SIMDIS
ANALYSIS & DISPLAY

| Version 10.0 SR7

User Manual

March 2020

U.S. Naval Research Laboratory
Electronic Warfare Modeling and Simulation Branch
Code 5770

This Page is Intentionally Left Blank

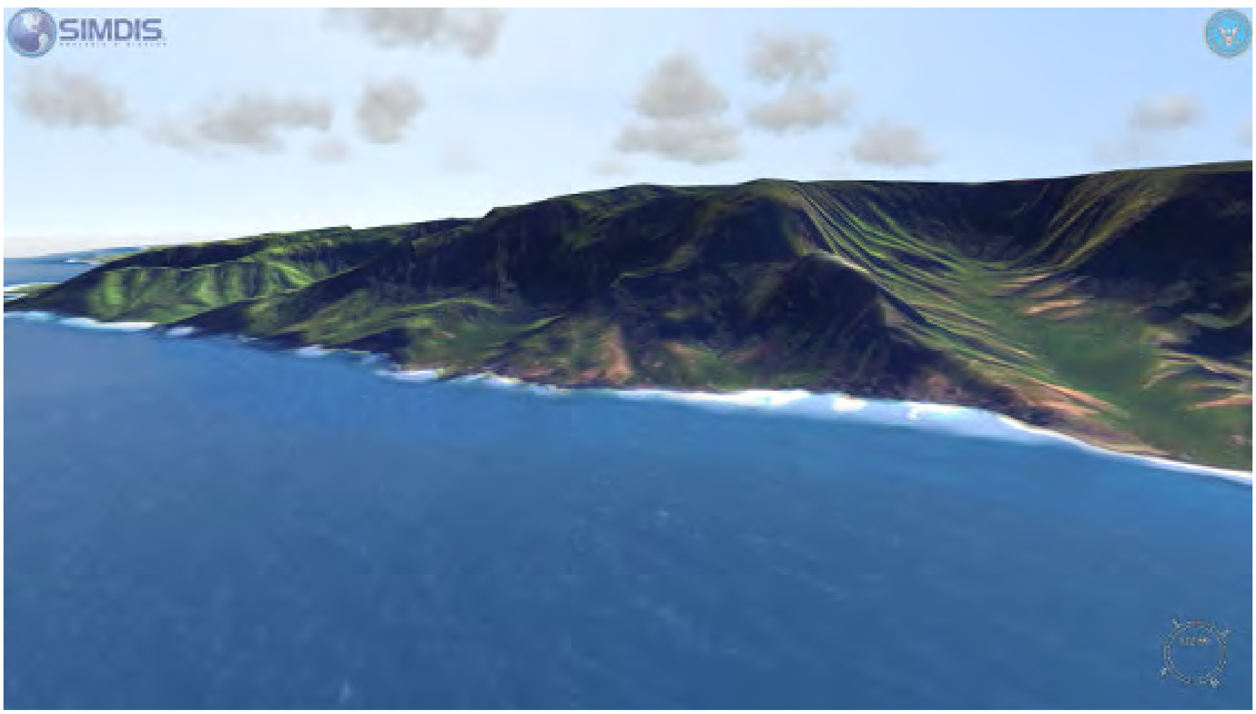
Table of Contents

List of Figures

Chapter 1

Introduction

1.1 What is SIMDIS?



The Naval Research Laboratory (NRL) Code 5770 has prototyped, produced, and delivered the SIMDIS 3D Visualization and Analysis Tool to the Department of Defense (DOD) community.

Offering an affordable analysis and visualization capability, SIMDIS provides support for high-fidelity analysis, and display of test and training mission data to a growing user base of over 23,000 users. This highly specialized visualization tool provides a unique capability for two and three-dimensional interactive data display and analysis. The SIMDIS toolset is designed for Windows and Linux platforms using hardware accelerated 3D graphics and that requires no additional Commercial-of-the-Shelf (COTS) products or license fees. SIMDIS provides identical execution, and **look and feel** for all supported platforms.

SIMDIS is in use at the NRL and other DOD sites. It has been used for numerous simulation and test applications. SIMDIS was used to assist Commander Operational Test and Evaluation Force (COMOPTEVFOR) in reconstructing data from the August 1998 Nulka Decoy operational test (OT). In this application, SIMDIS proved to be an invaluable asset in analyzing the vast test data in a common coordinate frame of reference. SIMDIS allowed the analysts to view the data faster and with greater fidelity than was previously possible.

SIMDIS is currently a key operational display system for:

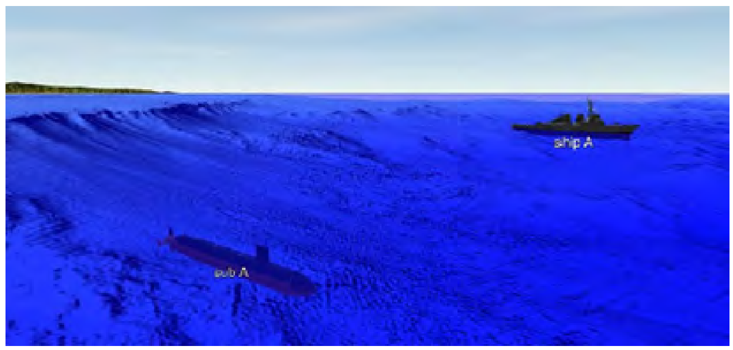
- Canadian Forces Maritime Warfare Centre (CFMWC)
- Fleet Area Control and Surveillance Facility, Virginia Capes (FFVC)
- Missile Defense Agency (MDA)
- Naval Undersea Warfare Center (NUWC)
- Pacific Missile Range Facility (PMRF)
- Southern California Offshore Range (SCORE)
- White Sands Missile Range (WSMR)
- Yuma Proving Ground (YPG)
- Test Resource Management Center (TRMC)

In addition to the DOD, SIMDIS has gained acceptance in other US and foreign organizations.

SIMDIS provides a live situational awareness and post-processed 3D analysis display capability on Ballistic Missile tests for both MDA and PMRF. In addition to Ballistic Missile tests, SIMDIS has been used for such diverse applications as the Rim of the Pacific (RIMPAC) Exercises, Developmental and Operational Testing (DT/OT) events, Fleet Synthetic Training (FST), After Action Reporting (AAR), vehicle performance assessment, live surveillance for range safety, situational awareness for large distributed test events as well as visualization of Modeling and Simulation (M&S) output.

SIMDIS is used for exercise control and as a post-mission debrief tool at SCORE and FFVC. During exercise control, real-time tracks are viewed within SIMDIS via a live data stream from the Range Operations Center. This allows various Navy/Marine command and control elements to monitor the progress of an exercise. Post-mission, SIMDIS is used to record and then playback an exercise for a participating unit.

Primary exercises to date have been the Composite Training Unit, Joint Task Force, Integrated Live Fire, Fast Attack Craft, Strike, and Anti-Submarine Warfare exercises. With the high-fidelity 3D, Helo squadrons can observe a torpedo transit through the water column and review their attack on a target. Strike groups review an air engagement in detail from any viewpoint.



SIMDIS was integrated directly as a subscriber utilizing various common object models for TENA. This combination of a common display tool and a common object model has provided users with a very powerful and cost-effective means to visualize and analyze both live and post processed range data. TENA enabled versions of SIMDIS have been used to support several distributed test exercises sponsored by the Joint Mission Environment Test Capability (JMETC).

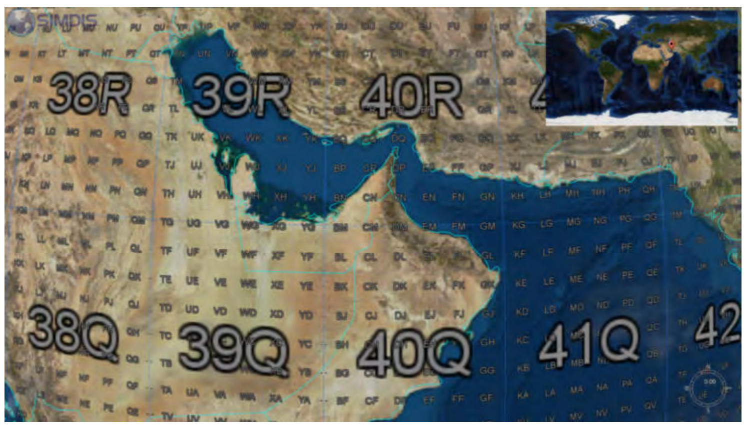
Current SIMDIS capabilities provide a needed and affordable analysis and visualization toolset to a growing user base. The current version provides abundant fully-tested and debugged capabilities. SIMDIS runs on all of the supported platforms without requiring additional COTS products or license fees. Furthermore, changes to the tool are freely shared among the community versus an additional toolkit charge or maintenance fee that is typically applied when using similar COTS products.

1.2 Why is SIMDIS Useful?

Leveraging the technology advances in graphics acceleration hardware, SIMDIS provides a powerful capability for interactively visualizing and analyzing simulation, and field test data from any viewpoint; i.e. from different platforms/sites or specified location.

SIMDIS provides a 3D display of the normally **seen** data, such as platform position and orientation, as well as the **unseen** data, such as the interactions of sensor systems with targets, countermeasures, and the environment. SIMDIS also provides tools for interactively analyzing data using custom tools for displaying equipment modes, spatial grids, ranges, angles, and antenna patterns. SIMDIS provides the capability to view time-synchronized 2D and 3D data on a single standalone workstation or across multiple networked platforms.

SIMDIS facilitates various modes of operation including live display, interactive playbacks, and a scripted multimedia mode, allowing it to apply to the needs of users including range operators, simulation users, analysts, and presenters. It is interactive in the sense that you can modify and interact with the presentation while it is running.



SIMDIS also provides bookmarking capabilities for creating scripted scenarios to aid in post-event analysis or debriefs. Bookmarks are timestamped markers used to manipulate and control the display in order to highlight events of interest during a playback. This capability allows users to create scripted scenarios that are saved with the data. Since the bookmarks are saved with the data, it makes them an attractive option for interactive debriefing and data product generation.

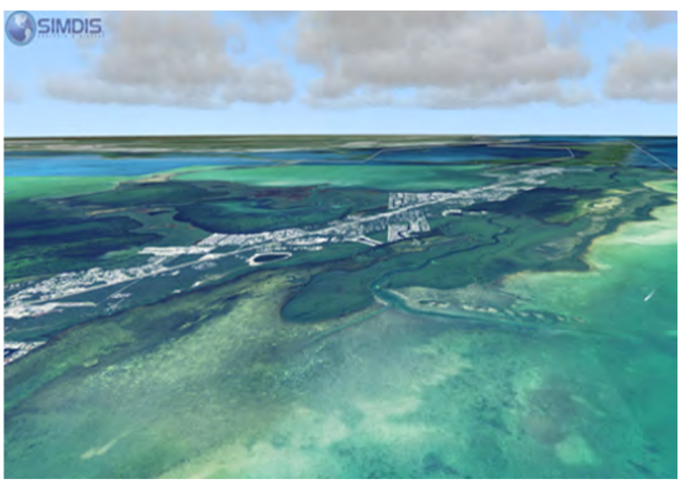
A key requirement for SIMDIS is to support the identical **look and feel** and execution across all supported platforms. SIMDIS is an object-oriented program written in ANSI standard C++ and OpenGL. SIMDIS uses a cross-platform graphical user interface to enable the identical **look and feel**, and execution across all supported platforms.

SIMDIS has been tested on the following operating system configurations:

- 64 bit Red Hat Enterprise Linux (RHEL) 6 and 7
- 64 bit Ubuntu 14.04 and 16.04 Long Term Support (LTS)
- 64 bit Windows 7, Windows 10, and Windows 10 DOD Secure Host Baseline (SHB)

Under the Consolidated Afloat Network and Enterprise Services (CANES), SIMDIS was successfully tested with: Host-Based Security System (HBSS) ePO version 5.3.1.

As a note, these are the system configurations that the SIMDIS team is equipped to compile and test. SIMDIS may work on other configurations, but they are not officially supported.

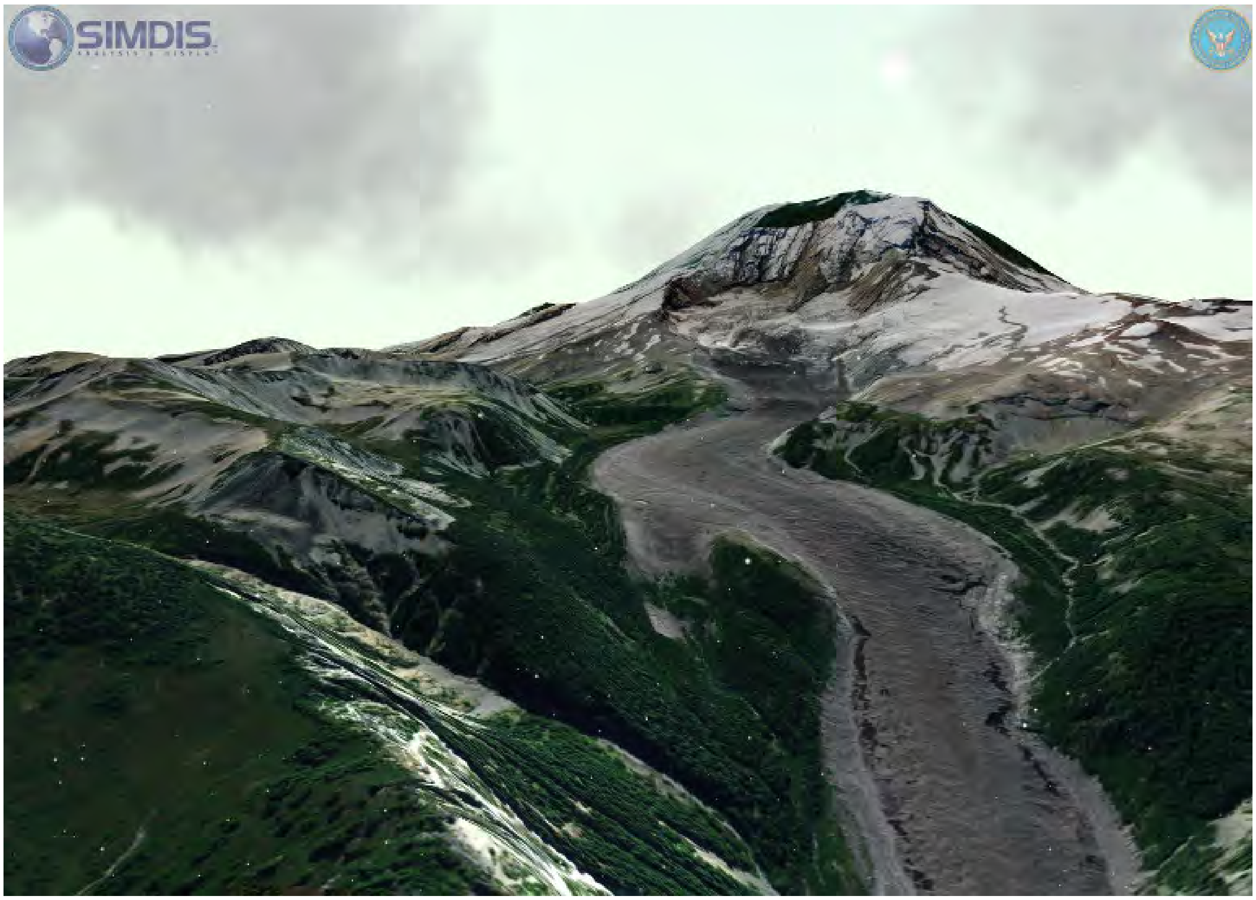


SIMDIS provides numerous methods for importing data including several filters that have been developed for specific range and simulation data formats. Additional free software libraries, the Data Client Server (DCS) library or the SIMDIS Plug-in API (Application Programming Interface), are also available on the SIMDIS web site for importing data. These ANSI standard C++ libraries allow users to design, develop and maintain their own software for importing their specific data formats into SIMDIS.

In addition to various forms of data entry, a large library of 3D models has been developed that includes many U.S. and foreign platforms including air, land, missile, satellite, ship, and undersea models. For geographic display, SIMDIS supports geocentric, geodetic, and topographic coordinate systems and allows display of map data including digital elevation data, shoreline data, ESRI Shape files, KML, and the generalized overlay graphics (GOG) format used at many Test and Training ranges.

The main file format of SIMDIS is the ASCII Scenario Input (.asi) file format. This format allows data to be both imported and exported. Using the export feature, platform and sensor data stored in SIMDIS memory space can be readily exported for quick 2D plotting and analysis.

SIMDIS also has a binary file format (.hdf5) that can be both imported and exported. The binary format is based on the Hierarchical Data Format (HDF) and is designed for flexible and efficient I/O and for high volume and complex data. However, as new versions of SIMDIS are developed, the ASCII format will be the only maintained upgrade path for prior SIMDIS data files to be read-in by newer versions of SIMDIS.



1.3 About This Document

This document is the user manual for the SIMDIS Toolset. It is intended as a general overview for those who want to familiarize themselves with the tool. The document also serves as a how-to guide and a general reference guide for the supported files loaded by SIMDIS.

This version contains detailed tool tips that are meant as a guide as you explore and operate the application. The tool tips are written as the primary user guide, thus allowing the manual to focus on broader topics. Figure 1.1 is an example of the SIMDIS tool tips.

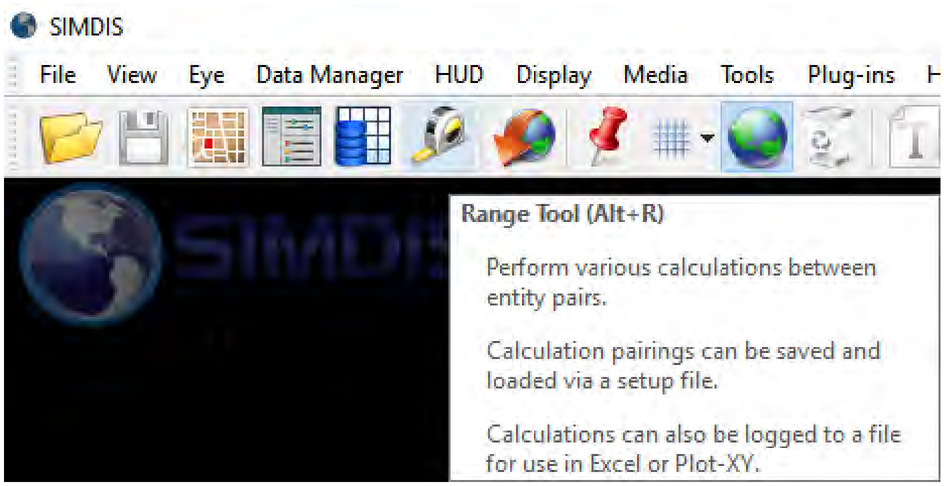


Figure 1.1: Tool Tips Example

Please direct all comments or questions you may have regarding this manual to the SIMDIS team via the SIMDIS Help Desk at https://simdis.nrl.navy.mil/jira.

Thank you and we hope that this document serves to clarify your questions and curiosity about the SIMDIS toolset.

The diagrams throughout the document come from a variety of platforms and versions of SIMDIS. Do not be surprised if they differ slightly from your version of SIMDIS, especially if you have an older version of the software.

1.3.1 Document Revision History

Revision 1.0	10.0 SIMDIS	Sep. 30, 2016
Revision 2.0	10.0 SIMDIS SR1	Mar. 30, 2017
Revision 3.0	10.0 SIMDIS SR2	Sep. 30, 2017
Revision 4.0	10.0 SIMDIS SR3	Mar. 30, 2018
Revision 5.0	10.0 SIMDIS SR4	Sep. 30, 2018
Revision 6.0	10.0 SIMDIS SR5	Mar. 30, 2019

This Page is Intentionally Left Blank

Chapter 2

Installation

SIMDIS is available for free download on the website at https://simdis.nrl.navy.mil. To download the SIMDIS software suite, register an account by going to the website and follow the Request Account link. This will take you to a page where you must fill out pertinent information. The main distribution does not require a U.S. Government point of contact. However, users looking to have access to certain tools (such as those used at NUWC, PMRF or SCORE) are required to have an official US Government contact and approval from one of the appropriate ranges.

To obtain a license for SIMDIS, select the Register System link when logged in to the SIMDIS website. You must be logged in with a valid account in order to register a system. Each system requires its own license key. For more information, browse our help desk, particularly https://simdis.nrl.navy.mil/jira/browse/FAQ-45.

2.1 System Requirements

SIMDIS supports two different platforms: **Linux** and **Windows**. The SIMDIS program supports multi-threading to take advantage of multiple core CPUs. The display frame rate should be high enough to support playback interaction from either a file or live data source. An acceptable frame rate is 10 Hz or 10 frames per second (fps). The frame rate can be shown via **HUD > Statistics > Frame Rate** from the menu bar or hot key Alt + F.

NOTES

- For Linux platforms, the SIMDIS license key is tied to the Ethernet Card's MAC (Media Access Control) address.
- Windows XP 32-bit is able to run SIMDIS using VC10 build with the same minimum specifications.
- The 32-bit version of SIMDIS has been installed and operates on 64-bit Windows OS.
- Some NVIDIA graphics card feature Threaded Optimization that can freeze SIMDIS. To avoid this from happening, turn off the **Threaded Optimization** in **NVIDIA Settings**.
- There is an OpenGL incompatibility with laptops running Windows 10 with **Switchable Graphics**. The solution is to disable the **Switchable Graphics** in the **BIOS**. For more information refer to FAQ-62.
- The Windows versions of SIMDIS have been installed and tested using virtualization software VirtualBox, VMWare Workstation Player, and VMWare Workstation Pro operating on either Windows or Linux. If you encounter issues running SIMDIS on a virtual machine refer to: SIMDIS-2552.

2.1.1 Minimum System Configuration

CPU	Dual-core processor 1.6 GHz, 2 MB L2 Cache, 667 MHz FSB
RAM	2 GB DDR-400
HDD	100 GB HDD Space
GPU	256 MB Texture Memory; OpenGL 2.0, GLSL 3.3 compliant
OS	Windows 7 or RHEL 6
Display	800x600 screen resolution
Peripherals	Mouse, Keyboard, Ethernet Card

2.1.2 Recommended System Configuration

CPU	Quad-core processor 3.0 GHz, 8 MB L2 Cache, 1066 MHz FSB
RAM	4 GB DDR2-800 SDRAM, 8 GB for 64-bit systems
HDD	256 GB Solid State Drive (SSD)
GPU	1 GB Texture Memory; OpenGL 3.0, GLSL 3.3 compliant
OS	Windows 10 or RHEL 6
Display	1920x1080 screen resolution
Peripherals	Three-button mouse with scroll wheel, Keyboard, Ethernet Card

---

NOTE: The OpenGL version is checked during SIMDIS startup and a message box will appear if minimum requirements are not met.

2.2 Core Install

The **Core** install is the refactored package to run SIMDIS with basic functionality. The installer is a **ZIP** file that contains minimal SIMDIS application with the required DLLs (Dynamic Link Library) or shared objects and files. It does not include programs such as Plot-XY, ASIEditor, Model Viewer, etc. It also does not include documentation.

The Core install is meant to be a stripped down version of the application for the end user to:

- Create custom plug-ins
- Create custom data sets
- Redistribute the package as a self-contained product

The Core package is on the order of 90 MB (210 MB unzipped). The Core package does not require an installation procedure, environment variables, or administrator/root privileges. However, there is still a license key requirement. It is intended for advanced users looking to customize or redistribute SIMDIS using as small of a footprint as possible. Users can create their own distributions by either removing undesired files from a full install or adding new files to a core install.

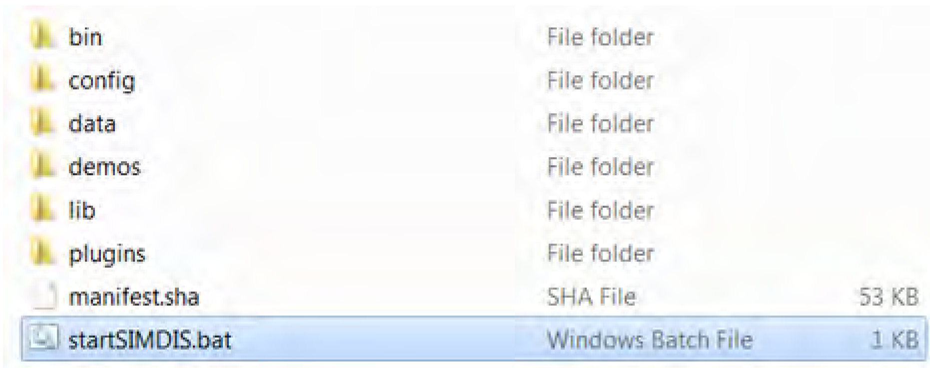


Figure 2.1: Core Install Directory

NOTE: The **User Install** application that ships with the core build can be used to create desktop icons and start menu shortcuts. You can also start SIMDIS by running the batch file **startSIMDIS.bat**, shown in Figure 2.1.

2.3 Full Install

The SIMDIS Full installation is a software suite loaded with plug-ins and extensions for general purposes. The installer is packaged as either an **.exe** for Windows or **.run** for Linux. SIMDIS does not require elevated privileges to either install or run. Elevated access is only required if installation is to a folder where the user account does not have write access.

The Linux installer will typically not have an execute permission after download. Prior to running the installer:

1. Right-click the installer file (**.run**).
2. Select **Properties**.
3. Select the **Permissions** tab.
4. Verify that the **Allow executing file as a program** is checked Figure 2.2.

For Linux, SIMDIS is compiled using the **GCC 4.4.7** in 64-bit version only. The **GCC 4.4** compiler supports RHEL 6-based and later, including versions of Ubuntu, CentOS, and Fedora operating systems.

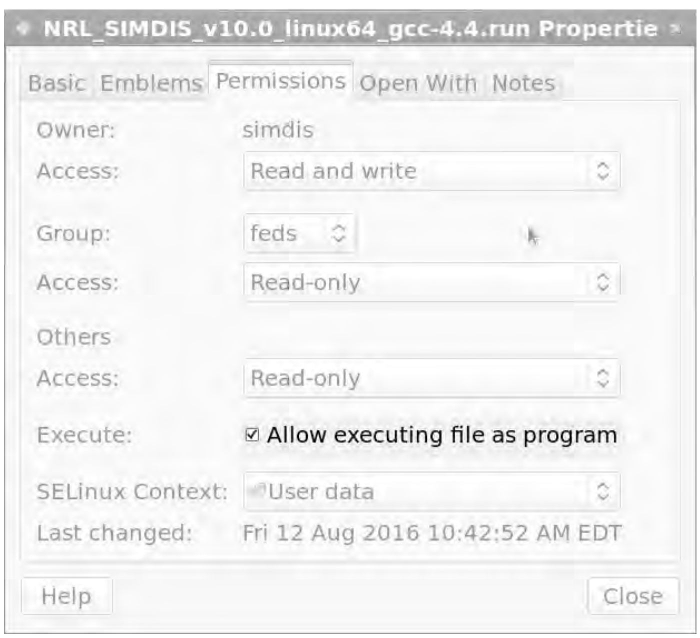


Figure 2.2: Linux File Properties Dialog

On Windows, the SIMDIS installer will detect the user's access level and suggest an appropriate directory for installation. If the user does not have elevated access, the default directory selected will be their local user folder, `C:\Users\username`. If the user has elevated access, the standard location of `C:\Program Files` is selected. The user also has the option to pick a writable directory of their choice.

To install SIMDIS on a machine, download and execute the installer. The installer will copy all the necessary files onto the computer and initiate an installation wizard. The full installer also accepts command line parameters, refer to Section C.2.

2.4 ZIP Install

The **ZIP** installation contains the same setup as Full Installer. The big difference is that the installer is a ZIP file and does not require elevated privileges to install or run.

To install:

1. Log in to the SIMDIS website and download **SIMDIS Distribution (ZIP)**.
2. Extract the downloaded zip file to a directory where you have write privileges (e.g. Desktop).
3. Navigate to the extracted directory and go to:
   - **Windows**: `\SIMDIS\bin\amd64-nt\`
   - **Linux**: `/SIMDIS/bin/amd64-linux/`
4. Run **UserInstall.exe**
5. Set the **PATH** Environment Variable, **Remove Per-User Persistent Settings File** in Clear Settings, and click the **Create SIMDIS shortcuts** radio button and check **Desktop**. Refer to Figure 2.3.
6. Click the **OK** button to apply.
7. Navigate to the **Desktop** and double-click the icon to start SIMDIS.

Details on the usage of the **UserInstall** program are covered in Section 2.5.

NOTE: The Linux zip installs include symbolic links. Make sure that the zip extraction application respects symbolic links when extracting the Linux zip install. The **UnZip 6.00** (“unzip" on **RHEL6**) and **RHEL7’s Archive Manager 3.14** are tested to support symbolic links. The **File Roller 2.28** (**RHEL6** GUI application) is tested to not support the symbolic links, and should not be used.

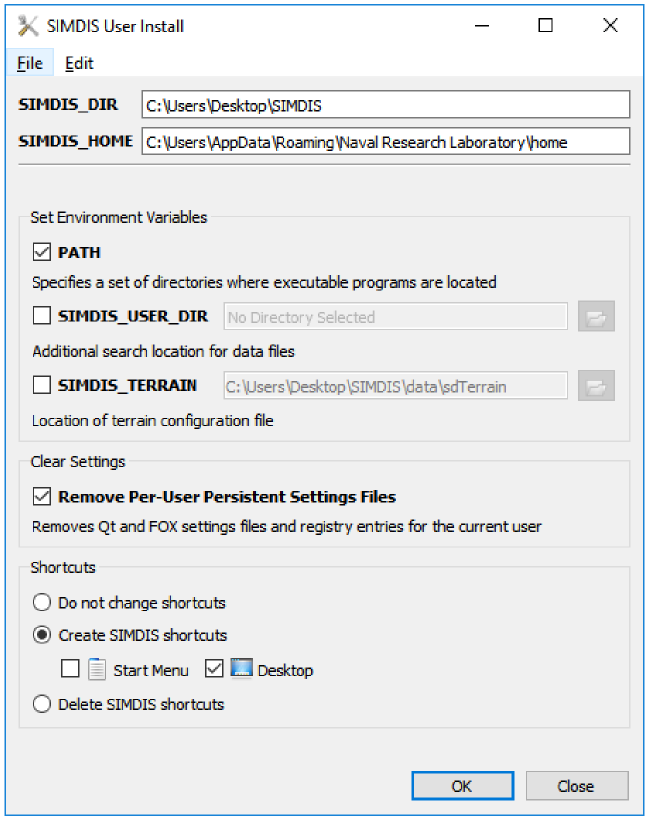


Figure 2.3: User Install Dialog - Zip Install Settings

2.5 User Install

The **User Install** program allows users to either validate or modify their individual environments with regard to the SIMDIS software suite. It configures per-user settings like your **PATH**. It is also used to uninstall SIMDIS and for clearing out things like old settings and cache files.

The User Install program performs the following functions:

Update the **PATH** environment	The **PATH** environment is used by the operating system to find programs. It is needed if using the command line shells to run programs. With the proper user privilege, installer will add **SIMDIS_DIR** to the **PATH**.

Update user-specific application environment variables.	The `SIMDIS_USER_DIR` and `SIMDIS_TERRAIN` are used by the SIMDIS software suite to look for configuration and other data. The parameters can be changed to use a different directory to store files.
`Remove Per-User Persistent Settings File`	Deleting the program settings will cause SIMDIS to forget any user settings values. This is useful to refresh SIMDIS to its default factory settings, or if issues arise with the current settings.
Create or delete shortcuts.	There are two types of shortcuts that the user can create under the **`Create SIMDIS shortcuts`**: on Start Menu and Desktop. Select the appropriate checkboxes to create shortcut/s or select the **Delete SIMDIS shortcuts** radio button to delete them.

The **removeUserConfigFiles.bat** batch file, located in the **bin** folder, will clear all persistent settings, by calling **UserInstall –uninstall**.

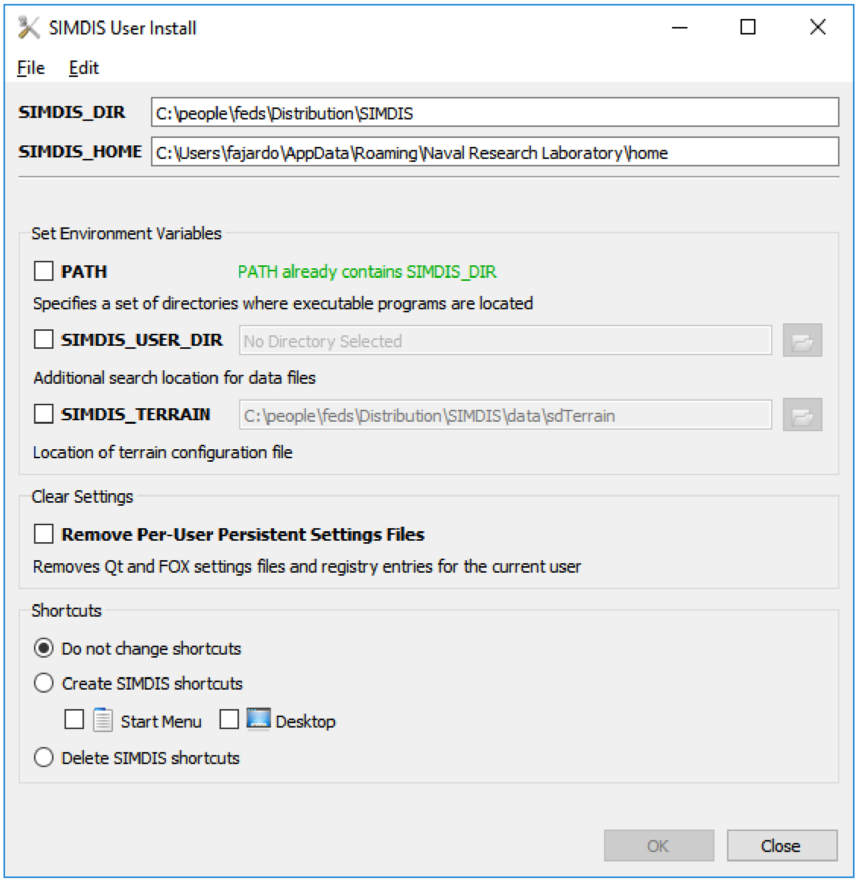


Figure 2.4: User Install Dialog

2.6 Uninstall

To uninstall SIMDIS on UNIX based systems, the uninstaller is found in the SIMDIS installation directory and is called **uninstall_SIMDIS**.

Uninstalling SIMDIS in Windows is accomplished either through:

- **Start Menu > SIMDIS**
- **Control Panel > Programs and Features**

NOTE: To uninstall the Core version, simply delete the folder.

Verify the following directories for persistent settings files:

Windows:

- **SIMDIS Qt Settings:** `%APPDATA%\Naval Research Laboratory`
- **SIMDIS FOX Registry:** in `regedit`, `HKCU\Software\Naval Research Laboratory`

Linux:

- **Qt Settings:** `~/.config/Naval Research Laboratory`
- **FOX Registry:** `~/.foxrc/Naval Research Laboratory`

Chapter 3

Starting SIMDIS

There are plenty of ways to start SIMDIS:

- Run the executable from the folder specific to the platform

  e.g. `$(SIMDIS_DIR)/bin` for **Linux**

- Double-click the SIMDIS icon  on the desktop

- Type **simdis** in command prompt (if **PATH** is configured)

- Create a batch file to customize SIMDIS. For more information see Section C.1

NOTE: For the core build, there are neither desktop icons nor application **path** settings, all files referenced will be relative to the execution location of SIMDIS. You can start SIMDIS by running the batch file **startSIMDIS.bat**. See Figure 2.1

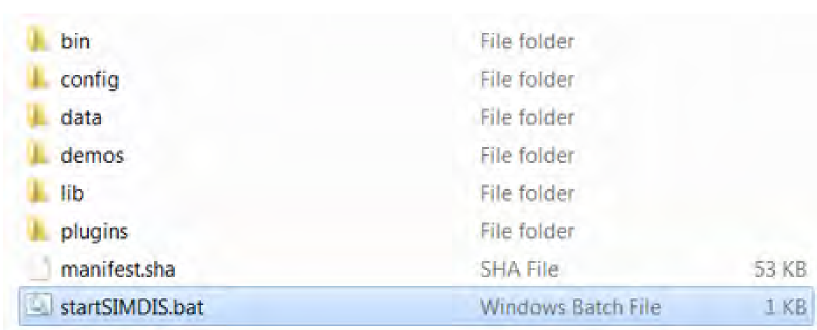


Figure 3.1: Core Install Directory

SIMDIS can be executed with a variety of optional command line arguments, these arguments affect the way SIMDIS is initialized. SIMDIS will run even if no arguments are specified. The options available through these arguments are all also available through the use of the various menu items and tools within SIMDIS. The available command line arguments can be shown by typing `simdis10 --help` into a command prompt.

SIMDIS initially displays the Earth in perspective mode centered off the coast of Africa in the Atlantic Ocean, as seen in Figure 3.2.

NOTES:

- The **Core Distribution** does not automatically create desktop icons. Use the **User Install** application to create desktop and start menu shortcuts, and set environment variables (with the proper permission). Refer to Section 2.5 for more information about the **User Install** application. All files referenced will be relative to the execution location of SIMDIS.
- With either the **Core** or **Full Install**, you can start SIMDIS by running the **startSIMDIS.bat** in the top level of the SIMDIS directory, see Figure 3.1.
- When SIMDIS is launched, it will check the configuration files in the `SIMDIS_HOME` directory. If there are missing configuration files, SIMDIS will copy the missing configuration files from the SIMDIS configuration directory (`SIMDIS/config/`) to the user's `SIMDIS_HOME` directory. To verify the `SIMDIS_HOME` directory in SIMDIS go to **Help > About**.

3.1 SIMDIS Display

Figure 3.2 identifies many of the key components in the main display of SIMDIS.

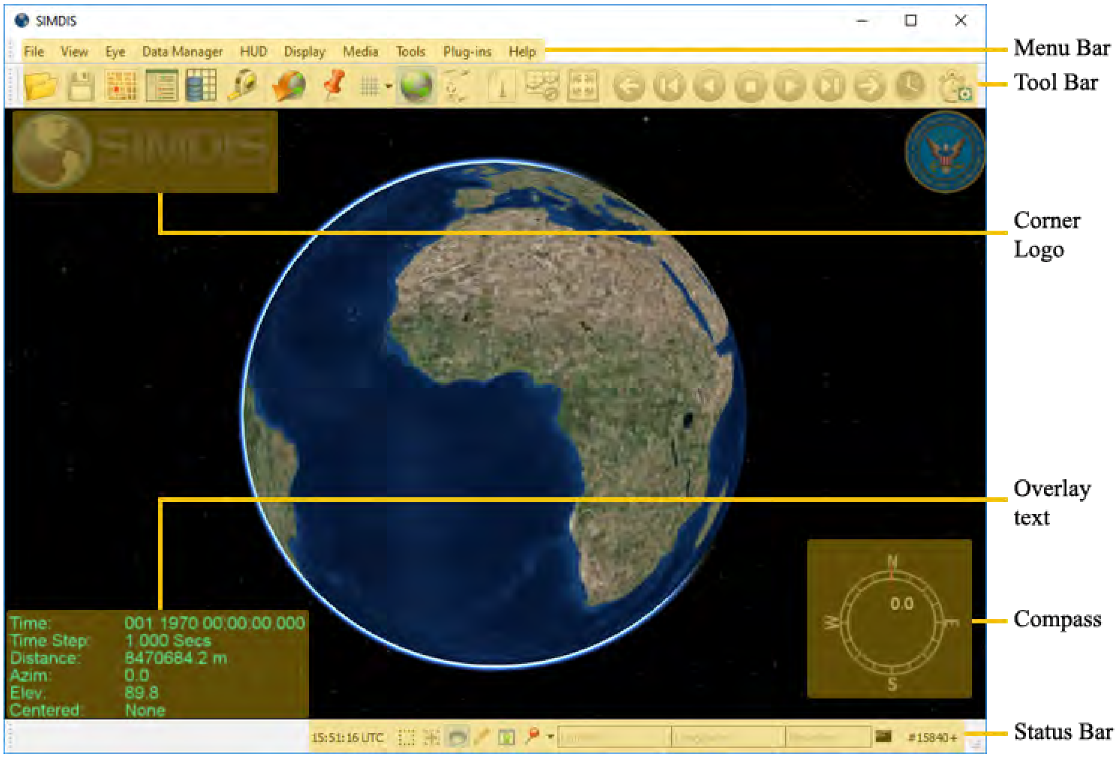


Figure 3.2: SIMDIS Display

3.1.1 Display Modes

SIMDIS has two display modes: Overhead and Perspective. Figure 3.3 shows an example of each mode. The default display mode is the **Perspective (3D)** mode. Toggle the **Overhead** mode via the button on the Toolbar or the hot key Ctrl + O.

Overhead Mode - orthographic projection pointing North, 2D view of the Earth.

- The **Eye** can pan and zoom around the Earth but cannot pitch, roll, or rotate.
- The ← or → keys will pan the display either West or East.
- The ↑ or ↓ keys will pan North or South.

Perspective Mode - a 3D view of the Earth.

- The **Eye** can pitch, roll, rotate, pan, and zoom.
- The ← or → key will control the azimuth angle of the view.
- The ↑ or ↓ keys will affect the elevation angle.

NOTE: Entities that are on the edge of the viewing field and far behind the view may not be visible in some extreme cases (e.g. satellites with high altitude) in **Overhead** mode.

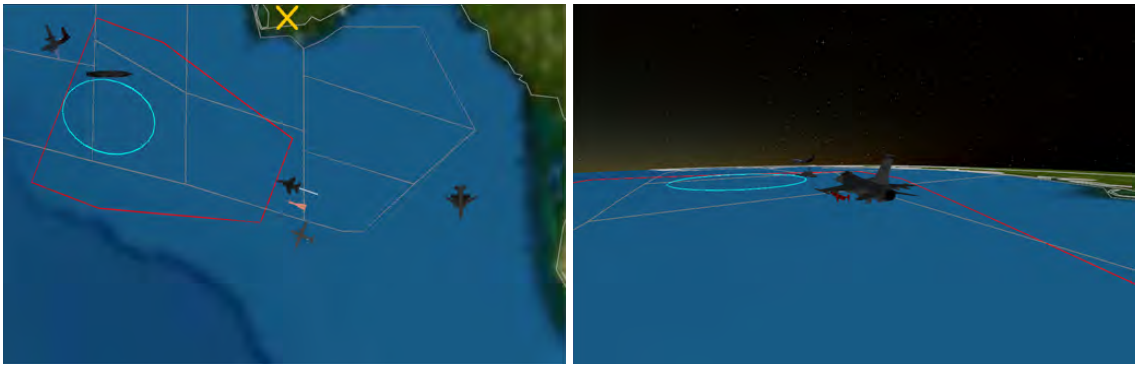


Figure 3.3: Overhead Mode vs. Perspective Mode

3.2 Mouse and Keyboard Controls

3.2.1 Mouse Navigation Modes

SIMDIS has two mouse navigation modes: SIMDIS Classic and GIS Mode.

- **SIMDIS Classic** - allows mouse controls similar to previous versions of SIMDIS. By default, mouse navigation mode is set to **SIMDIS Classic**.
- **GIS Mode** - allows mouse controls similar to popular GIS applications such as ArcGIS or Google Earth.

To change the mouse navigation mode go to: **Tools > Settings > Main Window > Mouse > Navigation**. The Table 3.3 is a quick reference guide for mouse and keyboard controls in SIMDIS.

Mouse Controls					
+	Rotate/Tilt View	+	Zoom In and Out	+	Grab and Pan

Keyboard Controls			
Shift + F	Full Screen	C	Center on next available platform
Ctrl + O	Overhead Mode	D	Dynamically scale all platforms
Ctrl + T	Top Down North		

Table 3.3: Basic Mouse and Keyboard Controls

By default, the **Mouse Navigation Modes** on the status bar are hidden. You can enable this feature via **Settings > Main Window > Mouse > Nav Modes on Status Bar**.

NOTES:

- You will need to restart SIMDIS to display additional mouse navigation modes on the status bar.
- Not all basic controls from Table 3.3 apply to the **Overhead Mode**.

3.2.1.1 Box Zoom

Holding **Ctrl** + **Shift** key while dragging the mouse with the left mouse button pressed will draw a box from the originally clicked position to the cursor's current position. When the mouse button is released, the display will translate to the center of the box and the range will zoom to display the contents of the box.

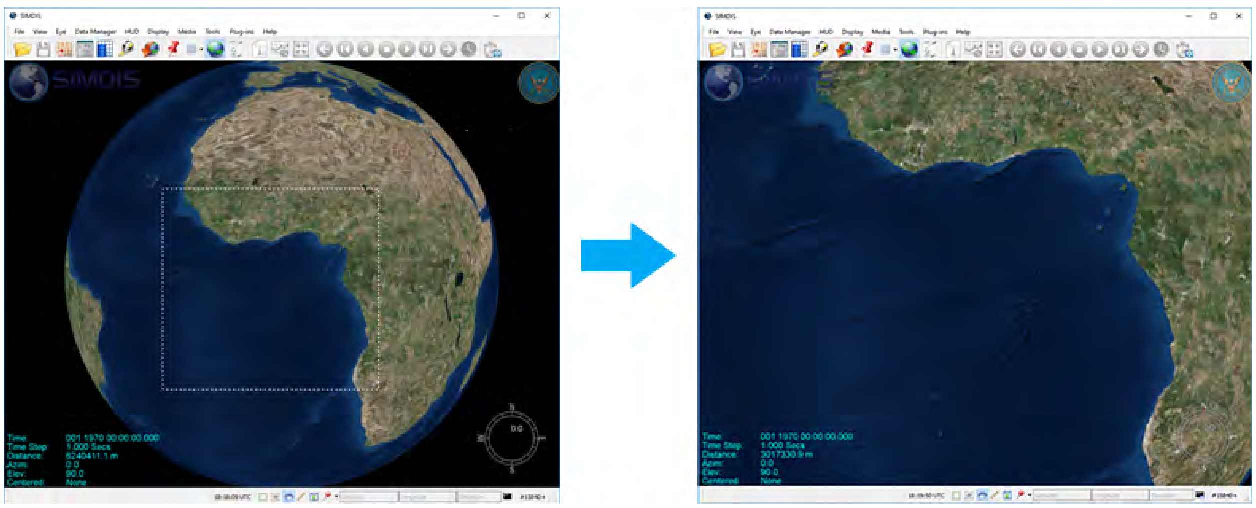


Figure 3.4: Box Zoom

3.2.1.2 SIMDIS Classic Mouse Navigation in Perspective (3-D)

The Table 3.4 below is a quick reference for mouse controls in Perspective Mode for **SIMDIS Classic** mouse navigation mode.

Left Button		Scroll/Middle Button		Right Button	
Action	**Operation**	**Action**	**Operation**	**Action**	**Operation**
Double Click	Re-center Display	Scroll	Zoom in or out	Click and Drag	Drag and Drop Globe
Click and Drag	Tilt or Rotate	Click and Drag	Glide in or out	Click on track	Track Functions

Table 3.4: SIMDIS Classic Mouse Navigation in Perpective

NOTE: Holding the Ctrl key while pressing the left mouse button will recenter the display at the mouse pointer location.

3.2.1.3 GIS Mode Mouse Navigation in Perspective (3-D)

Pressing the left mouse button, and moving the mouse or pressing arrow keys will pan or rotate the earth. The Table 3.5 below is quick reference for mouse controls in Perspective Mode for **GIS Mode** mouse navigation.

Left Button		Scroll/Middle Button		Right Button	
Action	**Operation**	**Action**	**Operation**	**Action**	**Operation**
Double Click	Glide in	Scroll	Zoom in or out	Click and Drag	Zoom in and out
Click and Drag	Rotate	Click and Drag	Tilt	Click on track	Track Functions

Table 3.5: GIS Mode Mouse Navigation in Perpective

3.2.2 Overhead Movement

The **Overhead** mode has the same controls in SIMDIS Classic and GIS Mode except the ability to **Tilt**, to control the elevation view angle.

3.3 Time Slider

When you load a scenario in SIMDIS, the **Time Slider** will appear (by default on the right side of SIMDIS display). The **Time Slider** will allow you to navigate through the timeline of the scenario.

You can set the position of the **Time Slider** simply by **right-clicking** on the slider and select the desired position/docking. Refer to Figure 3.5. The **Invert** option will shift the progression of the bar from bottom to top (if the slider is either docked left or right).

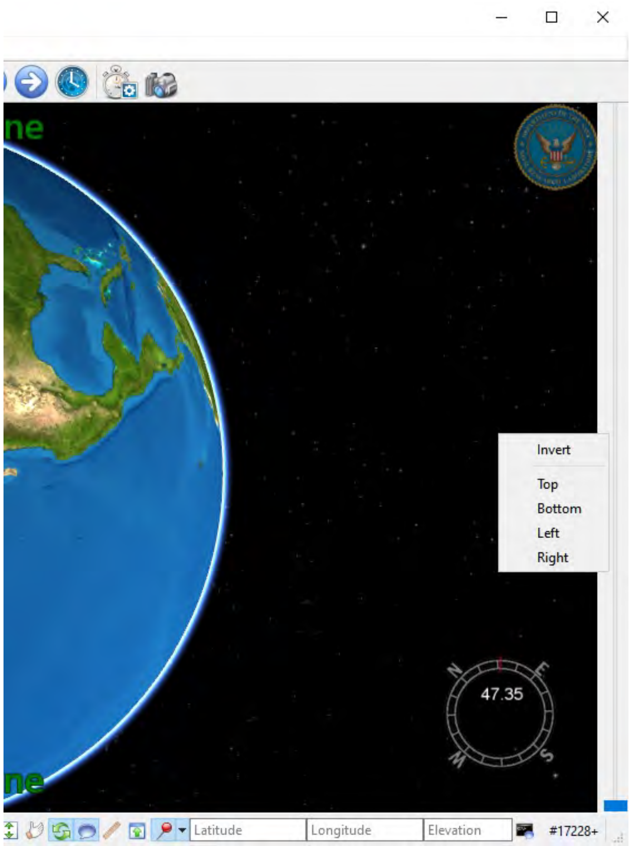


Figure 3.5: Time Slider Option

3.4 Window Behavior

Many of the dialog windows used to support SIMDIS tools may be docked and undocked from the main SIMDIS window. In order to determine if the dialog is dockable to the main SIMDIS window look at the upper right-hand corner of the dialog.

