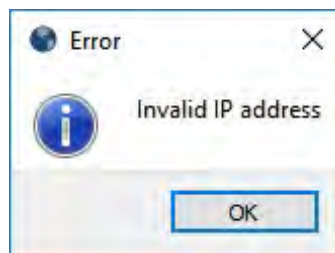


Figure 2.5: DIS Plug-in Error Notification



As seen in [Figure 2.6](#), the DIS plug-in contains six different categories:

1. Network Pane
2. Statistics Pane
3. Application Pane
4. DIS Pane
5. Models Pane
6. About Pane

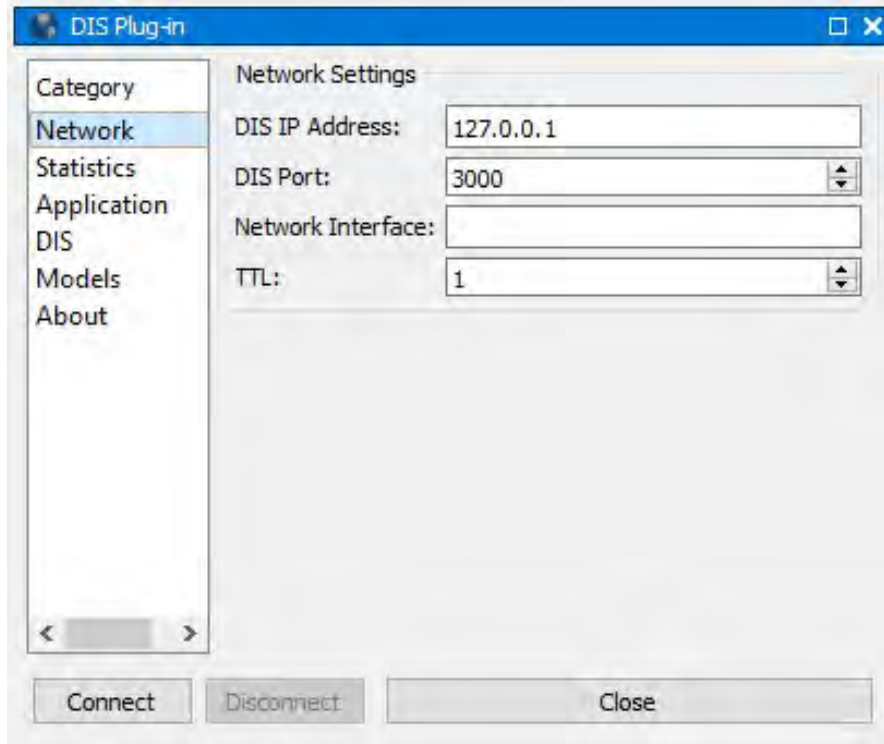


Figure 2.6: Plug-in Category Panes

The operation and settings for each of these category panes will be explained in detail.

## 2.4 Network Pane

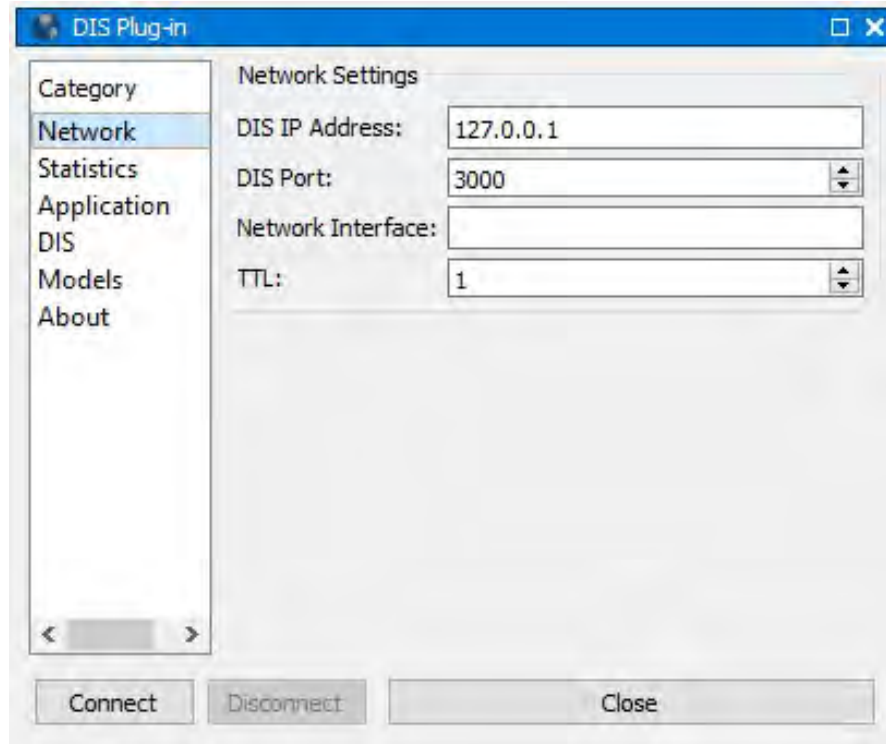


Figure 2.7: Network Pane

The starting category pane for the DIS plug-in is the **Network** pane. Within the Network pane is a section devoted to the options for the DIS Configuration. In this section, the user can make changes to the:

- DIS publisher IP Address. To connect to a unicast feed, use 127.0.0.1. An attempt is made to open a broadcast socket first. If the broadcast socket fails another attempt is made to open a multicast socket.
- DIS publisher Port. The supported range of ports is 1025 to 65535.
- Network Interface. The address to use for network reception. Empty value defaults to any/all interfaces. Useful for setting the interface on multi-homed machines.
- TTL is the time to live value for multicast data transmissions. 0 is restricted to the same host. 1 is restricted to the same subnet. 32 is restricted to the same site. 64 is restricted to the same region. 128 is restricted to the same continent. 255 is unrestricted.