In order to enable the dialog window to be dockable, **right-click** over the Window Title Bar as seen in Figure 3.6 for the GOG (Generalized Overlay Graphics) Tool. A drop-down window appears with the optional window states. If it is dockable then this option will be checked.

A drop-down window appears with the optional window states. If it is dockable then this option will be checked. It may be toggled **on** and **off** by selecting **Dockable**.

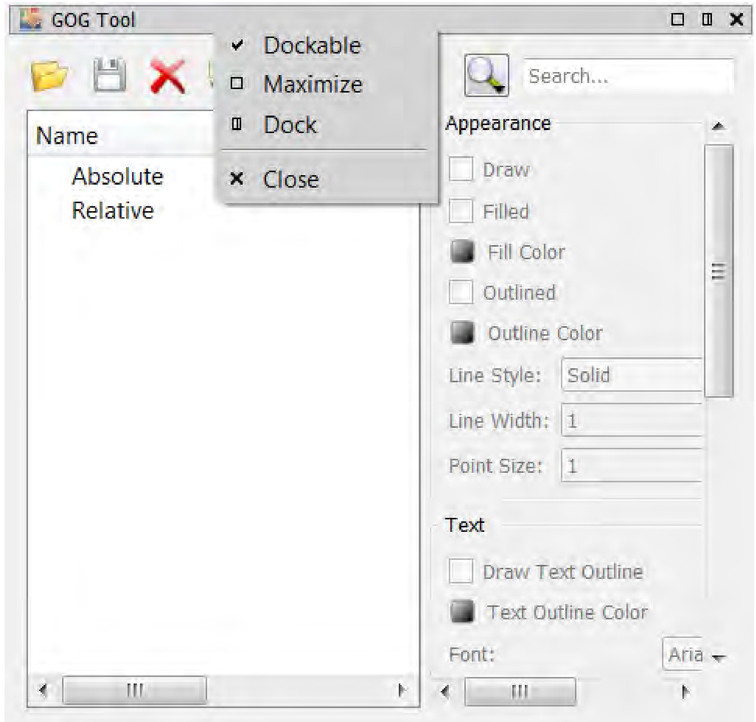


Figure 3.6: Example of Dockable Window

Once the dialog window has been enabled to be dockable, it may be docked to the **main window** in one of two ways:

- Left mouse-click over the Dock icon at the right of the dialog title bar.
- Left mouse-click-hold over the window title bar and drag the window to the side of the main window where you want to dock.
  1. The main window will split and a blue frame will appear as shown in Figure 3.7. This display will vary based on your system configuration and operating system.
  2. Release the mouse and the window will dock as shown in Figure 3.8.

Dialogs may be docked to top, bottom, left and right sides of the main window. More than one side of the main window may be used with multiple dialogs.

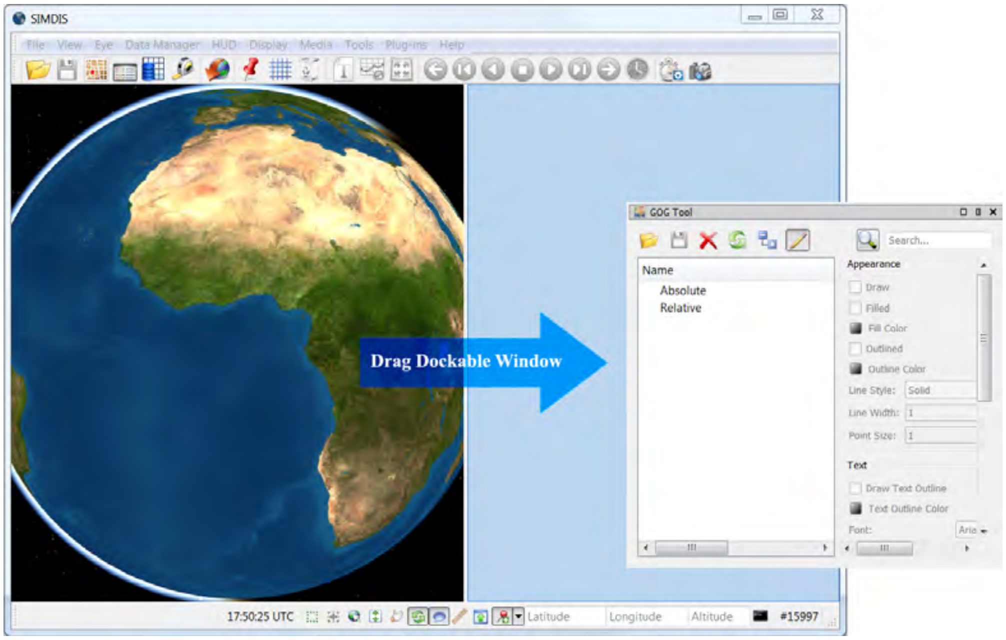


Figure 3.7: Manually Docking a Window

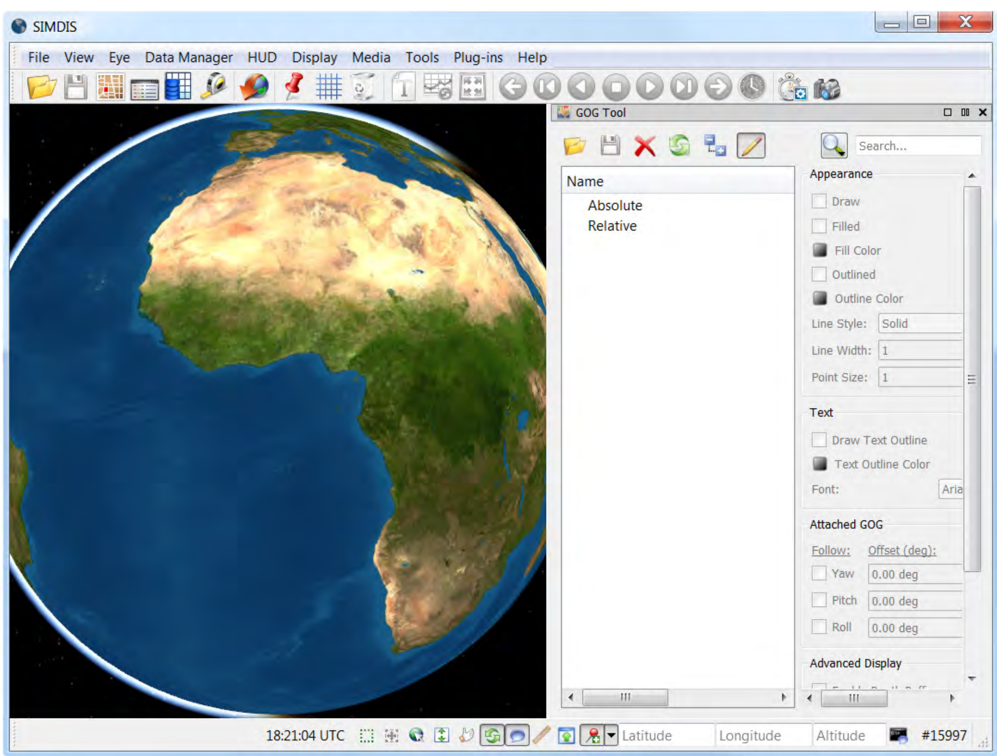


Figure 3.8: Docked Window

Multiple dialogs may be docked in which case each new window will add a tab to the bottom of the docking frame as shown in Figure 3.9. You may switch between docked windows by left mouse-clicking over the tab of the desired window.

The tabs at the bottom may be used to navigate between the docked windows.

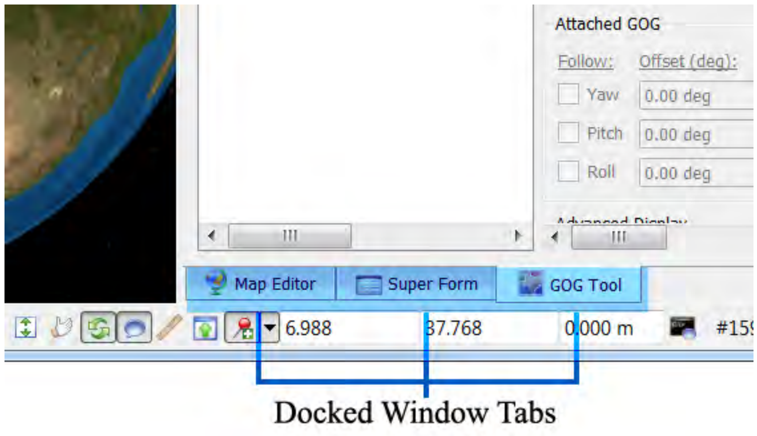


Figure 3.9: Multiple Docked Windows

The windows may be undocked by clicking the undock button over the dialog window title bar, double-click on the title bar, or by left mouse-click-drag over the dialog window title bar.

TIPS:

1. The title bar of the windows will change color based on your focus.
2. When you double-click the title bar of the window, it will maximize its size. If it is already maximized, the window will restore its last known size.

3.5 Layout Files

SIMDIS automatically saves layout information for itself and all associated dialogs when closed and restores that layout the next time it's opened. That layout information includes which dialogs are open as well as their position and size. Different layout files can be loaded via the command line option `--layout <file>`. Layout can be saved to different files via **Display > Save Layout As** from the Menu Bar.

By default, the layout is saved to and loaded from `$(SIMDIS_HOME)/SIMDISLayout.ini`

3.6 Configuration Files

SIMDIS use a configuration file **SIMDIS.ini** that is automatically saved upon exiting. It is not saved as part of the SIMDIS distribution but is saved in the user space. It is stored to either:

Windows: `%APPDATA%\Naval Research Laboratory\SIMDIS.ini`
Linux: `~/.config/Naval Research Laboratory/SIMDIS.ini`

The configuration file is in an ASCII format and can be read and edited (not recommended) by the user. The configuration is read by SIMDIS the next time it is started in order to return it to the user's previous state.

3.7 Settings

The primary means for modifying the configuration is via the **Settings dialog**, accessed via either **Tools > Settings** from the Menu Bar or the default Hot Key sequence
Ctrl + **Alt** + **O** .

Configurations can be loaded or saved from this dialog. Many settings take effect immediately, and some settings require a restart of SIMDIS. Check the tooltip for more information on each setting.

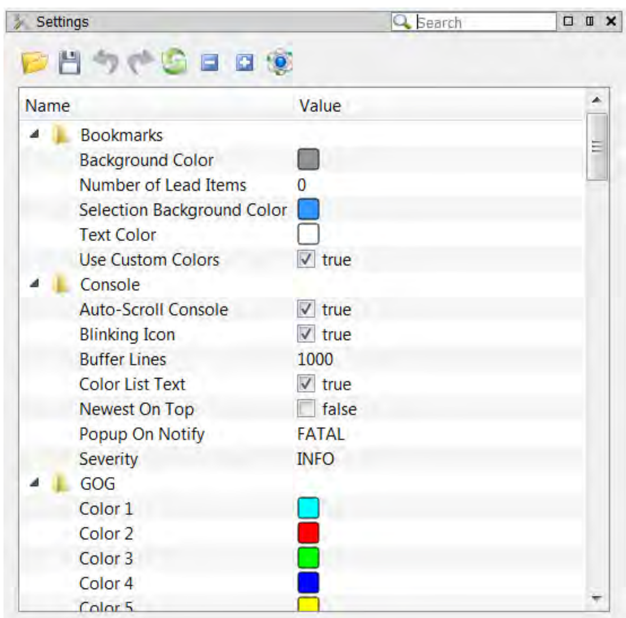


Figure 3.10: Settings Dialog

NOTE: The Settings Dialog influences the behavior of the SIMDIS display, but it is not necessarily the only influence. The values in Settings Dialog can be overridden externally through extensions, plug-ins, and environment variables. For example, certain behavior in OpenSceneGraph (OSG) and osgEarth can be overridden through the use of environment variables which have higher priority over Settings in SIMDIS.

3.8 Heads-Up Display (HUD)

The **HUD** menu controls the display of 2D overlay text and graphics in the main window.

3.8.1 Overlay Text Display

The **Overlay Text Display** is in the lower left of the main window that displays scenario information. The contents are available through the **HUD** menu. Refer to Section E.8 for the list of supported format for Overlay Text Display.

You can cycle through different **Overlay Text** formats through **HUD > Formats** or the default hot key O.

Figure 3.11: Overlay Text Display Formats

3.8.2 Compass and Wind Vane

Wind Vane W - This item will toggle the display of a wind vane and text in the bottom right corner. The direction from which the wind is blowing is indicated by the arrow and the text. For example, an easterly wind is blowing from the east and will be reported as a direction from 90 degrees. The wind speed is displayed below the vane in meters per seconds (**m/s**).

Compass Shift + W - This item toggles the display of the heading compass at the bottom right-hand corner of the main window. The heading compass displays the azimuth of the current view.

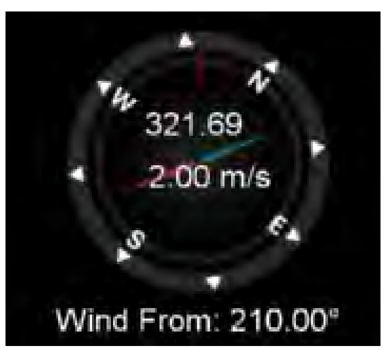


Figure 3.12: Compass and Wind Vane

3.8.3 Map Scale

The **Map Scale** is shown on the lower-right of the main display. It will allow you to easily visualize the distance or range displayed. The scale will dynamically adjust based on the zoom level of the main display. You can change the unit displayed on the scale via **Settings > Map Scale Units**.

Figure 3.13: Map Scale

3.8.4 Statistics

Frame Rate Alt + F - This item will toggle the frame rate display in the upper left-hand corner of the main window.

Viewer Statistics - This will toggle the advanced statistical display for performance metrics for debugging. This is useful for developers to benchmark and track performance bottlenecks.

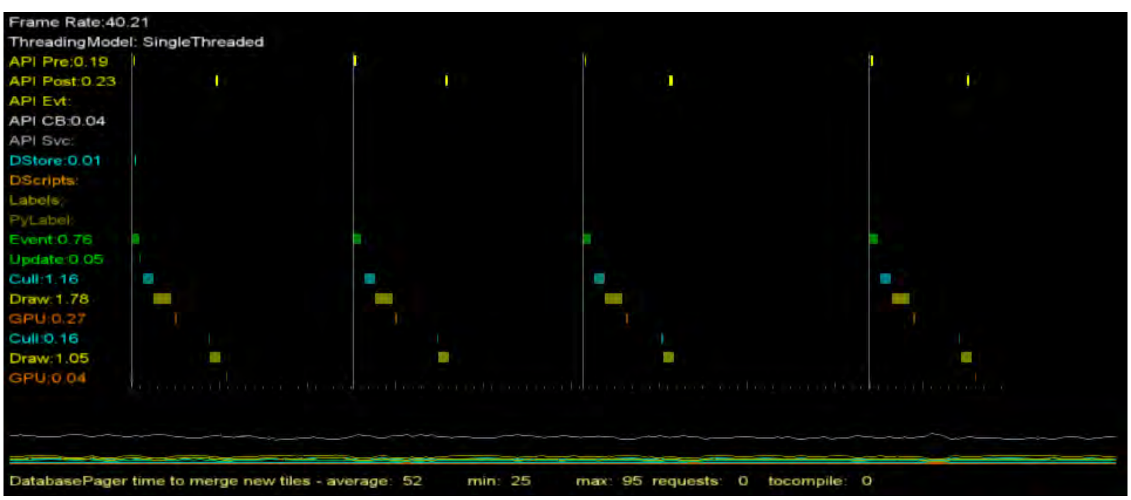


Figure 3.14: The Viewer Statistics

It will display the frame rate of SIMDIS and its process utilization. The graph displays series of vertical white lines. Between the two vertical white lines represents a frame. The loads in milliseconds are represented by horizontal bars on each frame.

The following is the list of the timing statistics:

Plug-in API

- **API Pre** - Plug-in Server pre-draw time
- **API Post** - Plug-in Server post-draw callback time
- **API Evt** - Plug-in Server event processing time
- **API CB** - Plug-in API callback time
- **API Svc** - Plug-in API methods processing time

Data Store

- **DStore** - SIMDIS Data Store scenario update time

Data Scripts

- **DScripts** - Active data scripts processing time

Entity Label

- **Labels** - Label draw time
- **PyLabel** - Python label interpreter processing time

osgEarth

- **Event** - Event traversal time
- **Update** - Update traversal time

The first block of **Cull**, **Draw**, and **GPU** represents the **Main Window**, subsequent blocks represent **Insets**.

- **Cull** - Cull traversal time
- **Draw** - Objects draw traversal time
- **GPU** - GPU draw time

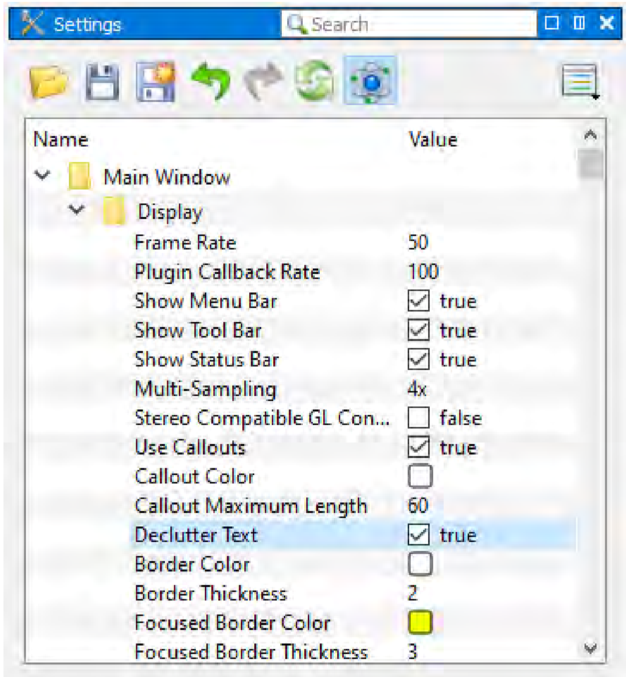


Figure 3.15: Declutter Settings

3.8.5 Text Decluttering

Text Decluttering is useful when multiple platforms are near enough such that their labels overlap each other. Use the setting at **Main Window / Display / Declutter Text** in the Settings dialog (see Section 3.7) to enable Text Decluttering.

The default decluttering method shows as many labels on screen at a time as possible without having any overlap (see Figure 3.16).

Figure 3.16: Default Text Declutter Method

Enable the setting **Main Window / Display / Use Callouts** to use callouts when decluttering. This decluttering method will show all labels and move them away from their associated platforms in an attempt to have none of the labels overlapping. The labels are connected to their platforms with a line (see Figure 3.17).

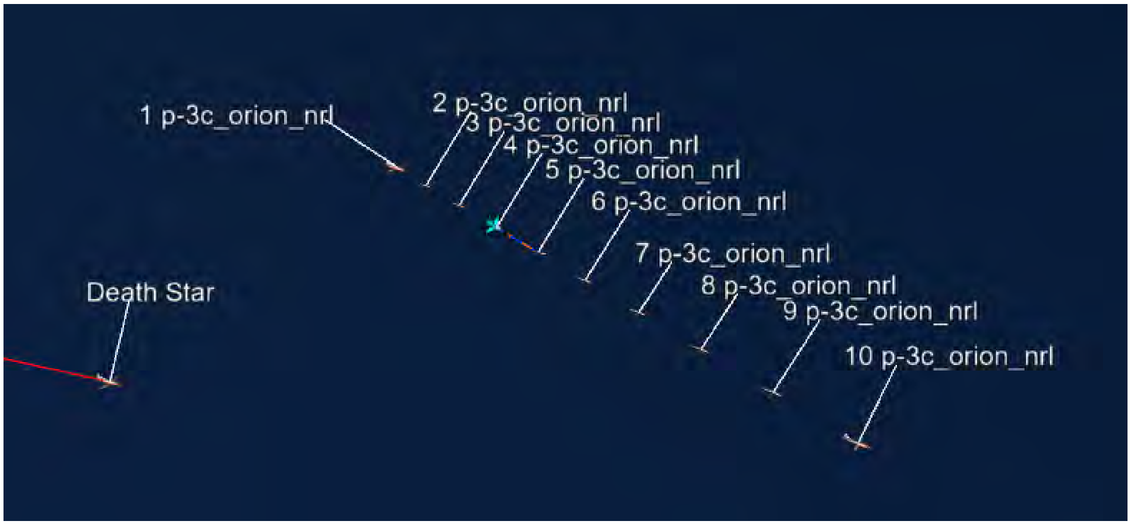


Figure 3.17: Text Decluttering Using Callouts

The settings **Main Window / Display / Callout Color** and **Main Window / Display / Callout Maximum Length** can be used to customize the callout lines.

3.9 Models/Icons

SIMDIS supports many different 3D model and 2D image file formats though use of the underlying OpenSceneGraph (OSG) library. The **.3db** file format is used to encrypt models that cannot be distributed outside of the SIMDIS application due to licenses that prevent redistribution. You can download additional 3D models from the SIMDIS website: https://simdis.nrl.navy.mil/Models.aspx.

The following is a list of model supported file formats that you can load in SIMDIS:

.dxf	AutoCAD DXF
.asc, .3ds	Autodesk 3D Studio File Format (old)
.3db	3-D Binary Model
.lwo .lws	Lightwave Objects and Scenes
.flt	Open Flight
.tmd, .lst	Media Player Video Icons
.iv	Open Inventor
.osg, .osga, .osgb, .ive	OpenSceneGraph (OSG) File Format
.obj	SGI Wavefront Objects
.png, .gif, .jpg, .jpeg, .rgb, .tif, .tiff, .bmp	Image Files

You can download models for free or purchase from the following sites:

AEgis Elements 3D Models	**https://aegiselements.com/**
Quantum 3D Facets	**http://quantum3d.com/facets-models/**
US Army Model Exchange	**https://modelexchange.army.mil/**

3.9.1 Automatic File Searching for Models/Icons

SIMDIS has a default file searching method to automatically find supported files to its library. It searches in the following order:

1. **Data File Locations**
   Location of the data file, and the current working directory of SIMDIS.
2. **Settings Locations**
   Icon, Texture, Default, etc.
3. **Environment Variables**
   `SIMDIS_DIR, SIMDIS_USER_DIR, SIMDIS_TERRAIN`

NOTE: When SIMDIS finds the files in **Settings Location** or **Environment Variables**, SIMDIS will cache the file locations for faster searches later. While this feature greatly improves system performance under normal circumstances, users that regularly add files with duplicate names to these locations may experience difficulty When you are in a situation that the intended files are not displayed properly in SIMDIS, you may need the clear the file cache. There are three ways to clear the file cache:

- Run SIMDIS with the `--clearFileCache` argument.
- Use the SIMDIS **UserInstall** application and check **Remove Per-User Persistent Settings Files**.
- Manually delete the file cache at **SIMDIS_HOME/SIMDIS10FileCache.xml**.

3.9.2 Models/Icon Scaling

By default, SIMDIS will load models/icons in real scale. You can turn on dynamic scaling for all models/icons in SIMDIS simply by clicking the **Dynamic Scale** icon. To set dynamic scaling for a single model/icon:

1. Open the **Prefs Tool**.
2. In the **Entity** tab, click the platform to highlight (You can also hold the Ctrl key to click and highlight multiple platforms).
3. On the **Platform** tab, under the **Appearance Settings**, check the **Dynamic Scale** option.

In **Appearance Settings**, you can also set the **Dynamic Scale Offset** and/or **Dynamic Scale Factor**.

Additionally, the **Settings > Main Window > Platform Dynamic Scaling** setting can be used to set the initial value of the **Dynamic Scale**  action on start up. Changing this setting requires a SIMDIS restart to take effect.

NOTE: You can modify the **Dynamic Scale Factor** value to get the desired scaling of the platform for your display.

3.10 SIMDIS Console

The SIMDIS menu bar contains a Help menu which provides access to help documents and the SIMDIS Console. The Console displays text message alerts from SIMDIS. The operator may filter the display using the drop-down menu at the left of the toolbar. Console display behavior may also be modified using the **Console Settings** button.

Error messages are displayed in red indicating that a particular function has not been implemented yet. One or more selected lines may be copied from the Console and placed on the Windows clipboard to facilitate saving in a file or filing an issue report on the SIMDIS Help Desk.

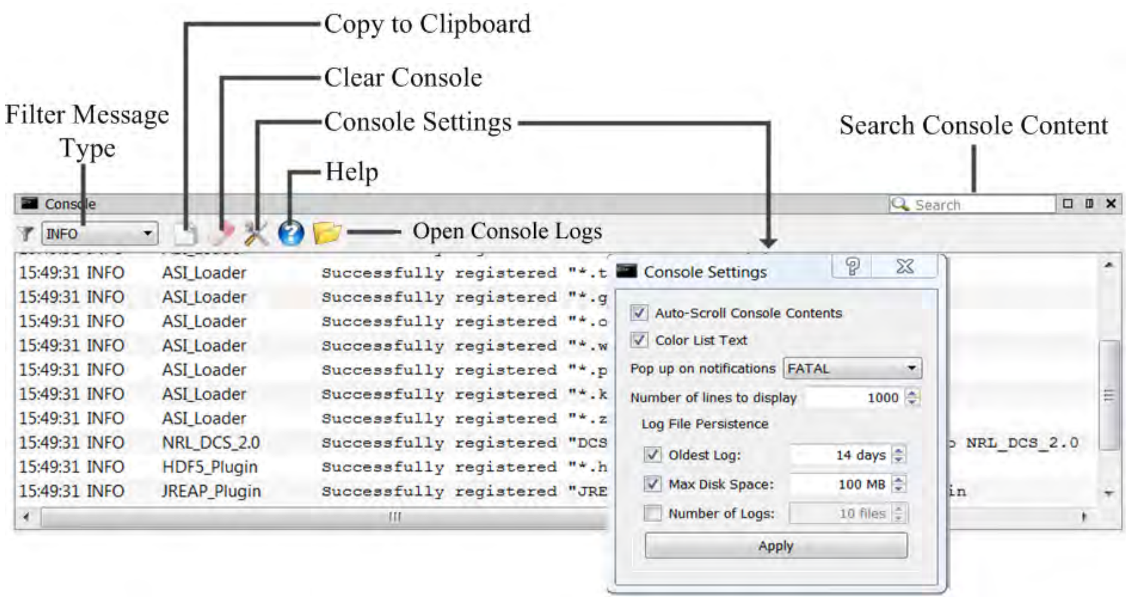


Figure 3.18: SIMDIS Console

The Console icon on the Status Bar at the bottom right of the main window may also be used to display the SIMDIS Console. If the console is not displayed a small light in the console icon will flash to indicate that something has been added to the console log.

3.11 Scenario

A scenario in SIMDIS is a collection of data that exists between a specific begin and end time. The data typically consists of Time, Space, and Position Information (**TSPI**) data of entities such as aircraft, boats, and/or vehicles. In addition to TSPI data, a SIMDIS scenario can also contain maps, 2D vector graphics, media, bookmarks, preferences, and viewing positions.

SIMDIS supports the following object types: Platform, Beam, Gate, Laser, Line of Bearing (LOB), Projector, and Custom Rendering. Ancillary data types are also supported: Generic, Category, and Data Table.

SIMDIS will limit the data to **N** number of previous points and/or drop duplicate data to reduce memory usage. The data reduction features are on by default. For more information refer to Data Limiting and Data Reduction.

There are two methods of data input for SIMDIS, file-based or network-based. File-based data is recorded data stored in a compatible format that can be read by SIMDIS. Network-based data is data streamed from a source, whether it is live or simulation based.

File-based data is loaded into SIMDIS using the existing file parsers or via plug-ins built to convert the data into a compatible format. Network-based data is injected into SIMDIS using either the Data Client Server (DCS) protocol or via plug-ins built to convert the data into a compatible format.

Figure 3.19: Loading Scenario Data

File-based data is loaded into SIMDIS using either existing file parsers or via plug-ins built to convert the data into a compatible format. Network-based data must be either converted to a SIMDIS compatible data format or processed and injected into SIMDIS by a user-created plug-in.

3.11.1 File-Based Scenario Data

All file-based formats are now supported via plug-ins (see Figure 3.20). SIMDIS does not have a native file format that it loads or saves. When a plug-in is loaded with a registered file handler, it will be available for use via the File Open folder icon on the SIMDIS toolbar.

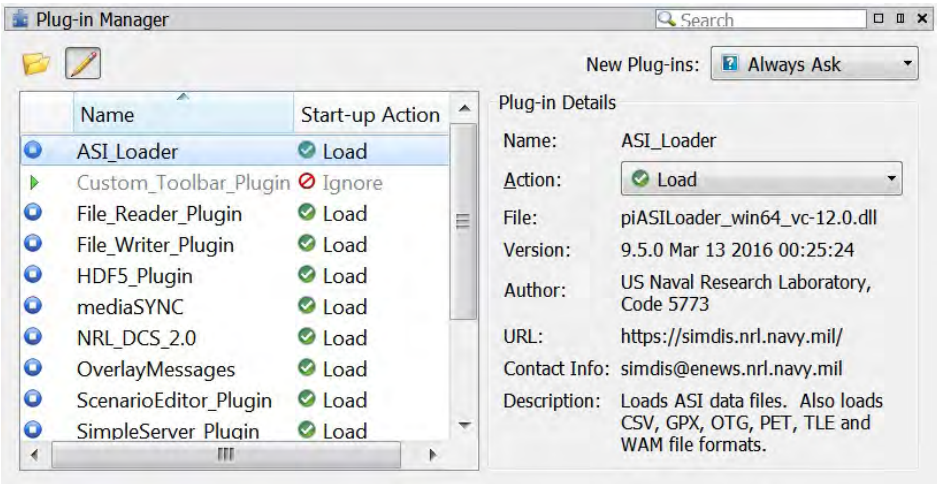


Figure 3.20: Plug-in Manager

To load a file, click the File Open icon and choose a file to load via the file selection dialog or drag the file from a file explorer into the SIMDIS main window. Many example files are included with the SIMDIS distribution in the `$(SIMDIS_DIR)/demos` folder.

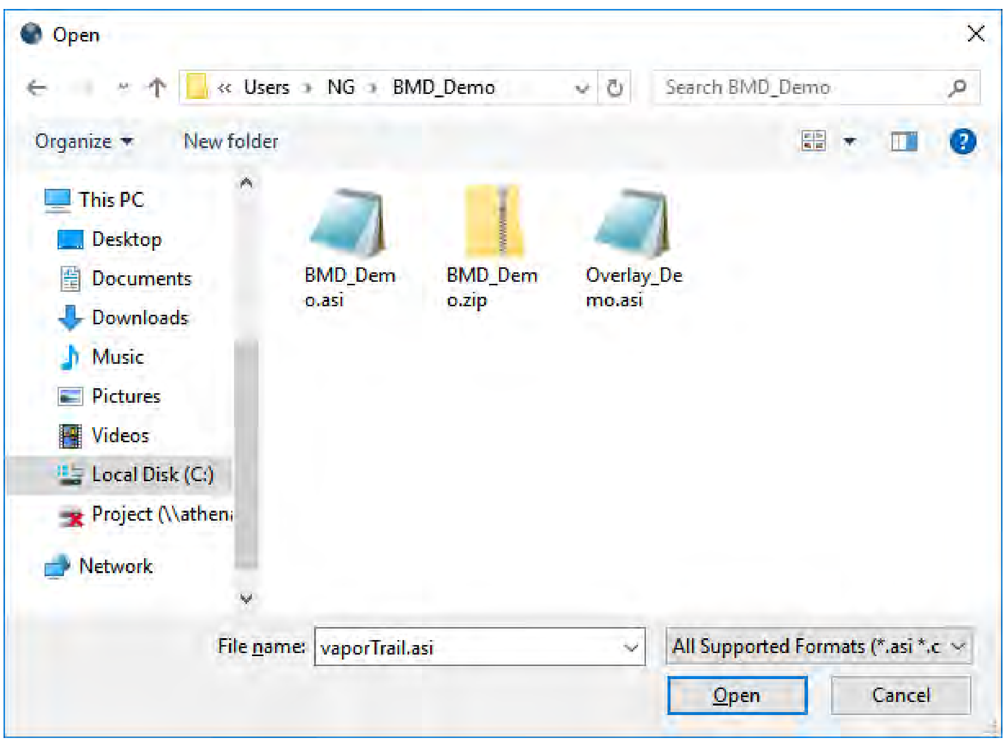


Figure 3.21: Open Scenario Data File

3.11.2 File Append

You can append multiple scenario files using the **Append** feature. You can use the Append feature to consolidate multiple files into a single scenario. The scenario with the earliest **Reference Year** must be loaded first.

To append a file, go to **File > Append** on the menu bar (Figure 3.22) or **Ctrl** + drag the file into the SIMDIS main window.

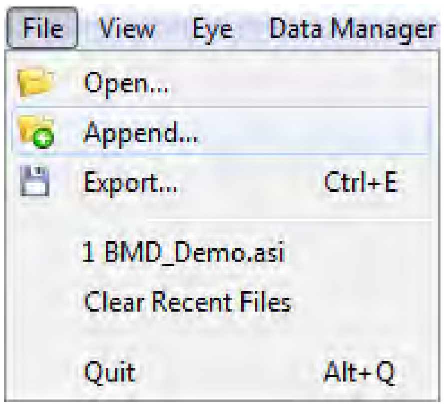


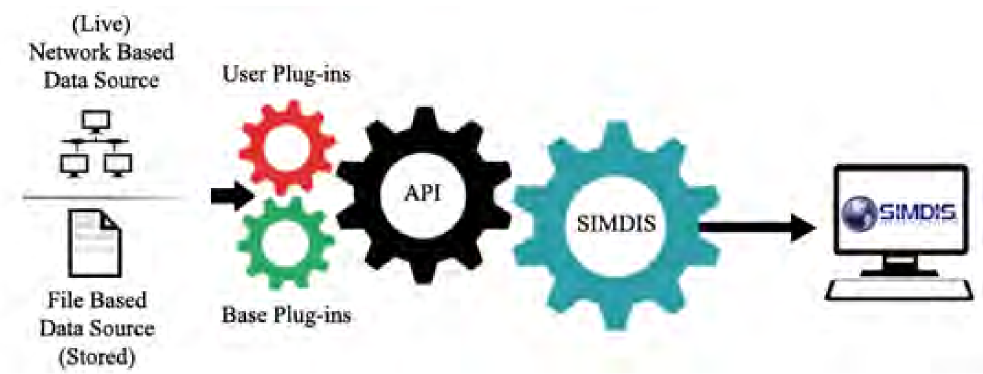
Figure 3.22: File Append

NOTES:

- You can append **.asi** files with different **Reference Years** (e.g. `1970` and current year).
- When the file-based scenario files are appended, the SIMDIS scenario timeline will extend based on the time of the earliest and latest data points from all loaded files.
- Dragging multiple scenario files into SIMDIS will always append all files after the first.

3.11.3 Network-Based Scenario Data and Plug-ins

Network-based scenario data can be loaded into SIMDIS via a plug-in built using the SIMDIS Plug-in Application Programming Interface (API). The SIMDIS Plug-in API supports a bi-directional communication path along with the capability to modify the user interface.



SIMDIS supports both new Qt and old FOX based plug-ins (backward compatible). There may be certain plug-ins that are not supported due to deprecated commands. The user will be notified of unsupported plug-ins and API commands in the SIMDIS Console.

Data conversion is a critical task that plug-ins written for SIMDIS can perform. A plug-in might register itself to handle a given file extension and feed information directly to SIMDIS through data manipulation routines. Similarly, a plug-in might be created to send live range data to SIMDIS in real-time. The distributed Simple Server plug-in is one such plug-in that is provided as an example which can simulate injecting either file or network based data into SIMDIS. The Simple Server plug-in is also distributed with the SIMDIS Plug-in API source code as a sample program.

3.11.4 Data Limiting

Data Limiting is applicable only in **Live Mode**. The default limit for data points is **1000**. The limits applies to the data of all entity types, Category Data, Generic Data, and Data Tables.

The **Data Settings** controls all the data associated with an entity via **Prefs Tool > Viewing > Data Settings**, see Figure 3.23. Data can be limited by time, by the number of points, or both.

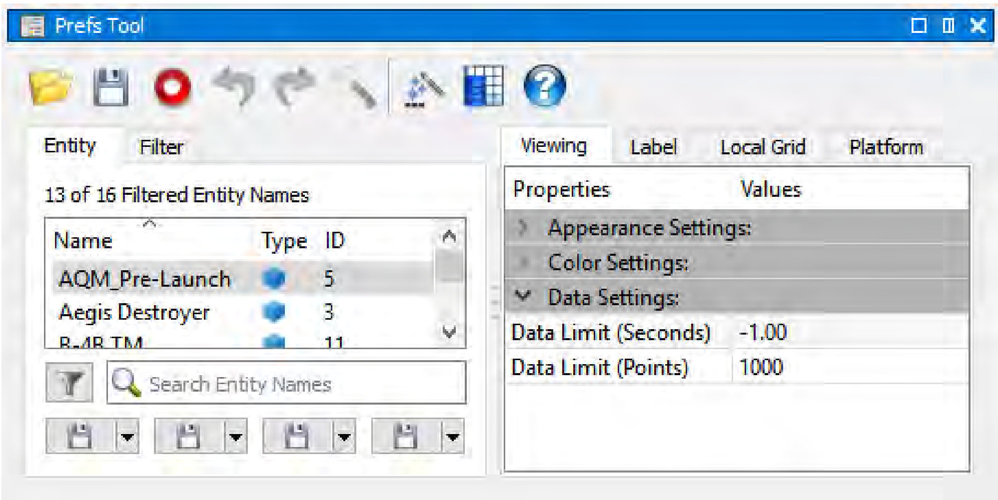


Figure 3.23: Data Settings

- A time data limit value of **-1** second disables the data limiting by time.
- A point data limit value of **0** disables the limiting of data points.

Changes to the **Data Limit** preferences will take effect when the preferences are applied. Data Tables and Generic Data that are associated with the scenario are data limited via **Settings > Scenario**, setting keys are **Data Limit Points** and **Data Limit Time**. Changes to the state of either setting key take effect immediately.

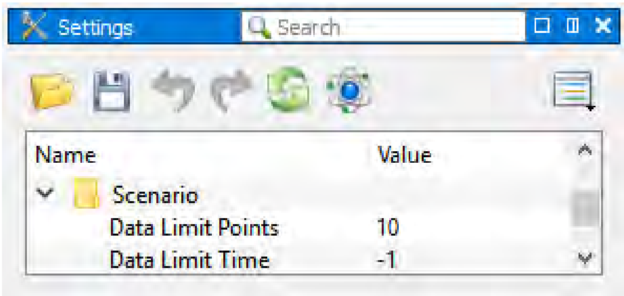


Figure 3.24: Data Limit under Settings

NOTES:

- For all but Data Tables, data limiting is applied when data is added. If a **Platform** is limited to 100 data points, once 100 data points is reached, the oldest point is dropped for every point that is added.
- Due to performance issues, **Data Tables** handle data limiting with higher granularity. Once the data limit is reached, half the data is dropped. For example, if the **Data Table** is limited to 100 rows, once 100 rows is reached, the 50 oldest rows will be dropped.

3.11.5 Data Reduction

In both **Live Mode** and **File Mode**, SIMDIS will drop duplicate data for Generic Data, Color Commands, and Status Commands by default. The detection of duplicated data is dependent on time of the data arrived. This feature should be turned off if data is not guaranteed to be given to SIMDIS in time order.

The **Data Reduction** feature is controlled by setting keys via **Tools > Settings > Scenario**. Changes to the state of the setting keys will take effect immediately.

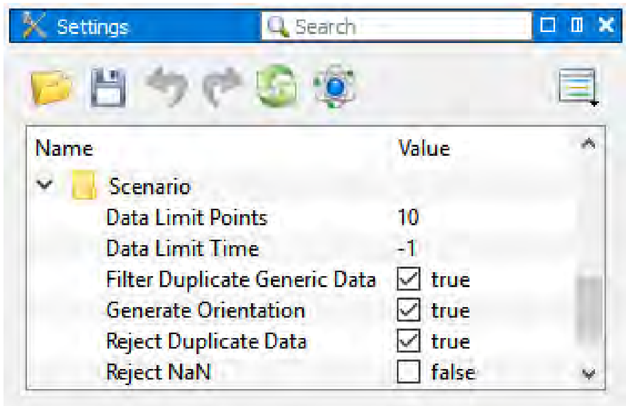


Figure 3.25: Data Reduction under Settings

- **Filter Duplicate Generic Data** will drop duplicate **Generic Data**.
- **Reject Duplicate Data** will drop duplicate **Color Commands** and **Status Commands**.

NOTES:

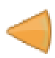
- Dropped duplicated data will not count against the data limits and will not prompt the removal of data.
- Turning off the **Data Reduction** features may increase memory usage and could lead to performance issues with the display.

3.11.6 Data Entity Types

Data loaded into SIMDIS falls into one of several entity types. The entity type will define a specific behavior as well as how the data is stored and retrieved from a file or over the network.

Platform: is the fundamental type that describes the **who**, **what**, and **where** of an entity.

- Entities in SIMDIS are jets, missiles, tanks, or ships.
- Valid entity must contain a timestamp and a position.
- A valid scenario must contain at least one platform entity.



Beam: A child entity of a **Platform**, this is an optional entity in SIMDIS.

- Describes a Radio Frequency (RF) emission based on time, range, and angle.
- Can also be used to portray other non-RF data based entities such as cameras or IR sensors.

Gate: is a child of a **Beam**, this is an optional entity in SIMDIS.

- Describes the range and angle of where an RF emitter is looking during a specified time interval.
- The term gate comes from the method in which pulsed signals are measured, i.e. gated.
- A gate entity can also be used to show a field of view (FOV) for EO/IR sensors as well as search fences for radars.

Laser: is a child of a **Platform**, this is an optional entity in SIMDIS.

- Describes a device that emits light via a focused source.
- A laser is similar to a beam; however, it is drawn as a narrow line instead of having horizontal and vertical extents that spread over distance.

Lines of Bearing (LOB): is a child of a **Platform**, this is an optional entity in SIMDIS.

- Represent receiver detections from an electromagnetic emission on a specified bearing at a given instant in time.

Projector: is a child of a **Platform**, this is an optional entity in SIMDIS.

- Represent imagery projection from a perspective (e.g. from a video camera) into the 3D scene.
- SIMDIS require the position, orientation, Field of View (FOV), and imagery of the projector to display in the manner similar to a video projector; projecting an image onto a screen.

Custom Rendering: can be either a stand-alone entity or a child of a **Platform**, this is an optional entity in SIMDIS.

- Represents custom specific graphics, as defined by a Renderer.
- Originally created for showing Error Ellipses and Ellipsoids, but are flexible to allow for alternate applications.
- Renderers are implemented in SIMDIS Extensions (see Section A.12.6).
- Supports track history feature similar to Platforms. For performance reasons, Live Mode history rendering is based on the initial data table update. Any table updates will not be rendered until the scenario transfers into File Mode.

3.11.7 Ancillary Data Types

In order to provide additional support to existing data entities, ancillary types are also used.

Generic Data is considered any non-positional timestamped data that can have a specific expiration time. Generic data can be associated to any of the entity types listed above as well as the SIMDIS scenario itself. The ASI file format documents all of the SIMDIS specific uses. If the generic data tag is not explicitly documented in the ASI file format, it is treated as textual data.

Category Data is considered any non-positional timestamped data that is used for filtering and display preferences. Category data can be associated to any of the entity types listed above.

Data Table data is considered any discrete parametric timestamped data. Typically, non-positional data such as telemetry (TM) and chat data are stored in a data table. Data tables can be associated to any supported entity type as well as the SIMDIS scenario itself.

3.11.8 Modes and Transitions Between Modes

SIMDIS has three modes of **Uninitialized Mode, Live Mode**, and **File Mode**.

Uninitialized Mode	SIMDIS has no data.
Live Mode	SIMDIS is actively receiving data from a plug-in and time is controlled completely by that plug-in.
File mode	SIMDIS is receiving bulk data, and the user can manipulate the clock once the data has been loaded.

Ending a Live Mode scenario will transition SIMDIS to File Mode. The Simple Server plug-in demonstrates examples of Live Mode, File Mode and the transition from Live Mode to File Mode (clicking the **Stop** button while in Live Mode). Deleting a scenario transitions SIMDIS to Uninitialized Mode. There is an implicit scenario delete when there is a transition from File Mode to Live Mode or from one File Mode to another File Mode. Deleting a scenario does the following:

- Delete all entities including their ancillary data.
- Delete all scenario level ancillary data.
- Delete all graphics like lines, circles, and 3D Landmarks.
- Delete Geo-Filters graphics, but the graphics will be redrawn once a new scenario is initialized.
- Delete all views and viewports.
- Clear out the scenario filename.
- Stop Live Logging, if active.

Preference Rules are not deleted when a scenario is deleted. Preference Rules must be explicitly deleted. Both the File Reader Plug-in and the HDF5 Plug-in will explicitly delete the Preference Rules, if and only if the file loaded by the plug-in contains a reference to a Preference Rule file.

NOTE: At the start of a Live Mode scenario any defined **DISCN** file is loaded.

This Page is Intentionally Left Blank

Chapter 4

Plug-ins

4.1 Plug-in API Support

4.1.1 FOX Toolkit

Plug-ins developed using the FOX Toolkit can be loaded and used in SIMDIS. However, since SIMDIS is based on Qt, the plug-in user interface may look dated. Additionally, FOX plug-ins are not able to take advantage of the docking capabilities that Qt provides.

Tool Tips **do not** work for FOX based plug-ins loaded in SIMDIS. Additionally, the “always on top" settings present in some plug-ins **will not work**.

4.1.2 Qt

Plug-ins developed using Qt can only be loaded in SIMDIS. Compared to the FOX toolkit, the Qt development framework makes developing any user interface quick and easy. It also provides a modern look and feel compared to former versions of SIMDIS. These plug-ins can take advantage of the docking capability provided through Qt.

4.1.3 PIGUI

Plug-ins developed using the PIGUI interface are automatically upgraded to use Qt instead of FOX. As such, these plug-ins are also dockable into the SIMDIS application. Users do not need to recompile PIGUI plug-ins for use with the latest version of SIMDIS.

However, PIGUI functions that provide direct access to FOX widgets do not work and will cause compiler warnings. The functions that will **NOT** work include:

- `PIGUI::PIButton::getFoxButton()`
- `PIGUI::PIGroupBox::getFoxGroupBox()`

Note that you should be able to continue using

- `PICommon::getApplication()`
- `PICommon::getMainWindow()`
- `PIGUI::PIObject::getFoxMainWindow()`
- `PIGUI::PIObject::getFoxApplication()`

The corresponding functions to retrieve the toolbar and plug-in menu may also not work.

4.2 Distributed Plug-ins

4.2.1 Chat Display: `piChatDisplay`

Manages, imports, and displays message data in the SIMDIS overlay. It is useful for timestamped text: chat, operator notes, and high-interest events.

4.2.2 Custom Toolbar: `piCustomToolbar`

Allows you to customize the user interface of SIMDIS or Plot-XY. The **CustomToolbarSample.xml** is distributed with SIMDIS and is the replacement for the former Easy Mode plug-in.

- Hide standard SIMDIS/Plot-XY toolbars.
- Plot-XY uses the FOX based piCustomToolbar.
- You can specify button icons using 14 built-in icons or custom image files. If the custom image file cannot be read, a default error icon is shown instead and an error message is printed to the console.
- You can create one or more toolbars.
- You can create three different types of buttons:

  Action - Trigger a hot key action when pressed.

  Category Filter - Toggle display of entities based on category data when pressed

  DISCN Terrain Config - Load configuration files from either a **.asi** or **.discn**. It is useful for generating multiple display configurations via buttons that involve the use of GOG, view, preference rule and map **.earth** files. An example file is shown below. Note that only the GOGFile, ViewFile, RuleFile and ITConfigFile commands are processed.

```
# SIMDIS ASCII Scenario Input (ASI) File Format
# Scenario Initialization Keywords
Version 21
RefLLA 33.0267 -118.579 0.
CoordSystem "LLA"
ReferenceYear 2014
GOGFile "States.gog"
GOGFile "!SOCAL_Overlays.gog"
ViewFile "10_SAW_VIEWPORT.svml"
RuleFile "10_SAW_RULE.rul"
ITConfigFile "config-SAW-LOW.earth"
```

4.2.2.1 Create a Custom Toolbar

You need to generate an **XML** (.xml) file to create a custom toolbar. Here's an example to create a basic button that toggles hide/show the SIMDIS toolbar:

```
<customToolbar>
  <options>
    <hideStandardToolbars>1</hideStandardToolbars>
  </options>
  <toolbar>
    <title>My Toolbar</title>
    <button type="Action">
      <text>Toggle All</text>
      <tooltip>Toggle All Tool Bars</tooltip>
      <actionName>Toggle All Bars</actionName>
      <iconData>TreeExpand</iconData>
    </button>
  </toolbar>
</customToolbar>
```

NOTES:

- The <hideStandardToolbars>1</hideStandardToolbars> line is used to hide the toolbar on load, either through the Custom Toolbar plugin or passed as an argument when starting SIMDIS.
- The icon of the button is indicated by the <iconData> line.
- **CustomToolbarSample.xml** is distributed with SIMDIS in the demos/Plugins/CustomToolbar folder.

4.2.3 Data Client Server (DCS): piDCSLoader

The plug-in acts as a DCS client for SIMDIS or Plot-XY. It detects, connects, and processes data sent via a DCS server.

4.2.4 File Reader: piFileReader

Imports/Appends ASI files into SIMDIS or Plot-XY. It also supports **.csv**, **.tle**, **.otg**, **.wam**, **.pet**, **.gpx**, **.kp**, and **.zip** file formats.

4.2.5 File Writer: piFileWriter

Scenario files can be exported using this plug-in. The plug-in supports exporting to ASI files, ZIP files, and to a directory. Exporting to ASI or ZIP is available via the **Export** icon  on the main toolbar or through **File > Export**. Exporting to a directory is available via **File > Export Directory...** menu option.

4.2.5.1 ASI

To export as an ASI file, select the ASI extension in the Export dialog. The dialog will be similar to:

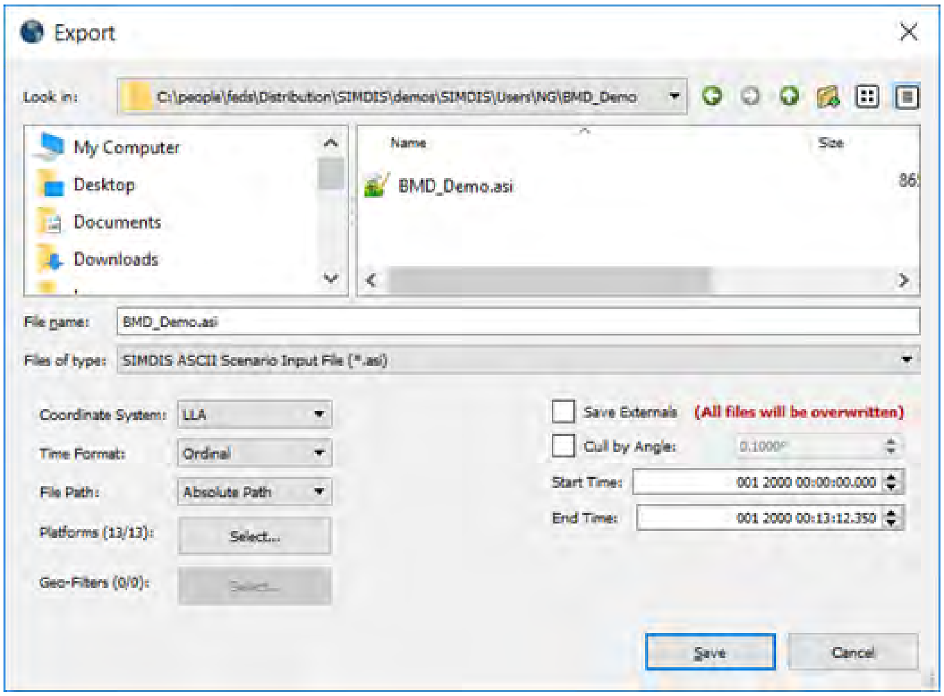


Figure 4.1: Export Scenario Dialog, ASI Selected

Fields include:

Coordinate System Coordinate system for the saved data.

Time Format Time format for saved data. The "Seconds" format exports and imports the fastest.

File Path Specifies the path, if any, for files referenced in the ASI file. The “Relative Path” and “Strip Path” are more portable between systems.

Platforms Display dialog to allow for platform selection. Selected platforms will be added to any platforms selected by Geo-Filtering.

Geo-Filters Export platforms with at least one point in the selected geo-filter(s). Only points in the geo-filter will be exported. Platforms passing the geo-filter(s) will be added to any manually selected platforms. For more information about geo-filters, see Section 5.22.

Save Externals Save modified external files like View file, Rule File, etc.

Cull By Angle Cull exported platform data using an angular tolerance between 0 and 180. Two composite angles are compared between 3 successive points. Angles less than the specified tolerance will cull the middle point.

Start Time The start time for the saved data.

End Time The end time for the saved data.

4.2.5.2 ZIP

To export as a ZIP file, select the ZIP extension in the Export dialog. The dialog will be similar to:

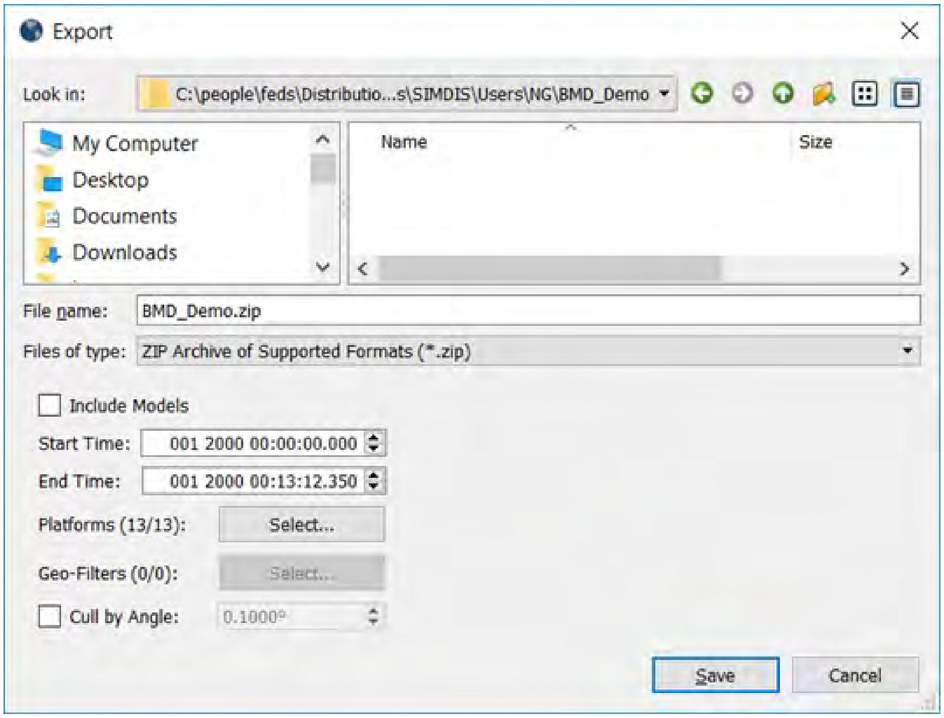


Figure 4.2: Export Scenario Dialog, ZIP Selected

The Start Time, End Time, Platforms, Geo-Filters and Cull by Angle fields have the same definitions as the ASI file. The icon models can optionally be included in the ZIP file.

4.2.5.3 Directory

To export as a directory, select the **File > Export Directory...** menu option or through the default shortcut Ctrl + Shift + E. This functionality allows saving all SIMDIS files related to the scenario (e.g. ASI, bookmarks, preference rules, and view files) in a single directory. The exported ASI file will share a name with the directory.

Using the **Re-Export Directory** action re-exports to the last directory without displaying a directory selection dialog. This action's default shortcut is Alt + Shift + E. If no last directory is set, the **Export Directory** dialog will display. Deleting the scenario clears the last directory and requires selecting a new directory the first time either action is triggered.

Exporting to a directory does not offer the same export options as ZIP export. When exporting to a directory, all platforms will be exported in the full scenario time range. Additionally, no models will be exported and the data will not be culled or geo-filtered.

4.2.6 Live Logging: `piFileWriter`

Live Logging functionality is included in the **File Writer** Plug-in. Live Logging writes all data received in Live Mode out to a file or series of files as it is received. This allows SIMDIS to capture all the data without having to first store all the data in memory. Supported file types include ASI, Zipped ASI, and HDF5. The File Writer Plug-in must be installed and active for Live Logging.

For the purposes of this section, the term Platform refers to a SIMDIS Platform entity as well as its children, Category Data, Generic Data, and Data Tables.

4.2.6.1 Dialog

The **Live Logging** dialog is found under the **Plug-ins** menu.

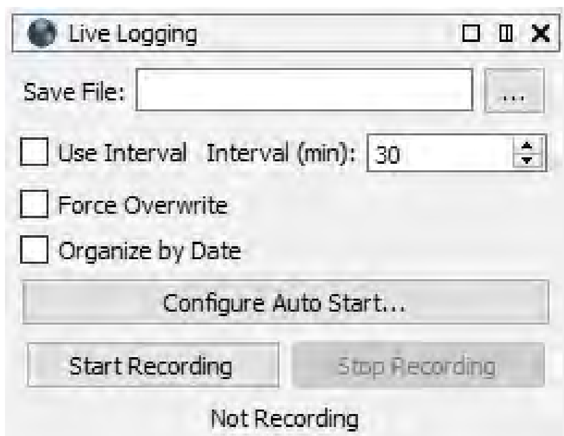


Figure 4.3: Live Logging Dialog

- **Save File:** The file name to use when writing the data. The supported file extensions are **ASI** (.asi), **Zipped ASI** (.zip), and **HDF5** (.hdf5). Filenames with a missing or unknown extension will default to ASI.
- **Use Interval:** If checked, Live Logging will divide the data into time-sequenced files based on the **Interval** value. When checked, a time stamp will be appended to the filename, using a format of `DDD_YYYY_HHMMSS`, where **D** is the ordinal date, **Y** is year, **H** is hours, **M** is minutes, and **S** is seconds. You can load multiple interval files into SIMDIS using the Append option of the File Reader Plug-in. For best performance load the files in chronological order.
- **Interval:** The time span (in minutes) to record before starting a new interval file. Pick a time span that balances manageable file sizes with a manageable file count.
- **Force Override:** If checked, Live Logging will overwrite any files with the same names. Logging will fail if this is not checked and a duplicate filename is encountered.
- **Organize by Date:** If checked, output files will be saved into a directory hierarchy based on timestamp in the ordinal format of **Year/Month/Day**.

- **Configure Auto Start:** Opens a dialog to set up the automatic starting and/or stopping of Live Logging. See the section below for more details.
- **Start Recording:** Starts a Live Logging recording. If no Start Time is specified, logging will begin when SIMDIS goes into Live Mode. If a Start Time is specified, then Live Logging will wait until Scenario Time passes the Start Time before starting recording.
- **Stop Recording:** Stops a Live Logging recording. Live Logging will automatically stop when the scenario transitions from Live Mode to File Mode or the specified Halt Time is reached.
- **Status Message:** Lists the current recording status, including error messages and the Time-To-Launch if Start Time is in the future.

4.2.6.2 Auto Start and Auto Halt

The **Live Logging Auto Start/Halt** dialog allows the user to configure when Live Logging should automatically start or stop recording.

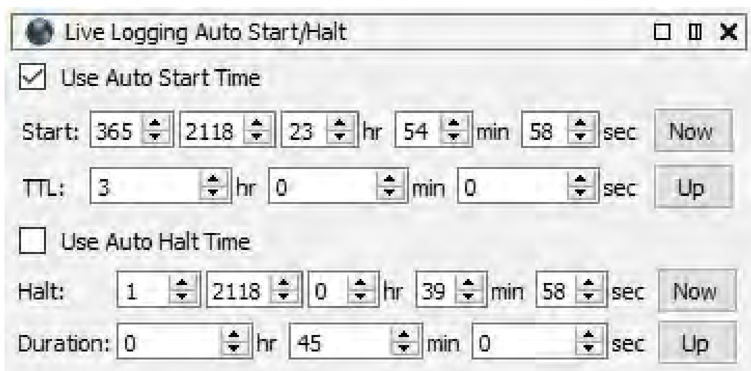


Figure 4.4: Live Logging Auto Start/Halt Dialog

- **Use Auto Start Time:** If checked, clicking **Start Recording** will schedule recording to begin at the Start Time set below.
- **Start:** The time at which to automatically begin recording. Listed as Day (Ordinal: 1-366), Year, Hours (0-23), Minutes (0-59), Seconds (0-59).
- **Now:** Sets the Start Time to the current Scenario Time.
- **TTL:** Time-To-Launch, as Hours (0-23), Minutes (0-59), Seconds (0-59) after the current Scenario Time.
- **Up:** Sets the Start Time to the current Scenario Time plus the TTL value.
- **Use Auto Halt Time:** If checked, clicking **Start Recording** will schedule recording to stop at the Halt Time set below.
- **Halt:** The time at which to automatically stop recording. Listed as Day (Ordinal: 1-366), Year, Hours (0-23), Minutes (0-59), Seconds (0-59).
- **Now:** Sets the Halt Time to the current Scenario Time.

- **Duration:** Duration to record for, as Hours (0-23), Minutes (0-59), Seconds (0-59) after the set Start Time.
- **Up:** Sets the Halt Time to the set Start Time plus the Duration value.

4.2.6.3 What Is Saved

All data received while in Live Mode is saved. When a recording is started, a snapshot of the Scenario information is saved to the file. This Scenario snap shot includes all the **Scenario Initialization** information listed in Appendix A. The expectation is that Scenario-level data is NOT changing during a recording. A new Scenario snap shot is taken at the start of each interval file.

NOTE: When appending a file most scenario data is ignored, with the exception of any Data Tables or Generic Data.

Live Logging in interval mode will create standalone, cohesive files. Active Platforms, Beams, and Gates are interpolated to the boundary between two interval files. This means active Platforms, Beams, and Gates will appear in SIMDIS at the start of an interval file if they have at least one data point both before and after the interval. LOBs, Lasers, and Custom Renderings Data Tables are not interpolated to the boundary between two interval files, which means LOBs, Lasers, and Custom Renderings will not appear in SIMDIS until there is data for them in the loaded interval file. Category Data and Generic Data are set to the state that existed at the start of the interval file.

Since data is written as it arrives, entities that are programmatically deleted will still show up in the recordings. It is possible to control which Platforms are recorded via Category Data. See the Category Data section for details.

Scenario time must increase; although Live Logging does tolerate data out of order by a few seconds, a large time jump backwards will cause Live Logging to stop recording and present an error message to the user.

4.2.6.4 Category Data

The list of Platforms saved to the data file(s) is controllable via Category Data. By default, all Platforms are saved to the data file.

Use the Category Tag “Live Logging" to override the default behavior. The corresponding Category Value defaults to “Active," and must be changed immediately after creating the Platform if alternate behavior is desired. Setting the value to “Ignore" will cause the Platform data to not be written to the data file. Setting the value to “Pending" gives the data source a few seconds before picking “Active” or “Ignore". All the Platform data and its children data are buffered while the value is “Pending". Once a value of “Active" or “Ignore" is set, the value cannot be changed. The Category Tag “Live Logging" is not necessary for the children of a Platform and will be ignored.

4.2.6.5 Program Interface

Live Logging is controllable via the SIMDIS Plug-in API. See the Plug-in API documentation for the methods `PICommon::startRecording` and `PICommon::stopRecording`. There is only one Live Logging session; if multiple plug-ins attempt to control Live Logging and/or the user uses the Live Logging dialog, whichever command is received last takes precedence.

4.2.7 Hierarchical Data Format (HDF-5) Import/Export: (piHDF-5)

The plug-in supports both import and export of HDF5 files and HDFZ files. The Hierarchical Data Format (hdf-5) is an efficient storage and retrieval format for high-volume data. The files are usually smaller and usually export and import faster than ASI. Append is supported for both file formats.

4.2.7.1 HDF5

To export as an **HDF5** file, select the HDF5 extension in the Export dialog, shown as Figure 4.5.

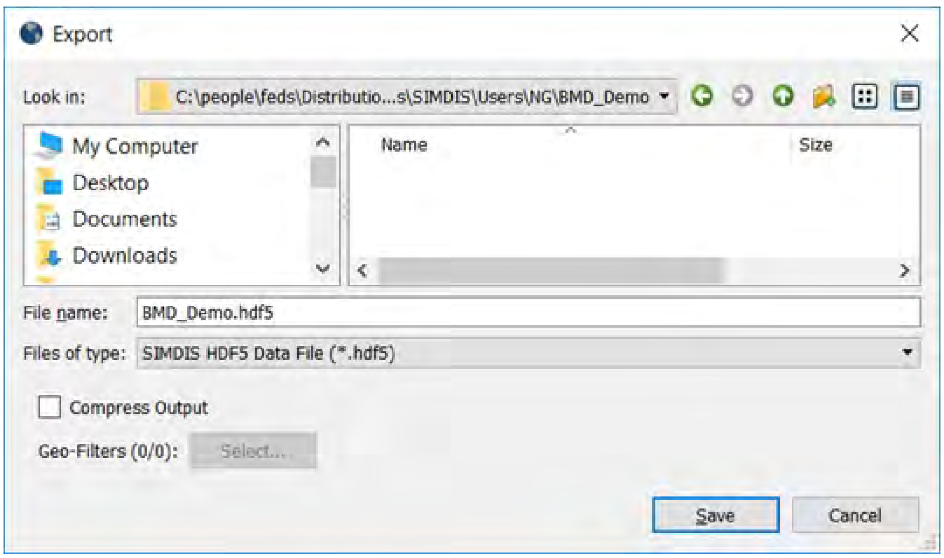


Figure 4.5: Export Scenario Dialog, HDF5 Selected

The file can optionally be compressed. Compression will reduce the file size, but increase export and import times. The Geo-Filters field has the same definitions as the ASI file.

4.2.7.2 HDFZ

The **HDFZ** format is described in Section 7.1.12. To export as an HDFZ file, select the HDFZ extension in the Export dialog. The dialog will be similar to:

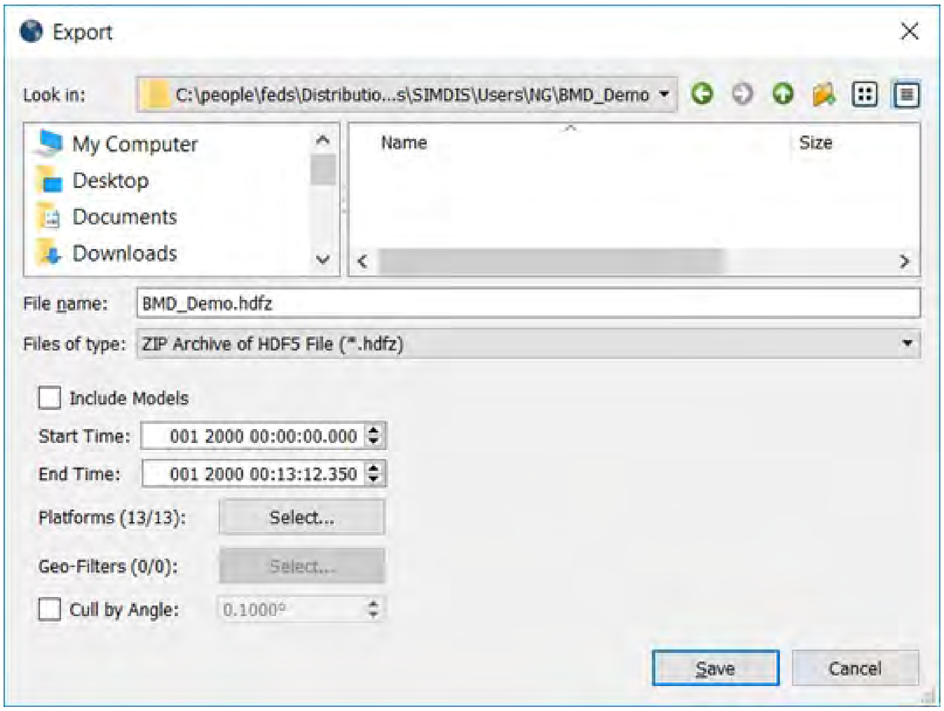


Figure 4.6: Export Scenario Dialog, HDF5 Selected

The Start Time, End Time, Platforms, Geo-Filters and Cull by Angle fields have the same definitions as the ASI file. The icon models can optionally be included in the HDFZ file.

4.2.8 Media Sync: `piMediaSync`

Automatically finds and synchronizes audio, images, video, and bookmark data files into SIMDIS.

4.2.9 Scenario Editor: `piScenarioEditor`

Provides file mode scenario editing capabilities through interactive right-click menus in the SIMDIS Data Browser and Prefs Tool and through a separate plug-in GUI.

4.2.9.1 Right-click Menus

When using Data Browser or Prefs Tool, the plug-in will add editing options to the right-click context menu when right-clicking on an entity or a group of entities. Option visibility depends on the entity selection when right-clicking.

The following are ways in which a user can edit entities via right-click menus in SIMDIS Data Browser and Prefs Tool:

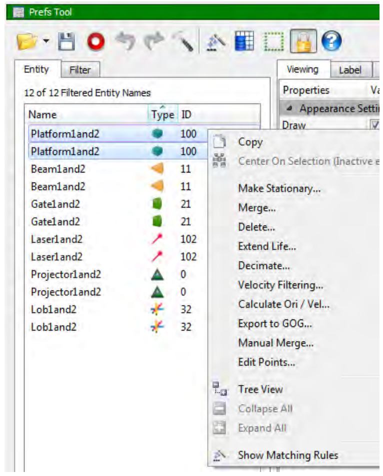


Figure 4.7: Scenario Editor Right-Click Context Menu

- **Calculate Ori / Vel:** Calculates orientation and/or velocity values for the selected platforms. Provides two calculation options. See the pop-up dialog for this option for more information.
- **Decimate:** Creates a new platform for each selected platform, with the number of TSPI data points reduced using one of several methods described in paragraph 4.2.9.1.1.
- **Delete:** Deletes the selected entity or entities (and their children) from the scenario.
- **Export to GOG:** Saves out a GOG created using the TSPI points of the selected platform(s). See paragraph 4.2.9.1.3.
- **Extend Life:** Duplicates the last TSPI data point and sets the time to the end of the scenario for each selected platform.
- **Make Stationary:** Makes the first TSPI data point static for all selected platforms, removing all later data points.
- **Merge:** Merges the selected platforms into a single entity with all the shared TSPI points. Only visible when two or more platforms with the same name are selected.
- **Velocity Filtering:** Filters out "wild" TSPI points for the selected platform(s) by removing points with velocities over the entered value. See paragraph 4.2.9.1.2.

The following are additional edits via right-click menu available in the Scenario Dialog:

- **Add Entity:** Adds an entity of any type to the scenario.
- **Copy Entity:** Copies the entities and any children of the entities to the global clipboard. Copy, cut and paste also supported by Drag and Drop.
- **Delete Scenario:** Deletes the scenario and starts a new empty scenario.
- **Paste Entity:** Pastes the entities from the global clipboard. Copy, cut and paste also supported by Drag and Drop.

- **Remove Duplicate Points:** Removes duplicate points from the selected entities. Picking the option when the Scenario is selected will remove duplicate points from all entities.
- **Transform Platform:** Allows for linear transformation of any platform field.

4.2.9.1.1 Decimating Platforms

There are several means of decimating SIMDIS platforms:

- **By Point:** Removes (n-1) out of every (n) data points.
- **By Time:** Removes points to keep data at or over a minimum data rate.
- **By Angle:** Removes points using an angular tolerance > 0 and < 180 degrees. Two composite angles are compared between 3 successive points. Angles less than the specified tolerance will cull the middle point.
- **By Distance:** Removes successive TSPI points until the distance traveled between two TSPI points is at least the specified distance.
- **By Area:** Reduces points by a percentage, implementing the Visvalingam line simplification algorithm.

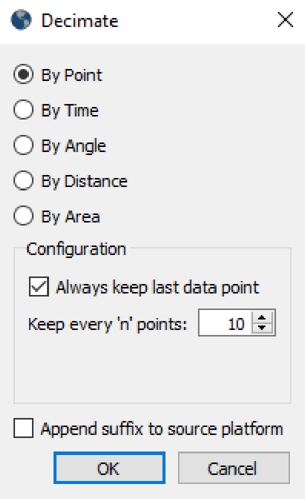


Figure 4.8: Scenario Editor Menu, Decimate Dialog

The "Configuration" box in the Decimate dialog (Figure 4.8) will change based on the selected method, to allow proper configuration.

The newly-created entity will have a suffix appended to its name based on the chosen decimation algorithm. If "Append suffix to source platform" is selected, the original entity will have "_orig" appended to its name instead.

4.2.9.1.2 Filtering Entity Velocities

The "Velocity Filtering" option can be used to remove TSPI data points with anomalous values by comparing the instantaneous velocity information with a set trigger value.

If instructed to use calculated velocity, SIMDIS will instead calculate the velocity required to travel between TSPI points and remove any that would result in a velocity above the configured limit.

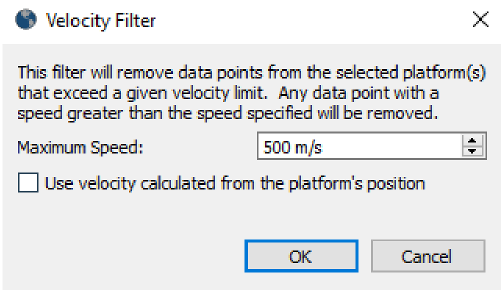


Figure 4.9: Scenario Editor Menu, Velocity Dialog

4.2.9.1.3 Export Entity to GOG

The "Export to GOG" option can be used to create a GOG based on the TSPI points of the selected platform(s). There are several options available to change the properties of the resulting GOG:

- **GOG Filename:** Filename of the resulting GOG.
- **Plot Style:** Determines how the TSPI points are converted to GOG points. For a detailed description of each option see below.
- **Line Style:** Line style of the resulting GOG.
- **Line Width:** Line width of the resulting GOG.
- **Default Color:** Color to use when not using track color.
- **X Offset:** Horizontal offset to add to each point. For LL and LLA plot styles, this is added in degrees. Altitude vs rnage, this is added in meters. For all other plot styles, this is added in seconds.
- **Use Track Color:** Toggle whether to get GOG color from the selected platform(s).
- **Use Depth Buffer:** Toggle the use of the depth buffer for the saved GOG.

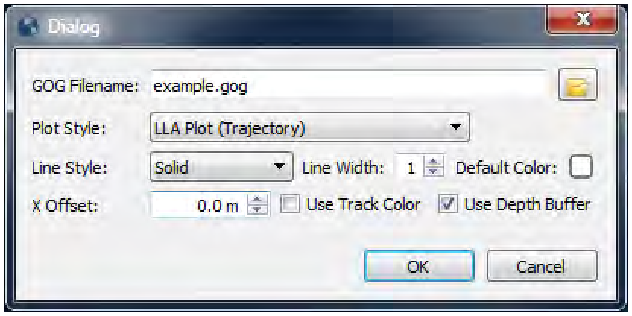


Figure 4.10: Scenario Editor Menu, GOG Dialog

The Plot Style used when exporting the GOG determines how the TSPI points are converted to GOG points. The options are:

- **LL Plot (Ground Range):** Assign the position of each TSPI point to an LL GOG point. Altitude is unused.

- **LLA Plot (Trajectory):** Assign the position of each TSPI point to an LLA GOG point including the altitude.
- **Altitude vs Time:** Each TSPI point is converted to an xy GOG point where the x is the time in seconds since ref year and y is the altitude in meters.
- **Altitude vs Range from Scenario Origin:**Each TSPI point is converted to an xy GOG point where the x is the distance from the scenario origin in meters and the y is the altitude in meters.
- **Velocity vs Time:** Each TSPI point is converted to an xy GOG point where the x is the time in seconds since ref year and y is the total velocity in m/s.

4.2.9.2 Plug-in GUI

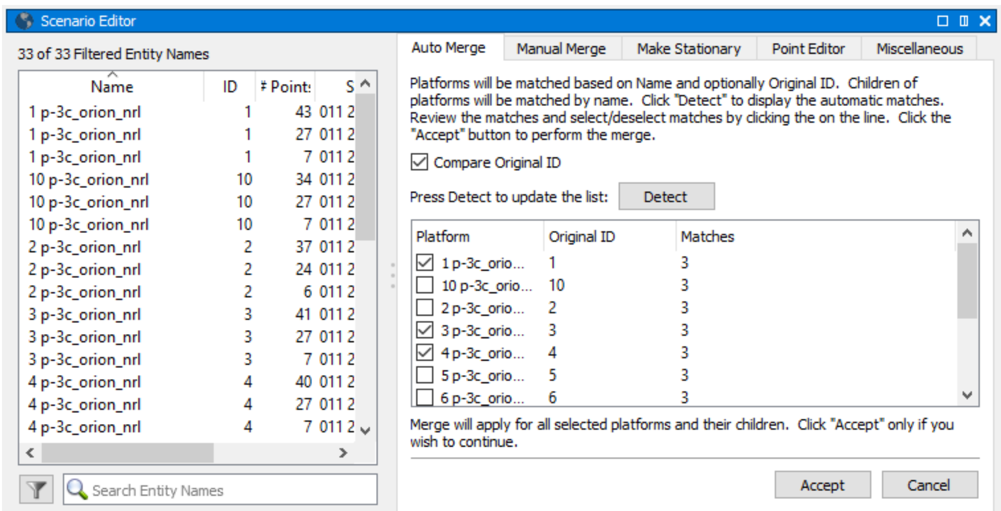


Figure 4.11: Scenario Editor GUI

The plug-in GUI offers more in-depth editing capabilities than the right-click menu options. Each tab in the GUI is briefly documented in the following sections.

4.2.9.2.1 Auto Merge

Automatically merges identically named platforms and their identically named children.

4.2.9.2.2 Manual Merge

Manually merge platforms into a new permanent platform to fill in gaps of missing data in a platform. Select and arrange a priority list of platforms to merge. Includes a customizable data dropout gap duration.

4.2.9.2.3 Make Stationary

Detect and make platforms stationary. Includes options for modifying how platforms should be detected.

4.2.9.2.4 Point Editor

Edit, add, and remove an entity's individual data points, Category Data, Generic Data and Data Tables.

4.2.9.2.5 Miscellaneous

Provides miscellaneous functionality to the plug-in. The following features are available:

- **Edit Scenario:** Allows editing of most scenario properties.
- **Edit Scenario Times:** Allows editing of the scenario reference year and start times. Shifting the scenario start time will shift the times for all the data in the scenario.
- **Units:** Allows changing the units used by the plug-in.
- **Validate Scenario:** Allows validating and fixing problems with the scenario (such as out of range data).

4.2.10 Remote API Server: `piRemoteApiServer`

Allows external programs to communicate with the Plug-in API. The external program can add data and retrieve data. Not all Plug-in API methods are available to the external program, but methods are added as time permits. The documentation for using the plug-in can be found at: `$(SIMDIS_DIR)/doc/SIMDIS/Remote_API.pdf`. Example Python code can be found at: `$(SIMDIS_DIR)/demos/RemoteAPI/Python`.

4.2.11 Simple Server: `piSimpleServer`

Demonstrates inserting both live and file-based data into SIMDIS and/or Plot-XY. The source code is provided in SIMDIS Plug-in API.

4.2.12 Snapshot Tool: `piSnapShot`

Take screenshots of the application window. In Plot-XY, will also capture all currently open tabs.

4.2.13 Time Client/Server (TCS): `piTCSPlugin`

Broadcast or receive time from a network address and port.

4.2.14 Video Recorder: `piVideoRecorder`

Records video from SIMDIS or Plot-XY to standard media formats; however, it does not support audio recording.

4.3 Python Plug-ins

Python Plug-ins use the embedded Python interpreter to allow writing Python code to interact with SIMDIS using the SIMDIS Plug-in API. Python Plug-ins are supported in SIMDIS 10 and Plot-XY Qt (Beta). They are not supported in the legacy FOX-based Plot-XY.

There are a number of advantages and disadvantages of Python Plug-ins when compared to regular C++ plug-ins:

Advantages

- Fast prototyping
- Reload plug-in without restarting SIMDIS
- Full access to Plug-in API, including PIGUI
- Cross-platform
- No waiting on compiling and building

Disadvantages

- Verbose callback implementations
- Sometimes harder to debug syntax errors
- CPU-heavy tasks might take longer

4.3.1 Getting Started

Below is a minimal Python Plug-in example displaying the minimum code required for a functioning Python Plug-in.

```
# Modified from Minimal.py (distributed with Plug-in API).

# Class definition, serves as PIApplication. Name *must*
# match 'class' property in call to registerPlugin().
Class Minimal():
    def StartUp(self, args): # maps to PIApplication::StartUp()
        return 0

registerPlugin({
    'class': 'Minimal' # Required. Value must match name of class above
})
```

The first step when writing a Python Plug-in is defining a class that serves as your `PIApplication`. Your `PIApplication` class has access to all methods available to C++ classes that inherit `PIApplication`, including `StartUp()`, `Close()`, and `Callback()`. For more information, the distributed example Python Plug-in `AllAppCallbacks.py`, which is distributed with the Plug-in API.

The next step is calling `registerPlugin()`, which takes a dictionary argument. The only required dictionary entry is `'class'` with a value matching the name of the `PIApplication` class. Other optional entries that are similar to the metadata from the C++ API's metadata functions like Name() and Author() are described below:

`'name'`	Name of the plug-in. If not specified, defaults to `PIApplication` class name.
`'description'`	Description of the plug-in displayed in Plug-in Manager.
`'author'`	Plug-in author information.
`'version'`	Version of the plug-in.
`'url'`	URL associated with the plug-in, displayed in Plug-in Manager.
`'contactInfo'`	Contact information for the plug-in author.
`'loadInSIMDIS10'`	Whether the plug-in is allowed to load in SIMDIS 10. 1 for true, 0 for false.
`'loadInPlotXY10'`	Whether the plug-in is allowed to load in Plot-XY Qt (Beta). 1 for true, 0 for false.

Python Plug-ins can be loaded like regular plug-ins via the command line (`-plugin`), registered (place file in `SIMDIS/plugins`), or through the Plug-in Manager GUI.

4.3.2 Advanced Details

The following are things to keep in mind when working with Python Plug-ins.

- PIApplication-based methods have the same return values as their C++ PIApplication counterparts. If nothing is returned, success is presumed.
- Various callbacks require correct return values, or SWIG complains in a verbose but unhelpful way. This is a common error.
- Not all routines from `PluginClient/Samples/Interfaces` are available. See `PluginClient/swig/Testing/TestPiInterfaces.py` for examples.
- Python interface is tied to the version of Python distributed with SIMDIS. Function signatures are subject to change as SIMDIS Plug-in API changes, so backwards compatibility is not guaranteed as it is with C++ Plug-ins.
- PIGUI is supported directly, but Qt is not.

4.3.3 Examples

A number of Python Plug-ins are distributed with the Plug-in API. `PluginClient/swig/Testing` contains sample Python files that test all Plug-in API methods, including all function signature variations. These are useful references for determining correct syntax. `PluginClient/Samples/Python` contains Python Plug-ins that perform various functions, including a Python example of the Simple Server sample plug-in. Other sample Python Plug-ins can be found in `SIMDIS/plugins` as part of the full SIMDIS distribution.

Chapter 5

Tools

The functionality in SIMDIS is abundant. One way to explore the various functions is to examine the tools. Each tool has different options and tasks. The following section explains each of the tools in order to enhance your experience with SIMDIS.

5.1 Viewports and Eye Positions

The Views Dialog has been replaced in SIMDIS by two separate dialogs: the **Viewports Dialog** and the **Eye Positions Dialog**.

5.1.1 Viewports Dialog

The Viewports Dialog is accessed via the **View** menu. It allows the user to create, delete, enable (make visible), and disable (hide) viewports in the SIMDIS window.

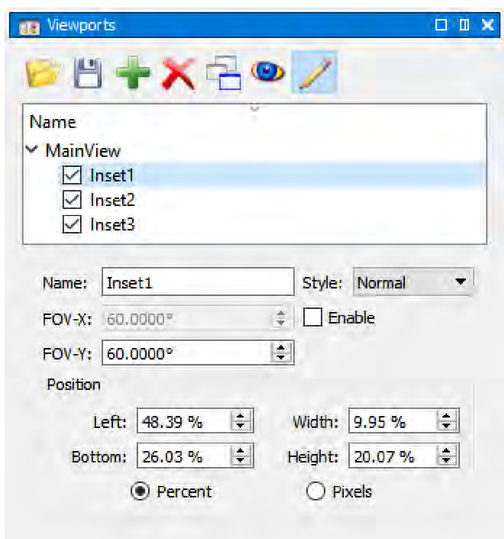


Figure 5.1: Viewports Dialog

Name	Name of the viewport, cannot be changed for the main view.
Style	Select from **Normal**, **Upper-Left**, **Upper-Right**, **Lower-Left**, and **Lower-Right**. Options other than Normal will lock the viewport to that corner of the SIMDIS window and prevent manual positioning.
FOV-X	If enabled, manually adjust the horizontal field of view (FOV) for the selected viewport. See Figure 5.2.
FOV-Y	Manually adjust the vertical field of view (FOV) for the selected viewport.
Left	Manually position the left side of the selected viewport (offset from the left side of the SIMDIS window). Cannot be changed for the main view or for viewports with an assigned style.
Width	Adjust the width of the selected viewport.
Bottom	Manually position the bottom side of the selected viewport (offset from the bottom of the SIMDIS window). Cannot be changed for the main view or for viewports with an assigned style.
Height	Adjust the height of the selected viewport.
Percent	When selected, all Position values will be listed as a percentage of the SIMDIS window size.
Pixels	When selected, all Position values will be listed as absolute pixel sizes.

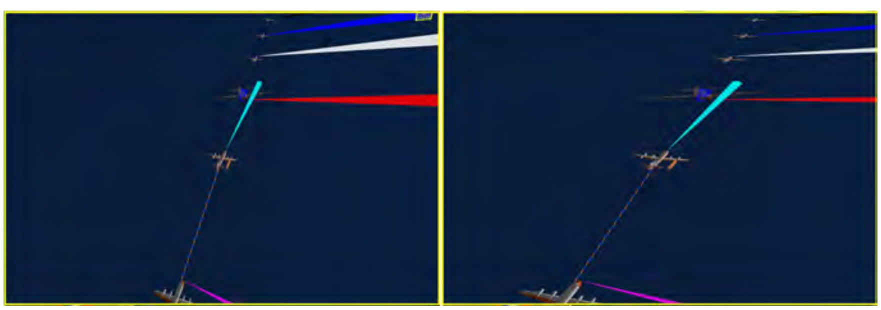


Figure 5.2: Default FOV-X vs. FOV-X of 45 degrees

NOTE: You can swap the camera positions of the main view and the active viewport via the hot key **Shift** + **V**.

5.1.2 Eye Positions Dialog

The Eye Positions Dialog is accessed via the **Eye** menu. It allows the user to view and edit details about the current and saved eye positions as shown in Figure 5.3.

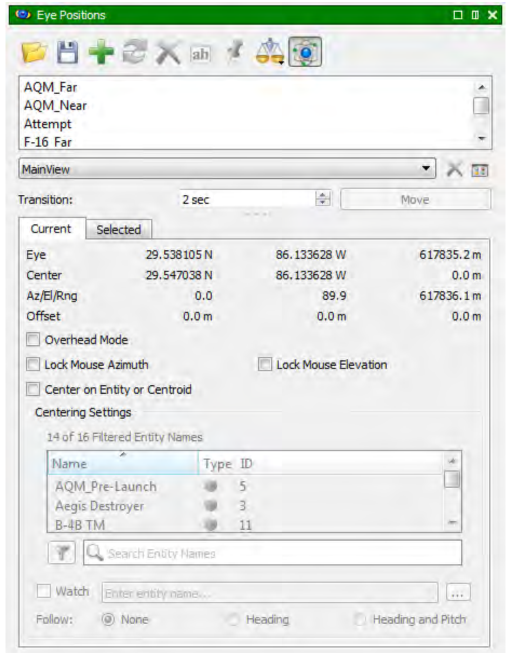


Figure 5.3: Eye Positions Dialog

You can move a viewport's eye to a saved eye position by selecting that viewport from the **Viewports** drop down, then either double clicking the desired eye position from the list or selecting it and pressing the **Move** button. Double clicking an eye position will move to it instantly, while the move button will transition smoothly over a period of seconds set in the **Transition** field.

The bottom section of the dialog presents a group of details and options for eye positions. The **Current** and **Selected** tabs switch this group between showing details about the current focused eye position and the selected eye position. This group contains a number of options that can be edited for the **Current** eye position:

Overhead Mode	Enable overhead mode for the eye position.
Lock Mouse Azimuth	Prevent the mouse from altering the eye position's azimuth.
Lock Mouse Elevation	Prevent the mouse from altering the eye position's elevation.
Center on Entity or Centroid	Lock the eye to the position of the selected entity or centroid of multiple selected entities. Enabling this option will enable the **Centering Settings**, allowing you to select the entity or entities to center on, set a watched entity, and set the eye position to follow the entity's heading and pitch in addition to position.
Watch	Orient the eye position to face a certain entity. This option is available when the eye position is centered on an entity or centroid.

Follow	Orient the eye position to match the selected attribute(s) of the centered entity. This option is not available when centered on a centroid or watching an entity.

5.2 Prefs Tool

The **Prefs Tool** provides management of the customizable preferences for all entities as well as preference rules. Preferences control various display attributes for each entity. SIMDIS preference rules are a means for defining preference values that SIMDIS will enforce for a specified class of entities. Preference rules can be saved and loaded using the file extension **.rul**.

You can manually edit individual **Preference Rules** by clicking the **Preference Rules** icon on the Prefs Tool toolbar. Preference Rules are automatically created and maintained by using the **Record** button on the toolbar.

For the list of supported Preference Rules visit the Supported Preference Rules FAQ.

There are multiple ways to access Prefs Tool:

1. The **Prefs Tool** button in the toolbar.
2. **Prefs Tool** menu item under the **Tools** menu bar.
3. The **Prefs Tool** button in **Data Browser** toolbar.
4. Hotkey Alt + S

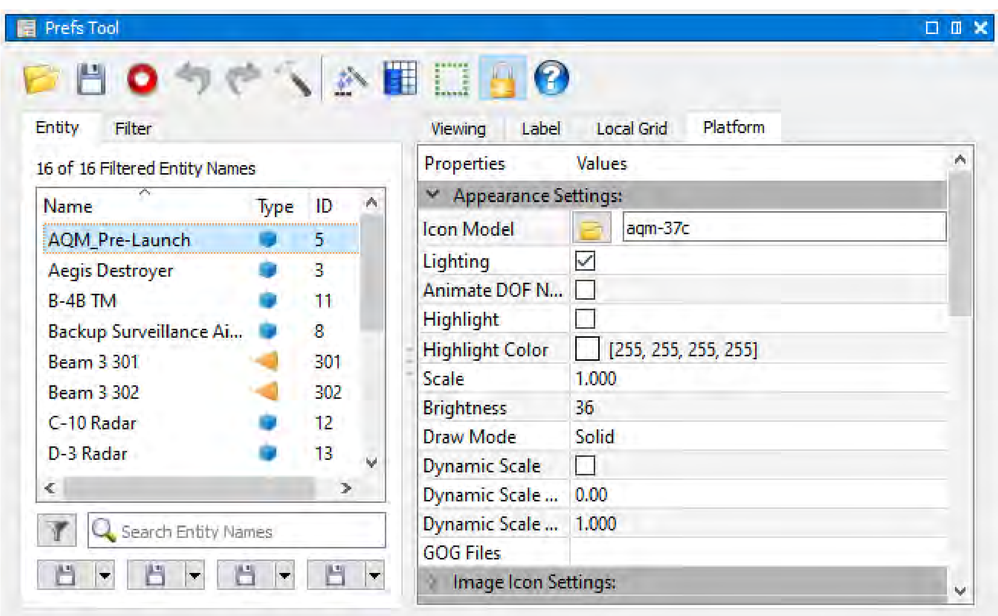


Figure 5.4: Prefs Tool Dialog Display Layout

When an entity is selected from the Entity List, the appropriate tabs specific to the entity selected will be enabled and a user can modify the fields that are in the selected tab. The **Platforms**, **Beam**, and **Gate** tabs contain fields that are used to display and modify their own specific settings.

If an entity is configured to display its alias in place of its name but it has no alias set, then its name will be grayed out in the Entity List.

While the **Fence Select** button is checked, the user can click and drag a box in the SIMDIS 3D display. Any platforms in the box on mouse release will be selected in Prefs Tool.

NOTE: Double-clicking on an entity in the list will center the SIMDIS view about that entity.

- **Viewing** tab - contains fields that are used to display and modify general viewing attributes of a selected entity.
- **Label** tab - contains fields that are used to display and modify various label-related settings of any selected entity(s).
- **Local Grid** tab - contains fields that are used to display and modify various attribute settings of any selected entity's local grid.

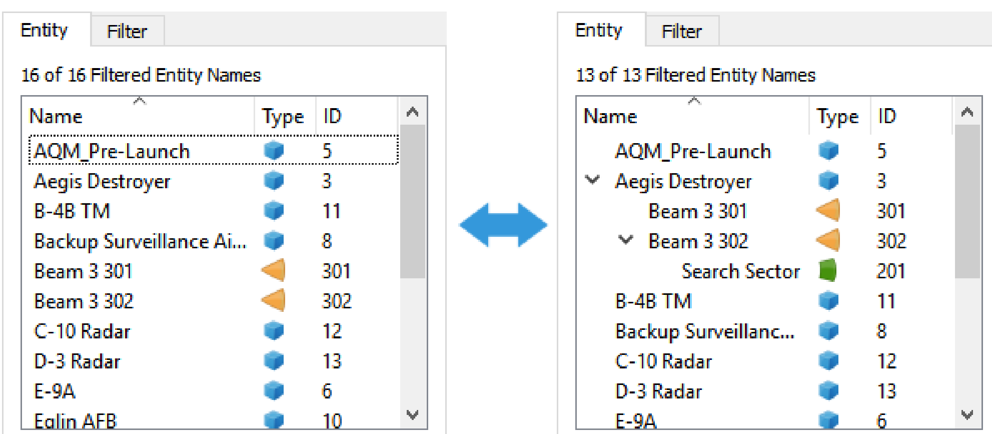


Figure 5.5: Prefs Tool Tree View Toggle

The **Tree View Toggle** button allows the operator to switch between a hierarchal view where entities that are associated are grouped together versus an alphabetical list and no hierarchy. See Figure 5.5.

The **Entity Filters** button pops up a window which allows the operator to filter the list (and subsequently the display) by entity types and category data. See Figure 5.6.

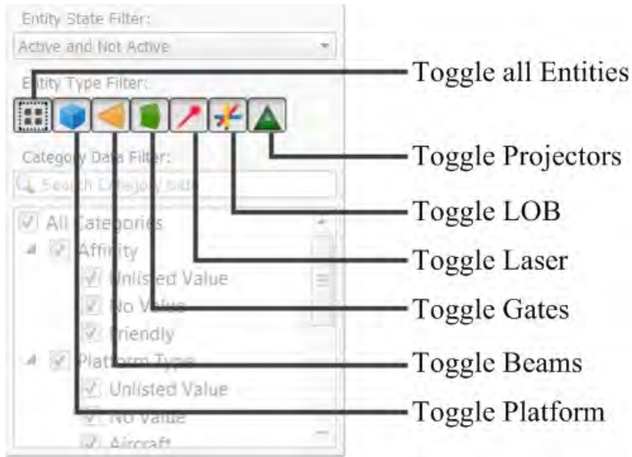


Figure 5.6: Entity Filters

5.2.1 Creating Preference Rules

The **Prefs Tool** not only allows you to edit preferences for selected entities or for filtered groups of entities, but also automates the creation of **Preference Rules**. **Preference Rules** tell SIMDIS how to apply preferences automatically to entities based on their entity type, name, and category values. This powerful feature allows for the automation of display configuration based on your expectation of what entities and category values you will see in the scenario. This is particularly useful for live mode scenario data.

Preference Rules are typically saved in a Rule file, with the extension **.rul**. They can be associated with ASI files using the `RuleFile` optional keyword. For more details, refer to Appendix A.

Rules are created by toggling on the **Record** button. When on, the red **RECORDING** banner is visible. Any preference changed in this mode will append a new rule. Newly created rules are automatically compressed, preventing accidental creation of duplicate rules. The type of rule created depends on the state of the **Entity and Filter** tabs on the left of the window:

- **Entity** tab - Rules created with this tab active will only match on name. All filter settings are ignored, and the rule is generated with a regular expression that matches each selected entity exactly.
- **Filter** tab - Rules created with this tab active will match the entity type, category value, and name expression selected in the tab. This selection may match one, many, or no entities in the scenario, depending on your data and your selections.

When the **Prefer Manual Edits** button on the Prefs Tool toolbar is checked, manual edits to entity preferences will not be lost by automatic application of preference rules. Any force applied preference rules will still override manual edits.

5.2.2 Preference Rules Editor

The **Preference Rules Editor** allows you to organize and modify preference rules. It is accessible by toggling the **Preference Rules** button on the Prefs Tool Toolbar.

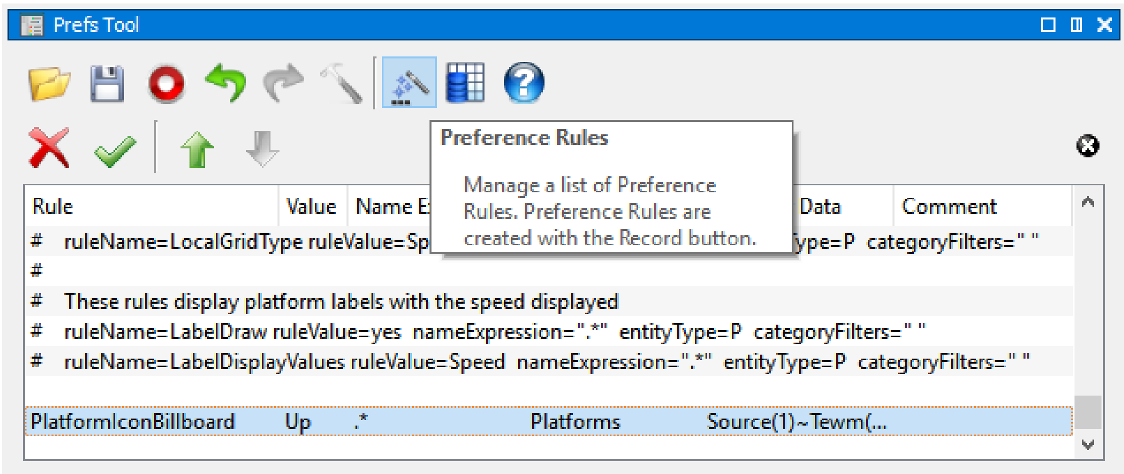


Figure 5.7: Preference Rules Editor

For examples, refer to Section 5.2.6 and Section 5.2.7. Refer to Figure 5.10 for the details about Preference Rules Editor Context (right-click) menu.