## 2.5 Statistics Pane

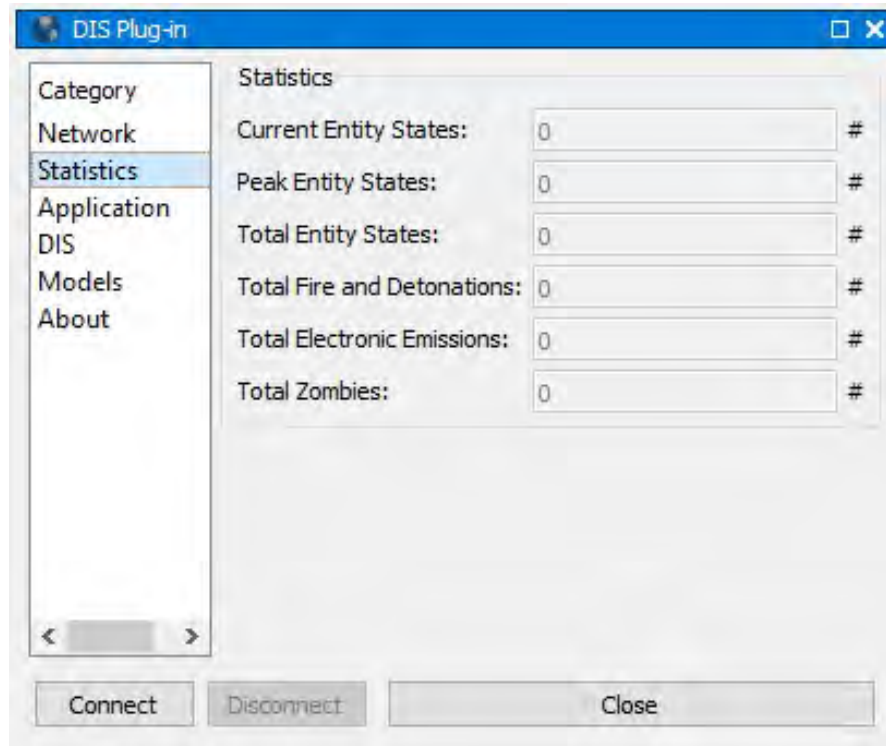


Figure 2.8: Statistics Pane

The next category pane for the DIS plug-in is the **Statistics** pane. In this section, the user is presented various statistics that are calculated as the plug-in receives data via DIS:

- Current Entity States lists the number of Entity State PDUs the plug-in is currently maintaining.
- Peak Entity States displays the peak count of Entity State PDUs the plug-in has managed at any point during an execution.
- Total Entity States displays a running count of the total number of Entity State PDUs seen during an execution.
- Total Fire and Detonations displays a running count of the total number of Weapon Fires and Detonation PDUs seen during an execution.
- Total Electronic Emissions displays a running count of the total number of Electronic Emission PDUs seen during an execution.
- Total Zombies displays a running count of the total number of zombie Entity State PDUs that were removed, based on the Platform Timeout value, during an execution.

## 2.6 Application Pane

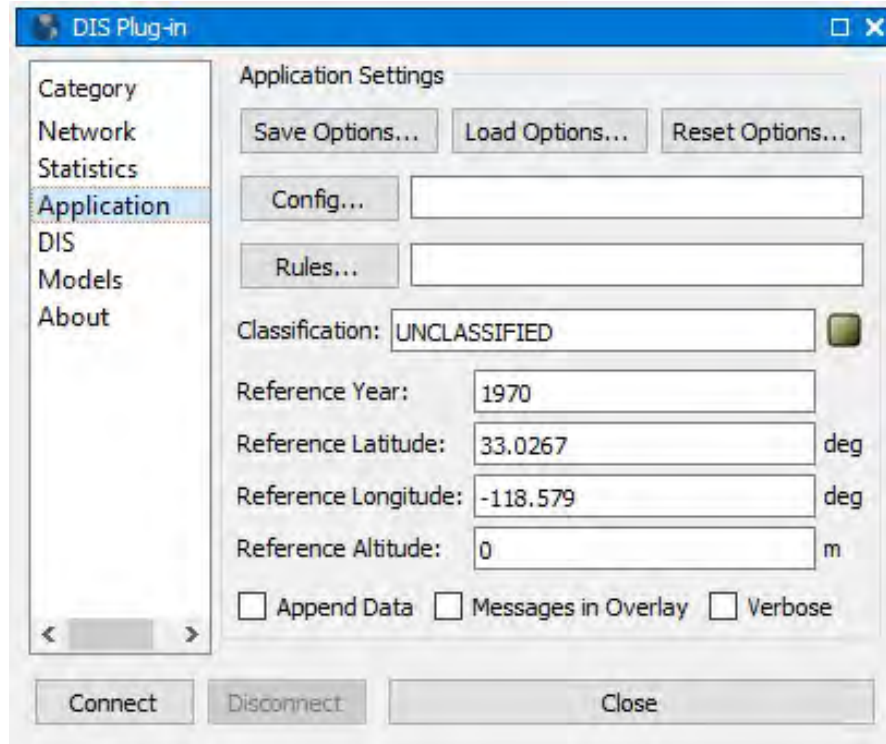


Figure 2.9: Application Pane

The next category pane for the DIS plug-in is the **Application** pane. In this section, the user can make changes to the:

- Save Options, once the plug-in is configured the user has the ability to save the settings. Upon exiting the plug-in automatically saves all the settings to the registry.
- Load Options allows the user to load preconfigured options from the registry
- Reset Options allows the user to reset all settings to a default “factory” configuration.
- Config... Allows the user to configure the plug-in via an input file. A sample configuration file, **sampleDIS.cfg** is included and the format is presented later.
- Rules... Allows the user to specify a SIMDIS preference rule file for application to new entities when data is read from the network. A sample SIMDIS rules file, **dis.rule** is included.
- Classification, allows the user to set the classification display in the SIMDIS overlay. The color well to the right of the text field allows a user to modify the classification color.
- Reference Year, allows the user to set the reference year for the SIMDIS scenario.

**NOTE:** This will only be settable when the "Append Data" option is unchecked.

- Reference Latitude, allows the user to set the default origin for the SIMDIS display. The input units are in degrees decimal.
- Reference Longitude, allows the user to set the default origin for the SIMDIS display. The input units are in degrees decimal.
- Reference Altitude, allows the user to set the default origin for the SIMDIS display. The input units are in meters.
- Append Data. When selected, the plug-in will attempt to append data to an existing live scenario. In order for append to work, the current scenario must be in a live mode state. The DIS plug-in will operate in the reference year set by the existing scenario, ignoring whatever is set in the Reference Year field.
- Messages in Overlay, displays weapon status messages as text in the screen overlay.
- Verbose. When selected verbose output status messages will be sent to the console window.

**NOTE:** On Windows the use of this option may affect performance due to the console messages.

## 2.7 DIS Pane

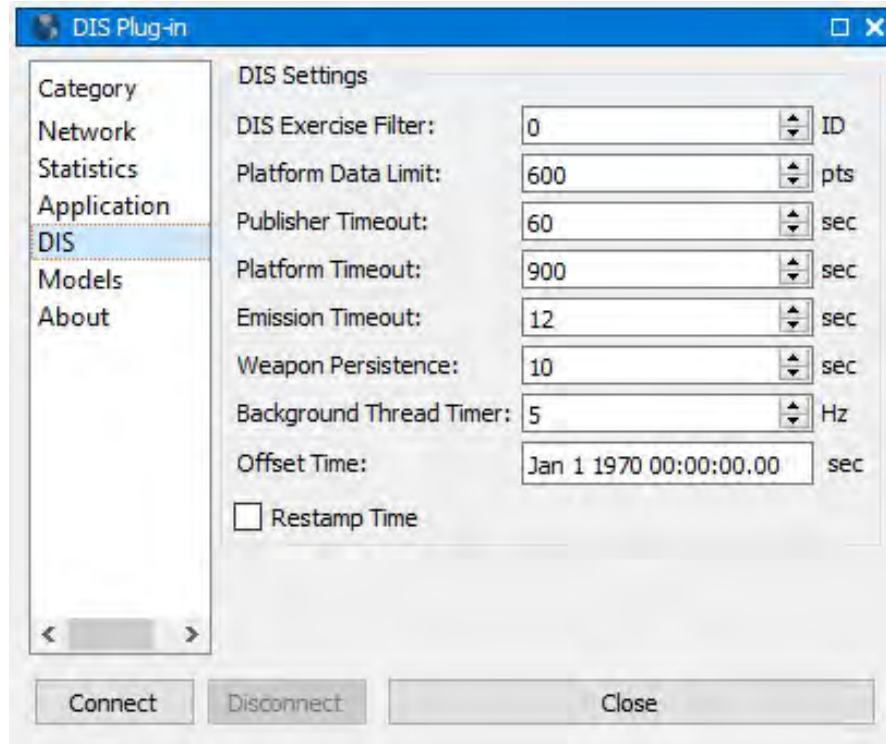


Figure 2.10: DIS Pane

The next category pane for the DIS plug-in is the **DIS** pane. In this section, the user can make changes to the:

- DIS Exercise Filter provides a mechanism to filter data processing and parsing based on the DIS Exercise ID. A filter value of 0 will allow all exercise IDs to be processed. A specific ID ranging from 1 to 255 will only process data from that exercise.
- Platform Data Limit controls the amount of data points a platform is logged in SIMDIS. Once a platform reaches the data limit older points are removed. The supported range is 5 to 1,000,000 points.
- Publisher Timeout value, in seconds, is used to notify SIMDIS if the plug-in fails to receive any DIS PDUs within the specified timeout value.
- Platform Timeout value, in seconds, is used to remove zombie tracks if the plug-in has not received an update in the specified interval. The supported range is 5 to 1,000,000 seconds.
- Emission Timeout value, in seconds, is used to turn off emission visualizations if the plug-in has not received an update in the specified interval. The supported range is 5 to 1,000,000 seconds.

- Weapon Persistence value, in seconds, is the lifetime of a weapon fire or detonation before it is removed from the display. The supported range is 5 to 1,000,000 seconds.
- Background Thread Timer value, in Hz, is the rate at which the background DIS thread is polled for data. The supported range is 1 to 30 Hz. During the polling, data structures from the DIS thread are synchronized with equivalent data structures in the application's main thread. All data received by the DIS thread is cached until thread synchronization is complete; once complete, the structures are cleared and the thread resumes processing.
- Offset Time, attempts to offset an incoming DIS relative timestamp to an absolute time stamp.

**NOTE:** This use of relative timestamps is not recommended and this use of this feature may produce unexpected results.

- Restamp Time. Instead of relying on the data time set from various sources, the user has the option to restamp incoming data using time of arrival. This feature is useful when playing back file data or receiving data that is real-time and reference to UTC where time of reception is nearly the same. When this option is selected, incoming data is restamped using the host machine's system clock. If this option is not selected, then the PDU time stamp is expected to contain the number of time units since the start of an hour if an absolute time marker is received. Relative time or non-real time modes that required input from additional simulation management PDUs are not fully-supported and may produce unexpected results.

## 2.8 Models Pane

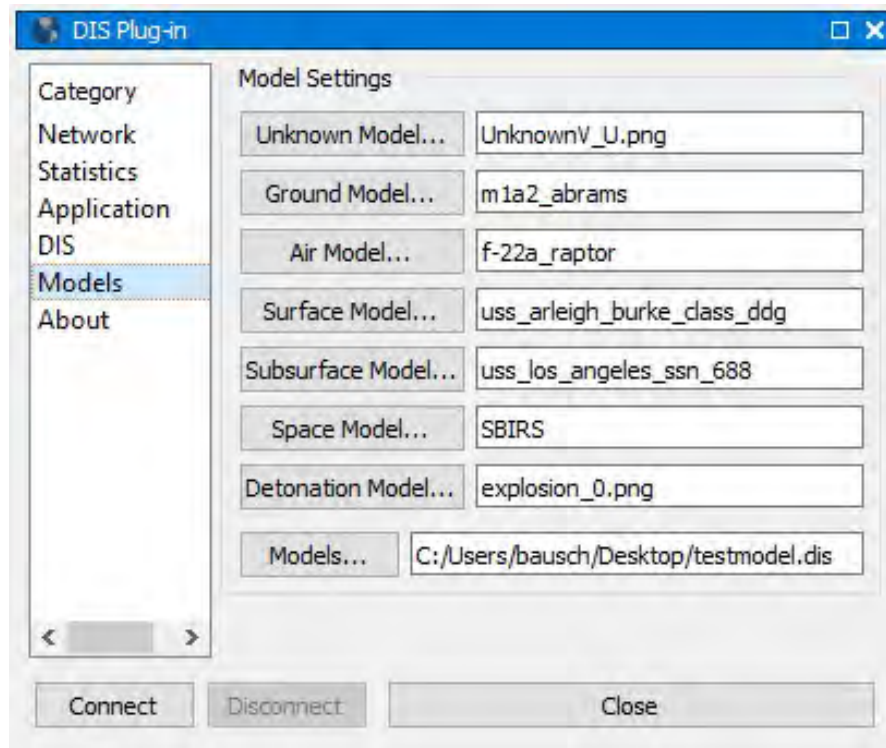


Figure 2.11: Models Pane

The next category pane for the DIS plug-in is the “Models” pane. In this section, the user can configure the type of SIMDIS model to associate to a particular platform type based on the DIS domain. Selecting one of the “Model...” buttons will launch a file selection dialog to allow the user to select a specific SIMDIS model. Supported DIS domains are:

- Unknown Domain
- Ground Domain
- Air Domain
- Air Domain
- Surface Domain
- Subsurface Domain
- Space Domain

In addition to supporting default SIMDIS models based on the DIS domains, a default SIMDIS model can be specified for the Detonation engagement message type.