The editor allows editing rule values in-place. To do so, double click on a row's value in the `Value` column and enter a new value.

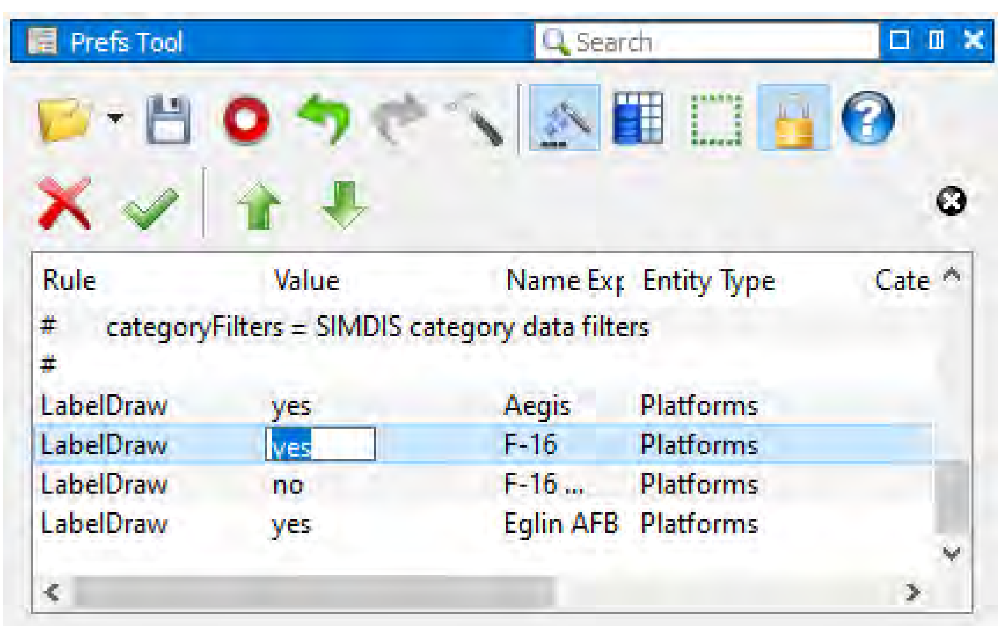


Figure 5.8: Editing Rules In-Place

5.2.3 Context Menu

The **Context Menu** (right-click menu) allows quick operations within the Prefs Tool Dialog.

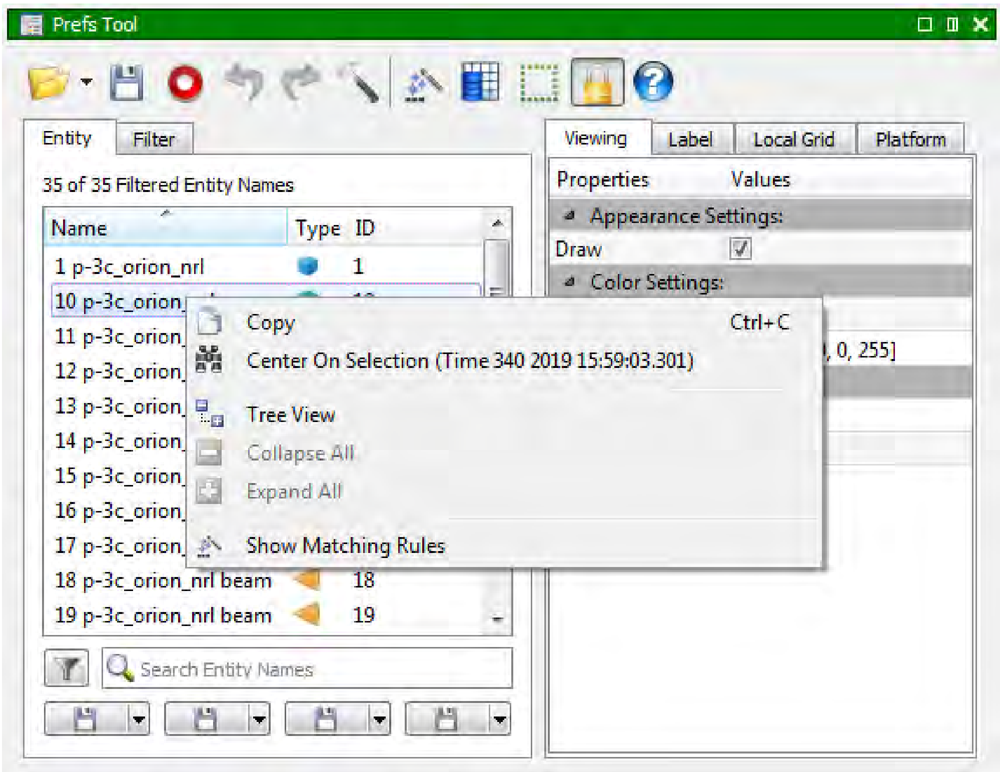


Figure 5.9: Entity Tab Context Menu

Copy	Copy the name of the selected entity.
Center On Entity	Center the Eye Position to the selected entity.
Tree View	Toggle the Entity list to Tree View. See Figure 5.5.
Collapse All	Collapse all the child entities in the Entity list (Only available in Tree View).
Expand All	Expand all child entities in the Entity list (Only available in Tree View).
Show Matching Rules	Open the pref rules list and show only the rules that match the selected entity.

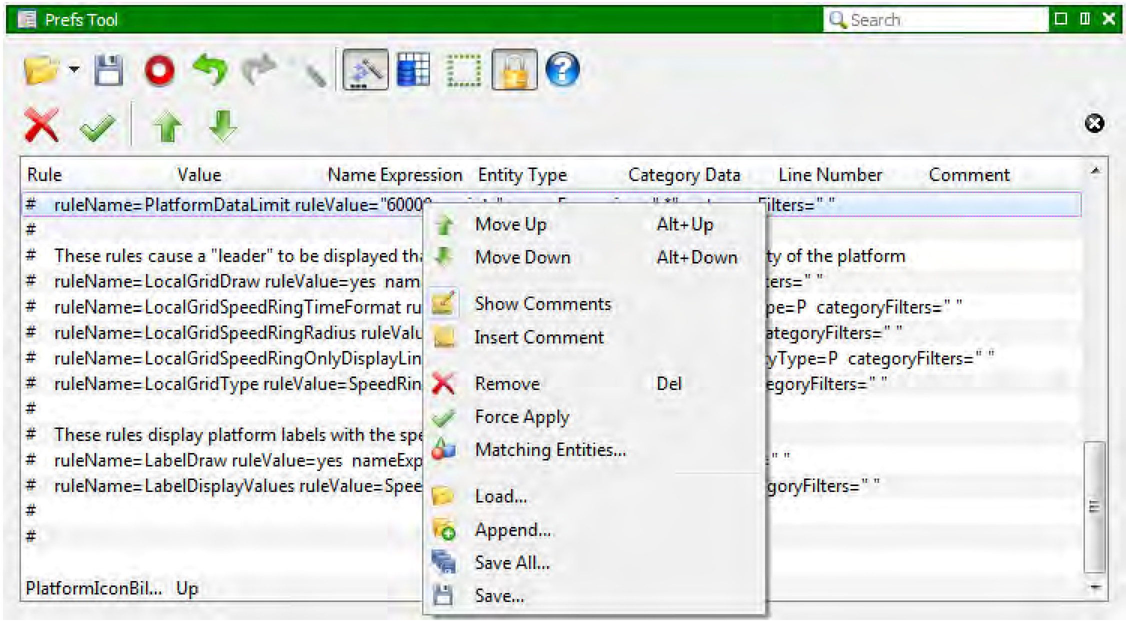


Figure 5.10: Preference Rules Context Menu

	Move Up	Move the selected rule up in the list.
	Move Down	Move the selected rule down in the list.
	Show Comments	Toggle to display the comments in the rules list.
	Insert Comment	Insert a single line comment. Note that the new comment is inserted on top of the selected line.
	Remove	Remove selected lines of comment and/or preference rules.
	Force Apply	Forces the application of the rule, overriding manually set preferences.
	Matching Entities	Show a list of entities affected by the selected rule.
	Load	Load a preference rules (**.rul**) file. Note that it will overwrite existing preference rules in the list.
	Append	Append a set of preference rules from a preference rules (**.rul**) file to the list.
	Save All	Save all the content of the preference rules list to a preference rule (**.rul**) file.
	Save	Save the selected preference rule(s) to a preference rule (**.rul**) file.

5.2.4 Entity Types

By default, the **All**  button in **Entity Type** bar is toggled. When you create a preference rule, it will affect all entities. To deselect all entity types, simply toggle the **All** button. The highlight for all Entity Type icons will disappear, see Figure 5.11.

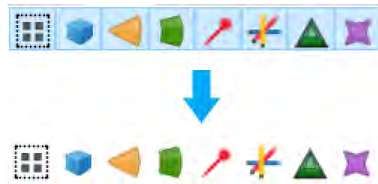


Figure 5.11: Toggle Off All Entity Types

Refer to Section 3.11.6 for more information about the **Entity Types**.

5.2.5 Category Filter

The **Category Filter** allows you to filter entities based on their category type, shown in Figure 5.12.

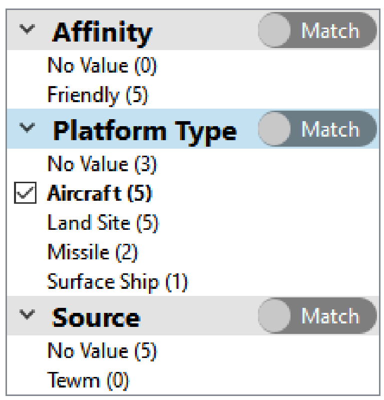


Figure 5.12: Category Filter

You can **Match** or **Exclude** category values using the slide toggle with their associated category type, shown in Figure 5.13. This feature allows flexibility when applying preference rules to specific category types.

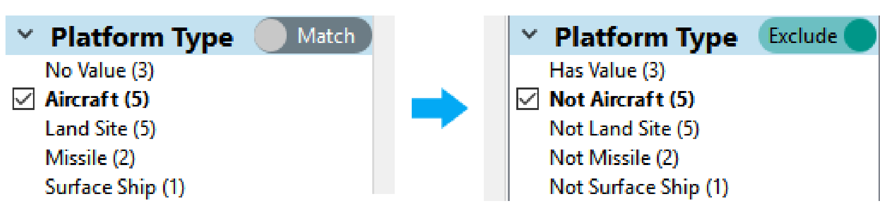


Figure 5.13: Matching and Excluding Category Values

TIPS: There are a few ways to manage category types in the filtered list:

- Use the **Search Category Data** search box to quickly filter the category types and/or values in the list.
- Use the right-click menu to **Collapse Values**, shown in Figure 5.14.

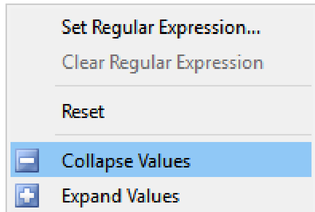


Figure 5.14: Category Filter Right-Click Menu

- Set **Regular Expression** to filter values of the category type. Click the button, located on the right of **Entity Name Filter** textbox, for more information about regular expressions. Section 5.2.9 is an example to guide you in using the regular expression to filter category values from a category type. You can also refer to Appendix G for detailed use of regex to filter entities.

When the category value is active, a checked-box will be shown to the left of the value and a breadcrumb will be shown above the Preference Tabs. This will aid you in seeing which category type values are selected. The **breadcrumbs** are displayed as an alternating shades of blue to easily discern between different category types, shown in Figure 5.15.

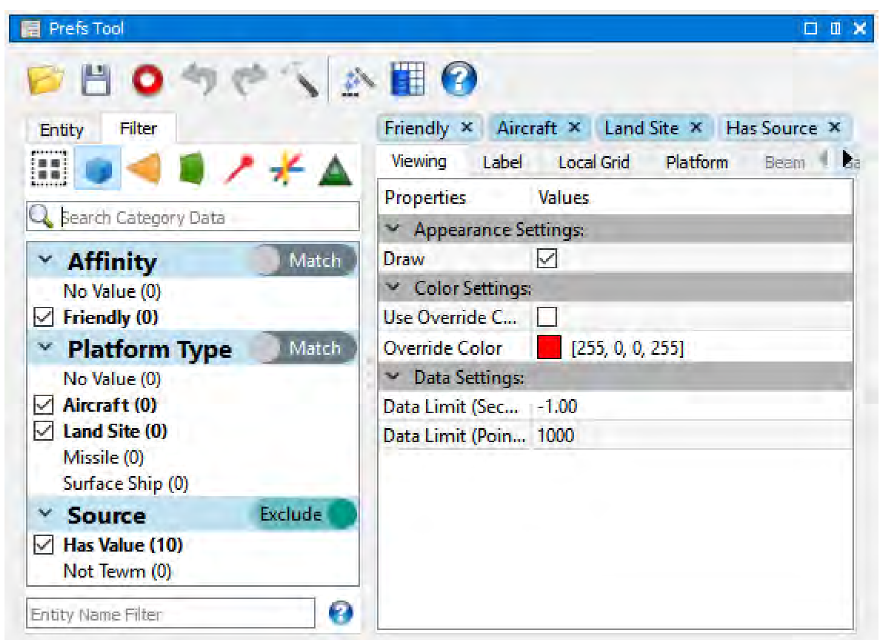


Figure 5.15: Active Category Type Values as Breadcrumbs

NOTE: Active category type value breadcrumbs are also displayed in **Data Browser**.

5.2.6 Platforms Preference Rules Example

For the following example, we will add preference rules to all platform's track history using **Track Draw Mode** and **Track Length**:

1. Click the **Filter** tab.
2. Deselect all entities by toggling the **All** button.
3. Toggle the **Platforms** icon.
4. Toggle the **Record** button. Make sure that the red **RECORDING** banner is visible.
5. Click the **Platform** tab and scroll all the way down to **Track History Settings:**
6. Change the **Track Draw Mode** to **Line**.
7. Change the **Track Length** value to **240**. The value represents a platform's history trail in seconds. See Figure 5.16 for reference.

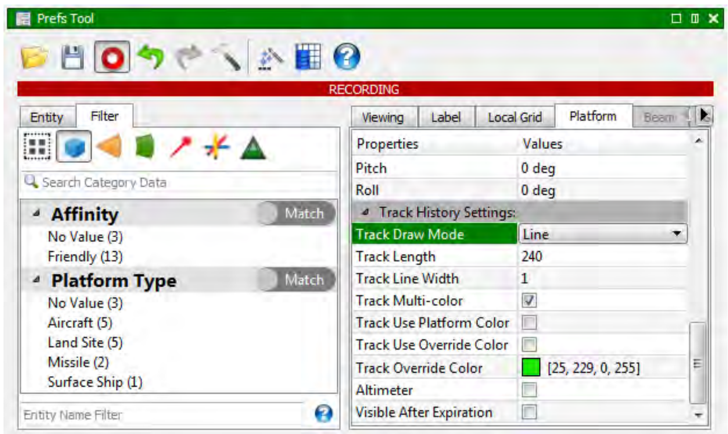


Figure 5.16: Preference Rules for Platform Track Line

8. Toggle the **Record** button. Make sure that the red **RECORDING** banner disappears.
9. Toggle the icon to display the preference rules list and verify added preference rules. See Figure 5.17 for reference.

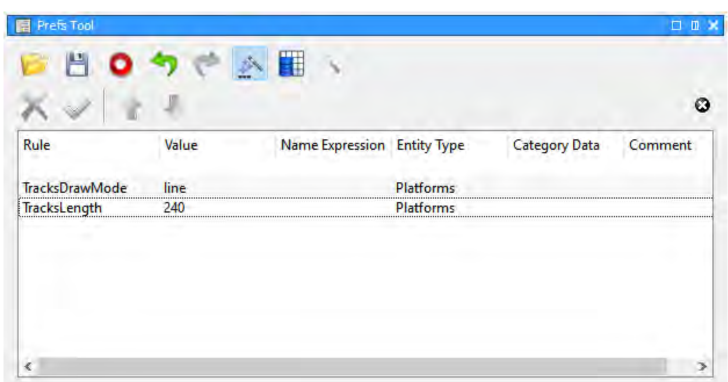


Figure 5.17: Added Preference Rules

5.2.7 Beams Preference Rules Example

For the following example, we will apply **Override Color** for all the beams:

1. Click the **Filter** tab.
2. Deselect all entities by toggling the **All** button.
3. Toggle the **Beams** icon.
4. Toggle the **Record** button. Make sure that the red **RECORDING** banner is visible.
5. Click the **Viewing** tab.
6. Under the **Color Settings**, change the **Override Color** to **Green**.
7. Click **Use Override Color** checkbox. See Figure 5.16 for reference.

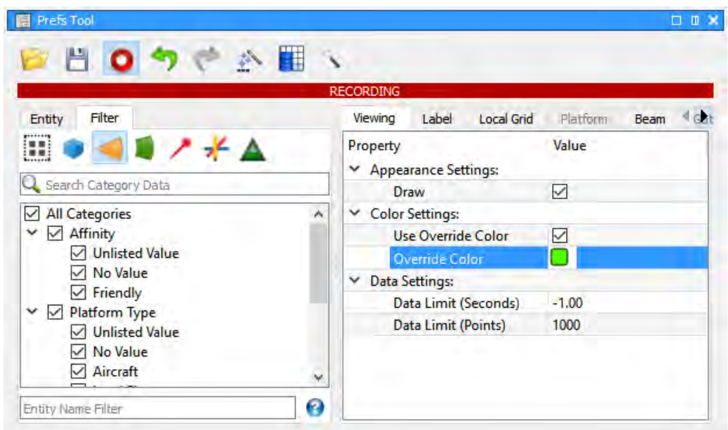


Figure 5.18: Preference Rules to Override Beam Color

8. Toggle the **Record** button. Make sure that the red **RECORDING** banner disappears.
9. Toggle the icon to display the preference rules list and verify added preference rules. See Figure 5.17 for reference.

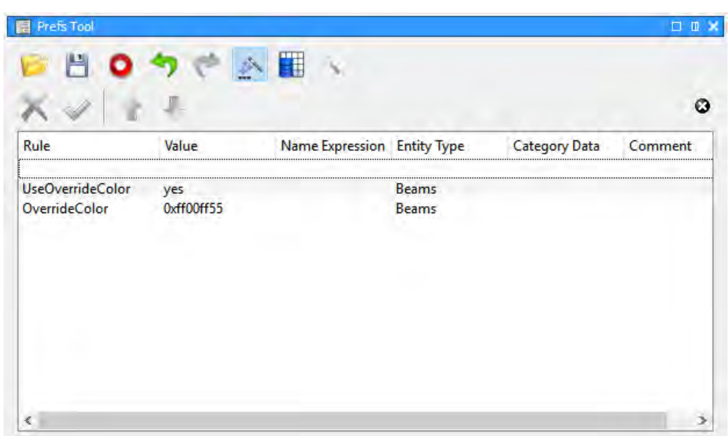


Figure 5.19: Added Preference Rules

5.2.8 Using Category Data

You can use **Category Data** to narrow down your preference rule. In Figure 5.20, the preference rule is to display the Velocity Vector of the y-axis and change the label color to green (using the hex color code `0xff7fff00`) for platforms with **Friendly** affinity. You can use the Section 5.2.6 and Section 5.2.7 as a guide for creating preference rules, and Figure 5.20 as a reference.

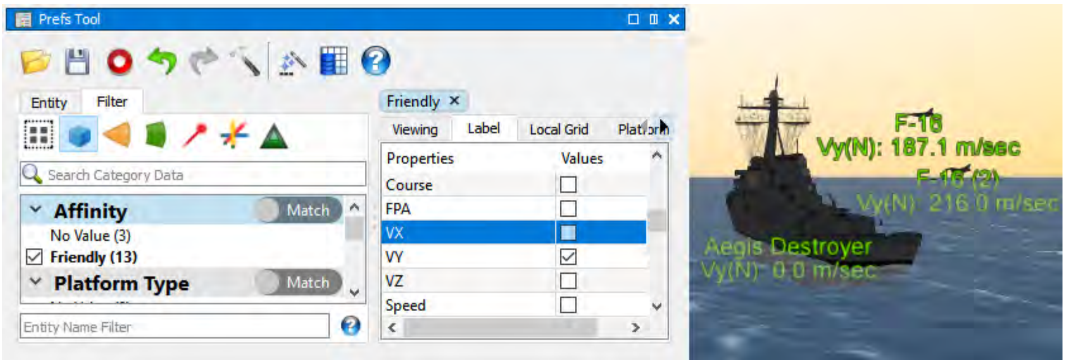


Figure 5.20: Preference Rule to Display Velocity Vector for Friendly Entities

The current altitude value is displayed as a label in SIMDIS for all aircraft platforms in a live scenario Figure 5.21. The rule **Z** corresponds to the **Altitude** of the platform.

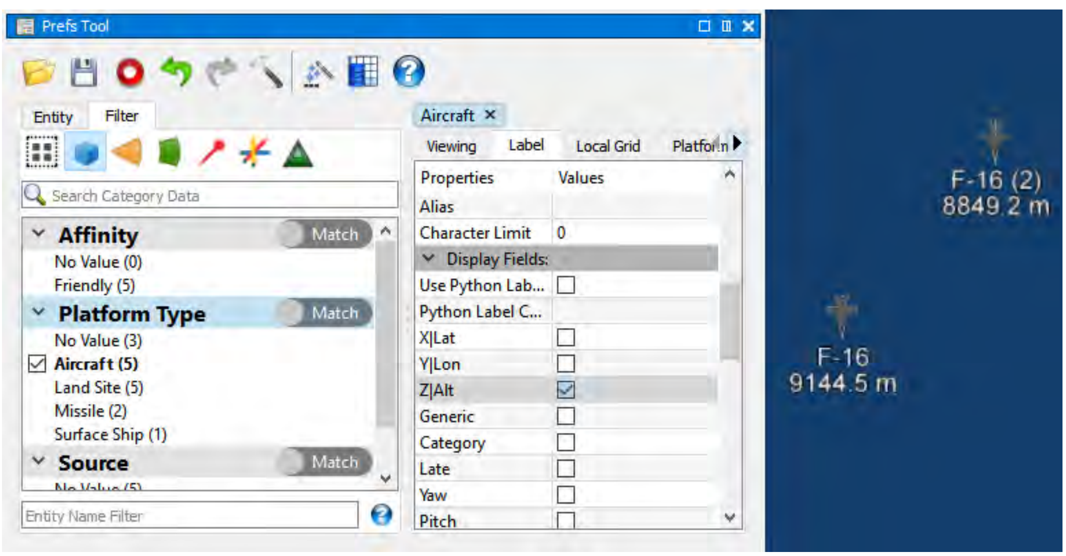


Figure 5.21: Preference Rule to Display Altitude for Aircrafts

5.2.9 Using Regular Expression in Category Data

The **Regular Expression** feature allows more options for filtering entities with category data. The following example will guide you through setting regular expressions on category types.

1. Load the **BMD_Demo.asi** in `$(SIMDIS_DIR)/demos/SIMDIS/Users/NG/BMD_Demo/`.
2. Click the **Prefs Tool** icon on the Main Toolbar or the default hotkey Alt + S.
3. On the **Prefs Tool**, click the **Filter** tab and right-click **Platform Type**.
4. On the right-click menu, click **Set Regular Expression**.
5. Type **^Land** in the **Set Regular Expression** textbox and click the **OK** button. The caret or circumflex symbol (**^**) is a regular expression operator that matches category type values at the beginning of the value name. In this example, we are trying to filter the platform type values that starts with **Land**.
6. Notice that the values under the **Platform Type** category is locked and the **Land Site(5)** value is active and as a breadcrumb displayed as `<Platform Type> ×`, shown in Figure 5.22. You can mouse-over the breadcrumb to display more information about the regular expression filter applied.

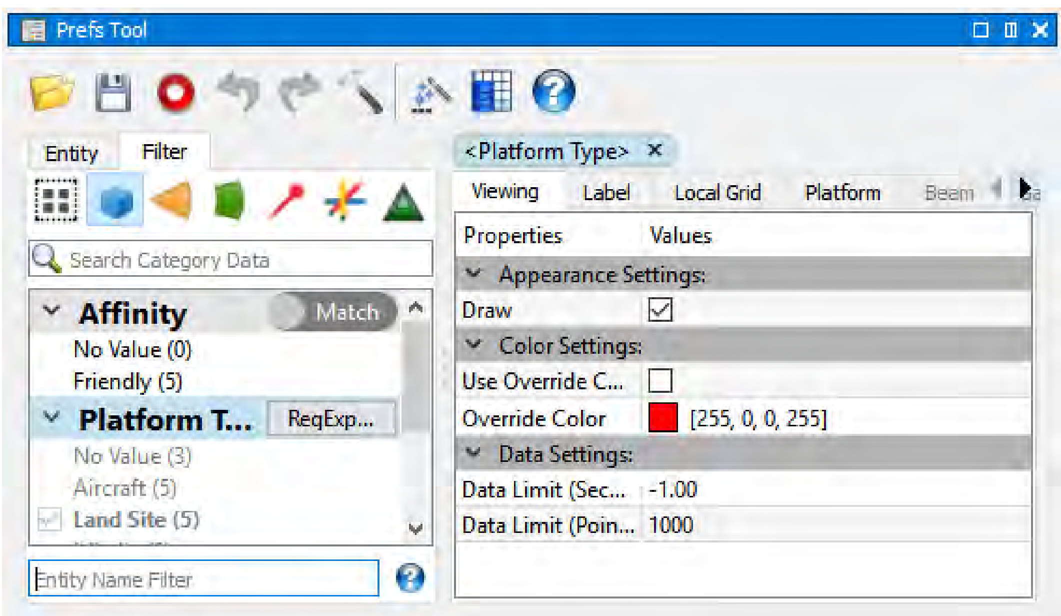


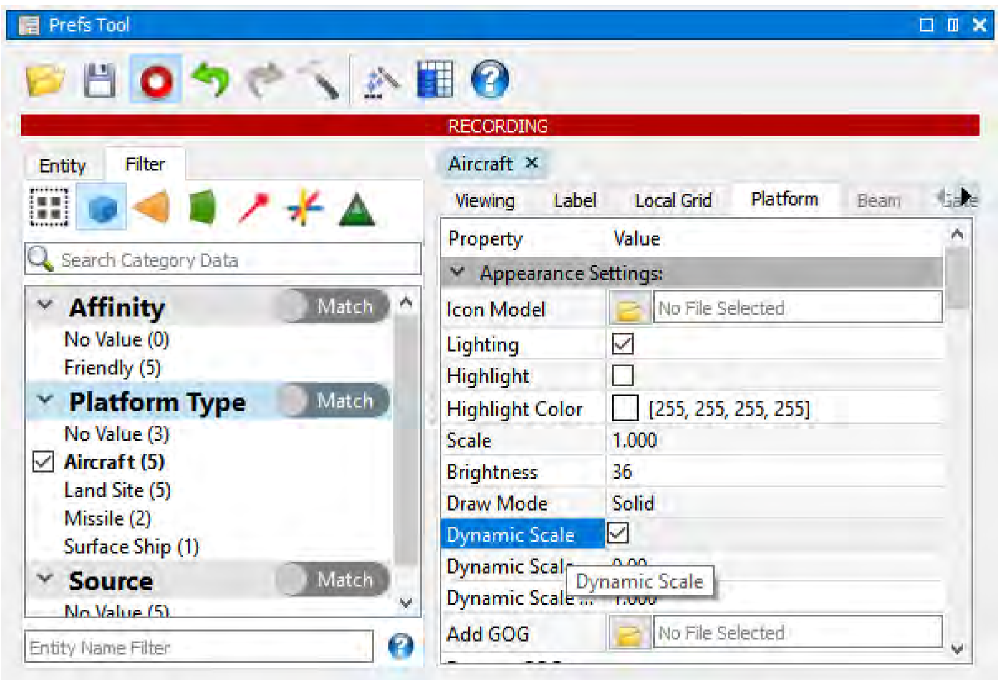
Figure 5.22: Category Filter Regular Expression

NOTE: Active category type value breadcrumbs are also displayed in **Data Browser**.

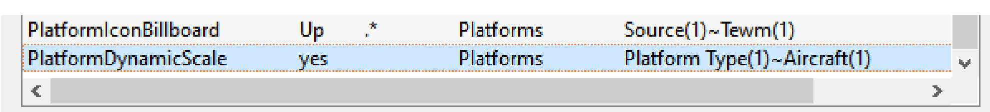
5.2.10 Save and Append Rules Example

You can customize your scenario with different presets by saving multiple preference rules (**.rul**) files. You can customize loaded entities to suit your display needs. The following is an example that will showcase customization of your scenario based on the **Platform Type**:

1. Load the **BMD_Demo.asi** in `$(SIMDIS_DIR)/demos/SIMDIS/Users/NG/BMD_Demo/`.
2. Click the **Prefs Tool** icon on the Main Toolbar or the default hotkey Alt + S .
3. On the **Prefs Tool**, click the **Filter** tab and click **Aircraft** under **Platform Type**. Verify that the slide toggle is in **Match**.
4. Click the to deselect all entities and toggle the **Platforms** icon.
5. Click the **Record** button to start applying rules.
6. Click the **Platform** tab and check **Dynamic Scale**. Your progress should be similar to the following screenshot:



7. Toggle the icon to display the preference rules list and verify the **PlatformDynamicScale** entry.



8. Click the **Record** button to stop recording preference rules changes.
9. Right-click the **PlatformDynamicScale** entry and click **Save** .

10. The **Save Preference Rules File** dialog will appear. Save the preference rule as **Aircraft.rul** file in a directory where you have write privileges (e.g. Desktop).

11. Right-click the **PlatformDynamicScale** entry and click **Remove** or the **Del** button to delete the entry.

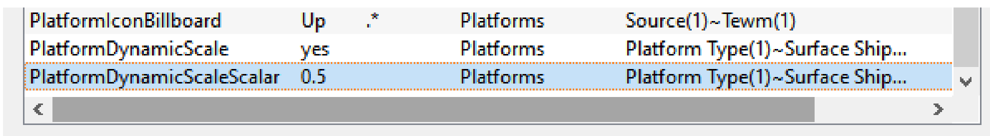
12. Toggle the icon and click the **X** button on the Aircraft × breadcrumb to remove the filter.

13. Click the **Surface Ship** under **Platform Type** then click **Record**.

14. Verify that the **Dynamic Scale** should be still checked. If not, select the **Dynamic Scale** and click the **Apply** button to apply the preference rule.

15. Change the **Dynamic Scale Factor** to **0.5**. Note that decrement to the **Dynamic Scale Factor** increase the dynamic scale size of the platform model.

16. Toggle the icon to display the preference rules list and verify the you have the same entries with the image below:



PlatformIconBillboard	Up	.*	Platforms	Source(1)~Tewm(1)
PlatformDynamicScale	yes		Platforms	Platform Type(1)~Surface Ship...
PlatformDynamicScaleScalar	0.5		Platforms	Platform Type(1)~Surface Ship...

17. Right-click any preference rules entry and click **Save All**.

18. The **Save Preference Rules File** dialog will appear. Save the preference rule as **Surface.rul** file in a directory where you have write privileges (e.g. Desktop).

19. Toggle the icon and click the **Entity** tab. You should only have **Aegis Destroyer** in the list.

20. Double-click on the **Aegis Destroyer** entry to center the eye position on the platform. Notice the increase of dynamic size of the platform.

21. Click the **X** button on the Surface Ship × breadcrumb to remove the filtering.

22. In the **Entity** tab, double-click the **F-16** or **F-16(2)** to center the eye position on the platform. If you notice, the aircraft platforms retain the **Dynamic Scale** even the preference rules is deleted/removed.

23. Reload the **SIMDIS** application by closing it and opening up again.

24. Click **File** on the Menu Bar then click **1 BMD_Demo.asi** to reload the demo.

25. Open **Prefs Tool** then click the **Open** button.

26. The **Load Preferences** dialog will appear. Load the **Aircraft.rul** preference rule file. Notice on your **Main Display** where the aircrafts are now visible because of the **Dynamic Scale**.

27. On the **Prefs Tool**, toggle the icon and verify the **PlatformDynamicScale** entry is in the list.

28. Right-click anywhere in the editor and click **Append**. The **Append Preference Rule** dialog will appear.

29. Open the **Surface.rul** preference rule file. Observe the **Main Display**, you should be able to see the scaled **Aircrafts** and **Surface Ships**. Your display should look similar to Figure 5.23.

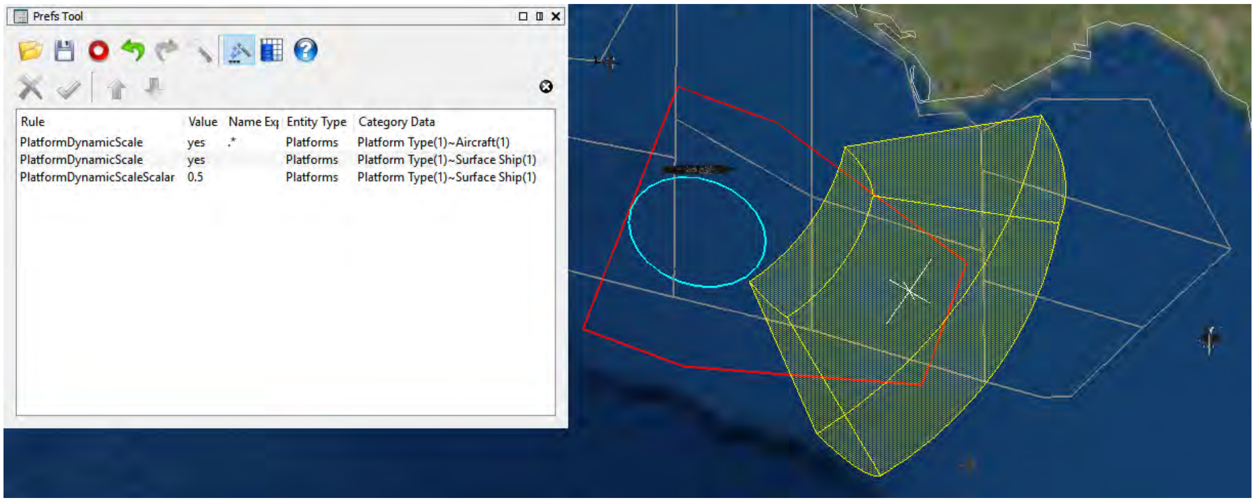


Figure 5.23: Appended Rules

5.3 Data Browser

Data Browser is the main interface dialog that provides access to all entity data. There are four different ways to access Data Browser:

1. The **Data Browser** button in the SIMDIS toolbar.
2. **Data Browser** menu item under the **Tools** menu on the SIMDIS menu bar.
3. The **Data Browser** button in Prefs Tool toolbar.
4. Default hot key Alt + D.

The Data Browser Tool Bar buttons toggle the visibility of different data sections of the dialog: Current Properties, Current Preferences, Current Data State, Data History, and Commands.

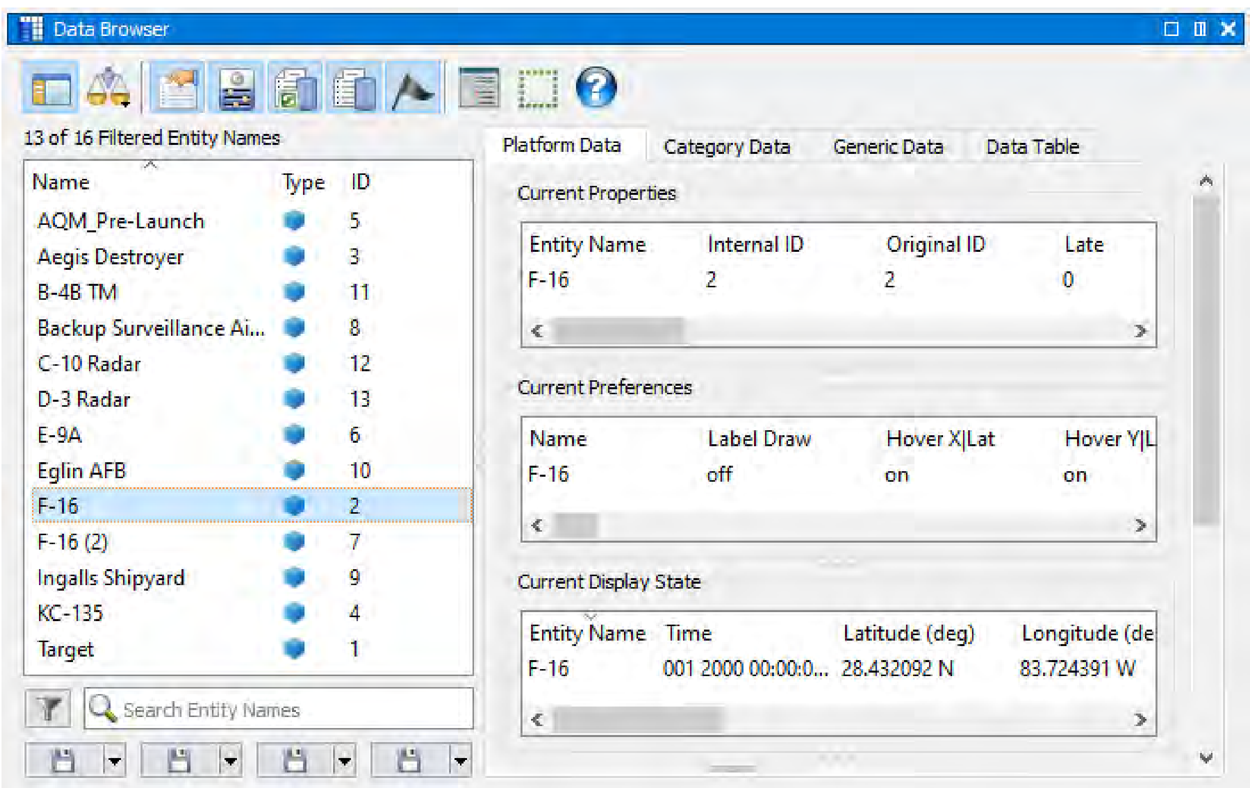


Figure 5.24: Data Browser - Platform Data

If a gate or beam entity is selected, then a Beam Data tab or Gate Data tab will be displayed in the place of the Platform Data tab. The data displayed in these tabs is relevant to the entity type selected.

Platform Data Tab	Contains five subdivisions: 1. Displays the current properties of the selected platform entity. 2. Displays the current preferences of the selected platform entity. 3. Displays the current display state of the selected platform entity. 4. Displays all the stored data points for the selected platform. 5. Display the Commands associated with this platform.
Category Data Tab	Contains fields with non-positional timestamped data that is used for filtering and display preferences. Category data can be associated with any of the entity types.
Generic Data Tab	Contains any non-positional timestamped data that can have a specific expiration time.
Data Table Tab	Displays all non-positional based timestamped data associated with the selected entity.

Different tables per entity can be selected via the highlighted drop down arrow button. Selecting one of the items in a table will jump the scenario to the point in time associated with that data. Telemetry, events, and chat are examples of data that does not include a positional component.

If an entity is configured to display its alias in place of its name but it has no alias set, then its name will be grayed out in the Entity List.

While the **Fence Select**  button is checked, the user can click and drag a box in the SIMDIS 3D display. Any platforms in the box on mouse release will be selected in Data Browser.

The values displayed in the data tabs will update in real time to reflect new data unless the number of selected entites exceeds the **Multi-Select Update Limit**. This setting defaults to 10 and can only be changed through the **Settings Dialog**. Be advised that raising this limit may cause reduced frame rates.

NOTE: Data Browser supports sorting on all columns of the data points table for all entity types, category data, generic data, and data tables.

5.4 Map Editor

To access the imagery and terrain data it is necessary to use the Map Editor Dialog (see Figure 5.25). Chapter 6 discuss detailed information about terrain and imagery. The tool may be accessed in two ways:

1. From the Tools menu on the SIMDIS Menu bar select Map Editor.
2. Using the Hot Key Alt + I

The **Open File** icon on the toolbar is used to load the .earth files. Selecting the icon will display the Load Terrain Configuration Dialog (see Figure 5.26) which displays both the available osgEarth **.earth** and SIMDIS 9 Terrain Configuration (**.txt**) files. It may be necessary to navigate to the desired directory where they are located.

The **Load New Layer** icon on the toolbar allows one to perform a **Quick-Add** for images, elevations, and layers. The appropriate osgEarth driver is selected based on file extension. Image or elevation layers can also be added by specifying their appropriate osgEarth driver directly.

Input files can also be loaded by dragging and dropping them onto SIMDIS. Similar to **Quick-Add**, the appropriate driver is selected based on file extension. Data files dragged onto the visualization are interpreted as imagery layers. If files are dragged onto the Map Editor, the user will be prompted to select the layer type, either imagery or elevation.

The Map Editor currently supports the following drivers:

DB	SIMDIS 9 database format
MBTiles	Map Box Tiles
GDAL	GDAL (Geospatial Data Abstraction Library)
WMS	OGC Web Map Service
TMS	Tile Map Service
ReadyMap	ReadyMap convenience driver
ArcGIS	ArcGIS Web Map Tile Service (WMTS) convenience driver
Weather	Various weather-related WMS feeds
USGS	High-Resolution Orthoimagery from USGS WMS feeds
XYZ	Generic Web Tile Maps (MapQuest)
Debug	Debug Display (tile outline with associated tile key)
Tile Package	ESRI 10.1 tile package compact cache driver
User Defined	Custom user specified imagery via `webMaps.config` file

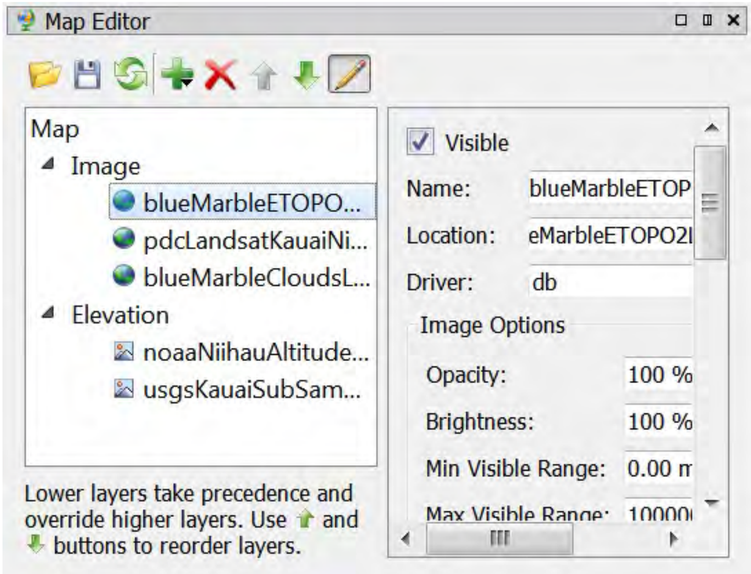


Figure 5.25: Map Editor Dialog

The currently loaded imagery layers are displayed when the dialog opens.

Several of the drivers will require additional parameters, which are stored in the Earth file when saved from Map Editor. Details on the various drivers can be found in osgEarth Documentation.

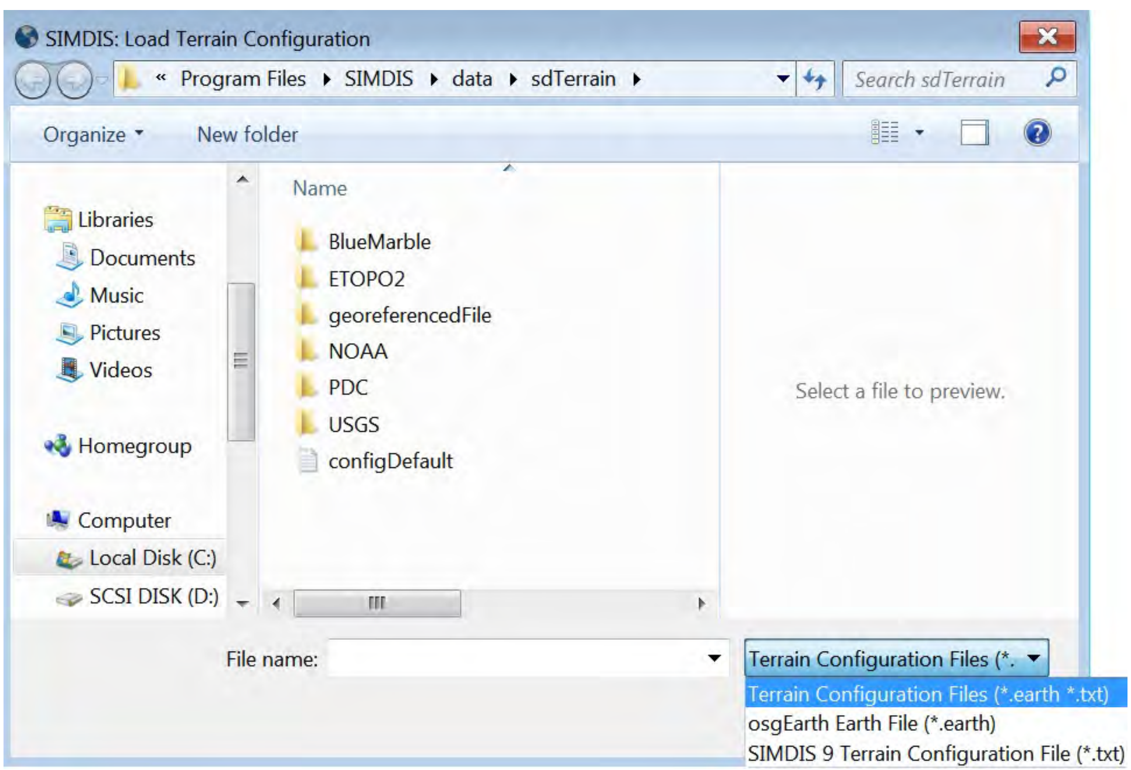


Figure 5.26: Load Terrain Configuration Dialog

5.4.1 Loading GeoTiff

The GDAL driver loads most of the geospatial image files in SIMDIS, the most common being **GeoTIFF**, **MrSID**, and **JPEG2000**. There are multiple ways to load GeoTIFF files in SIMDIS:

- Through an **.earth** file. Refer to Section 6.1.1 for loading geospatial image files.
- Through **Map Editor**
  1. Access Map Editor via **Menu Bar > Tools > Map Editor** or **(Alt + I)**.
  2. Click the icon on the Map Editor dialog.
  3. Select **Quick Add Image**
  4. **Load Image Files** dialog will show. Select **GDAL Files**.

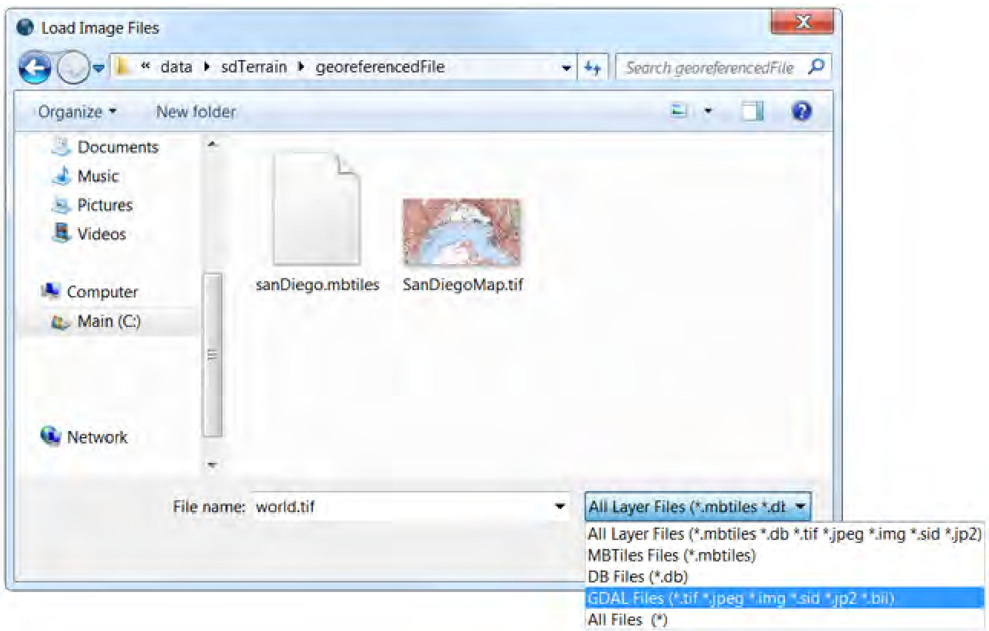


Figure 5.27: Select GDAL Files

  5. Select the appropriate **.tif** file and click the **Open** button.

- Convert it to **.mbtiles** using **osgearth_conv.exe**.
  Example, a **GeoTIFF** file named **sanFrancisco.tif** in the same directory:
  1. In the command line type:

```
osgearth_conv.exe \
--in driver gdal --in url ./sanFrancisco.tif \
--min-level 0 --max-level 32 \
--out driver mbtiles --out filename ./sanFrancisco.mbtiles \
--profile wgs84 --out format image/png
```

  2. Load the file **sanFrancisco.mbtiles** same as in Figure 5.27.

NOTE: The character \ splits single line command to a multi-line.

5.5 Environment Settings

Environment Settings is an optional tool that allows easy modification of the environment. This tool will let you change the environmental setting through several presets (**Fair**, **Overcast**, and **Stormy**) or you can customize the environment parameters using the **Advanced** settings. This tool can be downloaded as part of the **Triton_SilverLining Distribution** package.

When you toggle the **Advanced** settings the tabs **Sky** and **Ocean**, and **Wind Scenario** options will appear. You can change the ocean and sky models, and their respective parameters to achieve your desired environmental state.