- **Models** Allows the user to map models, native to SIMDIS, to the DIS Entity Type record. This gives the end user the flexibility to override the default Platform model settings. A sample configuration file, model.dis is included and the format is presented later.



## 2.9 About Pane

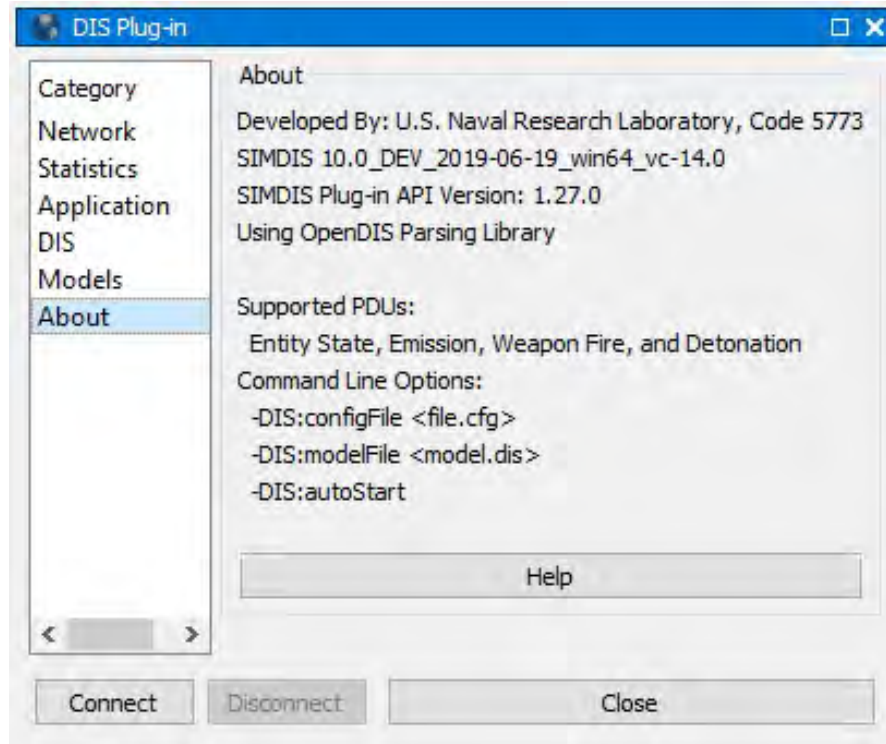


Figure 2.12: About Pane

The final category pane for the DIS plug-in is the **About** pane. This pane provides information regarding compatibility with the SIMDIS executable and the SIMDIS Plug-in API, supported PDUs and optional command line arguments for starting the plug-in. Finally, pressing the **Help** button will launch a PDF viewer to display this document.

The command line arguments can be used to pre-configure the plug-in via a start script. On Windows a start script would be a .bat file, on Linux the start script would be a Launcher. An example start script may contain the following:

```
cd c:/data/config/DIS
simdis -DIS:configFile sampleDIS.cfg -DIS:modelFile model.dis -DIS:autoStart
```

In this example the loaded files would be relative to the starting directory of the script.

This Page is Intentionally Left Blank

# Chapter 3

## Operation

### 3.1 Processing DIS Data

When the DIS Plug-in is loaded and connected to a DIS broadcast, the clock in the lower left and corner of SIMDIS will increment, and the SIMDIS display should appear centered about the reference latitude and longitude as set in the Plug-in dialog's Application Pane. If no data is being published then the display will be centered at 0 latitude and 0 longitude and the time display will not increment.

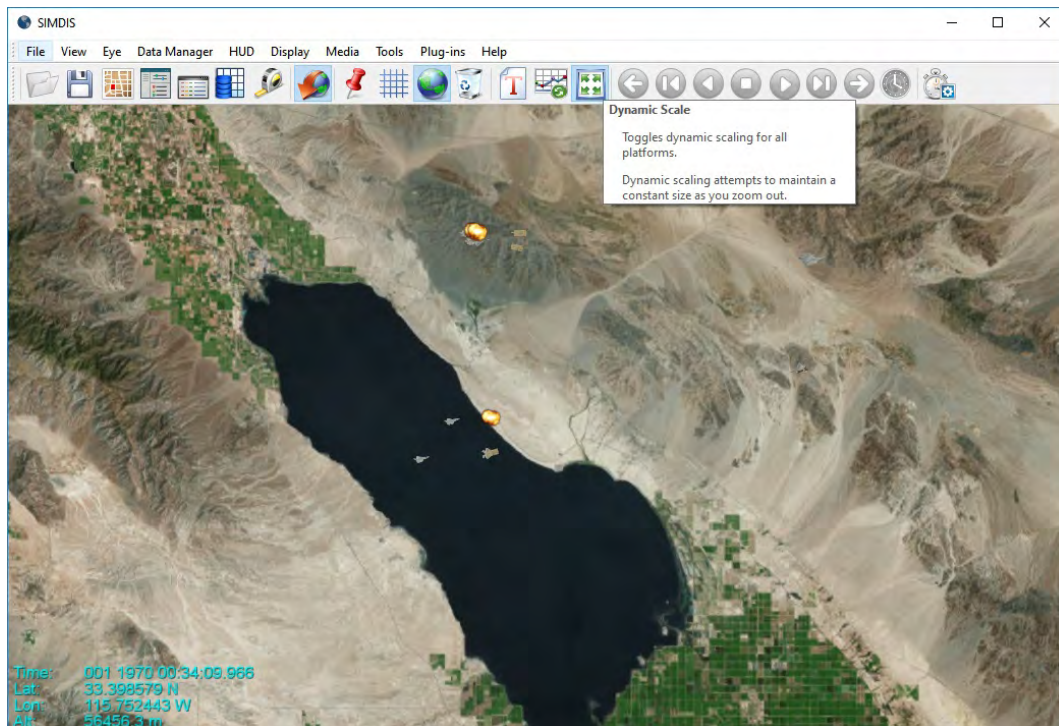


Figure 3.1: SIMDIS processing DIS Data

In the above view, the SIMDIS display is zoomed to the reference origin in addition to dynamic scaling of platforms was enabled (toggle D key). Helpful pop-up text boxes appear as you mouse over various toolbar buttons, in the above figure the mouse is over the dynamic scale toolbar button. Additional information and associated hot keys can be found inside the status bar along the bottom of SIMDIS.

Basic navigation tips for SIMDIS can be found in the Quick Start guide, found either from the Start Menu (on Windows) or via the Help menu pull down on the right side of the tool bar.

Once connected to a DIS feed (when not in Append Mode) the SIMDIS clock will be slaved to the first valid data point that was processed. The SIMDIS clock will then freewheel using the host computer's system clock offset by the time difference of the first valid data point that was received. Timing issues can arise when the first point that was received is not within range of other incoming data. Typically this occurs when publishers are not time synchronized or simulation data time is different from live data.

As previously mentioned, care must be taken when using SIMDIS as a stand alone logger. SIMDIS does not possess the capability to page data off to disk as it records data in real-time. Since all received data is stored in RAM, SIMDIS utilizes both data limiting and platform expiration policies for memory management. The data limiting settings were covered in the discussion of the DIS pane. SIMDIS has two methods for platform expiration, immediate and queued. The default behavior of the plug-in is to have SIMDIS perform an immediate deletion of platforms when a destruction callback is received. Users have the capability to override this setting via the "Platform Expiration..." menu option found in the Data Manager menu. If a user wishes to keep all received data during an execution, then a queued expiration policy in conjunction with a large data limit value must be used. Once again care must be taken not to exhaust the physical resources of the host computer, otherwise serious performance problems will arise once the computer begins to page in virtual memory.

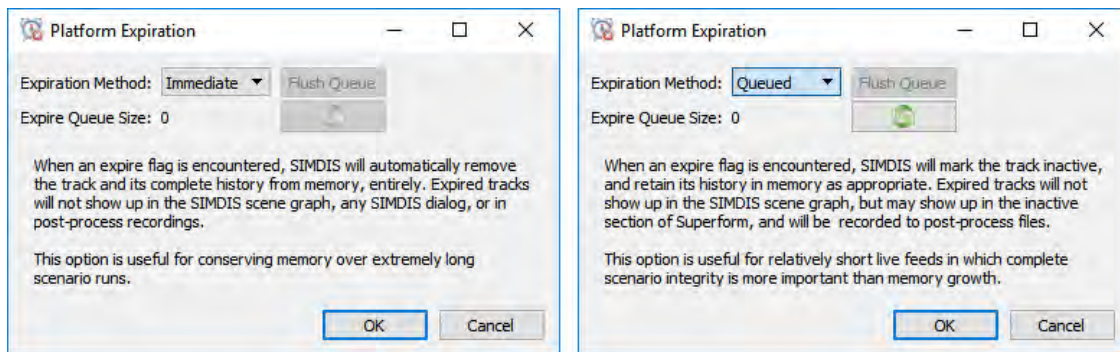


Figure 3.2: Platform Expiration Dialog

The DIS plug-in performs basic data validity checking. If the plug-in receives a data point that contains at a minimum position and time, then the information is forwarded to SIMDIS. If the plug-in detects an exception with the position conversion, the data is marked as invalid and not sent to SIMDIS. With the exception of geodetic position, position ranges are not checked for valid ranges. The plug-in will clamp latitude between the range of -90 and +90 degrees and longitude between -180 and +180 degrees.

The DIS plug-in also processes and displays other non-TSPI related information. This information includes:

- Affinity via Affiliation
- Platform Type via raw DIS Entity Domain or CSV File
- Context via DIS Entity Kind
- Detonation Type
- Warhead Type
- Damage State and Damage Percent
- DIS Entity State Record
- Platform IDs, Unique ID, application ID, site ID and entity ID
- Shooter
- Target
- Weapon Fire via an animated line drawn from Shooter to Target

Red animated lines are from Hostile platforms

Blue animated lines are from Friendly platforms

White animated lines are from remaining affiliations

- Detonation via an explosion icon at the impact location

Many important attributes from received PDUs are stored in SIMDIS via Category data. For entities, these include DIS Force, Entity Domain, Entity Kind, Country and Damage. For Fire and Detonation PDUs, included are Detonation Type and Warhead Type information. For Electronic Emission PDUs, the Emitter Function is stored. In all cases, the DIS Site, Application, Entity and Exercise IDs are stored. This storage mechanism allows for preference rule settings and data filters to be applied to the incoming platforms. Figure 3.3 depicts this data in the SIMDIS Data Browser and its Data Filter.

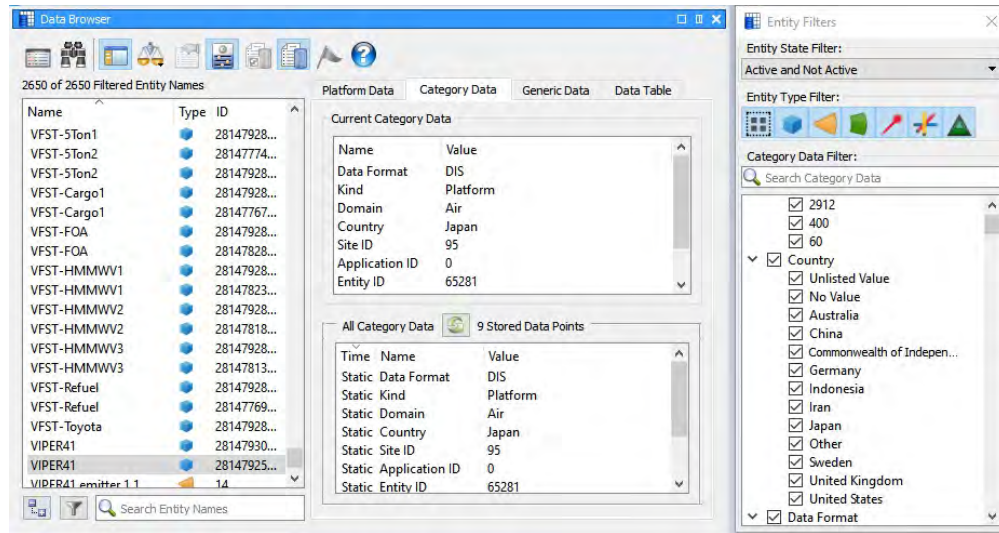


Figure 3.3: Data Browser Category Data Tab and Entity Filters



DIS Entity Record and ID information are stored in SIMDIS via Generic Data. The images below depict how Generic Data can be displayed in SIMDIS. In Figure 3.4, the display of Generic Data is enabled via the **Generic** attribute checkbox in the SIMDIS Preference Tool's Label tab.