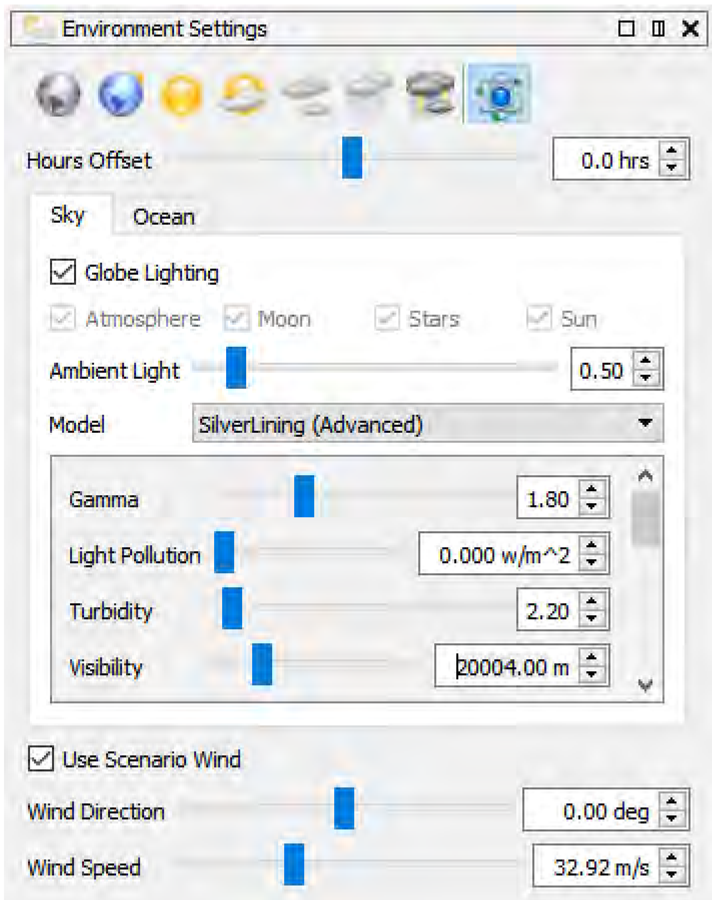


Figure 5.28: Environment Settings with Advanced Options

NOTE: You will have to zoom in less than 50,000 meters altitude and in perspective view to see the environmental effect changes.

5.5.1 Ocean Models

There are three ocean model options you can choose in SIMDIS: **No Ocean**, **Simple Ocean**, and **Triton**. Each model has unique characteristics and different resource requirements. To change the ocean model go to
Tools > **Environment** > **Advanced** > **Ocean**.

No Ocean	No ocean model is displayed.
Simple Ocean	It will render a moving body of water against the loaded terrain. It uses low computer resources and it is effective at simulating ocean environments from a high elevation.
Triton	The realistic ocean model. Features include: ■ Ocean sprays ■ Sun/God Rays ocean surface reflection ■ Choppiness ■ Realistic sea-state levels

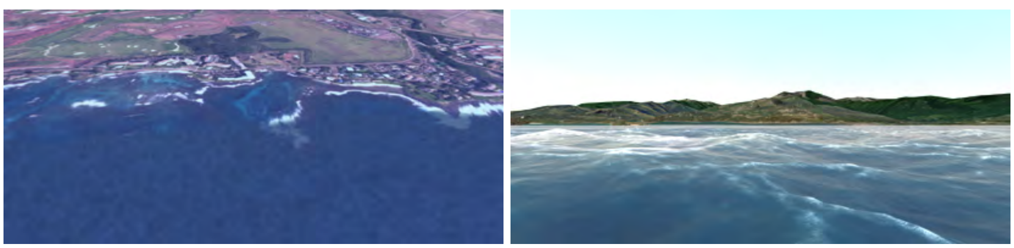


Figure 5.29: Simple Ocean vs. Triton

When certain low altitudes are reached, artifacts from underwater may appear. To avoid artifacts or image noise, you should supply an elevation layer that contains underwater bathymetry data.

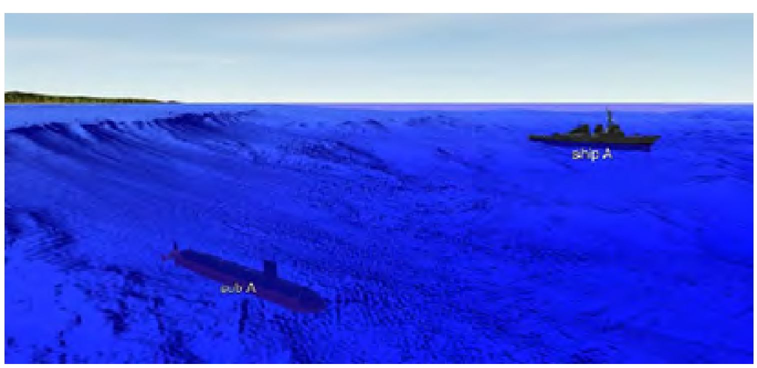


Figure 5.30: Underwater Rendering

5.5.2 Sky Models

SIMDIS has three available models of the sky: **Simple Sky**, **OpenGL Sky**, and **SilverLining**. Figure 5.31 displays the difference between Simple Sky and SilverLining. To change the sky model go to **Tools > Environment > Advanced > Sky**.

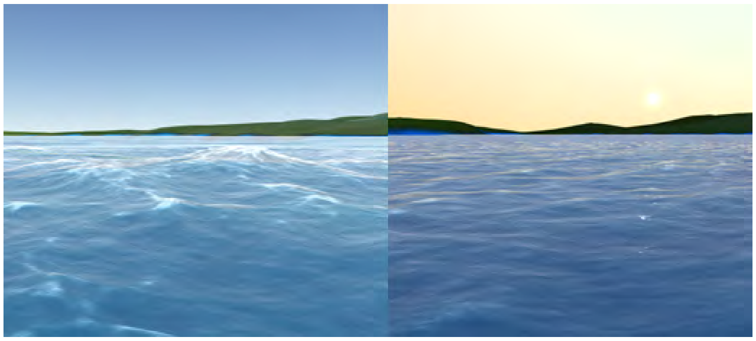


Figure 5.31: Simple Sky vs. SilverLining

Simple Sky	Sky model that implements simple atmospheric scattering and lighting according to the Sam O'Neil GPU Gems article.
OpenGL Sky	Sky model that implements OpenGL Phong shading.
SilverLining	Sky model that produces: ▪ Accurate skies with real atmospheric scattering simulation ▪ Ephemeris for accurate sun and moon position ▪ Dynamic time of day effects ▪ Volumetric 3D Clouds including wind animation ▪ Precipitation: rain, sleet, and snow

5.6 Bookmarks

Bookmarks are time-stamped markers used to indicate an event of interest at a specific point in time. They allow users to manipulate and control the display in order to highlight these events of interest during the playback of the scenario. This capability allows users to create scripted scenarios with either file or live data. Since the bookmarks are saved with the data, it makes them an attractive debrief option for easy distribution and replay. A 300 MB movie of a SIMDIS playback can be distributed in under 3 MB by generating a compressed export from SIMDIS.

Bookmarks can be accessed via **Menu Bar > Tools > Bookmarks**.

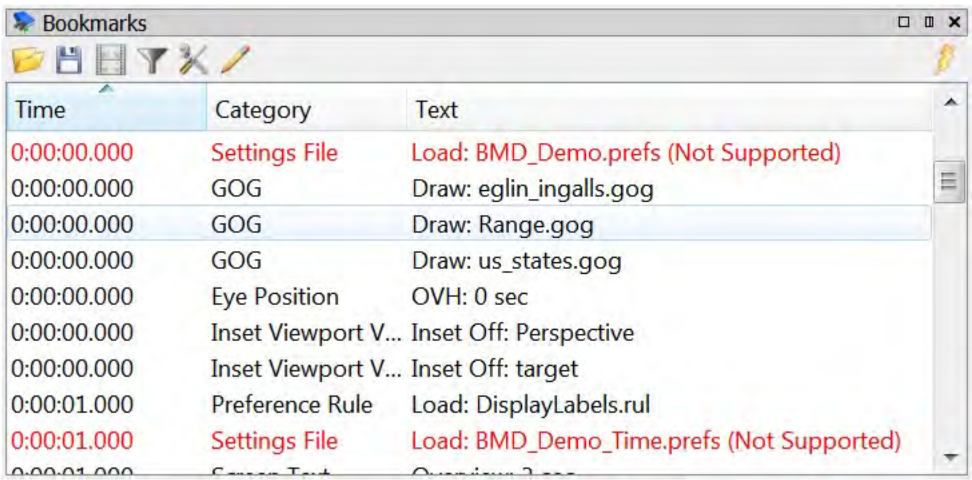


Figure 5.32: Bookmarks Dialog

NOTES:

- Unsupported bookmark events and events with invalid time range are highlighted in red, shown in Figure 5.32.
- You can easily remove the selected **Events** using the **Remove** right-click menu, shown in Figure 5.37.

5.6.1 Category Types

There are multiple category types in Bookmarks you can choose from, to enhance the scenario playback. To select category type simple drop-down the **Type** combo box via the Bookmarks dialog.

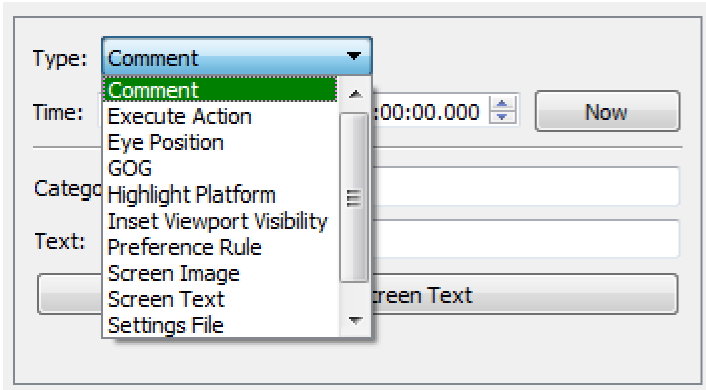


Figure 5.33: Bookmark Category Types

5.6.2 Comment

Comments allow you to insert any text into the bookmarks list for explanation, clarification, or marking significant events. They can be used to list filenames and type of media if **Generate from Files** was used.

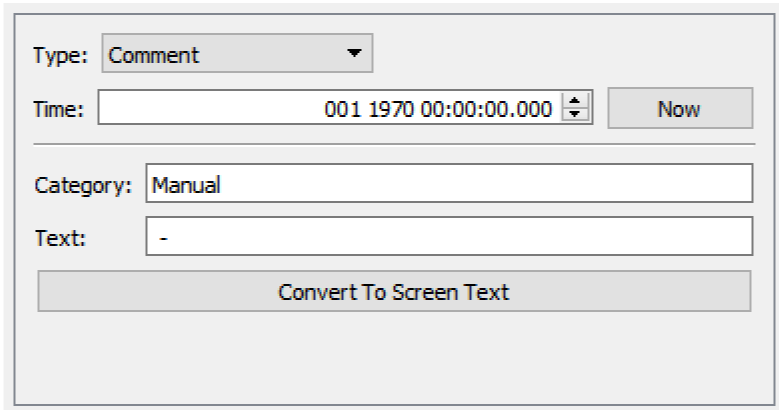


Figure 5.34: Comment Category Type

Time	Used to maintain sort order of comments and does not affect scenario during execution.
Now	Inserts current scenario playback time.
Convert To Screen Text	Convert the actual comment to **Screen Text**.

5.6.3 GOG

The GOG bookmark allows you to draw or hide any pre-loaded GOG file at a specified time. To ensure transportability, place GOG's in a directory that is relative to the bookmark **.toc** file.

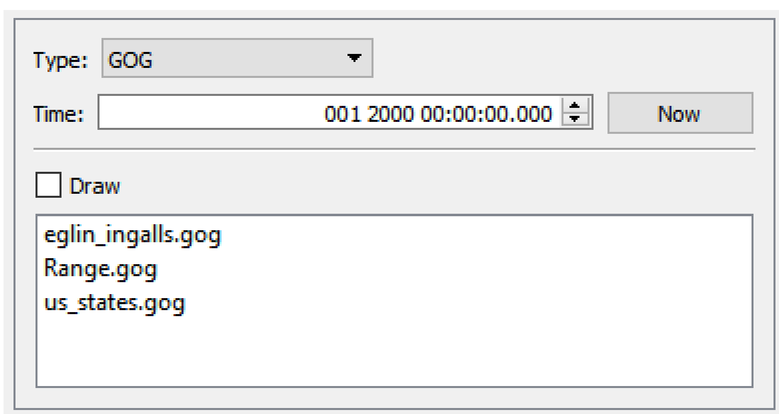


Figure 5.35: GOG Category Type

5.6.4 Eye Position

The Eye Position bookmark allows you to move the camera to a specific location and angle. The Eye Position options pane, shown in Figure 5.36, lists the available Eye Positions in a scrollable field from which you may select an eye position to execute. You may also use the **Create Eye Position** button to create a new Eye Position from the current settings.

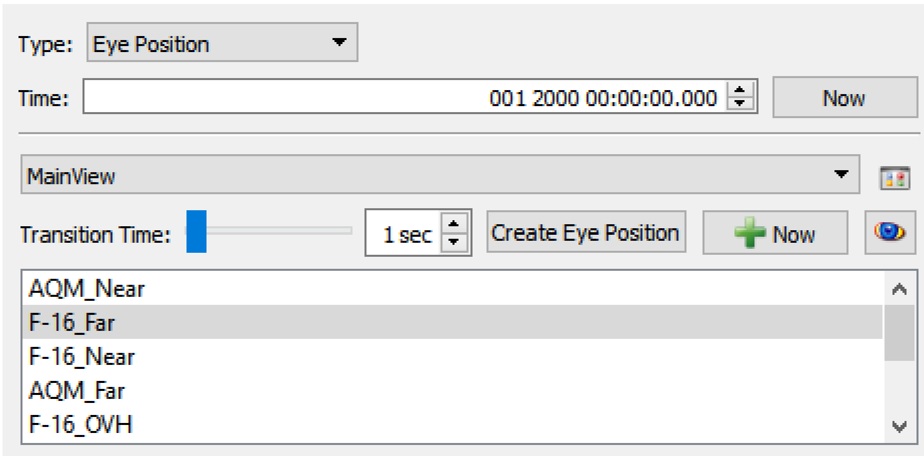


Figure 5.36: Eye Position Category Type

Transition Time	Specifies the duration time from the current view to the selected view.
Create Eye Position	Inserts current scenario playback time.
Now	Create an Eye Position from the current view.
	Open the **Viewport** dialog.
	Open the **Eye Positions** dialog.

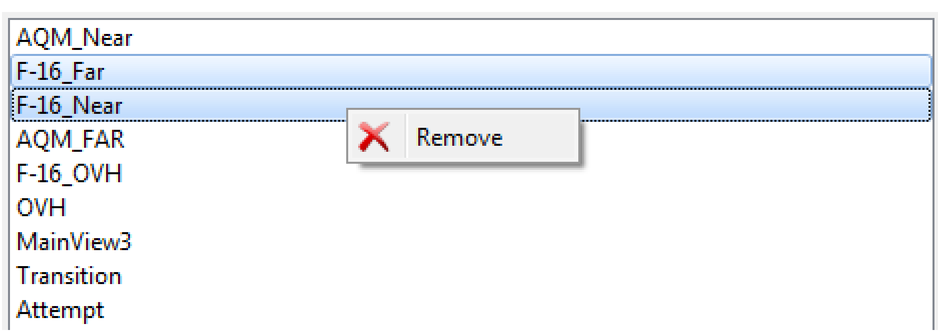


Figure 5.37: Eye Position Category Type

NOTE: You can easily remove the selected **Eye Positions** using the **Remove** right-click menu, shown in Figure 5.37.

5.6.5 Settings File

The Settings File category allows you to load a settings file to execute on a certain time frame Example: change the SilverLining clouds from **Fair** to **Mostly Cloudy** or change the overall font size through **Settings > HUD > Logo > Large Font Size**.

5.6.6 Screen Image

The Screen Image category lets you display images during scenario playback.

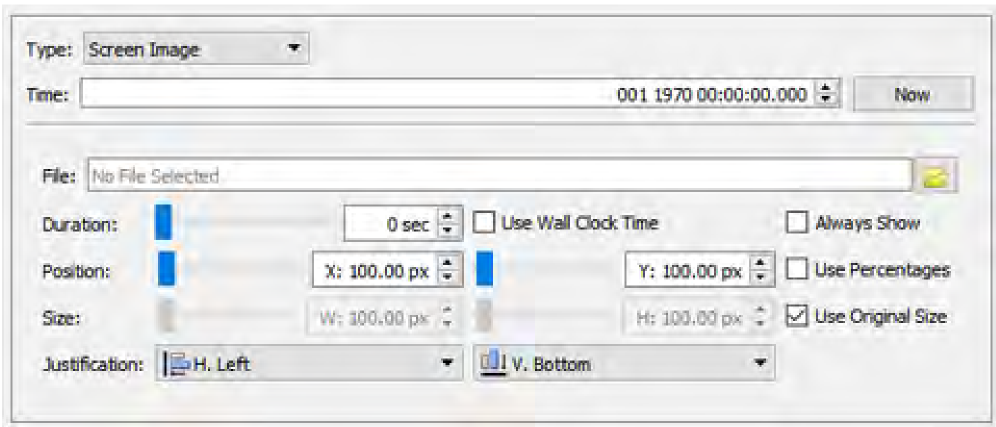


Figure 5.38: Screen Image Category Type

	Displays a file browser to select a file to open.
Position	Set the position of the **Screen Image** on the screen in pixel size.
Always Show	The **Screen Image** is shown for the entirety of the scenario. The **Time** and **Duration** options are not applicable.
Use Percentages	Set the position of the **Screen Image** using percentages. It is effective for displaying images for scenario playback in different resolutions.
Size	Set the size of the **Screen Image** in the screen in pixel size.
Justification	Set the horizontal (**H**) and vertical (**V**) alignment of the **Screen Image**.
Use Original Size	Use the actual resolution of the loaded **Screen Image**.

5.6.7 Screen Text

The Screen Text category lets you display text to appear on the screen during scenario playback.

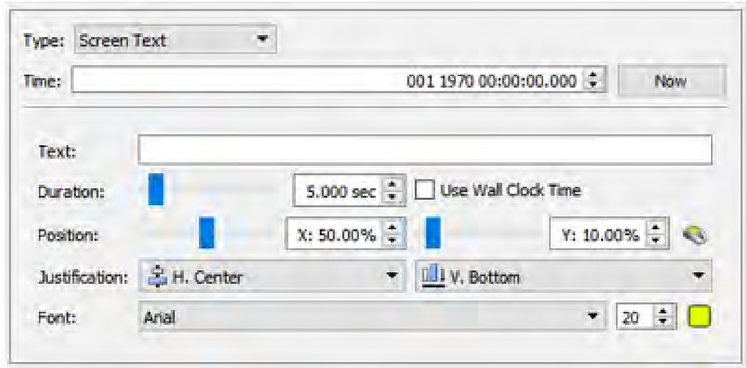


Figure 5.39: Screen Text Category Type

Text	Set the screen text.
Use Wall Clock Time	Force to show the text using the host's clock time and not the scenario playback time.
Position	Set the position of the text in the screen in percentage.
	Use cursor location to specify the position of the text.
Justification	Set the justification of the text horizontally and vertically
Font	Set the font type, size, and color of the text.

5.6.8 Preference Rule

The Preference Rule category allows you to clear, append, or load preference rules from a file in a scenario playback.

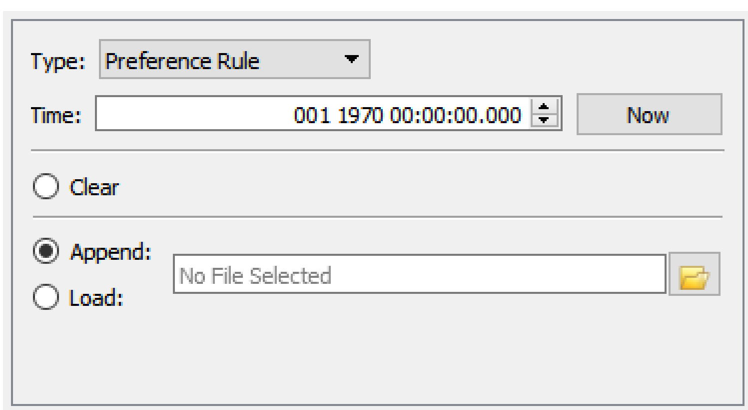


Figure 5.40: Preference Rule Category Type

5.6.9 Inset Viewport Visibility

The Inset Viewport Visibility category lets you select and toggle the available inset viewport. Figure 5.42 shows an example of two insets with different eye positions.

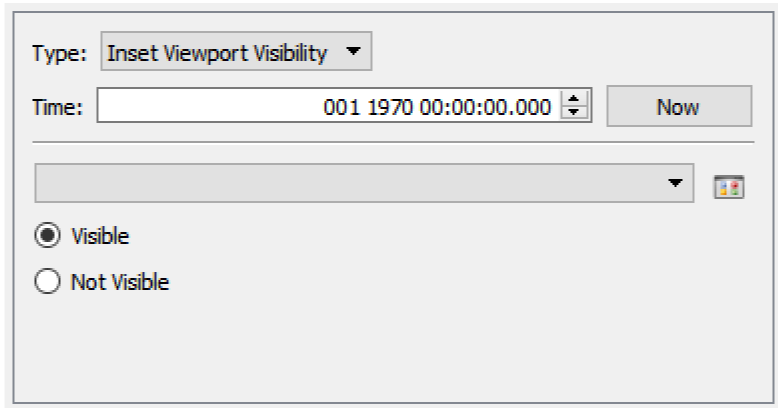


Figure 5.41: Inset Viewport Visibility Category Type

	Open the **Viewport** dialog.

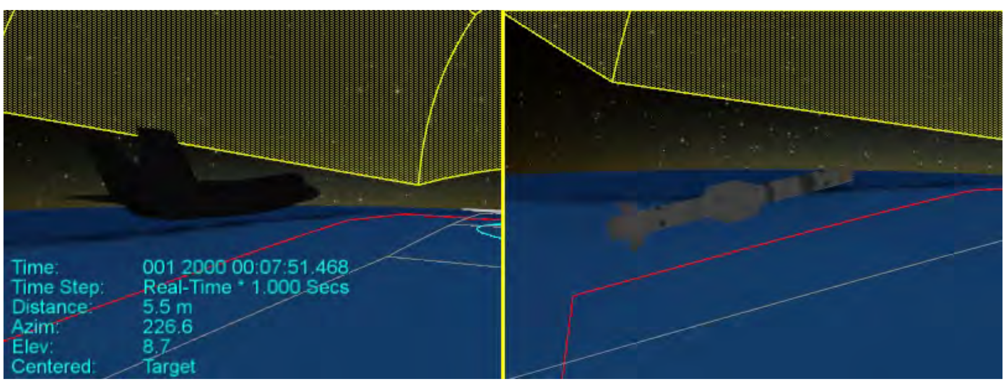


Figure 5.42: Inset Examples

NOTE: The inset viewport will not cover any text overlay on the main view HUD.

5.6.10 Time

The Time category allows you to add a bookmark to manipulate or control the time or the timing of the scenario playback.

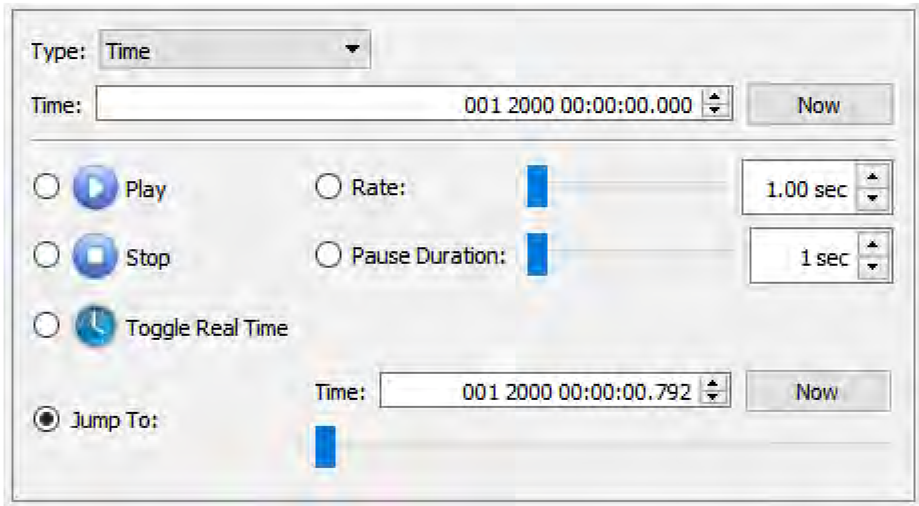


Figure 5.43: Time Category Type

5.6.11 Execute Action

The Execute Action category lets you add a bookmark to trigger any action that can be triggered via hot key.

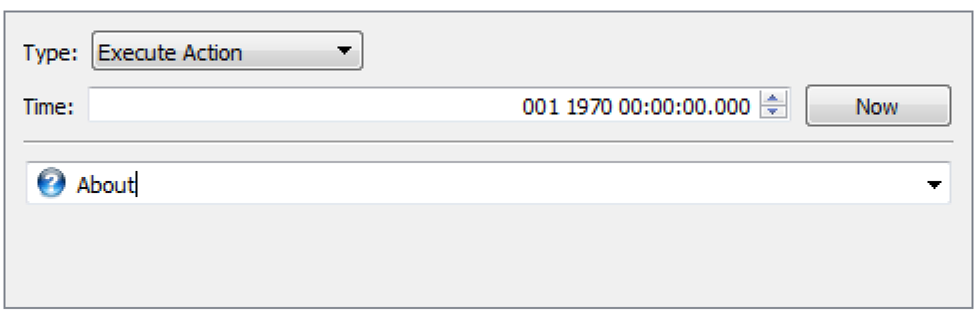


Figure 5.44: Execute Action Category Type

5.6.12 Highlight Platform

The Highlight Platform category lets you add a bookmark to draw a highlight around a specific platform. This highlight is drawn by setting the highlight pref for the platform. Edits made in the **Prefs Tool** may change the effect of this bookmark. For more information about prefs and the **Prefs Tool**, see Section 5.2.

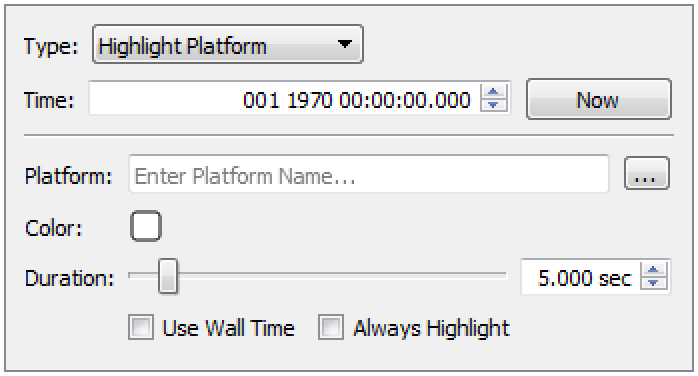


Figure 5.45: Highlight Platform Category Type

5.7 Ruler

The Ruler provides various measuring capabilities in the 3D display. Using the Ruler dialog, you can view the Ruler measurements and adjust its display behavior. For more information about the Ruler's capabilities, hover your mouse pointer over the respective icons in the Ruler dialog to display the tool tips.

Figure 5.46: Ruler Measuring Distance

5.7.1 Using Ruler

The **Ruler** can be used at any time, whether a scenario is loaded or not. There are three ways to activate the Ruler:

1. The **Ruler** button on the Main Toolbar. The Ruler dialog will not show if **Ruler > Show Dialog from Status Bar** is set to **false** in **Settings**
2. The **Ruler** menu item under **Tools** in the Menu Bar
3. A hot key, which can be bound in **Tools > Hot Keys**

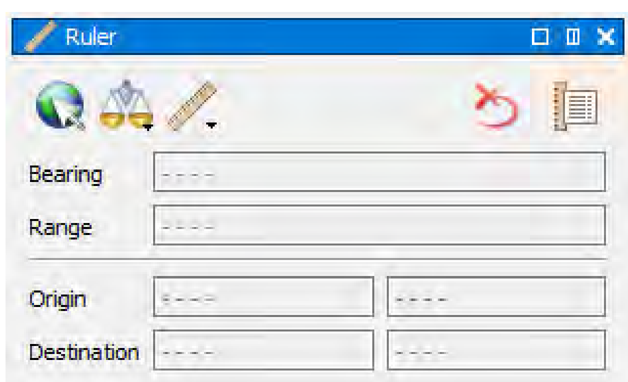


Figure 5.47: Ruler Dialog

Once Ruler is active, you can create one or more Ruler lines in the scene (see Section 5.7.2 for more information). The Ruler dialog (Figure 5.47) will show details about the most recently selected Ruler line, updating in real time as the Ruler line does. The **Delete All Rulers** button will delete all Ruler lines in the scene.

5.7.2 Ruler Mouse Modes

There are two ways to create Ruler lines in the scene, controlled by the **Allow Mouse Rotation** setting or the **Mouse Rotation** button in the Ruler dialog.

5.7.2.1 Without Rotation Mode

Without Rotation Mode is the default mouse mode, and offers simplified input options:

- Left-click in the scene and drag your mouse to start a Ruler line (replacing the last Ruler line you created) from your initial position to the point at which you released the mouse button.
- Hold the Ctrl key when you left-click on the map to create a new Ruler line as you drag without replacing the previous Ruler line.
- Hovering over either endpoint of a line will change your mouse cursor to an open hand icon. In this state, you can click on the endpoint and drag your mouse to place it at a new location.
- While editing or hovering over any portion of a line, you can right-click to delete that line.

5.7.2.2 With Rotation Mode

With Rotation Mode is a more advanced input mode which will allow you to rotate and pan the map while placing your Ruler line. Inputs in this mode are slightly modified:

- Left-click in the scene to start a Ruler line (replacing the last Ruler line you created) at your current mouse position. The destination of your new Ruler line will follow your mouse until you click again.
- Hold the Ctrl key when you left-click on the map to create a new Ruler line without replacing the previous Ruler line.
- Hovering over either endpoint of a line will change your mouse cursor to an open hand icon. In this state, you can click on the endpoint and it will follow your mouse until you click again.
- While editing or hovering over any portion of a Ruler line, you can right-click to delete that line.

Figure 5.48: Multiple Rulers

5.7.3 Platform Locking

One or both endpoints of a Ruler line can be locked onto a SIMDIS Platform, allowing the Ruler to update in real time as the Platform moves. The mouse cursor will change to a crosshair symbol when within range of a Platform, shown in Figure 5.49. If one or more Ruler lines are already locked onto a Platform, you may see the hand icon instead of the crosshair when creating a new line, but that will not stop your new line from locking on.

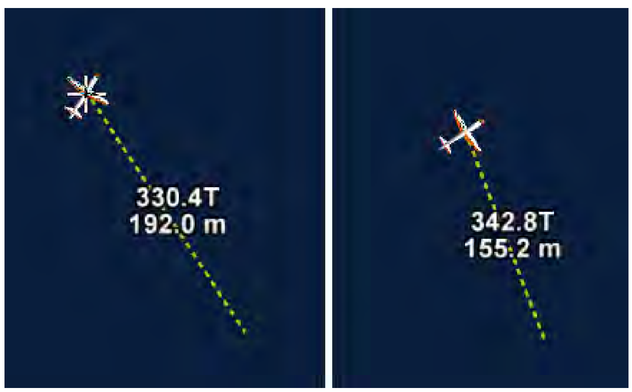


Figure 5.49: Locking a Ruler onto a Platform

NOTE: Hold the Alt key to bypass Platform locking and instead place your Ruler line directly on the map. However, disabling the **Lock To Platform** setting inverts this behavior, bypassing Platform locking except when you are holding the Alt key.

5.7.4 Ruler Configuration

The Ruler also offers a few other configuration options:

- **Edit Units** allows you to change the units used by Ruler lines. By default, Ruler uses the Units set in the global context of the **Unit Editor**.
- **Edit Label Template** allows you to change the label template used by the Ruler lines. Rulers can display either the **Bearing and Ground Distance** or the **Bearing, Slant Range, and Altitude** between the line's endpoints. The two options can be seen in Figure 5.50. You can also bind hot keys to each of the two template modes using the **Hot Keys** dialog.

Figure 5.50: Ruler Templates: Ground Distance (top) vs. Slant Range and Altitude (bottom)

5.7.5 Ruler List

The Ruler List display lists all Ruler lines in the scene by the latitude and longitude of each line's endpoints. The Ruler List display can be opened using the **Ruler List** button in the Ruler dialog.

The Ruler dialog will track the selected Ruler line in the Ruler List, rather than the last line modified. You can delete the selected Ruler line using the **Delete Selected Ruler** button.

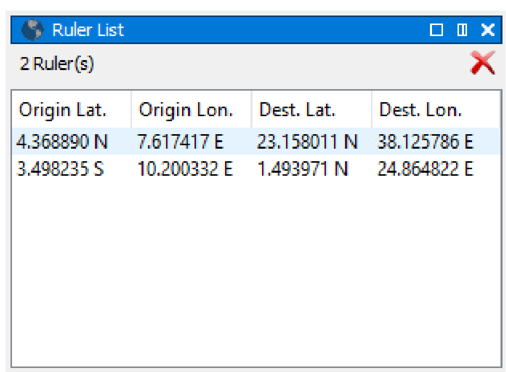


Figure 5.51: Ruler List

5.8 Range Tool

Range Tool is a comprehensive tool that includes a variety of measurements to calculate metrics between entities. The Range Tool dialog allows you to perform various types of calculations and measurements between two entities. The calculations are dependent upon the entity type and available entity state data. Appendix E contains detailed information about Range Tool units and calculations. For more information about the Range Tool's capabilities, hover your mouse pointer over the respective icons on the Range Tool dialog to display the tool tips.

Figure 5.52: Slant Range on Range Tool

5.8.1 Accessing Range Tool

A scenario is required in order to use the Range Tool. The scenario can either be file-based or network-based. There are three different ways to access the Range Tool:

1. The **Range Tool** button in the Toolbar
2. The **Range Tool** menu item under **Tools** in the Menu Bar
3. Use the hot key Alt + R .

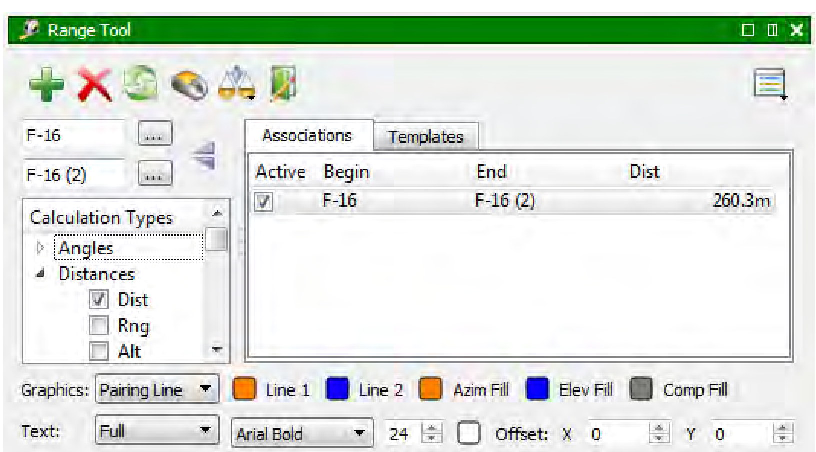


Figure 5.53: Range Tool Dialog

5.8.2 Pairing Entities

There are multiple ways you can pair or associate entities in Range Tool.

- You can type the entity in either the **Begin Entity Name** or **End Entity Name** text box and use the autocomplete feature, shown in Figure 5.54.

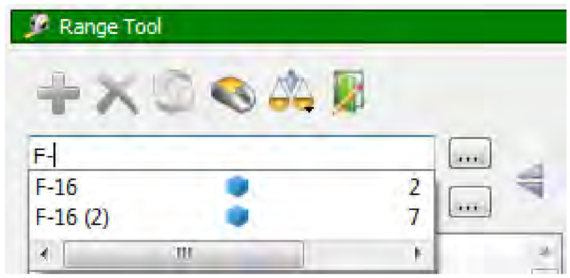


Figure 5.54: Range Tool's Autocomplete Feature

- Click the ... button next to **Begin Entity Name** or **End Entity Name** text box to open the **Select Entity** dialog, shown in Figure 5.57.
- Click the button on the Range Tool's toolbar to toggle **Select By Mouse** pairing. While in **Select By Mouse** mode, the cursor will change to a cross when over a platform to indicate it can be clicked. Left-click the **Begin Entity** and drag the line to the **End Entity** or left-click the **Begin Entity** and **Shift** + **Ctrl** + left-click on the **End Entity** to **connect** or **hook** the entities together.
- While the **Range Tool** is open, you can **right-click** an entity and set it as either **Begin Entity** or **End Entity**, shown in Figure 5.55.

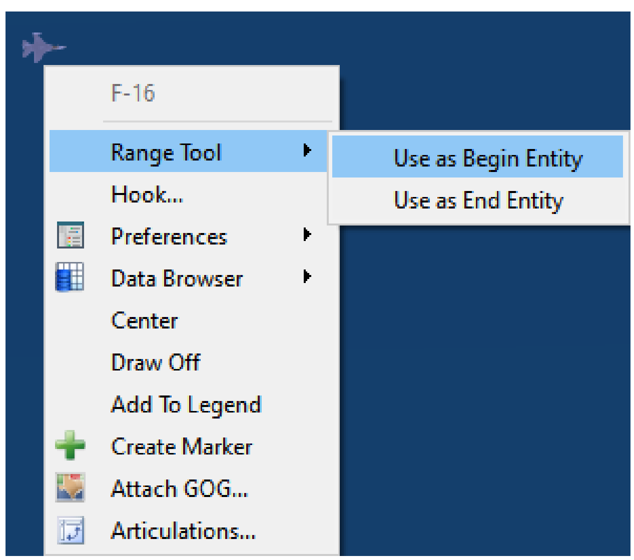


Figure 5.55: Range Tool's Right-Click Menu

Existing pairs can have one or both entities replaced either by using the button or by holding the Alt key while selecting entities by mouse.

Figure 5.56: Range Tool's Click-and-Drag Pairing Feature

5.8.3 Entity List and Filters

You can select an entity via the **Select Entity** dialog. Click the ... button on the Range Tool toolbar to open the **Select Entity** dialog shown in Figure 5.57

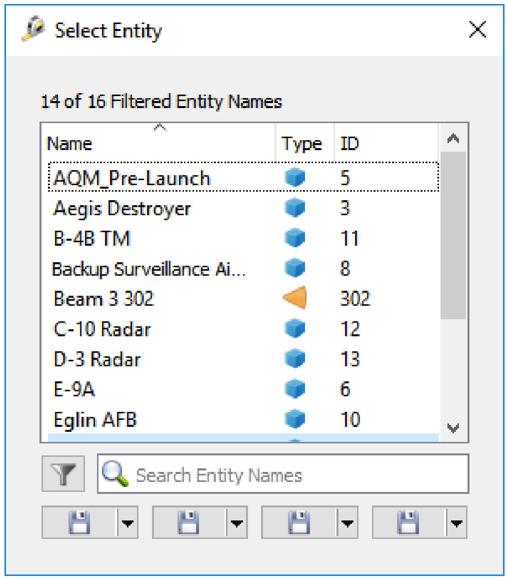


Figure 5.57: Select Entity Dialog

A filter can be applied to the list so that only certain types of entities are displayed. This filter is applied using **Entity Filters** dialog, shown in Figure 5.58. The **Entity Filters** dialog is displayed by clicking the button, left of the **Search Entity Names** search box. Refer to Section 5.8.7 for an example for filtering entities.

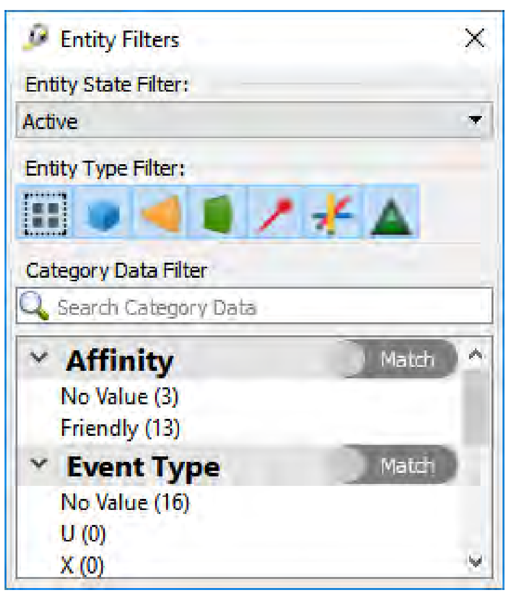


Figure 5.58: Entity Filters Dialog

5.8.4 Templates

Templates will allow the Range Tool to quickly switch formatted range calculation types. There are four templates that you can switch to and each template is assignable to a hotkey. You can access the templates in **Templates** tab, shown in Figure 5.59.

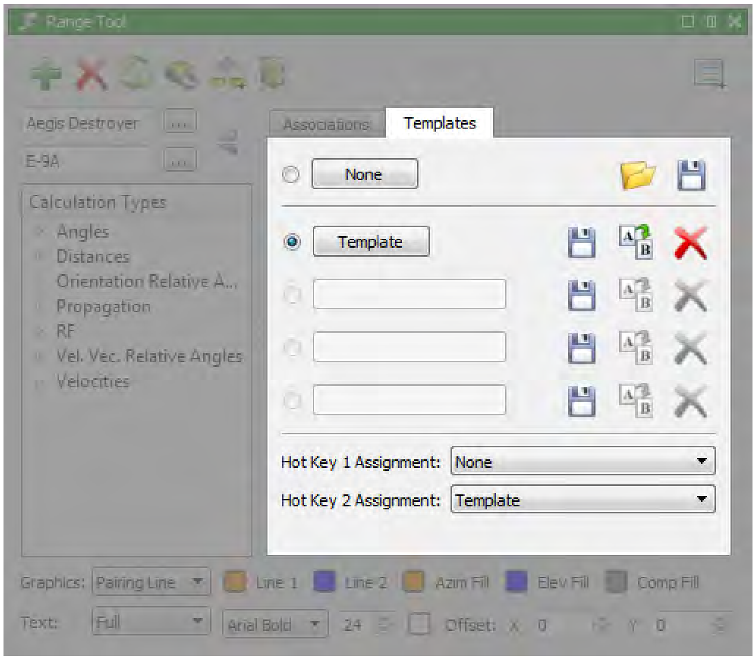


Figure 5.59: Range Tool's Template

You can automatically load saved template(s) in SIMDIS by using the command line argument **-rtt** and specify the **.rtt** file.

```
simdis -rtt <filename>
```

5.8.5 Configuration

The **Range Tool Configuration** will allow you to customize the **File Writer** behavior and the **Effective Earth Radius Scalar**.

You can access the **Range Tool Configuration** via **Configure** on the drop-down menu of the **Range Tool** .

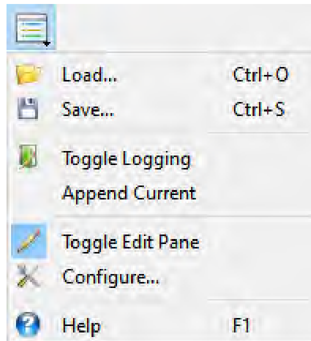


Figure 5.60: Range Tool Drop-Down Menu

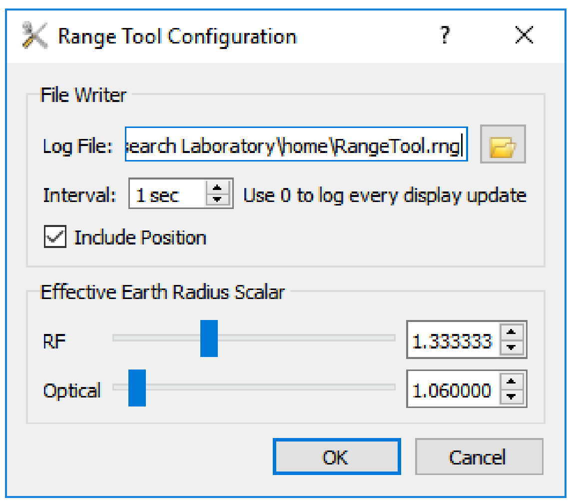


Figure 5.61: Range Tool Configuration

Effective Earth Radius Scalar

RF	This value is used to control the RF effective Earth radius for the calculation of the RF horizon. The default value of 1.33 is based on the standard 4/3's effective Earth radius.
Optical	This value is used to control the optical effective Earth radius for the calculation of the optical horizon. The default value of 1.06 is based on the value used by the EW Handbook.

5.8.6 Range Tool Example

This following example will guide you on how to setup a scenario display and use the Range Tool between platform entities.

1. Start **SIMDIS**.
2. Load `$(SIMDIS_DIR)/demos/SIMDIS/import/wake.asi`.
3. Click the button on the toolbar or press the D key to toggle the **Dynamic Scale**.
4. Click the button on the toolbar or press Alt + S keys to open the **Prefs Tool**.
5. Select the **Krivak** entity and click the **Platform** tab.
6. Change the **Dynamic Scale Factor** value to **0.5**, this will enlarge the platform entity icon twice its size when the **Dynamic Scale** is toggled.
7. Double-click **PC-12** to select and center on the PC-12 platform entity and change the **Dynamic Scale Factor** value to **0.5**.
8. Click the button on the SIMDIS toolbar or use the hot key Alt + R to open the Range Tool dialog.
9. Click the button on the **Range Tool** toolbar to select the pairing by click and drag to two platforms.
10. Mouse-over to the **PC-12** platform until your cursor changes to a cross. To display the platform labels, click the on the SIMDIS toolbar or press the N key.
11. Click and drag the line to the **Krivak** platform. Shown in Figure 5.62.

Figure 5.62: Range Tool's Click-and-Drag Pairing

12. In **Calculation Types**, expand **Angles** and check **Az(T)**. This will display the **True Azimuth** between **Krivak** and **PC-12** platforms.
13. In **Calculation Types**, expand **Distances** and check **Dist**. This will display the **Ground Range** between **Krivak** and **PC-12** platforms, shown in Figure 5.63
14. Click the button on the SIMDIS toolbar to play the scenario and notice the values change based on the platforms' movement.

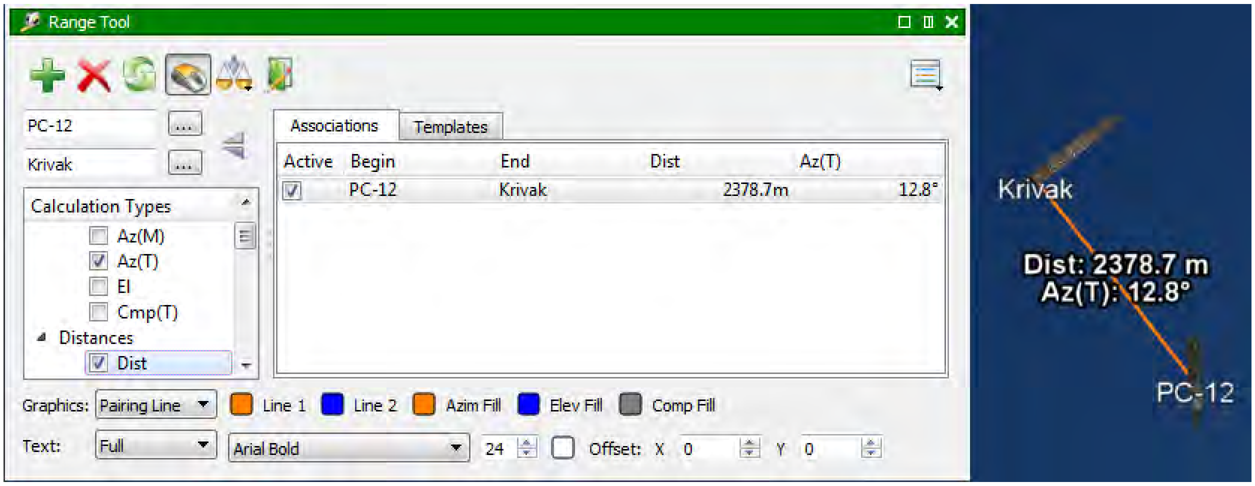


Figure 5.63: Range Tool's Paired Platforms

5.8.7 Entities Filtering Example

The following example will guide you on how to filter entities in Range Tool.

1. Start **SIMDIS**.
2. Load `$(SIMDIS_DIR)/demos/SIMDIS/Users/NG/BMD_Demo/BMD_Demo.asi`.
3. Click the button on the SIMDIS toolbar or use the hot key Alt + R to open the Range Tool dialog.
4. Click the ... button to open the **Select Entity** dialog.
5. Click the **Filter** button to open the **Entity Filters** dialog.
6. In the **Category Data Filter** list, click **Aircraft(5)** under the **Platform Type**. Notice that a checked-box will appear left of the **Aircraft(5)** platform type, shown in Figure 5.64.

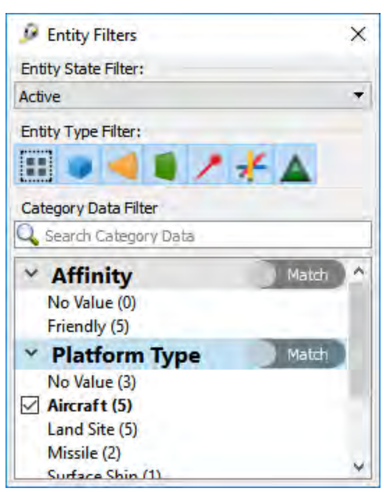


Figure 5.64: Entity Filter Example

7. Close the **Entity Filters** dialog. The entity list on the **Select Entity** dialog is updated with a list of aircraft platforms, shown in Figure 5.65.

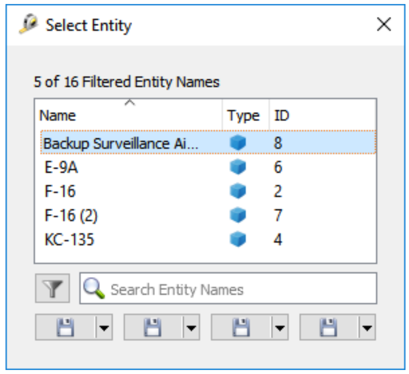


Figure 5.65: Select Entity Example

5.8.8 Templates Example

Templates allow you to switch different range calculations criteria and formatting. The following example will guide you to create and switch templates using the **Templates** tab in Range Tool and using the hot key.