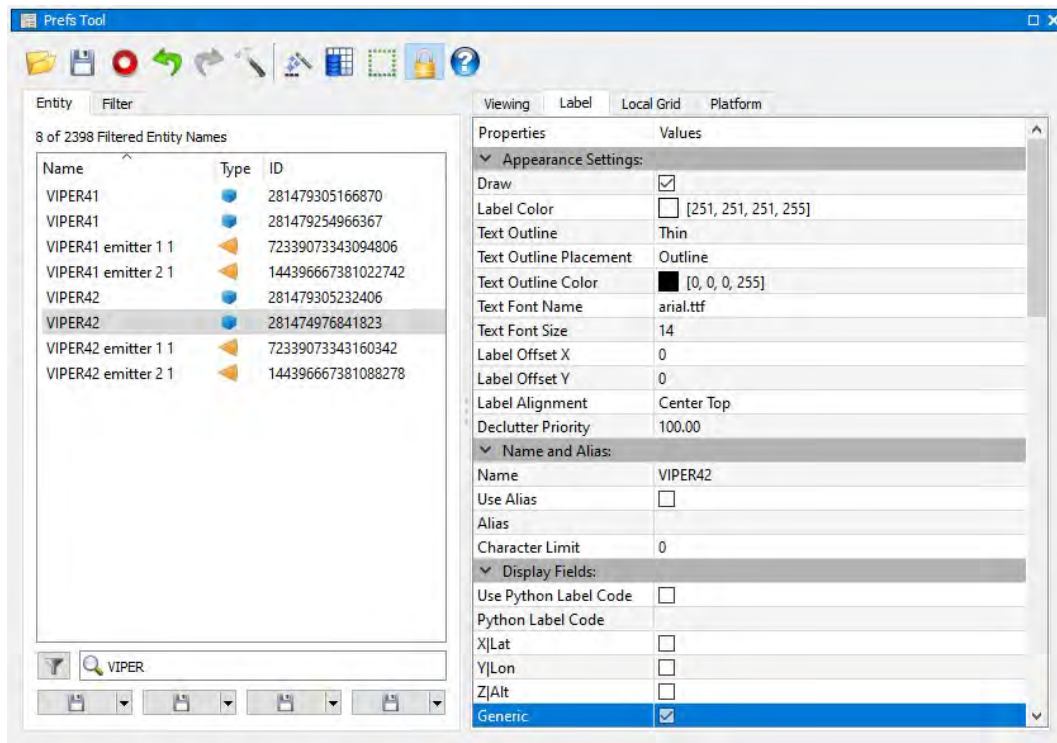


Figure 3.4: Platform Label Preferences

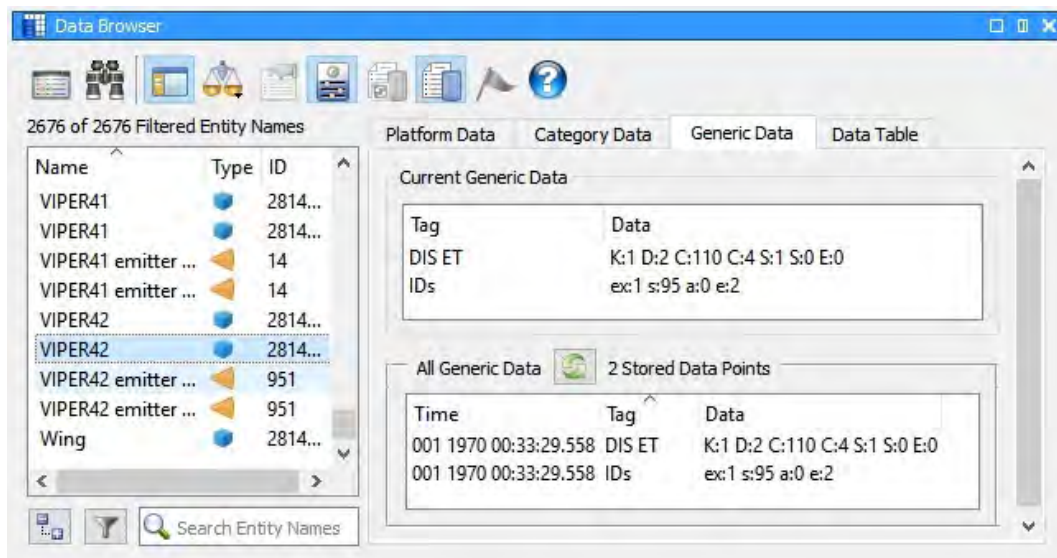


Figure 3.5: Data Browser Generic Data Tab

### 3.1. PROCESSING DIS DATA

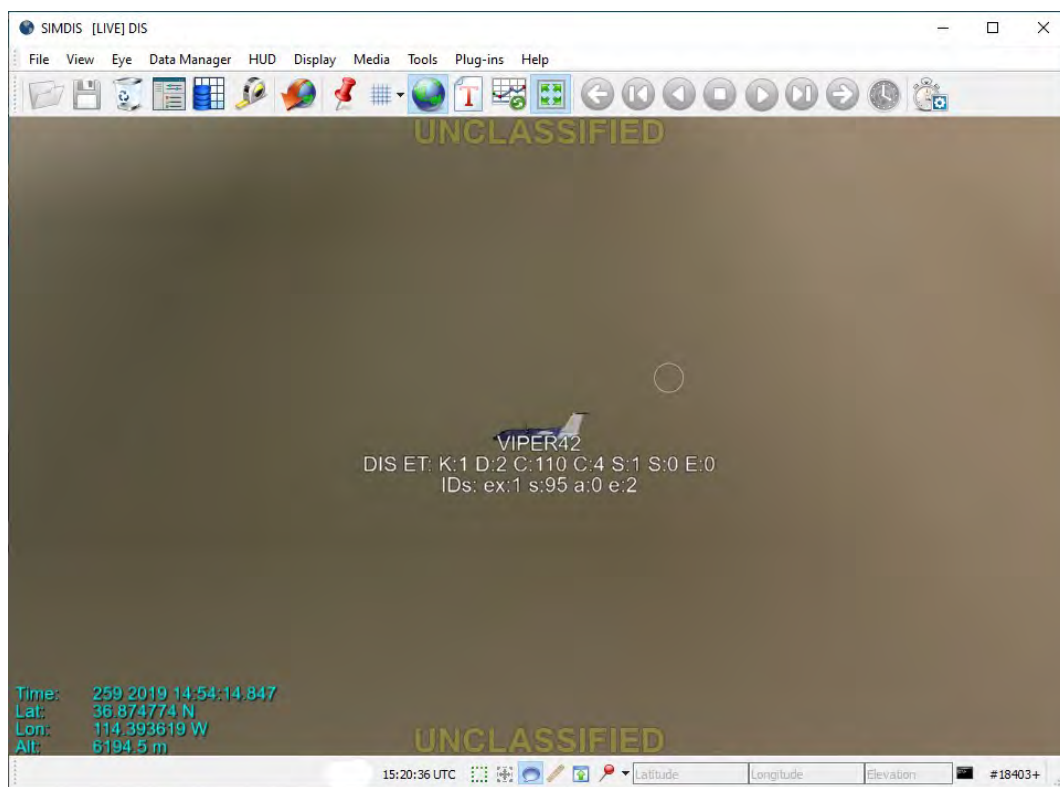


Figure 3.6: Platform Label Data



### 3.1. PROCESSING DIS DATA

When the Message Overlay check box is enabled, information pertaining to weapon messages will be displayed in the upper left screen overlay. Platform call signs of the Shooter and the Target are used in the weapon message overlay text. Figure 3.7 depicts the display of the overlay text. Also shown are animated lines representing weapon fire pairings between Shooter and Target, and resulting weapon detonations. Emission PDUs are also shown as semi-transparent beams that are color-coded based on Emitter Function.

Additional information regarding the configuration and operation of the Message Overlay window can be found in the SIMDIS User Manual.

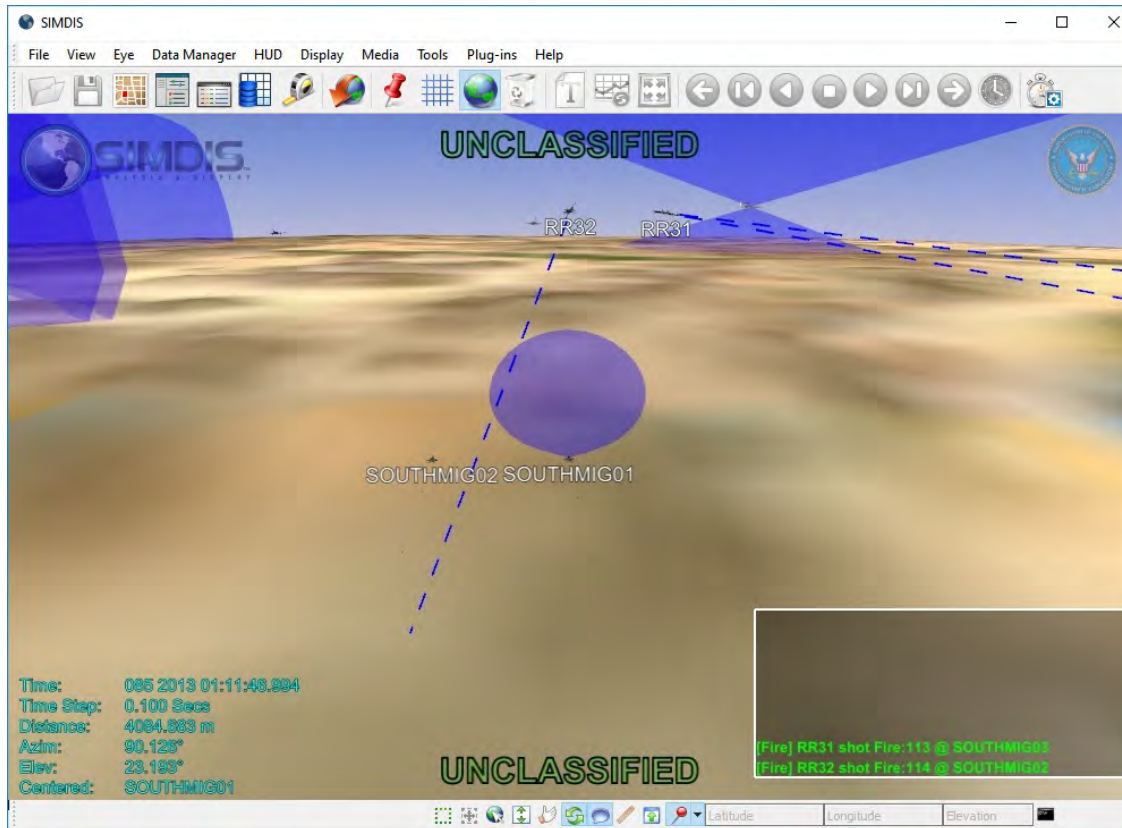


Figure 3.7: Weapon Fire Data

This Page is Intentionally Left Blank

# Appendix A

## DIS Configuration File

### A.1 Configuration File Format

The DIS plug-in also has the ability to read in a configuration file that pre-configures the plug-in. The plug-in configuration file format is presented below:

```