1. Start **SIMDIS**.
2. Load `$(SIMDIS_DIR)/demos/SIMDIS/Users/NG/BMD_Demo.asi`.
3. Click the button on the SIMDIS toolbar or use the hot key **Alt** + **R** to open the Range Tool dialog.
4. Click the ... button, right next to **Begin Entity Name** search box, to open the **Select Entity** dialog.
5. Double-click the **F-16** in the list.
6. Click the ... button, right next to **End Entity Name** search box, to open the **Select Entity** dialog.
7. Double-click the **KC-135** in the list.
8. Click the **Add Association** button on the Range Tool's toolbar. Notice the association of **F-16** and **KC-135** platforms, shown in Figure Figure 5.66.

Figure 5.66: Select Entity Example

9. In the **Calculation Types** list, expand **Angles** and click to check the **Az(M)**.
10. Click the **Templates** tab and click the first **Save Template** button to register the template.
11. Click the first **Rename** button to rename the template to **Magnetic**.
12. In the **Calculation Types** list, expand **Angles**, uncheck the **Az(M)**, and click to check the **Az(T)**.
13. Click the **Templates** tab and click the second **Save Template** button to register the template.
14. Click the second **Rename** button to rename the template to **True**.

15. Drop-down the **Hot Key 1 Assignment** combo box and select **Magnetic**.

16. Drop-down the **Hot Key 2 Assignment** combo box and select **True**.

17. Click the  button on the SIMDIS toolbar to open **Prefs Tool** or press the default hot key Alt + S .

18. In the **Entity List** double-click the **F-16** to center on the platform. The centered platform is displayed on the **Overlay Text**, bottom-left side of the Main Display.

19. Zoom-in until you see the **F-16** and **KC-135** platforms.

20. Press the hot keys Alt + 1 and Alt + 2 will switch between the **Magnetic** and **True** templates respectively, shown in Figure 5.67.

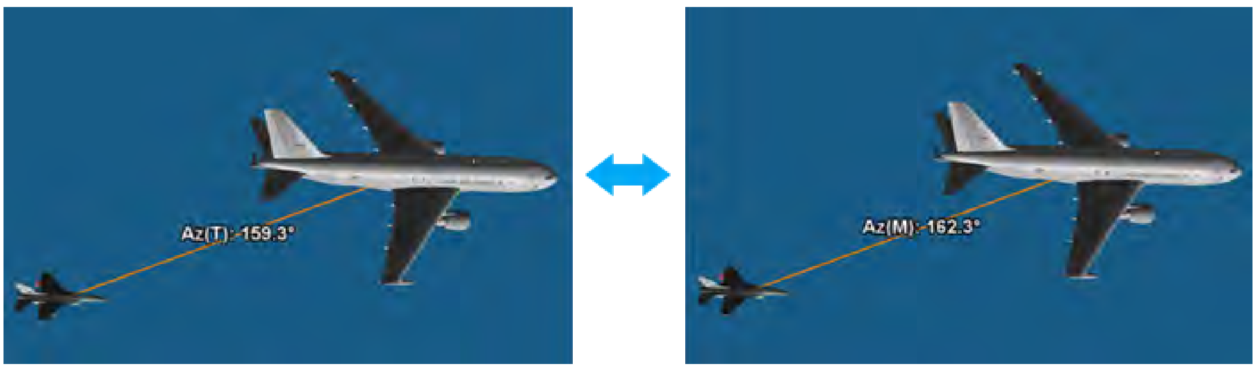


Figure 5.67: Switching Templates

5.9 Closest Point of Approach (CPA)

The Closest Point of Approach (CPA) Dialog is now implemented and is accessible via the **Tools** menu. The CPAs are calculated between two entities for a specific period of time. Filters may be applied to specify a minimum distance or a maximum number of calculated CPAs.

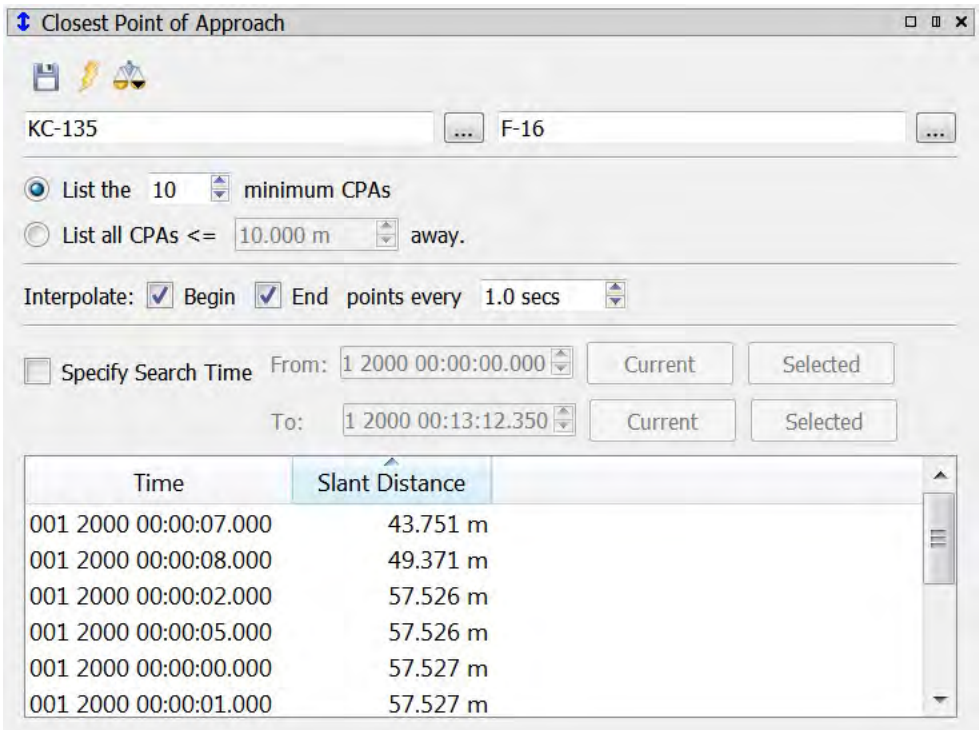


Figure 5.68: Closest Point of Approach (CPA)

5.10 Marker Tool

The **Marker Tool** allows you to drop markers in three-dimensional space on the globe. It is useful when you want to rapidly annotate target of interest (TOI), landmarks, and/or waypoints. The marker tool will enable you to drop markers on platforms in your loaded scenario. You can also set time duration in the **Time Span** section of the dialog.

The Marker tool supports both the geodetic (LLA) and MGRS format.

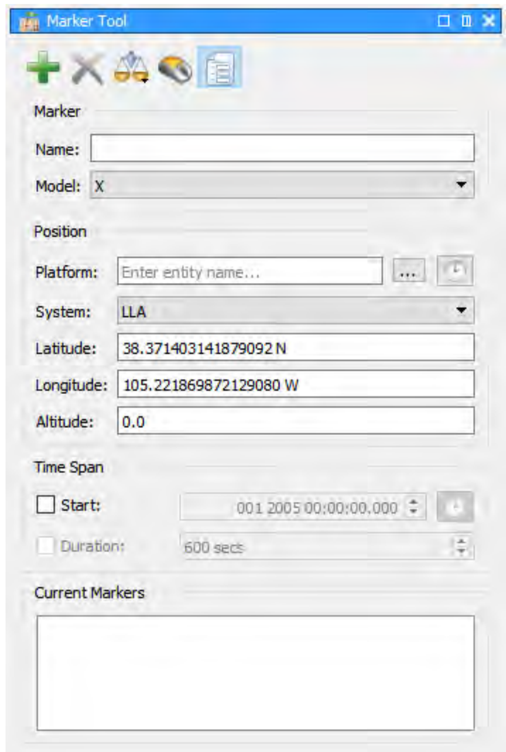


Figure 5.69: Marker Tool Dialog

5.10.1 Adding a Marker

Adding a marker is quick and easy:

1. Toggle the **Cursor Position** button.
   Notice that the Latitude and Longitude in the Marker Tool dialog will update as you move your cursor across the screen. The altitude will update if you have an elevation layer.

2. Click the specified location you want to add a marker.
   This will set the Latitude and Longitude (also Altitude with elevation) in the Marker Tool dialog.

3. Set a unique **Name** and select your desired **Model**.

4. Click the **Create** button to drop the marker.

NOTES:

- By default, the added markers are not dynamically scaled and therefore may be too small to see. To fix this, either set the **Dynamic Scale** of the markers in **Prefs Tool** or press the D key to dynamically scale all platforms (including markers).

- When **Cursor Position** is toggled, you can double-click in any area in the **Main Window** to drop a marker.

The markers have the same characteristics as a platform. You can customize the appearance of your markers in **Prefs Tool** under the **Platform** tab.

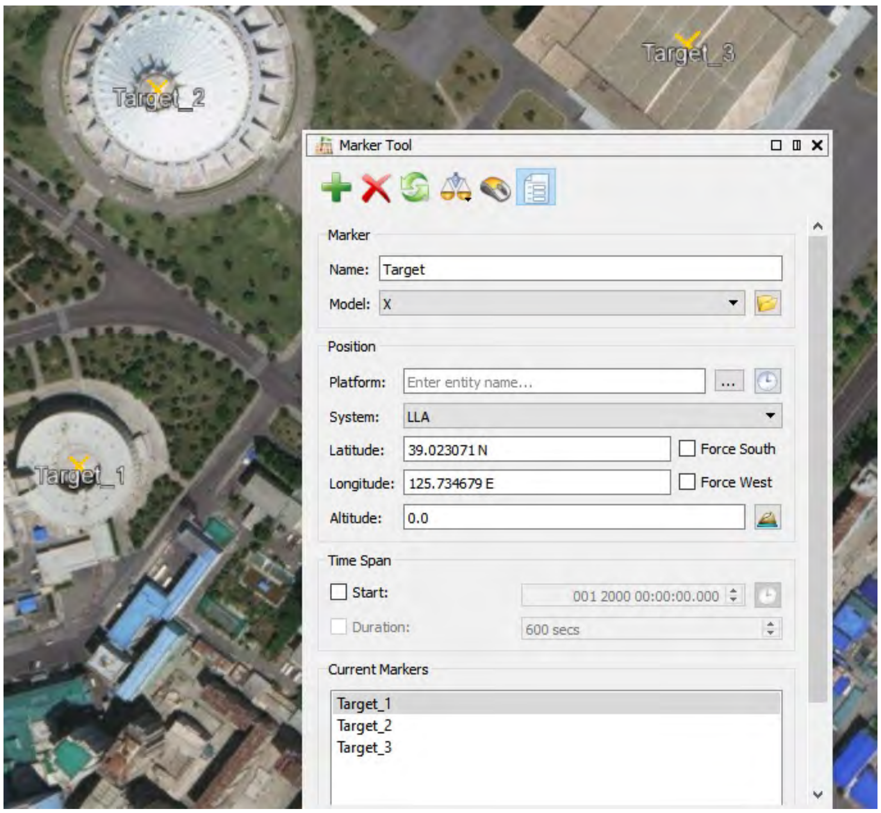


Figure 5.70: Example Markers

5.10.2 Deleting a Marker

1. Toggle the **Marker List** button. The **Current Markers** list will appear.

2. Highlight the marker by clicking it. If you want to select multiple markers, hold the Ctrl key while clicking desired markers to delete.

3. Click the **Remove** button to delete selected marker/s.

5.11 GOG Tool

The **GOG Tool** is available from the toolbar menu **Tools**. It allows loading, exporting, and changing display options for both absolute and relative overlay files. For detailed information about GOGs, refer to Chapter 8.

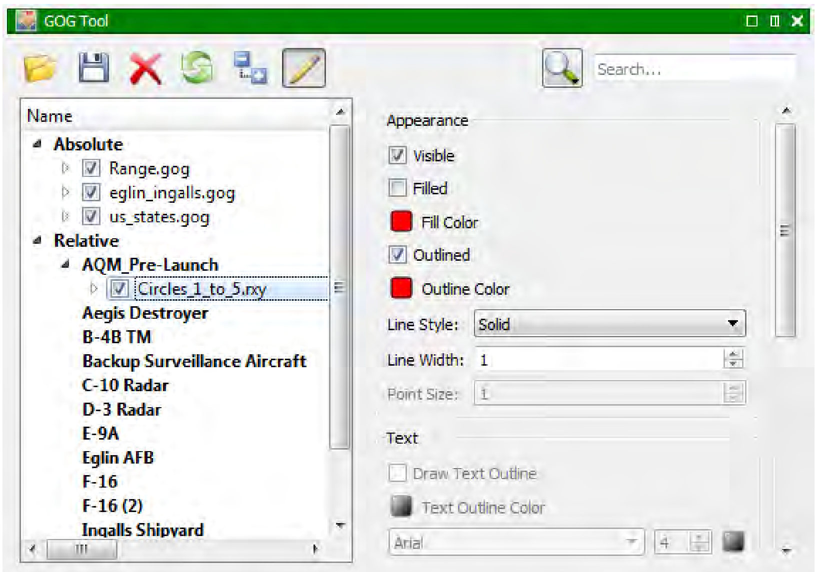


Figure 5.71: GOG Tool

Use the toolbar icon to load a GOG file. If a platform is selected in the tree when the load occurs, the GOG will be loaded as relative to that platform. Any absolute shapes contained in the GOG file will be rejected if loaded this way, as absolute shapes cannot be relative to a platform. Use the toolbar icon to export a GOG file. All GOGs and shapes selected in the tree when the save occurs will be saved into one file.

The GOG tool also has options to filter the GOG tree, as shown in Figure 5.72. To open the menu of filtering options, press the right-most button on the toolbar. The first three options control the behavior of the search bar to the right of the button. **Filename and Platform Name** is the default. While in this mode, any GOGs or platforms that don't match the text entered into the search bar will be filtered out, and the **Search Options** button icon will be set to . **Filename** will filter out only the GOGs that don't match the entered text and change the button icon to . **Platform Name** will filter out only the platforms that don't match the entered text and change the button icon to . **Hide Platforms without GOG** is a separate setting that filters out any platforms which have no relative GOGs attached.

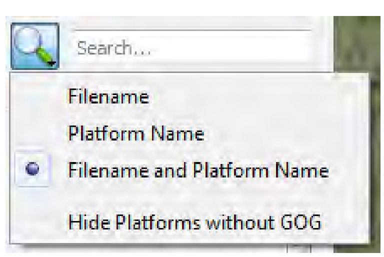


Figure 5.72: GOG Tool Filtering Options

5.12 Reference Grids

SIMDIS supports four reference grid systems:

- **Geodetic** - Latitude/Longitude Grid
- **GARS** - Global Area Reference System
- **MGRS** - Military Grid Reference System
- **UTM** - Universal Transverse Mercator

You can toggle the grid display by clicking on the **Reference Grid** icon in the **Main Toolbar**. There are two ways to change the Reference Grid:

1. Via **Settings > Reference Grid**.
2. Drop-down menu of the **Reference Grid** icon.

Figure 5.73: Reference Grid drop-down menu

Toggle the **Advanced** icon in the **Settings** dialog toolbar to customize the grid (such as grid lines size, color, and resolution profile).

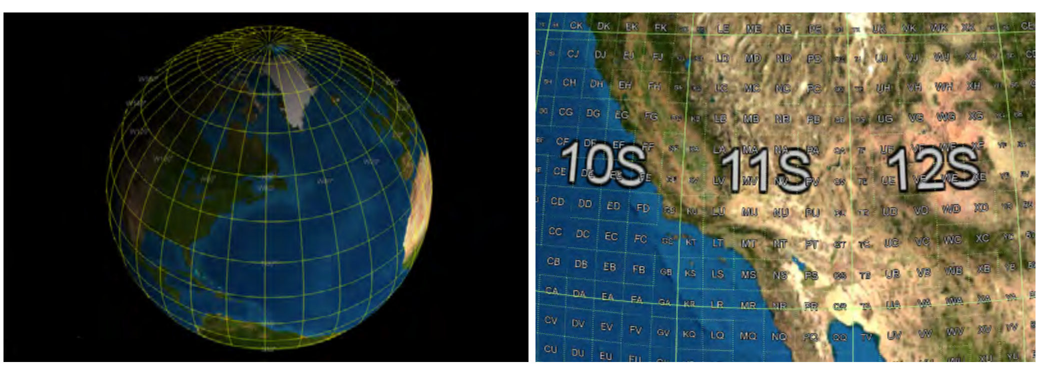


Figure 5.74: Geodetic vs. MGRS Grids

NOTE: Display options in MGRS may require a restart of SIMDIS to take effect.

5.13 Remote Control

SIMDIS supports the capability of remote-controlled instances of SIMDIS. An instance of SIMDIS is defined as one copy of the SIMDIS application running on one computer. This capability allows one SIMDIS application instance, known as the server, to control the appearance of other network connected SIMDIS applications instances, known as clients.

Remote Control Server is accessible via **Display > Remote Control Server** and Remote Control Client via **Display > Remote Control Client**.

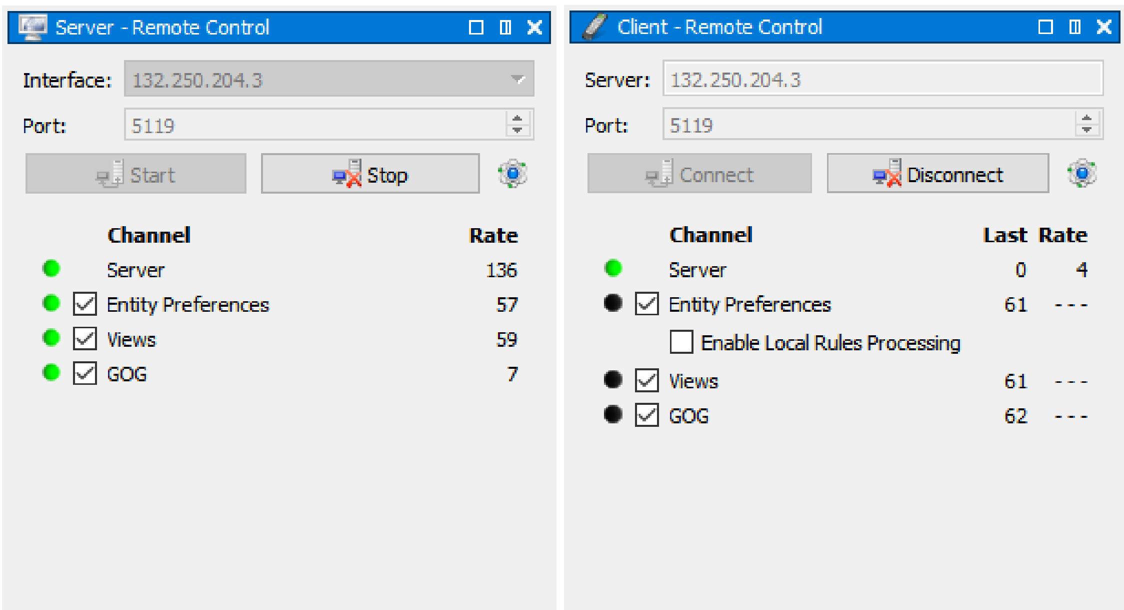


Figure 5.75: Remote Client-Server Connection

The advantages of this type of configuration are the reduction of operating cost as one operator now controls SIMDIS for multiple viewing sites. Fewer operators are required to control the display and fewer operators need to be trained to control the display. This would translate into a reduced operating cost as the number of SIMDIS operators would remain fixed as new sites increase.

5.13.1 Specification

The synchronization of multiple SIMDIS application instances is achieved through a client-server based communication infrastructure. The initial implementation provides TCP and UDP communication components however, the system design allows for use of different communication resource types.

Remote controlled SIMDIS employs a TV channel-like system where each designated SIMDIS server acts as a channel to which a client can subscribe. The channel-based system allows multiple SIMDIS control servers to coexist on the same network. Operators at remote sites may easily change the subscribed channel, or temporarily suspend a subscription to allow local manipulation of the display, through a simple user interface. The figure below illustrates the channel functionality.

You can control the data to publish and receive by toggling the checkboxes in the **Channel** column.

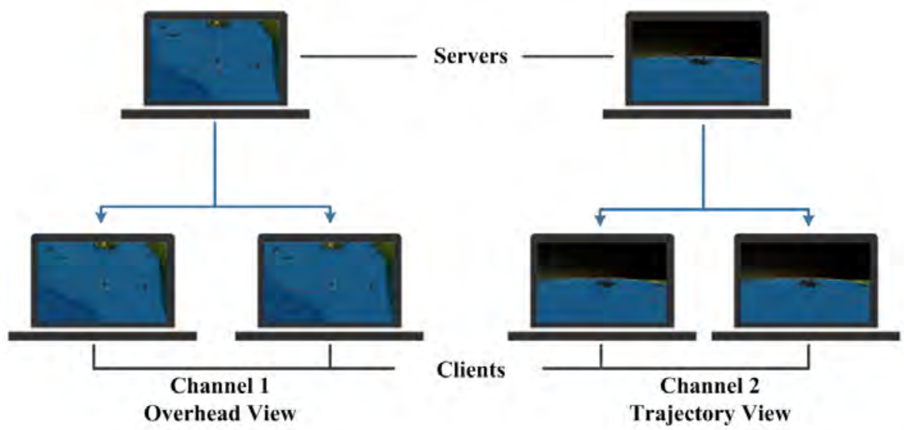


Figure 5.76: Remote Control Channel Functionality

The **Channels** allow for flexibility in amount and data type passed between **Remote Control** server and its client(s).

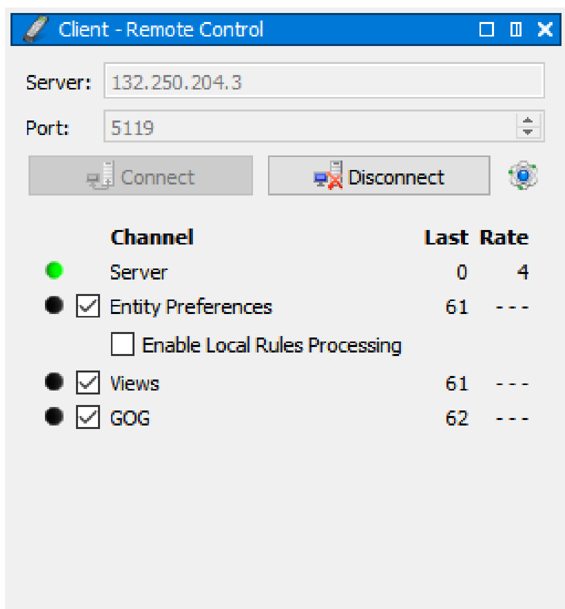


Figure 5.77: Client Connection

The **Enable Local Rules Processing** checkbox lets a **Client** toggle whether the local preference rules are processed. This is only useful when connected to the **Entity Preferences** channel. When it is checked, the local rules may cause conflicting interactions with the **Server** preference rule values.

5.14 RF Propagation

The **RF Propagation** tool allows visualization of supplied **AREPS** (Advanced Refractive Effects Prediction System) data and RF propagation data provided by the plug-in API, to visualize the spread of **Electromagnetic** (**EM**) data through the environment. The RF Propagation Tool provides efficient reuse and recalculation of propagation data based on radio frequency (**RF**) parameters, environmental conditions, antenna location, and pointing angle. For more details about the types of RF propagation, refer to Section E.7.

You can access the RF Propagation tool via **Tools > RF Propagation** or the default hot key Ctrl + Alt + R .

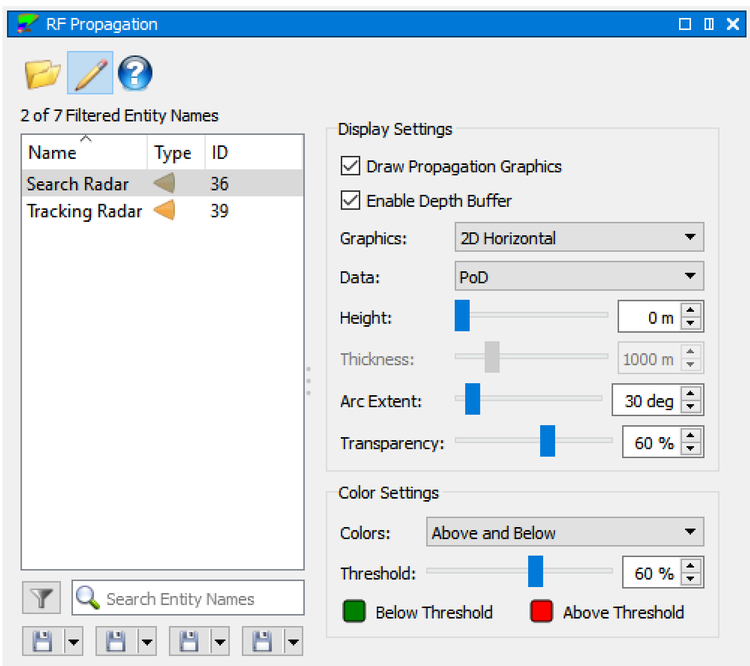


Figure 5.78: RF Propagation Tool

5.14.1 Propagation Graphics

The propagation graphics are recalculated based on the changes in the RF Propagation dialog. The radio frequency propagation is visualized by the type of graphics selected in **Graphics** combo-box. The values are reflected based on the **Threshold** parameter. The **Threshold** allows calculation of the propagation from **-300 dBsm** to **300 dBsm**.

NOTES:

- Drop-down the **Graphics** combo-box and mouse-over each type to see the tool tips.
- The **Transparency** works inversely with opacity, higher value means less visibility.

Figure 5.79: 2D vs 3D Propagation Graphics

5.15 Propagation Example

The following example will guide you on how to display RF propagation based on an AREPS file.

1. Start **SIMDIS**.
2. Load `$(SIMDIS_DIR)/demos/RFProp/AREPS/RCSnAREPS.zip`.
3. Open the **RF Propagation** dialog via **Tools > RF Propagation** or the Ctrl + Alt + R hot key.
4. Double-click on the **Search Radar** to center the view to the **Ship**, the beam origin.
5. Click the **Play** button.
6. Verify that the **Draw Propagation Graphics** checkbox is checked in the RF Propagation dialog. Notice the **Search Radar** beam will appear as the scenario plays.
7. By default, the **2D Horizontal** is selected in the **Graphics** combo-box and **PoD** (Probability of Detection) is selected in the **Data** combo-box.
8. Change the **Data** combo-box from **PoD** to **CNR** (Clutter-to-Noise Ratio). This will recalculate the propagation display, shown in the left image in Figure 5.79.
9. Switch the **Graphics** from **2D Horizontal** to **3D Texture**. The propagation displayed should be similar to the right image in Figure 5.79.
10. Change the **Colors** under the **Color Settings** from **Above and Below** to **Gradient**. The different colors displayed is the ratio of the clutter-power to the noise-level, shown in Figure 5.80.
11. Change the **Data** from **CNR** to **Loss**.
12. Change the **Thickness** to **150 m**. The **Below Threshold** color is evident on areas where the beam is projected on land. This display visualizes the propagation loss when obstructed.
13. When you have elevation data, in this example ReadyMap SRTM was used, the propagation loss is evident by the presence of land, shown in Figure 5.81. The propagation display will reflect the loss-of-propagation based on the **Thickness** parameter with elevation data.

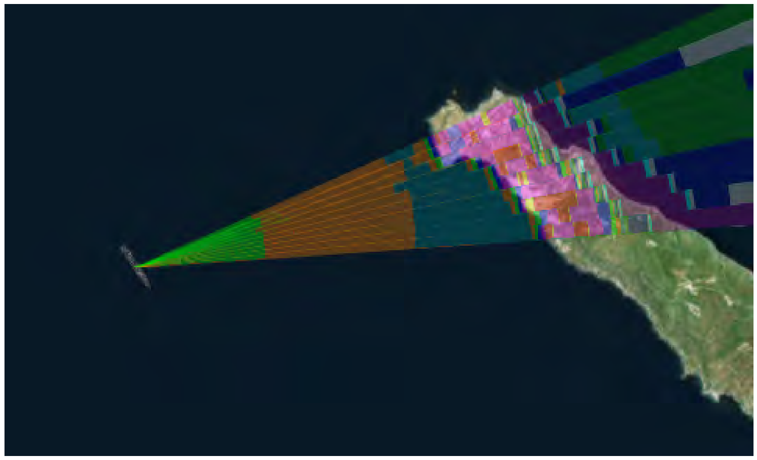


Figure 5.80: 3D Texture using Gradient color

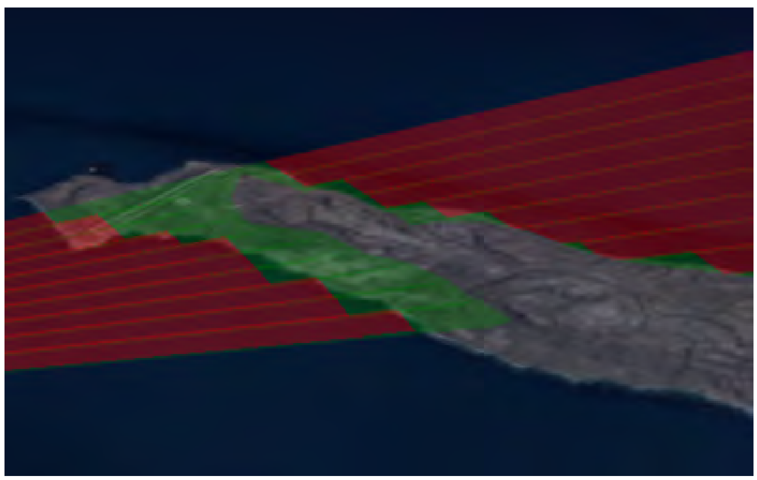


Figure 5.81: Loss with Elevation Data

5.16 Data Scripts

Data Scripts are a way to script your SIMDIS experience using **Python**. It is similar to the SIMDIS Scripting and Plug-in API in that it provides a way for you to customize SIMDIS by creating and modifying data, changing various display properties, and programmatically altering the presentation.

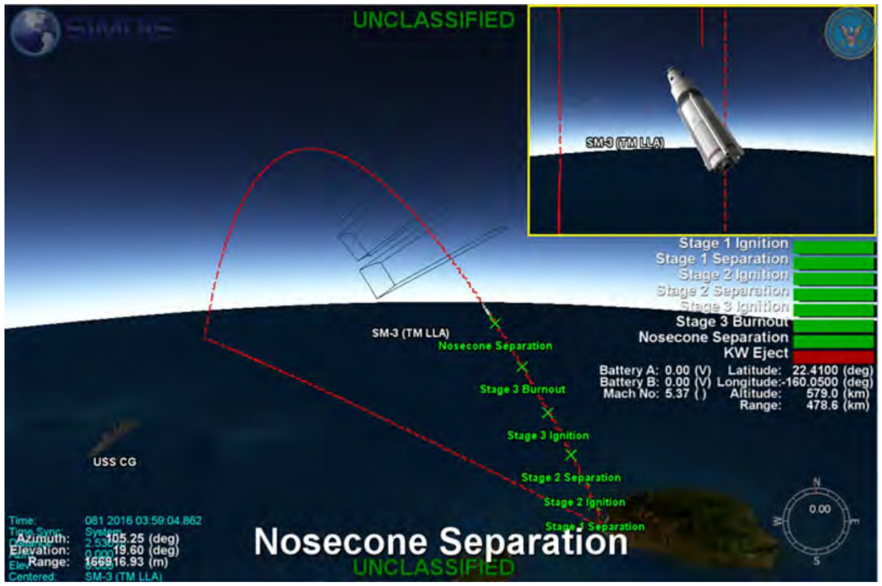


Figure 5.82: Screen Output from Scripting Example

Data Scripts provide a subset capabilities of the SIMDIS Plug-in API through a Python interface. The design goal of the SIMDIS Data Scripting API is to provide an interface to commonly used features of the Plug-in API for display customization without requiring recompilation of C++ code as using the Plug-in API requires.

With the user interface Figure 5.83 users are able to see which scripts are loaded and can stop loaded scripts from running. Refer to Data Script documentation for detailed instructions on implementing data scripts in your display.

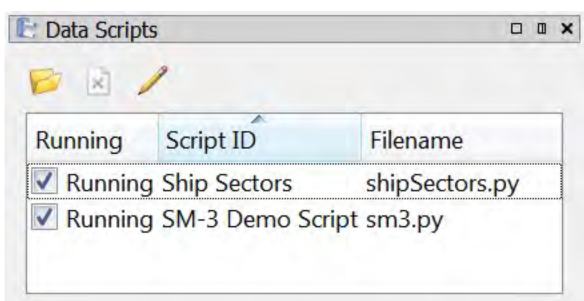


Figure 5.83: Data Scripts Dialog

5.17 Articulations

The **Articulations Tool** manages models with built-in articulation points that can be manually manipulated. Articulations Tool is accessible via **Tools > Articulations**.

There are three different types of control nodes supported:

- DOF Transform (e.g. Arm mount tilt)
- MultiSwitch (e.g. Payloads or damaged states)
- Sequence/Animation (e.g. Propellers)

To select a model with articulation nodes:

1. Open the Articulations Tool
2. Click the ... button, **Select Entity** dialog will appear
3. Select a model with articulation nodes

The articulation nodes will populate the **Articulations Tool**. Please refer to Figure 5.84 as an example.

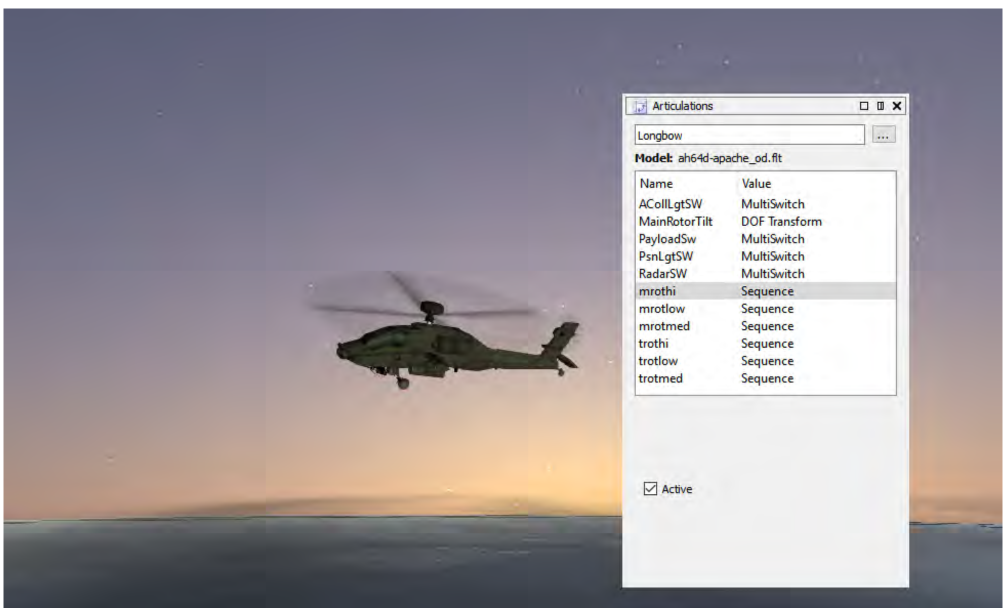


Figure 5.84: Articulations Tool

NOTE: The articulations are not saved and do not persist when SIMDIS is closed.

5.18 Stereoscopic Visualization

Support for 3-D Stereoscopic Visualization (via GL context enable/disable) has been added to SIMDIS.

Select Stereoscopic  from the Tool Bar **Views Menu** to display the **Stereoscopic Dialog** shown in Figure 5.85.

Figure 5.85: Stereoscopic Visualization Dialog

5.19 Hot Key Configuration

Hot keys management is accessible through the **Hot Keys** dialog from the toolbar menu **Tools**, allowing the user to easily search and configure the SIMDIS shortcuts, and also save and load hot key settings configuration files.

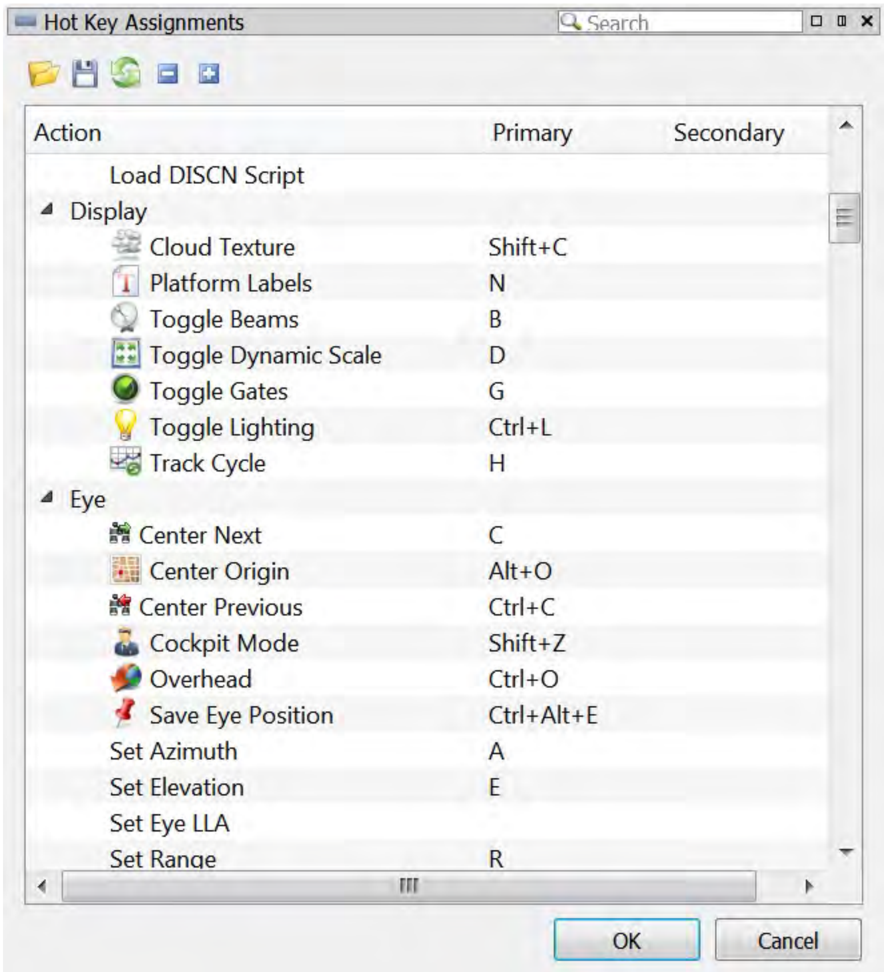


Figure 5.86: Hot Key Assignments Dialog

5.20 Python Label Editor

In Prefs Tool, you can edit and modify the display of the entity label using Python code. Toggling the **Use Python Label** as shown in Figure 5.87 and selecting the **Python Label Editor** under **Display Fields** executes the Python Label Code Editor extension. For Python beginners, templates are provided to assist you in composing syntactically correct Python code. This is possible for the entity labels and for Hover Text. The **Python Label Code Editor** supports syntax highlighting features.

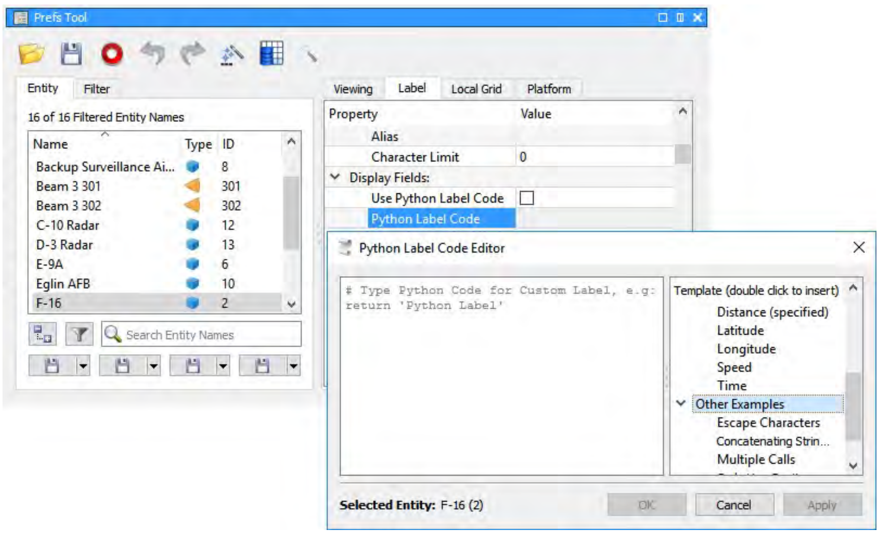


Figure 5.87: Python Label Dialog

5.20.1 Python Label Format

In the list of templates, you can choose the format of the platform's data. As an example Figure 5.88, the **alt** variable is used to store the altitude in **feet** and the **unit** variable stores the unit string of **ft**.

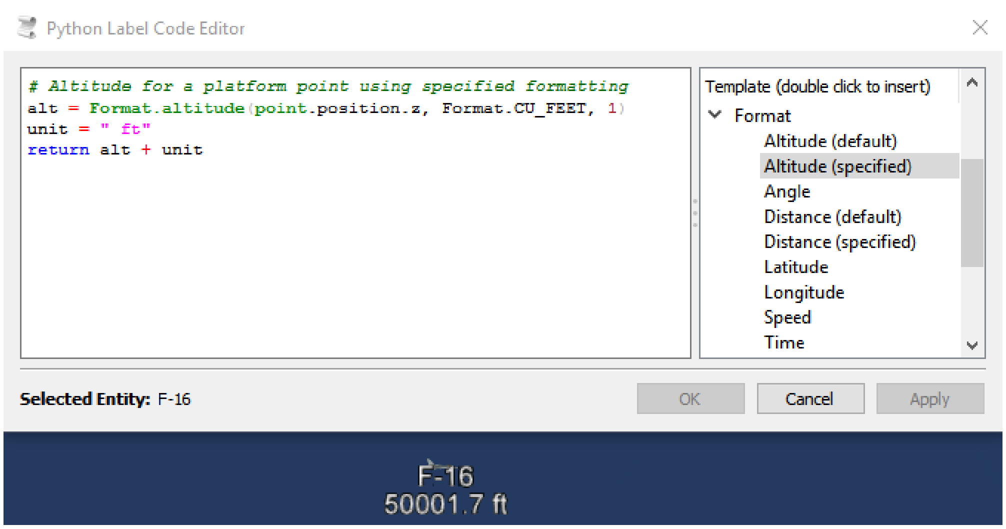


Figure 5.88: Python Label Custom Altitude Format

5.20.2 Local and Global Variables

There are local and global variables available for use, listed below:

- `id` **(LOCAL)** ID of the entity. See the template called "Entity ID" for example usage.
- `point` **(LOCAL)** Data point for the entity. Example usage can be found in many of the templates in the "Format" group.
- `GlobalCache` **(GLOBAL)** Python dictionary available to all Python label code. Keys added to the dictionary should be globally unique. The dictionary is cleared at the start of each scenario.

5.20.3 SIMDIS Python File

A user can create and store custom Python methods for use in the SIMDIS Python Label Code via the SIMDIS Python file. The Python methods defined in this file are referenced using the **simdis** namespace (e.g. **simdis.methodName()**). The file must be named **simdis.py** and must be in the `$(SIMDIS_HOME)` directory. The file can reference other modules in `$(SIMDIS_HOME)` directory. The file is loaded once when the first Python code is executed.

NOTE: If changes are made to the SIMDIS Python File (**simdis.py**) while SIMDIS is open, SIMDIS must be restarted in order for this file to be reloaded.

5.20.4 SIMDIS Python File Example

You can add or append custom entry to a platform label. This example will guide you to add a custom label using the SIMDIS Python File:

1. Open `$(SIMDIS_HOME)/simdis.py` in a text editor (e.g. Notepad). Create the file if it does not exist.
2. Add the following lines of code:

```
def f16Name():
  return "Fighting Falcon"
```

3. Save and close the **simdis.py** file.
4. Start **SIMDIS**.
5. Load `$(SIMDIS_DIR)/demos/SIMDIS/Users/NG/BMD_Demo.asi`.
6. Left-click the  button on the toolbar or press Alt + S keys to open the **Prefs Tool**.
7. In the **Entity** tab, double-click the **F-16(2)** platform.
8. Left-click the **Label** tab and collapse **Display Fields**.
9. Check the **Use Python Label Code** property check-box.
10. Left-click the **Python Label Code**, the **Python Label Code Editor** will appear.

11. Type the following line in the editor: `return simdis.f16Name()`.

12. Left-click the **OK** button to apply and close the **Python Label Code Editor** dialog.

13. Toggle the **Platform Labels** or press the N key, the default hot key.

14. Zoom-in to the **F-16(2)** platform, the display should be similar to Figure 5.89.

Figure 5.89: Custom Label

5.20.5 SIMDIS Python File Calculation Example

You can create a method to perform calculations based on inputs of the entity data. This example will calculate the actual altitude and additional custom offset to the platform:

1. Open `$(SIMDIS_HOME)/simdis.py` in a text editor (e.g. Notepad). Create the file if it does not exist.

2. Add the following lines of code:

```
def addAltitudeOffset(altitude):
  offset = 100 # Custom altitude offset
  withOffset = int(altitude) + offset
  return "Altitude: " + str(withOffset)
```

   NOTE: The python code **int()** parses the value to an integer and **str** to a string. The **str** is necessary to concatenate with other strings, in this example the word **Altitude:**.

3. Save and close the **simdis.py** file.

4. Start **SIMDIS**.

5. Load `$(SIMDIS_DIR)/demos/SIMDIS/Users/NG/BMD_Demo.asi`.

6. Left-click the button on the toolbar or press Alt + S keys to open the **Prefs Tool**.

7. In the **Entity** tab, double-click the **F-16(2)** platform.

8. Left-click the **Label** tab and collapse **Display Fields**.

9. Check the **Use Python Label Code** property check-box.

10. Left-click the **Python Label Code**, the **Python Label Code Editor** will appear.

11. Type the following lines in the editor:

```
actualAltitude =  Format.altitude(point.position.z)
return simdis.addAltitudeOffset(actualAltitude)
```

12. Left-click the **OK** button to apply and close the **Python Label Code Editor** dialog.

13. Toggle the **Platform Labels** or press the N key, the default hot key.

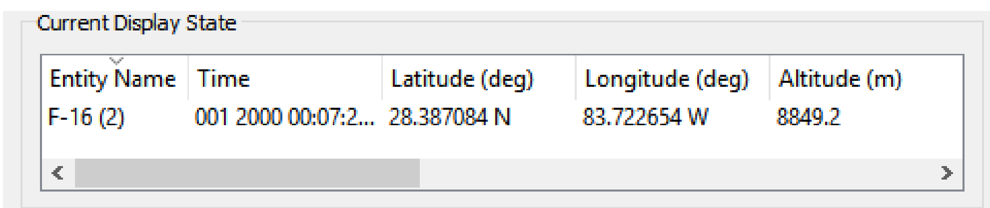
14. Zoom-in to the **F-16(2)** platform, the display should be similar to Figure 5.90.

Figure 5.90: Custom Label with Altitude Offset

15. To verify the added altitude offset, left-click the button or press the Alt + D to open the **Data Browser** dialog.

16. Left-click the **F-16(2)** platform and left-click **Platform Data** tab.

17. Under the **Current Display State**, verify the altitude. It should display the actual altitude (without the offset), shown in Figure 5.91.


Current Display State

Entity Name	Time	Latitude (deg)	Longitude (deg)	Altitude (m)
F-16 (2)	001 2000 00:07:2...	28.387084 N	83.722654 W	8849.2

Figure 5.91: Current Display State in Data Browser

NOTE: The altitude in this example of the **F-16(2)** platform does not change throughout the scenario. If the example is applied to a platform with varying altitude, the label will update accordingly.

5.21 Hook Track

The **Hook Track** provides right click context menu to hook and unhook entities. It pops up a distinct dialog with the information about the hooked entity as shown in Figure 5.92.

Figure 5.92: Hooked Track

The Hook Track dialog contents can be defined via **Prefs Tool > Label > Hook Text**.

5.21.1 Custom Hooked Track Content

You can create your custom hooked track content by using the Python Label Editor. For the following example, we will customize the **Latitude** and **Longitude** to `degrees-minutes` format and **Speed** in `knots` or `kts`.

1. Load a scenario with platforms

2. Open **Prefs Tool** 

3. Click **Entity** tab and select an entity (for this example we will use a platform)

4. Click the **Label** tab and scroll down to the **Hook Text:** property, as shown in Figure 5.93

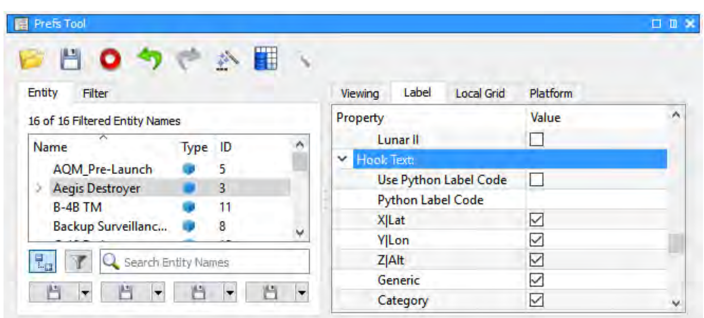


Figure 5.93: Hook Text in Label Tab

5. Uncheck all the checked checkboxes in the **Value** column

6. Check **Use Python Label**
7. Click the **Python Label Code**, the **Python Label Editor** will appear
8. Enter the following Python code in the **Python Label Editor**:

```
LATITUDE_STRING  = "Latitude:  "
LONGITUDE_STRING = "Longitude: "
SPEED_STRING     = "Speed:     "
SPEED_UNIT       = " kts"
lat = Format.latitude(point.position.x, Format.CU_DEGREES_MINUTES, 3)
lon = Format.latitude(point.position.y, Format.CU_DEGREES_MINUTES, 3)

# Speed for a platform point using default formatting
vx = point.velocity.x
vy = point.velocity.y
vz = point.velocity.z
spd = Format.speed((vx*vx+vy*vy+vz*vx)**0.5, Format.CU_KNOTS, 1)

hookLabel = '\n'      + LATITUDE_STRING + lat + '\n'
hookLabel = hookLabel + LONGITUDE_STRING + lon + '\n'
hookLabel = hookLabel + SPEED_STRING + spd + SPEED_UNIT + '\n'

return hookLabel
```

NOTE: The `'\n'` in python is newline.