# -*- Mode: sh -*-
#####
#
# SIMDIS DIS Plug-in Configuration File
#
# Lines with # are treated as comments
# thus are ignored by the parser
#

#####
#
# SIMDIS reference latitude origin
# units: degrees decimal
# values south of the equator are negative
#
DIS_ReferenceLatitude 33.026669

#####
#
# SIMDIS reference longitude origin
# units: degrees decimal
# values west of the prime meridian are negative
#
DIS_ReferenceLongitude -118.578636
```

## A.1. CONFIGURATION FILE FORMAT

```
#####  
#  
# SIMDIS reference altitude origin  
# units: meters  
#  
DIS_ReferenceAltitude 0  
  
#####  
#  
# DIS protocol IP  
# units: IP address  
#  
DIS_Ip 127.0.0.1  
  
#####  
#  
# DIS protocol port number  
# units: unsigned integer  
#  
DIS_Port 3000  
  
#####  
#  
# The time to live value for data transmissions  
# Indicates the scope or range in which a packet  
# may be forwarded  
# units: unsigned integer  
# 0 is restricted to the same host  
# 1 is restricted to the same subnet  
# 32 is restricted to the same site  
# 64 is restricted to the same region  
# 128 is restricted to the same continent  
# 255 is unrestricted  
DIS_TTL 1  
  
#####  
#  
# Multicast interface address  
# String specifying the network interface to use  
# for network transmission  
# units: IP address  
# Empty string defaults to any address  
#  
DIS_IFace
```