9. Click **OK** to apply and exit out of the **Python Label Editor**

Your hooked track dialog should look similar to Figure 5.94.

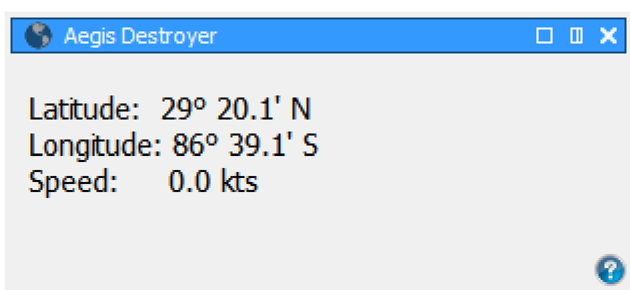


Figure 5.94: Custom Hook Track Dialog

5.22 Geo-Filters

Geo-Filters is now an integrated part of SIMDIS. **Geo-Filters** allows platforms to be included or excluded based on location. A common set of geo-filters are shared among all the features. **Geo-Filters** must have names and the names must be unique. A filter can be defined in one feature and accessed by the other features. Geo-filters can be defined in the Geo-Alerts plug-in, the Geo-Filter Service plug-in, and other customer specific plug-ins. Geo-filters can be used in the previously mentioned plug-ins plus the File Writer plug-in and the HDF5 plug-in. See the documentation for the plug-ins to learn how each plug-in uses **Geo-Filters**.

This Page is Intentionally Left Blank

Chapter 6

Terrain and Imagery

SIMDIS can efficiently render gigabytes of detailed terrain altitude and imagery data. This chapter will describe the details of how SIMDIS displays this data.

6.1 osgEarth

SIMDIS utilizes a geospatial SDK and terrain rendering engine called osgEarth that is built on OpenSceneGraph. The osgEarth SDK provides a host of built-in services to streamline the display of imagery and terrain files, and offers a wide range of options and support for most of the common types of geospatial data, as well as allowing for easy implementation of custom plug-ins for more specialized file types. By utilizing osgEarth, SIMDIS leverages all the extra features of a reliable open source terrain engine that is actively maintained to utilize the latest open mapping standards, technologies, and data. For more information about osgEarth, refer to the osgEarth Documentation.

6.1.1 Earth File

SIMDIS supports loading both the SIMDIS version 9 terrain configuration files, as well as the more generic **.earth** file format that osgEarth uses to describe geospatial data. The osgEarth earth file format uses simple XML tags to define the various image, elevation, and model layers to load into the 3D display. An Earth file allows you to specify:

- The type of map to create (geocentric or projected).
- The image, elevation, vector, and model sources to use.
- Where the data is cached.

NOTE: There are various configuration options for the **driver** type in the **.earth** file such as map lighting, tile size, overlay texture size, cache policy, location, and adjustments to the spatial reference system.

An example of an XML **.earth** file that reads data from a GeoTIFF file on the local file system and renders it as a geocentric round Earth scene:

```
<map name="MyMap" type="geocentric" version="2">
  <image name="bluemarble" driver="gdal">
    <url>world.tif</url>
  </image>
</map>
```

Additional information about the earth file format is found in the osgEarth Documentation.

6.1.2 Cloud Imagery

The cloud imagery can be added to SIMDIS using a WMS feed **Map Editor > Add Image Layer**, or overlaying a still image. Weather maps in Map Editor can be changed by editing the **weatherMaps.config** file in `$(SIMDIS\_DIR)/data/sdTerrain`.

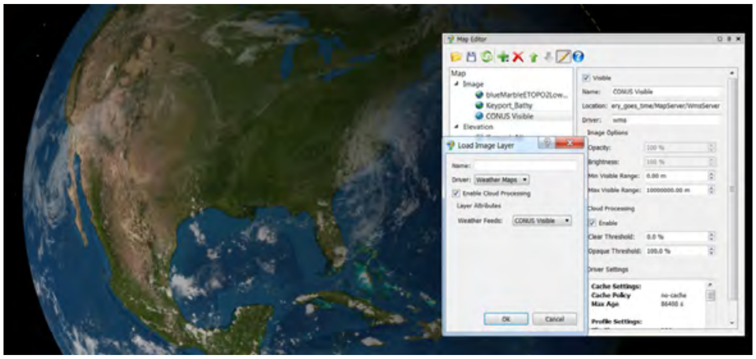


Figure 6.1: CONUS Weather Layer through Weather Maps

For multi-colored weather imagery, the `<chroma_key>` tag is used to make black or white background color transparent. We will use a NEXRAD image file of Hurricane Irma in the following example as an `<image>` in the **.earth** file:

```
<image>
  <driver>gdal</driver>
  <url>Irma.tif</url>
  <interp_imagery>false</interp_imagery>
  <name>Irma</name>
  <cache_policy>
    <usage>read_write</usage>
    <max_age>32400000</max_age>
  </cache_policy>
  <color_filters>
    <chroma_key r="0" g="0" b="0" distance=".1"/>
  </color_filters>
</image>
```

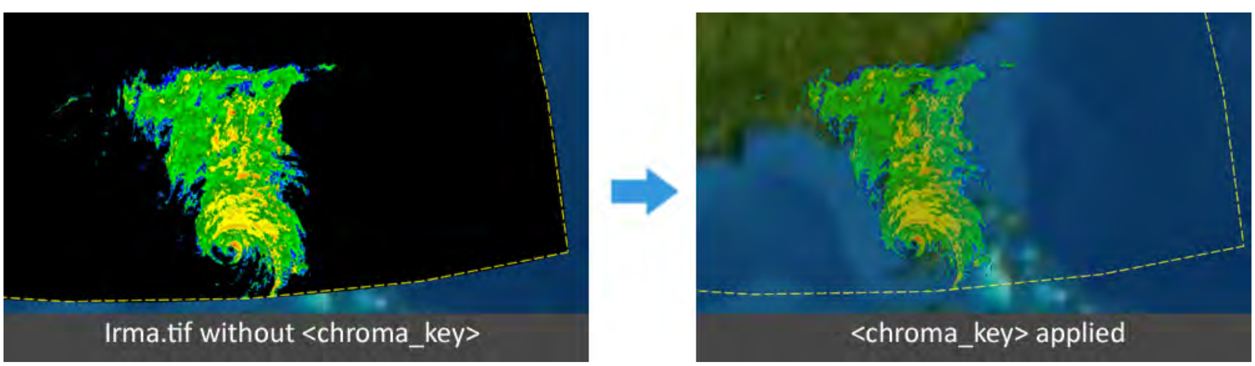


Figure 6.2: Remove Background Using Chroma

6.1.3 Time Stamped Imagery

The **.earth** file controls imagery via time using the <time> tag, that uses a UTC time reference. Time stamped imagery can be grouped together using the optional <time_group> tag. Timed images which don't specify a group are considered to be in the same group. At most one image is shown from each group depending on time. No timed images are shown before the earliest timestamp. The example below will show a Composite NEXRAD **GeoTIFF** via the GDAL driver starting at a UTC time of `2017-09-10 T16:00:00Z`.

```
<image>
  <driver>gdal</driver>
  <time>2017-09-10T16:00:00Z</time>
  <time_group>NEXRAD</time_group>
  <url>n0r_201709101600.tif</url>
  <name>n0r_201709101600.tif</name>
  <cache_policy>
    <usage>read_write</usage>
    <max_age>32400000</max_age>
  </cache_policy>
  <visible>true</visible>
  <opacity>0.6</opacity>
  <color_filters>
    <chroma_key>
      <r>0</r>
      <g>0</g>
      <b>0</b>
      <distance>0.1</distance>
    </chroma_key>
  </color_filters>
  <texture_compression>on</texture_compression>
</image>
```

The NEXRAD composite can be download from Iowa State University. The **.png** and **.tif** hosted from their site do not load in either SIMDIS or Global Mapper. You have to convert them to a **GeoTIFF** via a server-side conversion.

https://mesonet.agron.iastate.edu/request/gis/n0r2gtiff.php?dstr=2017091016

Where **dstr** specifies the `{year} {month} {day} {hour} {minute}` in UTC of the requested raster. Raster images are available in 5-minute increments. The return file will be a compress zip archive that contains the **.tif** image. The resulting **.tif** image will have a **"No Data"** areas that are black. In order to remove these, use the <chroma_key> filter to make it transparent. Refer to Section 6.1.2 for more information about <chroma_key>.

6.1.4 Periodically Refreshing Layers

Imagery and elevation layers can be configured to refresh their data from the source at a regular interval using the `<refresh>` tag. This tag takes an integer argument as the time in minutes between each refresh. For data that changes over time, such as weather data pulled from the internet, this allows the layer to stay up to date automatically.

```
 <WMSImage>
   <refresh>5</refresh>
   <name>Layer_name</name>
   <cache_policy>
     <usage>read_write</usage>
     <max_age>900</max_age>
  </cache_policy>
  <opacity>0.5</opacity>
  <url>http://example.com/WMSServer</url>
  <layers>1</layers>
  <format>png</format>
  <srs>EPSG:4326</srs>
  <transparent>true</transparent>
</WMSImage>
```

6.2 Preparing Data

If you have native geospatial data that you would like to view in SIMDIS, you can usually use the GDAL driver. If you plan on doing this, try loading the data as-is first. If the usability of SIMDIS is too slow then the data will need to be optimized for tiled access by osgEarth.

6.2.1 Converting Data

The **osgearth_conv** is a command line tool that converts the geospatial data from one driver format to another. This tool is useful for converting TMS repositories or SIMDIS 9 DB files into native MBTiles files. For example, to convert the **.tif** file to a more efficient **mbtiles**:

```
osgearth_conv --in driver gdal --in url world.tif --out driver mbtiles --out filename world.db
```

To download **WMTS** and **WMS** data for off-line use, refer to SIMDIS FAQ-70: Converting WMTS and WMS data for off-line display in SIMDIS.

NOTES:

- For detailed information about **osgearth_conv**, refer to the osgEarth Documentation.
- Downloading data from public servers means abiding by the provider's terms of service.

6.2.2 Reprojecting Data

SIMDIS, by virtue of osgEarth, will reproject your data on-the-fly if it does not have the necessary coordinate system. For instance, if you are trying to view a UTM image on a geodetic globe (epsg:4326). However, SIMDIS will run much faster if your data is already in the correct coordinate system. You can use any of the following tools to reproject data such as GDAL, Global Mapper or ArcGIS.

For example, to reproject a UTM image to geodetic using gdal_warp:

```
gdalwarp -t_srs epsg:4326 my_utm_image.tif my_gd_image.tif
```

6.2.3 Internal Tiles

Typically formats such as GeoTiff store their pixel data in scanlines. This generally works well, but because of the tiled approach osgEarth uses to access the data, you may find that using a tiled dataset will be more efficient as osgEarth does not need to read nearly as much data from disk to extract a tile.

To create a tiled GeoTiff using gdal_translate, issue the following command:

```
gdal_translate -of GTiff -co "TILED=YES" myfile.tif myfileTiled.tif
```

6.2.4 Overviews

Adding overviews, also called 'pyramids' or 'rsets', can sometimes increase the performance of a datasource in osgEarth. You can use the gdaladdo utility to add overviews to a dataset.

For example:

```
gdaladdo -r average myimage.tif 2 4 8 16
```

Another way to speed up imagery and elevation loading in SIMDIS is to build tile sets. In fact, if you want to serve your data over the network, this is the only way.

This process takes the source data and chops it up into a quad-tree hierarchy of discrete tiles that osgEarth can load very quickly. Normally, if you load a GeoTIFF, osgEarth has to create the tiles at runtime in order to build the globe. Doing this beforehand means less work for osgEarth when you run SIMDIS.

6.3 Finding Data

There are many sources of geospatial data. These sources span government agencies, academics, and private companies. The following is a partial list of online data sources.

6.3.1 Raster Data

- ReadyMap Tiles - 15m imagery, elevation, and street tiles for osgEarth developers.
- USGS National Map - Elevation, orthoimagery, hydrography, geographic names, boundaries, transportation, structures, and land cover products for the US.
- NASA EOSDIS - NASA's Global Imagery Browse Services (GIBS) replaces the agency's old JPL OnEarth site for global imagery products like MODIS.
- NASA BlueMarble - NASA's whole-earth imagery (including topography and bathymetry maps)
- NRL DMAP - US Naval Research Lab's Digital Mapping, Charting and Geodesy Analysis Program.
- Natural Earth - Free vector and raster map data at various scales
- Virtual Terrain Project - Various sources for whole-earth imagery
- Geospatial Data Gateway - Provides access to a map library of over 100 high-resolution vector and raster layers.
- National Oceanic and Atmospheric Administration - Provides navigational and nautical charts
- NOAA nowCOAST - NOAA's web mapping portal to real-time observations, forecasts, and warnings.
- GLCF - University of Maryland Global Land Cover Facility, provides access to 30m Landsat data.
- Digital Globe - Commercial imagery for sale.

6.3.2 Elevation Data

- CGIAR - World 90m elevation data derived from SRTM and ETOPO (CGIAR European mirror)
- SRTM30+ - Worldwide elevation coverage (including bathymetry) from the University California, San Diego.
- GEBCO - The General Bathymetric Chart of the Oceans.

6.3.3 Feature Data

- OpenStreetMap - Worldwide, community-sources street, and land use data (vectors and rasterized tiles)
- DIVA-GIS - Free low-resolution vector data for any country
- Natural Earth - Free vector and raster map data at various scales

Chapter 7

File Formats, Maps, and Time

The SIMDIS toolset is highly customizable and powerful. You can save and change data files and make script files to customize the display of your data your way. It is flexible supports multiple file formats to accommodate your needs.

The SIMDIS distribution also includes several sample files located in `$(SIMDIS_HOME)\demos\SIMDIS`. These files serve as examples to demonstrate various features of SIMDIS and are intended as a starting point for importing your own data into SIMDIS. The files types are described in the following sections.

7.1 File Formats

7.1.1 ASCII Scenario Input File Format (`.asi`)

The ASCII Scenario Input file format (**.asi**) is the primary file format for importing data into SIMDIS. The **.asi** files contain plain text definitions of scenario entities and their TSPI data. The **.asi** files can contain information relating to a scenario such as a reference origin, wind speed and direction, and associated sound or overlay files.

The tag value pair format uses keywords with data sets and is a ”readable” format to the human eye. The **.asi** format allows data to be both imported and exported. Appendix A describes in full detail how to write an .asi file along with providing detailed examples.

7.1.2 SCORE and PMRF Comma Separated Values (`.csv`)

Files with the **.csv** extension are loaded directly into SIMDIS. The **.csv** format is an ASCII comma separated value format. Currently, SIMDIS only accepts formats that are specific to the PMRF and SCORE ranges. On the other hand, CSV files that are saved from an Excel file can be loaded into Plot-XY for display provided they follow a specific format. The Plot-XY program expects time to be in the first column and cannot accept empty columns, or columns with textual data.

7.1.3 SIMDIS Data Initialization Scenario File (`.discn`)

The **.discn** file is a SIMDIS data initialization scenario file based on the ASI file format. The **.discn** files are useful because they can be used to load view files, terrain configuration files, preference rules and

GOG files in preparation for receiving data via DCS or a plug-in.

A **.discn** file, although it is an ASI file, is valid if no platforms are defined. All platforms loaded via a **.discn** file are treated as static placeholders until data is received from either DCS or a plug-in. Only platforms and their associated category data are supported. Beams, gates, lasers, LOBs, Custom Renderings, Data Tables, and generic data are not supported. In order for a placeholder platform to transition to a live entity, the call signs must be an exact match.

NOTE: If a placeholder call sign is not found in the live data stream, it will remain a static platform.

7.1.4 GPS Exchange Format (.gpx)

Files with the .gpx extension are the GPS Exchange format and can be loaded directly into SIMDIS. The **.gpx** is a lightweight XML based data format for the interchange of GPS data between applications and Web services on the Internet at http://www.topografix.com. Supported XML tags include **metadata**, **trk**, **trkseg**, **trkpt**, and **wpt**.

SIMDIS requires all **trkpt** data to contain the latitude and longitude attributes as well as the **ele** and **time** sub-trees. The **course** and **speed** sub-trees are optional attributes that SIMDIS will process, if available. SIMDIS also requires files containing **wpt** tags to also contain a metadata tree. The **time** sub-tree found in the **metadata** is used for the reference year setting in SIMDIS. All **wpt** tags become static points within SIMDIS.

7.1.5 Kepler Propagation Format (.kp)

Files with the **.kp** extension are used as input into the Keplerian Propagator application. The Keplerian Propagator application takes as input a reference origin, time and an Earth-centered Earth-fixed vehicle state vector and propagates its position and orientation using Kepler's third law, generating an **.asi** file.

7.1.6 NORAD Two-Line Element (`.tle`)

Files with the **.tle** extension are directly loaded into SIMDIS. These files are NORAD's Two-Line Element files. NORAD Two-Line Element is a set of data that can be used together with NASA's **SGP4/SDP4** orbital model to determine the position and velocity of an associated satellite.

The **.tle** files are loaded via the TLE plug-in. The TLE plug-in is found in the **Optional Plugins Distribution (ZIP)** under **Downloads** in the SIMDIS website.

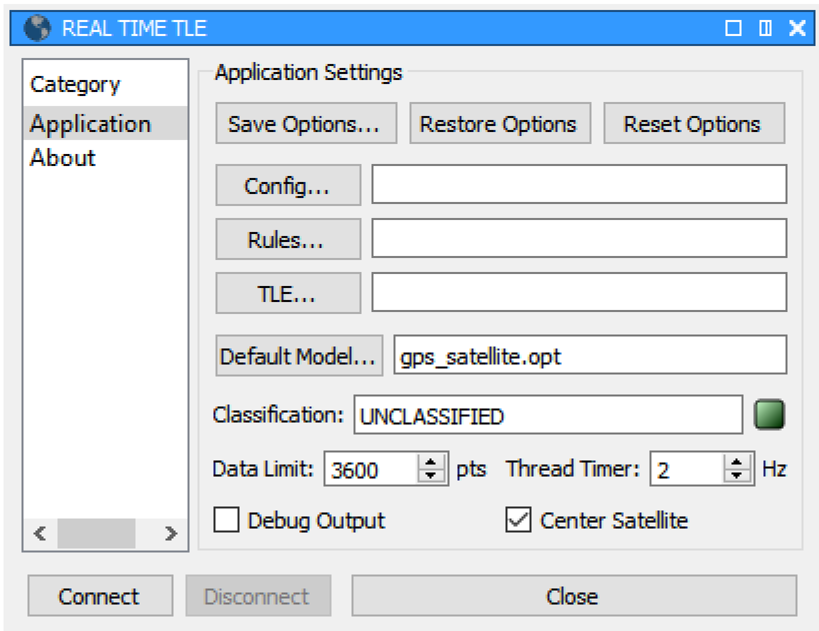


Figure 7.1: TLE Plug-in

Additional information and updated .tle files can be found at the following websites:

- http://celestrak.com/NORAD/elements/
- http://www.space-track.org

The **.tle** format accepted by SIMDIS is a three-line element set that includes the entity name in the first line. See format below:

```
Line 0: AAAAAAAAAAAAAAAAAAAAAAAA
Line 1: N NNNNNC NNNNNAAA  NNNNN.NNNNNNNN +.NNNNNNNN  +NNNNN-N +NNNNN-N N NNNNN
Line 2: N NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NN NNNNNNNNNNNN

        12345678901234567890123456789012345678901234567890123456789012345678 9
```

7.1.7 Over-the-Horizon Targeting Gold Revision D (`.otg`)

Files with the **.otg** extension are used as input into the ConvertOTG2ASI application. The **.otg** files are based on the Over-The-Horizon Targeting GOLD Revision D message format (OTG-D).

The OTG-D is the primary message format for Tactical Data Processor (TDP) to TDP information exchange on the Officer in Tactical Command Information Exchange System (OTCIXS) and Tactical Data Information Exchange System (TADIXS). It is designed to be a human-readable format for the non-TDP user.

NOTE: Normally OTG-D files either have a **.txt** or no extension. In order for SIMDIS to determine which parser to use, the file extension must be changed to **.otg**.

7.1.8 NSWC Corona Performance Evaluation Tool (`.pet`)

Files with the **.pet** extension are data files saved from the NSWC Corona Performance Evaluation Tool (PET). The PET tool saves data referenced to the beginning of day, so care must be taken when merging with other data sources.

7.1.9 SIMDIS HDF-5 Data File (`.hdf5`)

The Hierarchical Data Format (**.hdf-5**) is an efficient storage and retrieval format for high-volume data. The plug-in allows import and export of formatted **hdf-5** file format in/out of SIMDIS. Technical details on the format are covered in `$(SIMDIS_DIR)/doc/HDF5/HDF5_Plugin_User_Manual.pdf`.

7.1.10 Warfare Assessment Model (`.wam`)

Files with the **.wam** extension are used as input into the ConvertWAM2ASI application. The **.wam** files are ASCII based files output from the Warfare Assessment Model (WAM). The WAM was developed by NSWC Corona when it was known as the Naval Warfare Assessment Station (NWAS). The **.wam** file system is still in use today at various US Naval facilities.

7.1.11 Compressed ASI Archive (`.zip`)

Files with the extension **.zip** are a compressed archive containing one or more files supported by SIMDIS. SIMDIS can load ZIP archives in addition to saving them from the ASI Export dialog. ZIP archives saved from SIMDIS contain the ASI data file and external files such as views, rules, GOGs, models, textures and any media. This allows the ZIP to be used in conjunction with the SIMDIS Core as a standalone data product. Due to file sizes, SIMDIS does not export terrain and imagery files into the ZIP archive. SIMDIS does support loading terrain and imagery files found in a zip archive, provided the file paths are relative to an earth file in the archive. The timestamped imagery demo file `Irma NEXRAD.zip` includes one such example.

7.1.12 Compressed HDF5 Archive (`.hdfz`)

Files with the extension **.hdfz** are a compressed archive containing one or more files supported by SIMDIS. SIMDIS can load HDFZ archives in addition to saving them from the Export dialog using the **HDF5 plug-in**. HDFZ archives saved from SIMDIS contain the HDF5 data file and external files such as views, rules, GOGs, models, textures and any media. This allows the HDFZ to be used in conjunction with the SIMDIS Core as a standalone data product. Due to file sizes, SIMDIS does not export terrain and imagery files into the HDFZ archive. SIMDIS does support loading terrain and imagery files found in an HDFZ archive, provided the file paths are relative to an earth file in the archive.

7.1.13 SIMDIS Command Line File (`.scl`)

Files with the extension **.scl** are SIMDIS Command Line files intended to simplify starting SIMDIS with large numbers of command line arguments. An SCL file is a text file containing any number of command line arguments separated by whitespace. SCL files can also contain comments beginning with "**#**" or "**//**". Any text in the file not preceded by one of these comment tokens is treated as an argument. By using the **–scl** argument followed by the name of an SCL file, the contents of that file are added to the arguments used to start the program. Note that relative paths given in an SCL file are resolved the same as if they were given directly over command line, not relative the SCL file.

7.1.14 Bookmarks Output Format (`.toc`)

Files with the extension **.toc** are Table of Content files for Bookmarks. A TOC file describes all the events in a user editable text file. The file starts with a reference year followed by the events. Each event starts with the keyword Bookmark followed by a time stamp. After the time stamp there are a variable number of arguments. The number and types of arguments are dependent on the event type. An event must be defined on one line. An example file is located at `$(SIMDIS_DIR)/demo/SIMDIS/Users/NG/BMD_DEMO`. For definitions and examples for each event type refer to Appendix F.

7.1.15 Bookmarks Output Format (`.bml`)

Files with the extension **.bml** are XML Bookmark files, an XML equivalent of the Table of Content (TOC) files. The **.bml** file is a collection of Bookmark XML elements. The XML attribute defines the event type and the event time stamp. The bookmark XML elements contain one or more sub-elements as supported by the event types. All the names in the **.bml** files are the same names in the **.toc** file. An example file can be created by saving the `BMD_DEMO` to a **.bml** file. A snippet of the file will look like the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Bookmarks>
<ReferenceYear>1970</ReferenceYear>
<Bookmark type="Comment" time="001 2000 00:00:00">
<Category>Example</Category>
<Text>BMD Demo</Text>
</Bookmark>
<Bookmark type="Eye Position" time="001 2000 00:00:00">
<EyePos>OVH</EyePos>
<Duration>0</Duration>
<Viewport />
</Bookmark>
<Bookmark type="Time" time="001 2000 00:00:00">
<RealTime>True</RealTime>
</Bookmark>
```

7.2 Coordinate Systems and Reference Frames

Position, orientation, and velocity data can be provided to SIMDIS in several well-defined coordinate systems. These systems range from local to Earth-based, each having its own use with origins defined at a specified location and their axes aligned in a specific way. This section will address how each one is used in SIMDIS.

The Earth coordinate system forms the **base** on which all other coordinates are defined. Typically, the origin is located at the center of the Earth, and the principle axes are aligned with directions like North/South, East/West, and Up/Down. This is the coordinate system in which a majority of the entities in SIMDIS are defined. Within SIMDIS there are three types of coordinate systems: Cartesian, Geocentric, and Spherical.

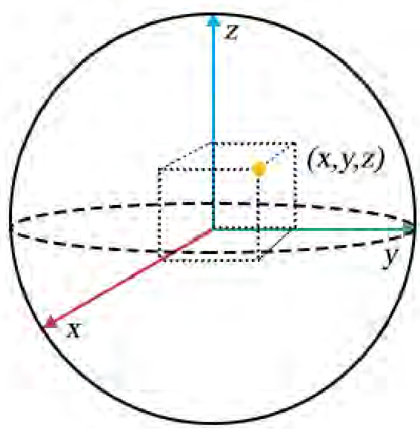


Figure 7.2: Cartesian

The **Cartesian** coordinate system is defined as a **flat** coordinate system placed on the surface of the Earth at a specific origin. The three Cartesian axes defining the system are perpendicular to one another. This system is typically used in simulation models as well as acoustic tracking systems. All Cartesian systems used in SIMDIS are "right-handed", with the XY-plane tangent to the Earth's surface at the specified origin. Cartesian systems used in SIMDIS are: X-East tangent plane, generic tangent plane, and a scaled flat Earth system.

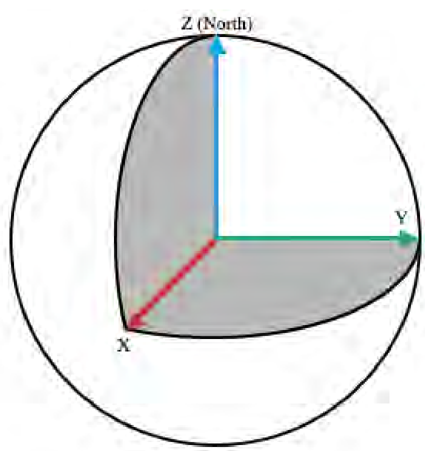


Figure 7.3: Geocentric

The **Geocentric** coordinate system is based on a Cartesian X, Y, and Z coordinate system with the origin at the center of Earth. This coordinate system is typically used in missile and satellite position and navigation systems as well as simulation-based protocols such as Distributed Interactive Simulation (DIS) and High-Level Architecture (HLA). Geocentric systems used in SIMDIS include: Earth-Centered Earth-Fixed (ECEF) and Earth-Centered Inertial (ECI).

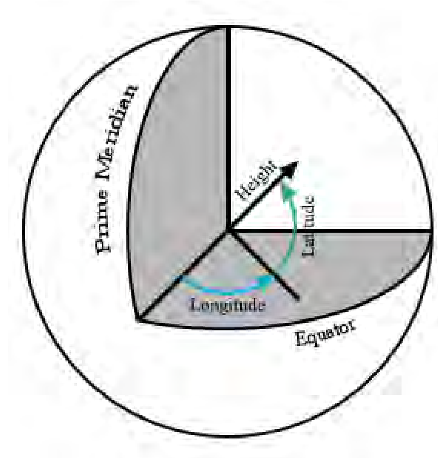


Figure 7.4: Geodetic

The **Spherical** or **Geodetic** coordinate system is probably the best known as it has easy to understand concepts for East, North, and Up. It is based on angles relative to the Prime meridian and Equator that are referred to as Longitude and Latitude. Heights within this system are given relative to the Earth's surface as defined by a particular vertical datum.

7.2.1 Earth-Centered, Earth-Fixed (ECEF)

The **ECEF** system used in SIMDIS is based on the **WGS-84** ellipsoidal Earth model. The WGS-84 Earth model is a geocentric right-handed rectangular coordinate system in which the origin is the center of the Earth.

- The **+X-axis** lies in the equatorial plane and points toward the Greenwich meridian.
- The **+Y-axis** lies in the equatorial plane and points toward 90 degrees East longitude.
- The **+Z-axis** is coincident with the Earth's polar axis and is directed toward the North Pole.

Figure 7.5 depicts the Earth-Centered Earth Fixed coordinate system. The ECEF system in SIMDIS uses the **Cartesian System** to describe global positions based on axes X, Y, and Z.

The velocity vector, when given in geocentric coordinates, is defined by a magnitude and two flight path angles relative to a local geocentric horizon coordinate system, shown in Figure 7.6. The origin of this coordinate system is located at the vehicle's center of mass. The local geocentric horizon z-axis is aligned with the position vector r, but in the opposite direction so that it points toward the center of the Earth.

The **Local Geocentric Horizon Plane** is the X and Y axes in a plane that is perpendicular to vector **r**.

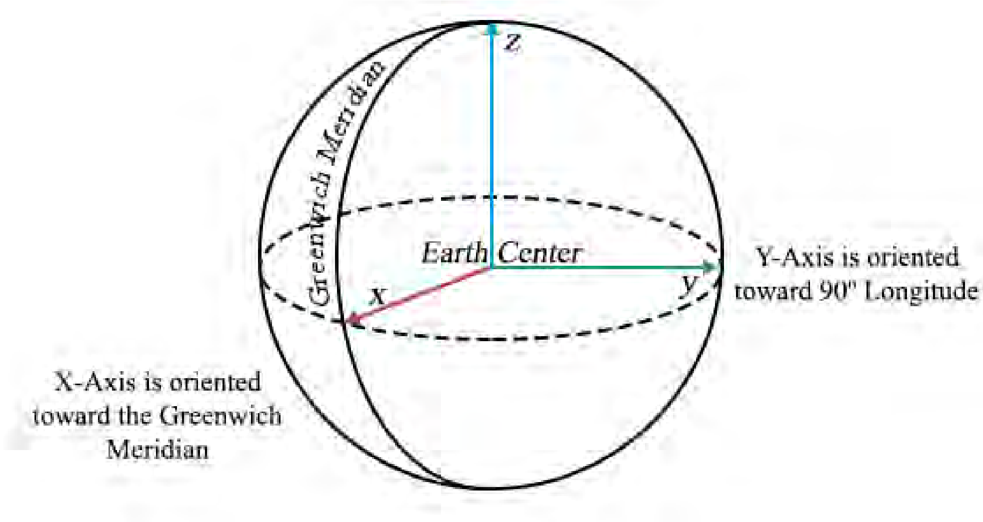


Figure 7.5: Earth-Center, Earth Fixed (ECEF) System

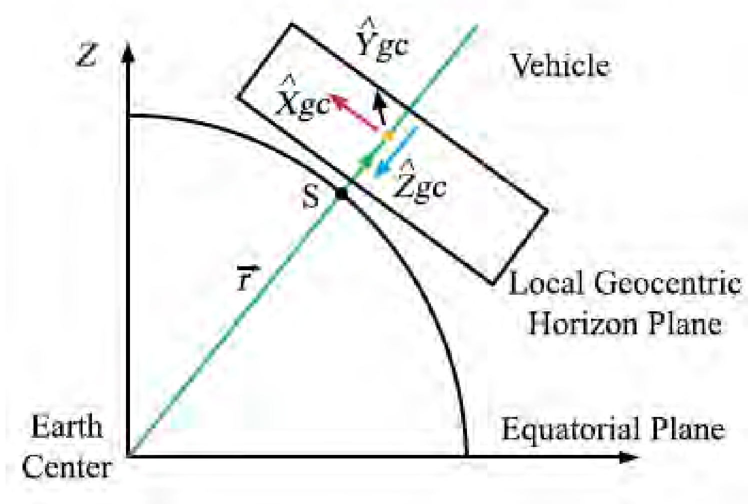


Figure 7.6: Local Geocentric Horizon Coordinates

In the **North-East-Down** or **NED** frame:

- The X-axis points toward the North Pole.
- The Y-axis points East making the Y-axis parallel to the equator.

A set of Euler angles based on a **front-right-down** system with respect to a geocentric body fixed ZYX orientation, also known as the **yaw-pitch-roll** rotation sequence, is used in SIMDIS. The yaw-pitch-roll rotation sequence is used to convert from the local geocentric horizon system to the body system. The direction of the positive rotation for each axis is clockwise from the perspective of the vehicle.

- The X-axis of the body coordinate system points toward the front of the vehicle as shown in Figure 7.7. It is often along the centerline or longitudinal axis of the vehicle.

- The Y-axis is oriented toward the right wing.
- The Z-axis points downward to form a right-handed system.

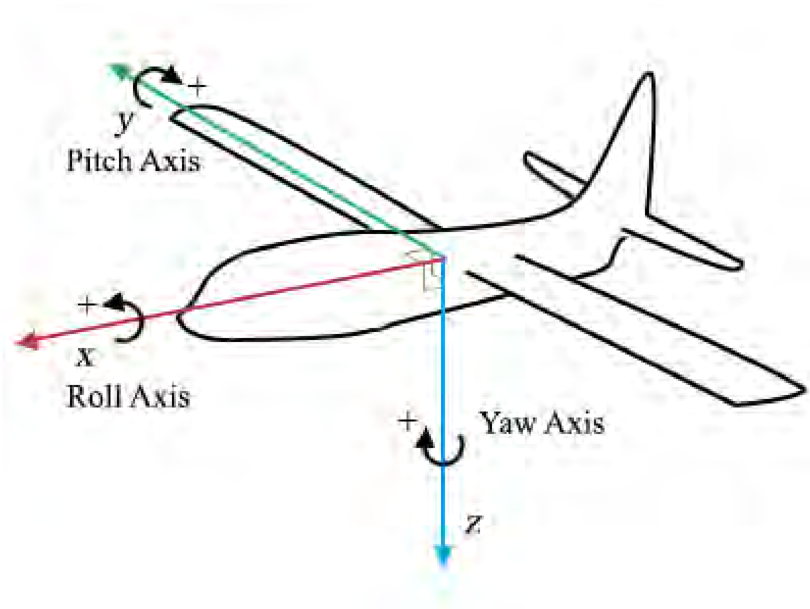


Figure 7.7: Geocentric Body Fixed Coordinates

The three angles used to describe the orientation of the vehicle are composed of a set of three successive rotations about three different orthogonal axes. This system is an Institute of Electrical and Electronics Engineers (IEEE) standard specified in the Distributed Interactive Simulation (DIS) protocol. The DIS standard states that three Euler rotations are implemented by first rotating around the Z-axis by angle psi (ψ), then around the new Y-axis by angle theta (θ), and finally around the newest X-axis by angle phi (ϕ) . This order is useful because it corresponds to physical situations; for example, ones in which an aircraft might maneuver by changing its heading about its Z-axis, then pitching about its new Y-axis, and finally rolling about its newest X-axis.

The following direction cosine matrix (DCM) is used to perform the transformation from the inertial ECEF frame to the geographic NED reference frame:

$$T_{I2G} = \begin{bmatrix} -\sin u \cos l & -\sin l & -\cos u \cos l \\ -\sin u \sin l & \cos l & -\cos u \sin l \\ \cos u & 0 & -\sin u \end{bmatrix}.$$

Where u = latitude, l = longitude

The platform body frame is aligned X-Forward, Y-Starboard (right), and Z-Down. The sequence of rotations conventionally used in the aerospace industry to describe the instantaneous attitude with respect to a reference frame (NED) is as follows:

- Rotate about the Z-axis, nose right (positive yaw ψ)
- Rotate about the new Y-axis, nose up (positive pitch θ)
- Rotate about the new X-axis, right wing down (positive roll ϕ)

The following DCM is used to perform the transformation from the geographic (NED) reference frame to the platform body frame:

$$T_{G2B} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}.$$

Where ψ = yaw, θ = pitch, ϕ = roll

The overall transformation from the inertial (ECEF) frame to the platform body frame is denoted by the following matrix:

$$T_{I2B} = T_{I2G}T_{G2B}$$

Conversely, if moving from the platform-body frame to the inertial frame, the rotations are reversed such that a roll, pitch and yaw sequence is followed. This is accomplished by multiplying the transpose of the above matrices.

7.2.2 Earth-Centered Inertial (ECI)

The **ECI** system used in SIMDIS is also based on the **WGS-84** ellipsoidal Earth model.

- The Z-axis runs along the Earth's rotational axis pointing North.
- The X-axis points in the direction of the vernal equinox.
- The Y-axis completes the right-handed orthogonal system.

The vernal equinox is an imaginary point in space which lies along the line representing the intersection of the Earth's equatorial (X-Y) plane and the plane of the Earth's orbit around the Sun called the ecliptic. The axes defining the ECI coordinate system are **fixed** in space and do not rotate with the Earth. The axes of the ECEF coordinate system rotate with the Earth, so they always point along fixed meridians. The two coordinate systems are aligned in SIMDIS when the ASI keyword `ReferenceTimeECI` value is zero (**t = 0**). Figure 7.8 depicts the ECI coordinate system.

This means that the ECI coordinate system can be defined from the ECEF coordinate system with a rotation about the z-axis by the angle Ω given by the following:

$$\Omega = \omega \cdot t$$

Where ω is the earth's rotation rate of **7.292115 x 10^{-5} rad/sec** and ***t*** is the `ReferenceTimeECI`.

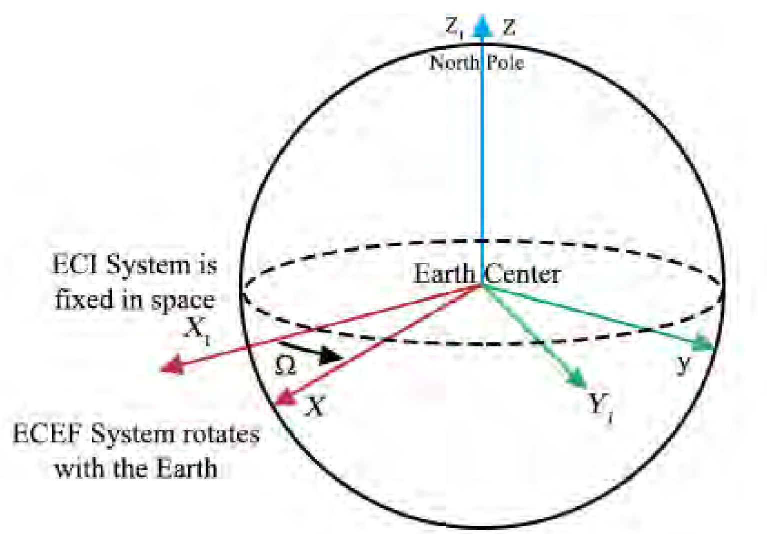


Figure 7.8: Earth Centered Inertial (ECI) System

7.2.3 Geodetic Local Coordinate Frame

Two geodetic local coordinate systems are used in SIMDIS. Both are Cartesian with their origins at the geometric center of the vehicle. The North–East–Down or NED frame with local directions of North, East, and Down define a right-handed set of three axes. The second system is also Cartesian with axes in the local East–North–Up or ENU frame. In both sets of axes, the North and East directions define a plane tangent to Earth's surface. At any point, North points along the local meridian, while East points along the local small circle of constant latitude. The vertical direction, zgd, is perpendicular to this plane. The vertical direction does not intersect Earth's center for geodetic systems, it makes an angle of the latitude, δ_{gd} , with the equatorial plane, as seen in Figure 7.9 below.

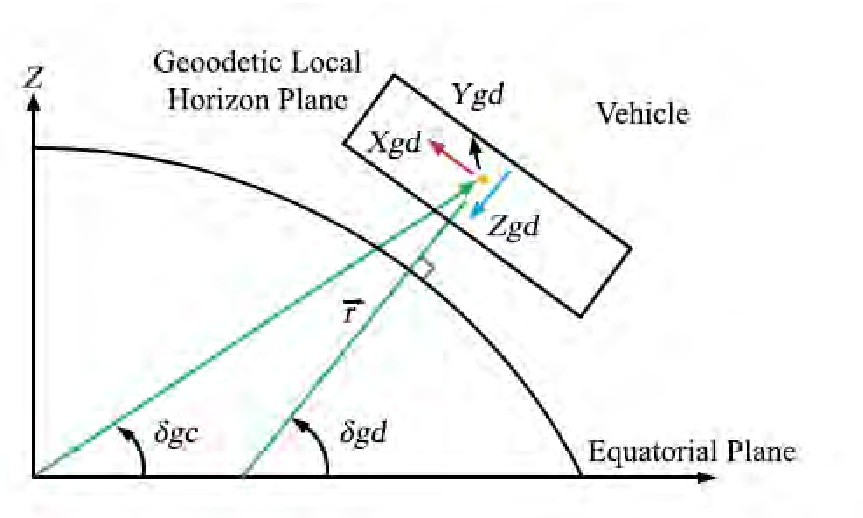


Figure 7.9: Geodetic Local Coordinate System

An entity in SIMDIS (e.g. a 3D model) is also defined in its own local coordinate system. The local coordinate system is used to place transformable entities such as Radar beams, antenna patterns and GOGs, relative to a host entity. The local system allows precise placement of entities relative to one another that would otherwise be difficult or numerically impossible due to finite decimal ranges in an Earth-based coordinate system. A real-world example would consist of offsetting the local origin of a ship based on a sensor's shipboard location. The local origin of a 3D ship model used in SIMDIS is typically located at the waterline and the geometric center of the length (X-axis) and the beam (Y-axis). Position reporting sensors are typically located in high areas that have an unobstructed view of the sky. In this example, if an offset were to be used, the placement of the ship model would be realistic with respect to the reported datum, versus hovering above it.

7.2.4 Geodetic (WGS-84)

The **Geodetic** coordinate system is similar to the geocentric coordinate system except that the positions are defined with respect to the surface of an ellipsoid instead of the center of the Earth.

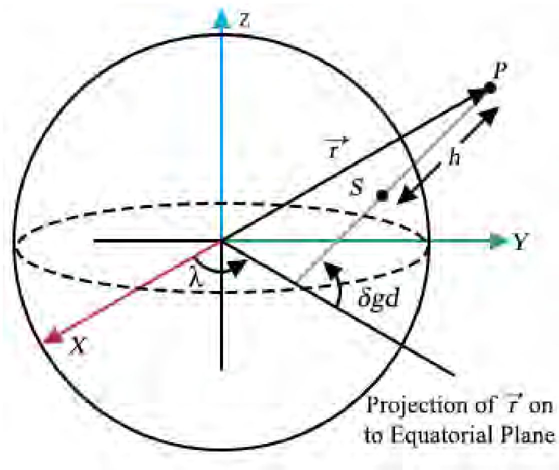


Figure 7.10: Geodetic System

The geodetic system used in SIMDIS defines a position referenced to WGS-84 ellipsoidal Earth model in terms of geodetic:

- Latitude δ_{gd} - the angle between the equatorial plane and a line passing through the vehicle's position and normal to the ellipsoid. Latitude is positive when the vehicle is above or north of the equatorial plane.

- Longitude λ - the angle between the x-axis and the projection of the position vector onto the equatorial plane. Longitude is positive when measured counter-clockwise or east from the x-axis.

- Altitude h - the distance from the vehicle to the surface of the ellipsoid (the distance between points P and S). It is measured along a line passing through the vehicle's position and normal to the ellipsoid. Altitude is positive when the vehicle is located above the Earth's surface.

Position data from live instrumentation are usually given in geodetic coordinates. In the simulation world position data obtained from DIS and High-Level Architecture (HLA) is in geocentric coordinates. In order to handle the frequent transformations between the two systems in SIMDIS an efficient algorithm developed by Ralph Toms is used for the conversion. This algorithm sacrifices precision for speed, a one-centimeter resolution is the best precision that can be obtained when converting from geodetic to geocentric.