## A.1. CONFIGURATION FILE FORMAT

```
#####  
#  
# Exercise ID Filter  
# ID of an exercise to include (excluding all others)  
# 0: no filtering  
#  
DIS_ExerciseIdFilter 0  
  
#####  
#  
# Background DIS thread polling frequency  
# Rate at which to poll processed DIS PDUs  
# units: Hz (1-100)  
#  
DIS_ThreadTimer 5  
  
#####  
#  
# Platform Timeout Value  
# Used to remove Zombie Entity State PDUs  
# units: seconds  
#  
DIS_PlatformTimeout 120  
  
#####  
#  
# Emission Timeout Value  
# Used to expire Electromagnetic Emission PDUs  
# units: seconds  
#  
DIS_EmissionTimeout 20  
  
#####  
#  
# Platform Data Limit Value  
# Used to manage SIMDIS memory space  
# units: # track points to keep  
#  
DIS_PlatformDataLimit 600
```

## A.1. CONFIGURATION FILE FORMAT

```
#####
#
# Debug Output
# Controls debug output to console
# boolean:  TRUE or FALSE
#
DIS_DebugOutput FALSE

#####
#
# Append Scenario
# Controls appending DIS data to a live scenario
# boolean:  TRUE or FALSE
#
DIS_AppendScenario FALSE

#####
#
# Publisher Timeout Value
# Indicates a loss of contact with DIS publishers if
# data has not been received within specified interval
# units:  seconds
#
DIS_PublisherTimeout 60

#####
#
# classification color
# Plugin colors are stored as 32 bits: 0xRRGGBBAA
#
# RR : red component,  00 - FF (0-255) in hex
# GG : green component, 00 - FF (0-255) in hex
# BB : blue component, 00 - FF (0-255) in hex
# AA : alpha component, 00 - FF (0-255) in hex
#
# WHITE = 0xFFFFFFFF
# GREEN = 0X00FF00FF
# BLUE  = 0x0000FFFF
# RED   = 0xFF0000FF
# BLACK = 0x000000FF
# GRAY  = 0x80808050 (semi-transparent)
#
DIS_ClassificationColor 0X00FF0050
```

## A.1. CONFIGURATION FILE FORMAT

```
#####
#
# classification string
#
DIS_Classification UNCLASSIFIED

#####
#
# useTimeOfArrival
# boolean:  TRUE or FALSE
# Incoming data times are restamped based on host
# computer's system clock at the moment they are
# read from the network
DIS_UseTimeOfArrival FALSE

#####
#
# Default Model
# Default SIMDIS model to use if one is not specified
#
DIS_DefaultModel UnknownV_U.png

#####
#
# Default Ground Model
# Default SIMDIS model to use for DIS Ground Domain
#
DIS_DefaultGroundModel m1a2_abrams

#####
#
# Default Air Model
# Default SIMDIS model to use for DIS Air Domain
#
DIS_DefaultAirModel f-22a_raptor

#####
#
# Default Surface Model
# Default SIMDIS model to use for DIS Surface Domain
#
DIS_DefaultSurfaceModel uss_arleigh_burke_class_ddg
```

## A.1. CONFIGURATION FILE FORMAT

```
#####  
#  
# Default Subsurface Model  
# Default SIMDIS model to use for DIS Subsurface Domain  
#  
DIS_DefaultSubsurfaceModel uss_los_angeles_ssn_688  
  
#####  
#  
# Default Space Model  
# Default SIMDIS model to use for DIS Space Domain  
#  
DIS_DefaultSpaceModel SIBRS  
  
#####  
#  
# Default Detonation Model  
# Default SIMDIS model to use for Detonation Messages  
#  
DIS_DefaultDetonationModel explosion_0.png  
  
#####  
#  
# Lifetime of weapon fire and detonation PDUs before  
# they are removed from SIMDIS (sec)  
#  
DIS_WeaponPersistence 15  
  
#####  
#  
# Message Overlay Output  
# Controls display of fire and detonation messages  
# in the SIMDIS overlay  
# boolean: TRUE or FALSE  
#  
DIS_MessageOverlay FALSE  
  
#####  
#  
# Animation speed (Hz) of weapon fire line  
#  
DIS_AnimatedLineSpeed 10
```



```
#####
#
# Line width (pixels) of weapon fire line
#
DIS_AnimatedLineWidth 2

#####
#
# Model file to load; must be a fully specified path
#
DIS_ModelFile C:\Program Files\SIMDIS\config\DIS\model.dis

#####
#
# Rule file to load; must be a fully specified path
#
RuleFile C:\Program Files\SIMDIS\config\DIS\dis.rul
```

## A.2 Entity Type Model Mapping File Format

The DIS plug-in also has the ability to read in a configuration file that will allow a user to pre-configure SIMDIS models based on the DIS entity type record. This file is loaded via the **Models** button in the Application Settings pane. The model mapping file format is presented below, additionally a sample file (model.dis) is also distributed with the plug-in. The DIS plug-in supports a flexible matching scheme: if an exact match to the full K.D.C.C.S.S.E is not found, the plug-in will iteratively attempt to find a match to K.D.C.C.S.S.0. then, K.D.C.C.S.0.0 and so on until a match is found.

```
#####
#
# 3-D model assignment is handled via the
# the DIS entity type record
#
# Example with DIS entity type record
#
# DIS 1.2.225.1.6.1.0 f-22a_raptor.opt
#
# NOTE: There are no spaces or wild cards allowed
#       in the DIS entity type record. The order
#       is based on the following:
#       K = kind
#       D = domain
#       C = country
#       C = category
```

## A.2. ENTITY TYPE MODEL MAPPING FILE FORMAT

```
#          S = subcategory
#          S = specific
#          E = extra
#
#####
DIS 1.2.45.1.3.0.0 f-7m_chengdu.opt
DIS 1.1.101.4.15.0.0 sa-10_tel_launcher.opt
DIS 1.1.102.4.13.2.0 c-802_launcher.opt
DIS 1.1.165.4.71.0.0 sa-10_tel_launcher.opt
DIS 1.1.205.28.1.1.0 ram_launcher.opt
DIS 1.1.222.1.2.2.0 t72_tank_des.opt
DIS 1.1.222.2.7.5.0 acrv_1v12_des.opt
DIS 1.1.222.2.13.0.0 btr-70_apc_des.opt
DIS 1.1.222.2.2.0.0 brdm-2_at-5_spandrel_des.opt
DIS 1.1.222.28.1.1.0 sa-2_launcher.opt
DIS 1.1.222.28.1.2.0 sa-2_launcher.opt
DIS 1.1.222.28.1.3.0 sa-2_launcher.opt
.
.
.
```