```
"An Improved Algorithm for Geocentric to Geodetic Coordinate Conversion",
Ralph Toms, February 1996 UCRL-JC-123138.
```

Euler angles, giving the vehicle's body angular orientation with respect to local geodetic horizon coordinate system are computed using a **North-East-Down** (NED) coordinate system.

7.2.5 Scaled Flat Earth

In a Scaled Flat Earth system, the Earth's surface is projected (warped) onto an X-Y plane of a Cartesian coordinate system, with the reference origin specified at a latitude and longitude surface location. SIMDIS supports three scaled flat Earth systems: East-North-Up (**ENU**), North-East-Down (**NED**), and North-West-Up (**NWU**). In these projections, the Earth is not flat in the sense the projections follow the Earth's curvature along latitude.

The scaling of the latitude and longitude values into the flat Earth system is based on the values of the reference origin. The resulting scaled flat Earth only maintains proper scale, direction, distance, and area within a short range of the reference origin. Additionally, the altitude is referenced to the height above the Earth's surface instead of the height above a plane tangent to the surface.

The ENU system is represented as:

Positive X-axis points East
Positive Y-axis points North
Positive Z-axis points Up (positive altitude is positive Z)
True North [000] is looking down the Y-axis
East [090] is looking down the X-axis
Positive yaw/course is clockwise from North

The NED system is represented as:

Positive X-axis points North
Positive Y-axis points East
Positive Z-axis points Down (positive altitude is negative Z)
True North [000] is looking down the X-axis
East [090] is looking down the Y-axis
Positive yaw/course is clockwise from North

The NWU system is represented as:

Positive X-axis points North
Positive Y-axis points West
Positive Z-axis points Up (positive altitude is positive Z)
True North [000] is looking down the X-axis
East [090] is looking down the negative Y-axis
Positive yaw/course is clockwise from North

7.2.6 Tangent Plane

SIMDIS supports two tangent plane projections:

- **X-East** - aligned with the Earth such that +X is East, +Y is North, and +Z is up.
- **Generic Tangent Plane** (**GTP**) - the reference origin point that does not have to be at this tangential point, but can lie at some other selected point on the x-y plane. Furthermore, the X-Y axes do not have to be oriented so that the +Y-axis points to True North; they can be rotated to any desired angle. The x-offset position is the True East distance; the y-offset position is the True North distance in meters of the desired origin as seen from the tangential point.

  The x-offset is positive if the origin lies to the east of the tangential point, and it is negative if the origin lies to the west of the tangential point. The y-offset is positive if the origin lies to the north of the tangential point, and it is negative if the origin lies to the south of the tangential point.

  The rotation angle is the desired angle in degrees to apply to rotate the x-y plane. Positive angles perform a clockwise rotation referenced to True North. If no translation or rotation offsets are applied, the GTP system becomes an X-East system.

Both projections define a local rectilinear coordinate system based on a flat plane tangent to the Earth's surface at a specific reference origin. Lines of equal distortion are concentric about the tangent plane's reference origin. Further the distance from the origin, the greater the distortion.

The tangent plane velocity vector is the velocity of the vehicle's center of mass relative to the tangent plane. The X component of the velocity vector is in the x-axis direction, the Y component of the velocity vector is in the y-axis direction and the Z component of the velocity vector is in the z-axis direction.

Euler angles, giving the vehicle's body angular orientation with respect to the tangent-plane coordinate system, are computed using a North-East-Down (NED) coordinate system (local geodetic horizon coordinate system) located at the tangent plane origin and aligned with the tangent-plane azimuth. This means that the tangent plane Euler angles are the same as geodetic Euler angles except they are rotated by the tangent-plane azimuth and with respect to the tangent-plane origin. In the case where a tangent plane is aligned with the X-axis oriented East, the Euler angles will be the same as the geodetic.

7.3 Magnetic Variance

Within SIMDIS a user can specify the magnetic variance (declination) between Magnetic North and True North. The variance is considered positive east of True North and negative when west of True North. SIMDIS supports three options for specifying magnetic variance data:

Value	Meaning
WMM	Variance is computed based on the World Magnetic Model.
TRUE	No variance, yaw values relative to True North. **[Default]**
14.2	User-specified variance in DD, DM or DMS.

If a magnetic variance is not specified, then all yaw values will be referenced to True North. WMM data is only valid for a five-year period. Presently, SIMDIS can compute the magnetic variance using the WMM for data within the ranges of 1985 to 2025. Additional information regarding the WMM can be found on the National Geophysical Data Center's (NGDC) web site:

http://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml.

For a user-specified variance, the magnetic declination is considered as the angle between the local magnetic field, the direction the North end of a compass points and True North. The declination is positive when the Magnetic North is east of True North. Units for the user specified variance must be in degrees.

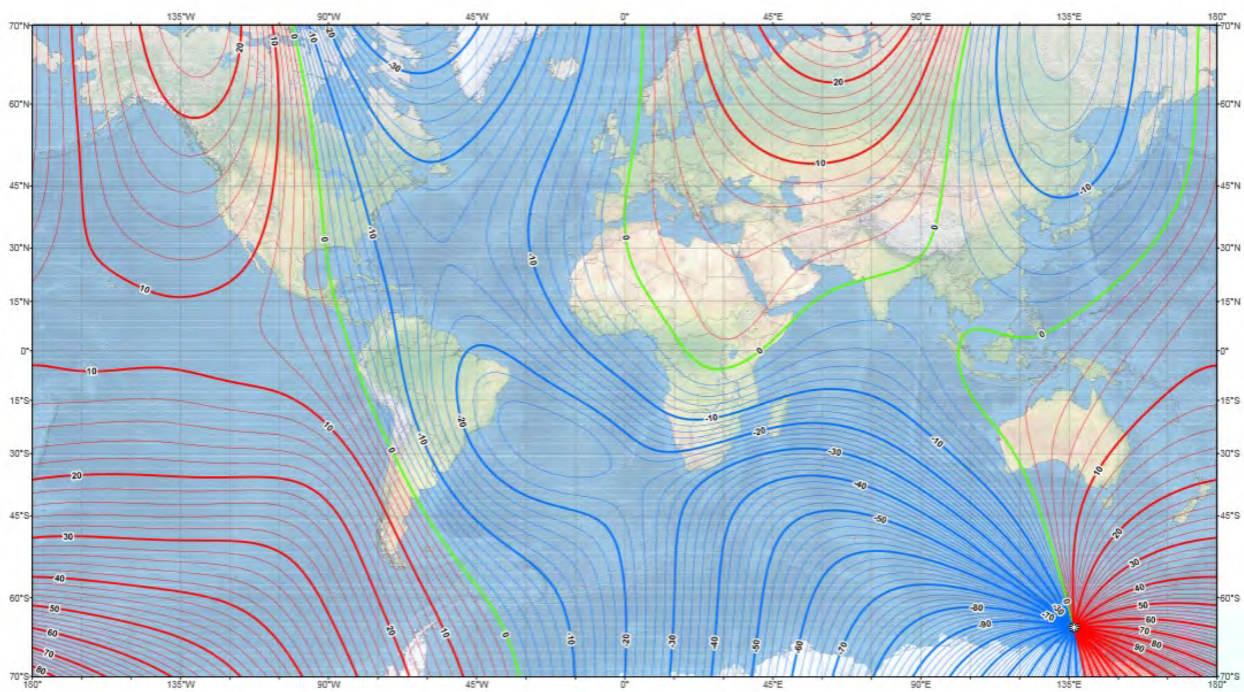


Figure 7.11: US/UK World Magnetic Model (WMM), Epoch 2015, Main Field Declination

7.4 Time

The time reference for all time stamped data in SIMDIS is the **Portable Operating System Interface** (**POSIX**) based on the **IEEE Standard 1003.1**. The IEEE 1003.1 POSIX time is represented by the elapsed non-leap seconds since the epoch date of 00:00:00 Coordinated Universal

Time (UTC) January 1, 1970. SIMDIS does not support the concept of time zones; all POSIX times should be referenced to Greenwich Mean Time (GMT).

SIMDIS can support time precision to the nanosecond level for file-based data. In order to handle this level of detail time values are referenced to a specified Gregorian calendar year. This year is known as the "reference year" in the SIMDIS vernacular. Furthermore, SIMDIS does not support negative time values relative to a given reference year. Negative time values are a special case in SIMDIS to denote a static platform that exists for the lifetime of a scenario. A static platform does not move and will remain fixed at the specified position. If multiple data points for a static platform are defined, only the first is used.

Finally, SIMDIS can only store a given data point at a specific instance in time. If an incoming file or data source specifies more than one position value for a given time, SIMDIS will use the last value encountered.

Chapter 8

Generalized Overlay Graphics (GOG)

The Generalized Overlay Graphics (GOG) file format is a vector graphics format that was originally designed for the NTADS (Naval Tracking and Display System) program. NTADS is currently in use at the Pacific Missile Range Facility. Subsequent to the inception of the GOG format with NTADS, the format has been ported for use in SIMDIS.

Several new SIMDIS specific features as well as support for KML, ESRI Shapefiles, Tsunami overlays, NOAA ENC, and SDTS Data Descriptive Files (DDF) have been added to the original GOG format to create an updated version 2.0 of the GOG specification. The files defined in the original format still work in both NTADS and SIMDIS; however the newer version GOG files as well as those that contain SIMDIS specific features are known only to work with SIMDIS and Plot-XY.

8.1 General Format

All GOG command keywords are case-insensitive. Each GOG object is defined by blocks of commands that are within a required start command and finished with an end command. Comments are indicated by the comment keyword. Additionally, SIMDIS allows the use of both C++ (`//`) and shell script style (`#`) comment indicators.

The two basic type of object are absolute and relative objects. Absolute versions of GOG are referenced to static geodetic (Latitude, Longitude, and Altitude) locations. Any GOG object created using the `ll` position command will be absolute. Relative versions of GOG make use of the `xy` position command. Relative GOG files can be attached to either reference points or platforms within SIMDIS and Plot-XY. Relative GOG objects that are not attached to a platform or do not use a reference point are drawn referenced to a location off of the Pacific Missile Range Facility called BARSTUR Center. The coordinates of this location are:

```
22 07 9.9811 N      159 55 10.1957 W
```

8.2 Units

There are specific units associated with all the value types found in version 1.0 GOG files. In version 2.0 of GOG, units are defined by altitudeUnits, rangeUnits, angleUnits, and timeUnits keywords; if keywords are not specified then the following table delineates these units:

Measurement	Units
Latitude and Longitude (all versions)	Degrees decimal: -77.5147453 Degrees minutes decimal:"-77 30.88472" Degrees minutes seconds: "-77 30 53.083"
Relative xy position values (all versions)	Yards
Altitude and z values (all versions)	Feet
Angles values (all versions)	Degrees
Time values (version 2)	Seconds

NOTE: Double quotations are required for values separated by white space.

8.3 Colors

Color names are linked to colors (**color1** - **color7**) found in the **simdis.prefs** file as well as the Options dialog. The color name has no relation to the color that is displayed. The colors are stored as persistent settings in the **SIMDIS.ini** file, the Options dialog, or the GOG Tool.

Below are the available fill colors, along with the default color1 through color7 assignments for SIMDIS. The keywords **color1** through **color7** and the color names are valid GOG commands.

color1: cyan (SIMDIS only)

color2: red

color3: green

color4: blue

color5: yellow

color6: orange

color7: white

NOTE: The color commands **black** and **magenta** were added in version 2 of the GOG specification.

SIMDIS and Plot-XY have the additional functionality of accepting a color value in a hexadecimal format. The value must be placed in the third column of an entry line. All hexadecimal colors are specified as `0xAABBGGRR`:

Value	Meaning
AA	Alpha component, range 00 to FF (0-255) in hexadecimal
BB	Blue component, range 00 to FF (0-255) in hexadecimal
GG	Green component, range 00 to FF (0-255) in hexadecimal
RR	Red component, range 00 to FF (0-255) in hexadecimal

For example the color **Magenta** would be specified using the following syntax:

```
linecolor hex 0xFFFF00FF
```

8.4 Commands

A detailed list of all GOG commands and GOG Object Examples can be found in Appendix B. Additionally, several distributed files contain useful examples:

Version 1 commands: `$(SIMDIS DIR)/data/GOG/sample.gog`

Version 2 commands: `$(SIMDIS DIR)/data/GOG/sample2.gog`

8.4.1 Creating a GOG file

It is simple to create a GOG file. You can refer to Appendix B for the structure, types, positions, and modifier. The GOG files can be created or generated with any text editor such as Microsoft Notepad. You will need to save the file as **.gog**. Here is an example of creating a line GOG in SIMDIS:

```
start
line
ref 40.50,-89.36
xy 0 0
xy 10000 0
xy 10000 10000
xy 0 10000
xy 0 0
linewidth thin
linecolor white 0xFF606060
end
```

NOTE: When you add a GOG file, make sure they have unique filenames. SIMDIS will automatically reject GOGs with same filenames.

8.4.2 Create a GOG File with Custom Outline

When you create a GOG without the modifier **outline**, it will create a GOG with a **red** outline. Here is an example to customize your GOG's outline and color.

This example will display thicker dashed green GOG using modifiers **linewidth** 3px, green **linecolor** using hex, and dashed **linestyle**. Figure 8.1 represents the GOG with the default outline to the left and the custom GOG to the right using line modifiers.

```
start
circle
linewidth 3
linecolor hex 0xFF00FF00
linestyle dashed
centerll 0 0
radius 2500
end
```

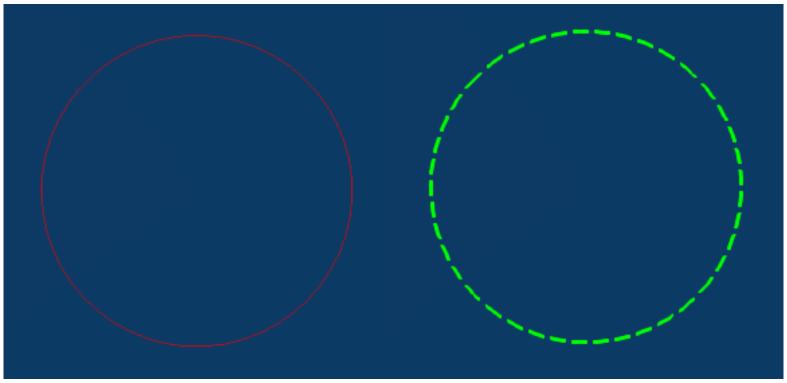


Figure 8.1: Default GOG vs GOG with Line Modifiers

NOTES:

- When you set **outline** modifier to **false**, the outline will disappear. The GOG will not be visible if it is not filled. Check the color's alpha when investigating GOG's visibility. Please refer to Section 8.3 for GOG color options.
- Double click the created GOG in the **GOG Tool** to center it to your display.

8.5 NOAA Coastline Extractor

GOG files can be created from NOAA coastline segments files using their GEODAS-NG software. Their website, http://www.ngdc.noaa.gov/mgg/geodas/geodas.html depicted below, contains a link to their GEODAS-NG Desktop Windows Software which then allows a user to extract coastline segments bounded by a specified latitude/longitude range.

To extract a coastline, follow the steps outlined below the snapshot of the NOAA Coastline Extractor website.

NOTE: Python must be installed on your system to convert NOAA coastlines to GOG files. Python is available from www.python.org. Internally, SIMDIS uses the Python version 2.7.3; the scripts have been verified to work with versions of Python up to and including 2.7.3. Due to a Python API change, SIMDIS scripts will not work with Python version 3.x.

After installation of GEODAS-NG, you may need to download additional coastlines data files such as the Global Self-consistent, Hierarchical, High-resolution Geography Database (GSHHG) or the legacy GEODAS Coastlines files.

http://www.ngdc.noaa.gov/mgg/shorelines/data/gshhg/latest/

http://www.ngdc.noaa.gov/mgg/geodas/geodas.html

1. Launch the GEODAS-NG Smart Start program, then select the Coastline Extractor.
2. Select the coastline database (GEODAS, GSHHG or Shapefiles).
3. Navigate to a region of interest.
4. Perform a **Save As** using the Mapgen Coast Format
5. Once the coastline has been extracted, save it to your local disk.
6. Convert the Mapgen format to a GOG file, use the following python script (**7794.dat** was extracted from the **7794.zip** file).
7. Open a cmd shell, then cd to the directory that contains the extracted data: **usgsToGOG.py 7794.dat**
8. The resulting output will be named **7794.gog**. Rename the file to something more meaningful.

The **World Coastlines and Lakes** database is a bit rough, but has worldwide coverage. The data is suitable for map scales of 5,000,000 or smaller.

The **World Data Bank II** database has fair resolution with worldwide coverage. The data is suitable for map scales of 2,000,000 or smaller. It also contains international and internal political boundaries and rivers.

The **World Vector Shoreline** database has better resolution for worldwide coverage. This data is suitable for map scales of 1:250,000 or smaller.

NOAA/NOS Medium Resolution Digital Vector Shoreline has the best resolution overall. It has United States coverage digitized from NOAA nautical charts. This data is suitable for map scales of 1:70,000 or smaller. It is noteworthy, however, that there may be a few odd data points in this version of the dataset (September, 1994).

8.6 Vector to GOG Converter (Vec2GOG.exe)

Vec2GOG is a file conversion program that reads in various vector formats and translates them to GOG files. The supported formats are:

Arc/Info Binary Coverage	**Microstation DGN**	**UK .NTF**
SDTS	**U.S. Census TIGER/Line**	**IHO S-57 (ENC)**
Mapinfo File	**OGDI Vector**	**ESRI Shapefile**

The file conversion program automatically converts the coordinates into WGS84 based geode- tic values. Some of the above file formats (Microstation DGN and ESRI Shape file) do not contain coordinate system description information, therefore there is no coordinate transformation performed on these types of files.

NOTE: Python must be installed on your system to convert NOAA coastlines to GOG files. Python is available from www.python.org. Internally SIMDIS uses the Python version 2.7.3; our scripts have been verified to work with versions of Python up to and including 2.7.3. Due to a Python API change, SIMDIS scripts will not work with Python version 3.x.

Some of the file formats listed above contain extra information in each line or point within the file. The Vec2GOG program defines defaults for how to convert this information. More information on changing these defaults is provided below.

The following websites contain information about various data sources.

http://www.census.gov/geo/maps-data/data/tiger.html
http://mcmcweb.er.usgs.gov/sdts/
http://www.gisdatadepot.com/catalog/index.html
http://www.hawaii.gov/dbedt/gis/
http://www.esri.com/data/
http://nationalmap.gov/small_scale/
http://www.mapmart.com/
http://www.dmoz.org/Science/Social_Sciences/Geography/Geographic_Information_Systems/Data/

8.6.1 Modifying the Vec2GOG Conversion Process

There is a Python file called **GetFeatureAttributes.py** that Vec2GOG uses during the file conversion process. This file contains one Python function for each of the supported file formats. If you want to modify the conversion process for one of the supported file formats, then you could edit the Python file and modify the corresponding function. There is documentation contained within the Python file that explains what types of modifications are expected.

8.6.2 Running the Vec2GOG program

Once a data file has been acquired, the program can be executed in a cmd shell. A sample ESRI shape file is distributed with SIMDIS and located in the `$SIMDIS DIR/demos/SIMDIS/import` folder.

If the user changes their directory location to the import directory, then:

```
Vec2GOG  ./rr lines clark county.shp
```

The program can also be run on multiple files and file types at once. In this example all ESRI Shapefiles and all SDTS files within the directory are converted:

```
Vec2GOG  *.shp *.ddf
```

Appendix A

Writing an ASI File

What is an ASI file?

The ASI data file format is the default input data file of SIMDIS. ASI files store data about a scenario, including the hierarchy of its platforms, beams, gates, lasers, lines of bearing, and custom renderings.

How is Generic Data used with an ASI file?

As the ASI file format has evolved, features such as Generic Data have been added for the sake of convenience. These features should be used at the user's discretion; an ASI file should be an accurate representation of the actual data for a scenario. Although these features can be used to simulate "scripted" events over time, we recommend against using Generic Data in this manner. Since Generic Data is considered data, it takes precedence over similar user preference settings and can result in undesired display behavior. If you desire to change the way actual data is presented, we recommend using the SIMDIS Python scripting interface.

How can I create an ASI file from scratch or modify an existing one?

ASI files can be created or edited using any plain text editing. SIMDIS utilizes a file parser to match keywords within incoming data and recognizes white space (e.g. tabs and spaces) as delimiters. The file parser also skips all blank lines and lines beginning with shell script (#) or C++ style (//) comments. For example, the following lines would be skipped in the parser:

```
# This is skipped
// This is skipped
```

Furthermore, all quoted strings must contain a value. The parser will register any empty quoted string ("") as an error.

Throughout this section, [**REQUIRED**] fields will be boldface and color-coded red, while [**OPTIONAL**] fields will be boldface and color-coded teal. Special note sections will be marked in boldface type. Pseudo code and file format examples will be displayed using a fixed width font.

An ASI file requires the following to be valid:

- A valid scenario definition (see Section A.1)
- At least one valid non-static platform (see Section A.2)

There is one [**OPTIONAL**] keyword available to all entity types:

`OriginalID` `UniqueID OriginalID`

`UniqueID`	Unique ID of the entity being referenced.
`OriginalID`	Original ID to assign to the referenced entity.

Example: **`OriginalID`** `10 42`

`OriginalID` is an ID that can be used to map an entity to an original data source. It does not have to be unique. SIMDIS plug-ins will be able to access the `OriginalID` value, but will not be able to access an entity's `UniqueID` as set in the ASI file. Additionally, the `UniqueID` might change as a file is converted to and from ASI, but the `OriginalID` field will stay the same.

The <**`OriginalID`**> keyword should only be used after an entity has been appropriately defined.

A.1 Scenario

Three keywords are [**REQUIRED**] to be specified when defining a scenario.

`Version Num`

Num ASI version used by this file.

Example: `Version 23`

The <**Version**> keyword enables multiple versions of the file format to be maintained.

NOTE: In order to parse multiple versions, it is required that the Version keyword be the first keyword encountered in the ASI file. If the keyword is not the **first**, a fatal error will be generated and the file will not be parsed.

All required keywords must precede any optional keywords.

Version 1 is no longer supported. Versions 2 through 23 are supported.

`RefLLA Lat Lon Alt`

- `Lat` Reference latitude (in degrees) of flat Earth or tangent plane coordinate system for the scenario.
- `Lon` Reference longitude (in degrees) of flat Earth or tangent plane coordinate system for the scenario.
- `Alt` Reference altitude (in meters) of flat Earth or tangent plane coordinate system for the scenario. Only used for flat Earth or tangent plane systems.

Example: `RefLLA 22.0 -160.0 0.0`

Latitude and longitude can be specified in degrees decimal (DD), degrees minutes (DM), or degrees minutes seconds(DMS). For both the DM and DMS notations, values must be enclosed in quotes, e.g.:

```
// degrees minutes (DM) notation
RefLLA "21 34.88243" "-159 12.47353" 0.0
// degrees minutes seconds (DMS) notation
RefLLA "21 34 52.946" "-159 12 28.412" 0.0
```

`CoordSystem System`

- `System` Type of coordinate system in which data points are represented. See Table A.1 for supported formats. Used for all data points unless an entity defines a different coordinate system for its data points.

Example: `CoordSystem "LLA"`

The following [**OPTIONAL**] keywords can be used to further define the scenario.

`ScenarioInfo Info`

`Info` Text description of the scenario.

Example: `ScenarioInfo "Data from various sources"`

`WindAngle Angle`

`Angle` Direction (in degrees) from which the wind is blowing, referenced to True North.

Example: `WindAngle 130.4172209`

`WindSpeed Speed`

`Speed` Speed of the wind (in meters/sec).

Example: `WindSpeed 1.313837`

`ReferenceYear Year`

`Year` The year to which time is referenced. All time values are assumed to be seconds since the beginning of a year. Default year is 1970.

Example: `ReferenceYear 2006`

`Classification Label Color`

`Label` Classification label of the data in the scenario.

`Color` Color of the label. See Section A.12.2 for information on color formats. The default setting is presented in the example below.

Example: `Classification "Unclassified" 0x8000FF00`

`GOGFile File(s)`

`File(s)` GOG filename(s) to load and display in SIMDIS. If specifying more than one filename, separate with a space. See `$(SIMDIS_DIR)/data/GOG` for example files.

Example: `GOGFile "us_states.gog"`
Example: `GOGFile "bsure.gog" "swtr.gog"`

`ViewFile File`

`File` View (eye position) filename to load in SIMDIS. Only one file is permitted.

Example: `ViewFile "19991109.view"`

`MediaFile File`

`File` SIMDIS media player file (.tmd, .lst, .pst) to load in SIMDIS. Only one file is permitted.

Example: `MediaFile "missile.lst"`

`ITConfigFile File`

`File` Imagery and terrain configuration file to load in SIMDIS. Only one file is permitted.

Example: `ITConfigFile "configPMRF.txt"`

`RuleFile File`

`File` Preference rule file to load in SIMDIS. Only one file is permitted.

Example: `RuleFile "simdis_gps.rul"`

`File` `File`

`File` Configuration file to load in SIMDIS. How the file is loaded is determined by its extension. Currently, only Range Tool Configuration files ("*.xml") can be loaded using this keyword.

Example: `File "rtConfig.xml"`

`DegreeAngles` `UseDegrees`

`UseDegrees` If set to 1, Platform, Beam, Gate, Laser, and LOB angles are interpreted as degrees. If 0, radians.

Example: `DegreeAngles 1`

NOTE: The DegreeAngles setting does not apply to the latitude and longitude values of RefLLA, which are always specified in degrees.

`MagneticVariance` `MagVar`

`MagVar` Magnetic variance (declination) between magnetic north and true north. Considered positive east of true north and negative when west. See Table A.2 for supported options. "TRUE" is the default option.

Example: `MagneticVariance "TRUE"`

If a magnetic variance is not specified, then all yaw values will be referenced to true north. WMM data is only valid for a five-year period. Presently, SIMDIS can predict the magnetic variance using the WMM for data within the ranges of 1985 to 2025. Additional information regarding the WMM can be found on the National Geophysical Data Center's (NGDC) web site:

http://www.ngdc.noaa.gov/geomag/WMM/back.shtml.

For a user specified variance, the magnetic declination is considered as the angle between the local magnetic field, the direction the north end of a compass points and true north. The declination is positive when the magnetic north is east of true north. Units for the user specified variance must be in degrees.

`VerticalDatum` `VertDatum`

`VertDatum` Zero surface to which elevations or heights are referenced. See Table A.3 for supported options. "WGS84" is the default option.

Example: `VerticalDatum "WGS84"`

In order to approximate Mean Sea Level, SIMDIS uses the WGS-84 Earth Gravitational Model 1996 (EGM96). This model consists of a height file with a 0.25 degree grid of point values in the tide-free system. Bi-linear interpolation is used to obtain the associated height value based on a specified latitude and longitude. Additional information on the EGM96 can be found on the National Geospatial-Intelligence Agency's website:

http://earth-info.nga.mil/GandG/wgs84/gravitymod/index.html.

If a vertical datum is not specified, then all height values will be referenced to the WGS-84 ellipsoid. See Table A.4 for information on the availability and reference surface of the vertical datum offsets per coordinate system.

ReferenceTimeECI `Time`

`Time` Time at which the Earth-Centered Inertial (ECI) reference frame is defined (in seconds).

Example: `ReferenceTimeECI "0.0"`

The reference time is also known as the Greenwich Mean Sidereal Time (GMST), and is the time since the ECI system was aligned with the vernal equinox and the ECEF system. The reference time has a valid range of values from 0 to 86400 seconds. If this value is not specified, zero will be used. SIMDIS uses the ECI reference time to calculate an elapsed time which is multiplied by the Earth's angular velocity in order to determine the corresponding ECEF position.

TangentPlaneOffset `XOffset YOffset Rotation`

`XOffset` X translation position (in meters) offset from the Generic Tangent Plane (GTP) coordinate system.

`YOffset` Y translation position (in meters) offset from the Generic Tangent Plane (GTP) coordinate system.

`Rotation` Rotation angle (in degrees) applied to the x-y plane.

Example: `TangentPlaneOffset -14063.024 5641.235 13.145999`

The X offset position is true east distance; the Y offset position is the true north distance in meters of the desired origin as seen from the tangential point. The X offset is positive if the origin lies to the east of the tangential point, and is netative if the origin lies to the west. The Y offset is positive if the origin lies to the north of the tangential point, and is negative if the origin lies to the south. The rotation angle is the desired angle in degrees to rotate the X-Y plane. Positive angles perform a clockwise rotation referenced to true north. If this value is not specified, all values will default to zero.

Value	Meaning
NED	NED Scaled Flat Earth
NWU	Scaled Flat Earth
ENU	Scaled Flat Earth
LLA	Latitude, Longitude, Altitude (Geodetic)
ECEF	Earth-Centered Earth Fixed, WGS-84 Ellipsoid
ECI	Earth-Centered Inertial, WGS-84 Ellipsoid
XEAST	X-East Tangent Plane
GTP	Generic Tangent Plane

Table A.1: Supported coordinate systems

Value	Meaning
WMM	Variance is computed based on the World Magnetic Model (WMM).
TRUE	No variance, yaw values relative to true north.
14.2	User specified variance in DD, DM, or DMS.

Table A.2: Available options for magnetic variance

Value	Meaning
WGS84	Surface referenced to the WGS-84 ellipsoid.
MSL	Mean sea level surface based on the EGM96 model.
7.4	User specified vertical datum, in meters.

Table A.3: Supported vertical datum options

Coordinate System	WGS84	MSL	User
Tangent Plane	No offsets applied	N/A	Yes, datum offset from the reference altitude setting
Scaled Flat Earth	No offsets applied	N/A	Yes, datum offset from the reference altitude setting
LLA (Geodetic)	No offsets applied	Yes, datum offset from EGM Geoid	Yes, datum offset from the surface of the ellipsoid
Earth-Centered	N/A	N/A	N/A

Table A.4: Availability and reference surface of vertical datum per coordinate system

A.2 Platform

A.2.1 Platform Initialization

There are two keywords that are [**REQUIRED**] to be specified when defining a platform.

`PlatformID ID`

`ID`	Unique ID assigned to the platform on creation. This ID is used to associate data with this platform. Must be a positive non-zero integer.

`PlatformName ID Name`

`ID`	Unique ID of the platform associated with this name.
`Name`	Name or description of the platform. For strings separated by spaces, use double quotes.

For example, to create two unique platforms:

```
PlatformID 1
PlatformName 1 Tanker
PlatformID 2
PlatformName 2 "CG-70 Lake Erie"
```

The following [**OPTIONAL**] keywords can be used to further define a platform.

`PlatformType ID Type`

`ID`	Unique ID of the platform referenced by this keyword.
`Type`	Category of which the platform is a member (integer or string). See Table A.5 for available categories.

Example: `PlatformType 2 "ship"`
Example: `PlatformType 2 1`

This keyword adds Category Data to the platform. The two examples above are equivalent to:
`CategoryData 2 -1 "Platform Type" "ship"`

`PlatformFHN ID FHN`

`ID`	Unique ID of the platform referenced by this keyword.
`FHN`	Friendly, Hostile, Neutral designation character. See Table A.6 for accepted values.

Example: `PlatformFHN 2 "F"`

This keyword adds Category Data to the platform. The example above is equivalent to:
`CategoryData 2 -1 "Affinity" "Friendly"`

`PlatformOffset ID XOffset YOffset ZOffset`

`ID`	Unique ID of the platform referenced by this keyword.
`XOffset`	Distance (in meters) of the X body offset with respect to the platform center.
`YOffset`	Distance (in meters) of the Y body offset with respect to the platform center.
`ZOffset`	Distance (in meters) of the Z body offset with respect to the platform center.

Example: **`PlatformOffset`** `2 0.0 0.0 1.5`

`PlatformOriOffset` `ID Yaw Pitch Roll`

`ID`	Unique ID of the platform referenced by this keyword.
`Yaw`	Yaw orientation offset (in degrees) with respect to the platform body.
`Pitch`	Pitch orientation offset (in degrees) with respect to the platform body.
`Roll`	Roll orientation offset (in degrees) with respect to the platform body.

Example: **`PlatformOriOffset`** `2 90.0 10.0 0.0`

`PlatformRCS` `ID RCSFile`

`ID`	Unique ID of the platform referenced by this keyword.
`RCSFile`	Radar cross-section (RCS) filename. RCS format determined during loading. See Table A.7 for supported file formats.

Example: **`PlatformRCS`** `2 "fake_rcs_3.rcs"`

`PlatformAttachedGOG` `ID GOGFile`

`ID`	Unique ID of the platform referenced by this keyword.
`GOGFile`	GOG filename of the relative GOG file to attach to the platform. See `$(SIMDIS_DIR)/data/GOG` for example files.

Example: **`PlatformAttachedGOG`** `2 "t2.rxy"`

`PlatformInterpolate` `ID OnOff`

`ID`	Unique ID of the platform referenced by this keyword.
`OnOff`	If set to 1, position, orientation, and velocity data is interpolated between discrete data points during display in SIMDIS. If 0, no interpolation is performed.

Example: **`PlatformInterpolate`** `2 1`

`PlatformRefLLA` `ID Lat Lon Alt`

`ID`	Unique ID of the platform referenced by this keyword.
`Lat`	Reference latitude (in degrees) of flat Earth or tangent plane coordinate system for the platform.
`Lon`	Reference longitude (in degrees) of flat Earth or tangent plane coordinate system for the platform.
`Alt`	Reference altitude (in meters) of flat Earth or tangent plane coordinate system for the platform. Only used for flat Earth or tangent plane systems.

If not specified, the scenario's reference LLA values are used.

Example: **`PlatformRefLLA`** `2 22.0 -160.0 0.0`

`PlatformCoordSystem` `ID System`

`ID`	Unique ID of the platform referenced by this keyword.
`System`	Type of coordinate system in which the platform's data points are represented. See Table A.1 for supported formats. If not specified, the scenario's coordinate system value is used.

Example: `PlatformCoordSystem 2 "ECEF"`

`PlatformMagneticVariance ID MagVar`

`ID`	Unique ID of the platform referenced by this keyword.
`MagVar`	Magnetic variance (declination) for the platform between magnetic north and true north. Considered positive east of true north and negative when west. See Table A.2 for supported options. "TRUE" is the default option.

Example: `PlatformMagneticVariance 2 "TRUE"`

WMM data is only valid for a five-year period. Presently, SIMDIS can predict the magnetic variance using the WMM for data within the ranges of 1985 to 2025. Additional information regarding the WMM can be found on the National Geophysical Data Center's (NGDC) web site:

http://www.ngdc.noaa.gov/geomag/WMM/back.shtml.

For a user specified variance, the magnetic declination is considered as the angle between the local magnetic field, the direction the north end of a compass points and true north. The declination is positive when the magnetic north is east of true north. Units for the user specified variance must be in degrees. If this value is not specified, the scenario's magnetic variance value is used. If a magnetic variance is not specified by either the scenario or the platform, then all yaw values will be referenced to true north. This setting will be ignored if the input data is in one of the Earth-Centered coordinate systems.

`PlatformVerticalDatum ID VertDatum`

`ID`	Unique ID of the platform referenced by this keyword.
`VertDatum`	Zero surface to which elevations or heights for this platform are referenced. See Table A.3 for supported options. "WGS84" is the default option.

Example: `PlatformVerticalDatum 2 "WGS84"`

In order to approximate Mean Sea Level, SIMDIS uses the WGS-84 Earth Gravitational Model 1996 (EGM96). This model consists of a height file with a 0.25 degree grid of point values in the tide-free system. Bi-linear interpolation is used to obtain the associated height value based on a specified latitude and longitude. Additional information on the EGM96 can be found on the National Geospatial-Intelligence Agency's website:

http://earth-info.nga.mil/GandG/wgs84/gravitymod/index.html.

If this value is not specified, the scenario's vertical datum value is used. If a vertical datum is not specified by either the scenario or the platform, then all height values will be referenced to the WGS-84 ellipsoid. See Table A.4 for information on the availability and reference surface of the vertical datum offsets per coordinate system.

`PlatformReferenceTimeECI ID Time`

`ID`	Unique ID of the platform referenced by this keyword.
`Time`	Time at which the Earth-Centered Inertial (ECI) reference frame for this platform is defined (in seconds).

Example: `PlatformReferenceTimeECI 2 "0.0"`

The reference time is also known as the Greenwich Mean Sidereal Time (GMST), and is the time since the ECI system was aligned with the vernal equinox and the ECEF system. The reference time has a valid range of values from 0 to 86400 seconds. If this value is not specified, the scenario's reference time value is used. If neither the scenario nor the platform reference value is specified, then zero will be used. SIMDIS uses the ECI reference time to calculate an elapsed time which is multiplied by the Earth's angular velocity in order to determine the corresponding ECEF position.

`PlatformTangentPlaneOffset` `ID XOffset YOffset Rotation`

`ID`	Unique ID of the platform referenced by this keyword.
`XOffset`	X translation position (in meters) offset from the Generic Tangent Plane (GTP) coordinate system for this platform.
`YOffset`	Y translation position (in meters) offset from the Generic Tangent Plane (GTP) coordinate system for this platform.
`Rotation`	Rotation angle (in degrees) applied to the x-y plane.

Example: **`PlatformTangentPlaneOffset`** `2 -14063.024 5641.235 13.145999`

The X offset position is true east distance; the Y offset position is the true north distance in meters of the desired origin as seen from the tangential point. The X offset is positive if the origin lies to the east of the tangential point, and is netative if the origin lies to the west. The Y offset is positive if the origin lies to the north of the tangential point, and is negative if the origin lies to the south. The rotation angle is the desired angle in degrees to rotate the X-Y plane. Positive angles perform a clockwise rotation referenced to true north. If this value is not specified, the scenario's tangent plane offset values are used.

`PlatformUsesQuaternion` `ID ScalarPos`

`ID`	Unique ID of the platform referenced by this keyword.
`ScalarPos`	Position of the quaternion scalar value with respect to the quaternion vector. See Table A.8 for available positions. 0 is the default.

Example: **`PlatformUsesQuaternion`** `2 0`

Internally, SIMDIS will convert the four element quaternion into the equivalent three Euler angle rotations based on a yaw, pitch and roll sequence recovered from a direction cosine matrix. The formula for converting quaternions to Euler angles is derived from From "Aircraft Control and Simulation 2nd Edition", by B. Stevens & F. Lewis 2003, ISBN 0-471-37145-9, using equations 1.3-24 and 1.3-32.

`PlatformIcon` `ID IconFile`

`ID`	Unique ID of the platform referenced by this keyword.
`IconFile`	3D model file used to depict the platform. See `$(SIMDIS_DIR)/data/models` for available files. File extension is optional. If this keyword is not specified, no icon is used.

Example: **`PlatformIcon`** `2 "uss_arleigh_burke_class_ddg"`
Example: **`PlatformIcon`** `2 "uss_arleigh_burke_class_ddg.3db"`

Platform also supports the <**`OriginalID`**> keyword. See the documentation for information.

Value	Meaning
`0 or "unknown"`	Unknown
`1 or "ship"`	Ship
`2 or "submarine"`	Submarine
`3 or "aircraft"`	Aircraft
`4 or "satellite"`	Satellite
`5 or "helicopter"`	Helicopter
`6 or "missile"`	Missile
`7 or "decoy"`	Decoy
`8 or "buoy"`	Buoy
`9 or "reference"`	Reference
`10 or "vehicle"`	Vehicle (cars, trucks, tanks, etc.)
`11 or "landsite"`	Land Site (buildings, airfields, Radar installations, etc.)
`12 or "torpedo"`	Torpedo
`13 or "contact"`	Contact (Radar, sonar, visual, etc.)

Table A.5: Available categories for Platform Type

Value	Meaning
F	Friendly
H	Hostile
N	Neutral

Table A.6: Available values for the Platform FHN

Value	Meaning
LUT	ASCII lookup table file format, see appendix for more details
SADM	Ship Air Defence Model file format, see appendix for more details
XPatch	XPatch (SAIC/Demaco) RCS file format, see appendix for more details

Table A.7: Supported RCS file formats

Value	Quaternion form
0	q0 + q1i + q2j + q3k
1	q3k + q0 + q1i + q2j
2	q2j + q3k + q0 + q1i
3	q1i + q2j + q3k + q0

Table A.8: Supported positions of the quaternion scalar

A.2.2 Platform Data

Data points for a platform can be defined after the platform is defined. The required <**PlatformID**> keyword is used to define data points for a platform. The minimum definition of a PlatformData is as follows:

Position only:

`PlatformData ID Time XPos YPos ZPos`

`ID`	Unique ID of the platform referenced by this keyword.
`Time`	Time at which this data is valid.
`XPos`	X or latitude position of the platform (meters, radians, or degrees).
`YPos`	Y or longitude position of the platform (meters, radians, or degrees).
`ZPos`	Z or altitude position of the platform (meters).

Example: `PlatformData 2 0.0 15.0 32.8 10.0`

When only position data is specified, velocity is calculated based on an instantaneous velocity between the last data point and the next data point such that vel = delta position / delta time. Orientation is then calculated based on flight path angles derived from the velocity vector. Note that only points that have already been defined will be used for this calculation. Out-of-order points defined later in the ASI file will not be considered even if they have an earlier time.

The following values can optionally be appended to the minimum PlatformData definition, depending on the desired format. Previously documented input parameters are not re-documented unless required.

Orientation only:

`PlatformData ID Time XPos YPos ZPos XOri YOri ZOri`

`XOri`	Yaw/Psi orientation of the platform (radians or degrees).
`YOri`	Pitch/Theta orientation of the platform (radians or degrees).
`ZOri`	Roll/Phi orientation of the platform (radians or degrees).

Example: `PlatformData 2 0.0 15.0 32.8 10.0 0.4 0.2 1.1`

Orientation and Speed:

`PlatformData ID Time XPos YPos ZPos XOri YOri ZOri Speed`

`Speed`	Speed of the platform (meters/sec).

Example: `PlatformData 2 0.0 15.0 32.8 10.0 0.4 0.2 1.1 5.0`

If only speed is specified, a velocity vector will be created based on orientation.

Orientation and Velocity:

`PlatformData ID Time XPos YPos ZPos XOri YOri ZOri XVel YVel ZVel`

`XVel`	X component of the platform's velocity vector (meters/sec).
`YVel`	Y component of the platform's velocity vector (meters/sec).
`ZVel`	Z component of the platform's velocity vector (meters/sec).

Example: `PlatformData 2 0.0 15.0 32.8 10.0 0.4 0.2 1.1 1.0 2.0 2.0`

Quaternion and Speed:

`PlatformData ID Time XPos YPos ZPos Q0 Q1 Q2 Q3 Speed`

- `Q0` 0 value of the quaternion vector.
- `Q1` 1 value of the quaternion vector.
- `Q2` 2 value of the quaternion vector.
- `Q3` 3 value of the quaternion vector.

Example: `PlatformData 2 0.0 15.0 32.8 10.0 1 3 2 5 5.0`

If only speed is specified, a velocity vector will be created based on orientation.

Quaternion and Velocity:

`PlatformData ID Time XPos YPos ZPos Q0 Q1 Q2 Q3 XVel YVel ZVel`

Example: `PlatformData 2 0.0 15.0 32.8 10.0 1 3 2 5 1.0 2.0 2.0`

Note: The <`PlatformUsesQuaternion`> keyword should be set when using a <`PlatformData`> format that includes quaternion values.

All angle units are radians unless the scenario keyword <`DegreeAngles`> is set.

If either an ECEF or ECI coordinate system is used, orientation is represented by the Euler angles psi (ψ), theta (θ) and phi (ϕ). The orientation representation of the Euler angles is presented below.

SIMDIS supports three scaled flat Earth systems, ENU, NED and NWU. In a Scaled Flat Earth system, the Earth's surface is projected (warped) onto an X-Y plane of a Cartesian coordinate system, with the reference origin specified at a latitude and longitude surface location.

The ENU system is represented as:

Positive X axis points East
Positive Y axis points North
Positive Z axis points Up (positive altitude is positive Z)
True North [000] is looking down the Y-axis
East [090] is looking down the X-axis
Positive yaw/course is clockwise from North

The NED system is represented as:

Positive X axis points North
Positive Y axis points East
Positive Z axis points Down (positive altitude is negative Z)
True North [000] is looking down the X-axis
East [090] is looking down the Y-axis
Positive yaw/course is clockwise from North

The NWU system is represented as:

Positive X axis points North
Positive Y axis points West
Positive Z axis points Up (positive altitude is positive Z)
True North [000] is looking down the X-axis
East [090] is looking down the negative Y-axis
Positive yaw/course is clockwise from North

All three systems supported are right-handed.

For all flat Earth projections, the local rectilinear coordinate system is a scaled flat Earth system centered with the origin at a specified reference latitude, longitude, and altitude. Altitude is referenced to height above the WGS-84 ellipsoid. The scaling of the latitude and longitude values into the flat Earth system is based on the values of the reference origin. The resulting scaled flat Earth only maintains proper scale, direction, distance and area within a short range of the reference origin. The reference origin is specified by either the <**PlatformRefLLA**> or the <**RefLLA**> keywords.

SIMDIS also supports a geodetic coordinate system, LLA. Geodetic coordinates are latitude, longitude, and altitude referenced to the ellipsoidal height (height above the WGS-84 ellipsoid). Latitude ranges are +90.0 (North) to -90.0 (South). Longitude ranges are +180.0 (East) to -180.0 (West). Positive altitude is considered +Z values. If the <**DegreeAngles**> value is set, latitude and longitude can be entered in degrees decimal (D.D), degrees minutes decimal (D M.D) or degrees minutes seconds (DMS), otherwise radians are the input angle unit. If either the degrees minutes decimal or degrees minutes seconds notation is used, the values must be enclosed within quotes, e.g.:

```
// degrees minutes decimal notation
PlatformData 1 0.0 "21 34.88243" "-159 12.47353" 0.0
```

```
// degrees minutes seconds notation
PlatformData 1 0.0 "21 34 52.946" "-159 12 28.412" 0.0
```

SIMDIS also supports two tangent plane projections, XEAST and Generic Tangent Plane (GTP). Both projections define a local rectilinear coordinate system based on a flat plane tangent to the Earth's surface at a specific reference origin. Lines of equal distortion are concentric about the reference origin. Further the distance from the origin, the greater the distortion. The XEAST system is aligned with the Earth such that +X is East, +Y is North, and +Z is up. For the GTP system, the reference origin point does not have to be at this tangential point, but can lie at some other selected point on the x-y plane. Furthermore, the x-y axes do not have to be oriented so that the +y axis points to true north; they can be rotated to any desired angle. Either the keyword <**PlatformTangentPlaneOffset**> or the keyword <**TangentPlaneOffse**> is used to specify the position translation and rotation offset. The "raw" tangent plane values are first translated by x and y, then the rotation is applied. If no offsets are applied, the GTP system becomes a XEAST system. The reference origin is specified by either the <**PlatformRefLLA**> or the <**RefLLA**> keywords.

SIMDIS also supports two geocentric coordinate systems, Earth-Centered Earth Fixed (ECEF) and Earth-Centered Inertial (ECI). Both the ECEF and ECI systems used in SIMDIS are based on the WGS-84 ellipsoidal Earth model.

A.3 Beam

A.3.1 Beam Initialization

There are four keywords that are [**REQUIRED**] to be specified when defining a beam.

`BeamID` `HostID ID`

`HostID`	Unique ID of the host platform for the beam.
`ID`	Unique ID assigned to the beam on creation. This ID is used to associate data with this beam. Must be a positive non-zero integer.

`BeamType` `ID Type`

`ID`	Unique ID of the beam associated with this type.
`Type`	Beam type for this beam. Controls the method of processing when parsing the <**BeamDataRAE**> command. See Table A.9 for available beam types.

`VertBW` `ID BW`

`ID`	Unique ID of the beam referenced by this keyword.
`BW`	Vertical beam width (in degrees) of the beam, which is the angular width of a radar beam between the half-power points in the vertical plane. Valid range is (0, 180].

`HorzBW` `ID BW`

`ID`	Unique ID of the beam referenced by this keyword.
`BW`	Horizontal beam width (in degrees) of the beam, which is the angular width of a radar beam between the half-power points in the horizontal plane. Valid range is (0, 360].

For example, to create two unique beams:

```
BeamID 1 101
BeamType 101 "ABSOLUTE"
VertBW 101 5.0
HorzBW 101 5.0
BeamID 2 102
BeamType 102 "TARGET"
VertBW 102 3.4
HorzBW 102 3.4
```

The following [**OPTIONAL**] keywords can be used to further define a platform.

`BeamDesc` `ID Desc`

`ID`	Unique ID of the beam referenced by this keyword.
`Desc`	Brief text description of the beam.

Example: **`BeamDesc`** `"Seeker"`

`AzimOffset` `ID Offset`

`ID`	Unique ID of the beam referenced by this keyword.
`Offset`	Azimuth angle offset (in degrees) relative to True North. Positive offsets are clockwise.

Example: `AzimOffset 101 5.0`

`ElevOffset ID Offset`

`ID`	Unique ID of the beam referenced by this keyword.
`Offset`	Elevation angle offset (in degrees) relative to the horizon. Positive offsets are upward.

Example: `ElevOffset 101 5.0`

`RollOffset ID Offset`

`ID`	Unique ID of the beam referenced by this keyword.
`Offset`	Roll angle offset (in degrees). Positive offsets are to the right when looking from the host platform down the axis of the beam.

Example: `RollOffset 101 0.0`

`BodyOffset ID XOffset YOffset ZOffset`

`ID`	Unique ID of the beam referenced by this keyword.
`XOffset`	X body offset (in meters) with respect to the host platform center.
`YOffset`	Y body offset (in meters) with respect to the host platform center.
`ZOffset`	Z body offset (in meters) with respect to the host platform center.

Example: `BodyOffset 101 0.0 0.0 -10.0`

`BeamMissileOffset ID`

`ID`	Unique ID of the beam referenced by this keyword.

Example: `BeamMissileOffset 101`

In ASI versions 2 through 8, a beam would automatically be offset by SIMDIS when the host platform was a missile. This automatic offset would enable the beam to emit from the nose instead of center of the platform. This functionality is supported for old versions for backwards compatibility. As of ASI version 9, this keyword allows SIMDIS to automatically adjust the beam position to the "nose" of the host platform, regardless of type.

`AREPSPattern ID File`

`ID`	Unique ID of the beam referenced by this keyword.
`File`	ASCII Advanced Refractive Effects Prediction System (AREPS) pattern (.txt) or Advanced Propagation Model (APM) input (.in) file to load. SIMDIS supports AREPS ASCII file version 3.x.

Example: `AREPSPattern 101 "file_APM_0_15_15.txt"`

NOTE: More than one AREPS file can be loaded and associated to a beam.

For additional information regarding AREPS and APM, see the Space and Naval Warfare Systems Center Atmospheric Propagation Branch's web site: http://areps.spawar.navy.mil/

The SIMDIS RF Propagation Tool is used to manage and display either the AREPS or APM data.

AntennaPattern `ID PatternType PatternFile`

`ID`	Unique ID of the beam referenced by this keyword.
`PatternType`	Antenna pattern for the beam. See Table A.10 for available pattern types.
`PatternFile`	Antenna pattern file for the beam, of the type specified by `PatternType`.

Example: `AntennaPattern 101 "TABLE" "bogey_ant.aptf"`

AntennaAlgorithm `ID Algorithm`

`ID`	Unique ID of the beam referenced by this keyword.
`Algorithm`	Antenna pattern algorithm for the beam. See Table A.11 for available pattern alrogithm types.

Example: `AntennaAlgorithm 101 "PEDESTAL"`

NOTE: The <AntennaPattern> and <AntennaAlgorithm> keywords are mutually exclusive. You can only use one or the other per beam.

If either <AntennaPattern> or <AntennaAlgorithm> is specified, then the <AntennaGain> <AntennaPeakPower> and <AntennaFrequency> keywords are **required** in order to correctly display the antenna pattern.

AntennaGain `ID Gain`

`ID`	Unique ID of the beam referenced by this keyword.
`Gain`	Main lobe antenna gain (in dB) with respect to the antenna pattern of the beam.

Example: `AntennaGain 101 30.0`

AntennaPeakPower `ID Peak`

`ID`	Unique ID of the beam referenced by this keyword.
`Peak`	Peak power output (in watts) of the beam.

Example: `AntennaPeakPower 101 20000.0`

AntennaFrequency `ID Freq`

`ID`	Unique ID of the beam referenced by this keyword.
`Freq`	Transmit frequency (in MHz) of the beam.

Example: `AntennaFrequency 101 1000`

AntennaPolarity `ID Polarity`

`ID`	Unique ID of the beam referenced by this keyword.
`Polarity`	Polarization of the beam. See Table A.12 for available polarization values.

Example: `AntennaPolarity 101 "UNKNOWN"`

BeamInterpolate `ID Interpolate`

`ID`	Unique ID of the beam referenced by this keyword.
`Interpolate`	If set to 1, interpolation is performed between discrete beam data points. If set to 0, no interpolation is performed.

Example: `BeamInterpolate 101 1`

Beam also supports the <`OriginalID`> keyword. See the documentation for information.

NOTE: If any antenna pattern parameter is specified, SIMDIS will create an initial pattern that uses a Sin(x)/x algorithm pattern with a gain of 20 dB, peak power of 2000 W, a frequency of 7000 MHz and an unknown polarity. Subsequent commands will override these initial settings.

A.3.2 Beam Data

Data points for a beam can be defined after the beam is defined. There are three [**REQUIRED**] keywords used to specify beam data input.

`BeamOnOffCmd ID Time OnOff`

`ID`	Unique ID of the beam referenced by this keyword.
`Time`	Time at which this data is valid.
`OnOff`	If set to 1, the beam is turned on. If set to 0, the beam is turned off.

Example: `BeamOnOffCmd 101 1.0 1`

`BeamColorCmd ID Time Color`

`ID`	Unique ID of the beam referenced by this keyword.
`Time`	Time at which this data is valid.
`Color`	Color of the beam. See Section A.12.2 for information on accepted color formats.

Example: `BeamColorCmd 101 1.0 yellow`

The third required keyword depends on the beam type specified by the <`BeamType`> keyword. For `ABSOLUTE`, `BODY`, or `UNKNOWN` type beams:

`BeamDataRAE ID Time Az Elev Range`

`ID`	Unique ID of the beam referenced by this keyword.
`Time`	Time at which this data is valid.
`Az`	Azimuth (in radians or degrees) of the beam, measured clockwise from north.
`Elev`	Elevation (in radians or degrees) of the beam. Positive values are measured above the horizon.
`Range`	Range (in meters) of the beam from the host platform.

Example: `BeamDataRAE 101 1.0 0.1745329 0.1 10000.0`
`Az` and `Elev` angle units are radians unless the scenario keyword <`DegreeAngles`> is set.

For `TARGET` type beams:

`BeamTargetIDCmd ID Time TargetID`

`ID`	Unique ID of the beam referenced by this keyword.
`Time`	Time at which this data is valid.
`TargetID`	Unique ID of the tracked target. The target is used to calculate the pointing angles for the beam.

Example: **BeamTargetIDCmd** 101 1.0 23

The actual active state of a beam in SIMDIS at a given time is dependent on the presence of a valid host platform, a <**BeamOnOffCmd**> command as well as either a valid <**BeamDataRAE**> data point or <**BeamTargetIDCmd**> command. All three items are required to exist at the given time in order for the beam to be considered active and drawn.

Type	Beam data parsed as
ABSOLUTE	an absolute position
BODY	relative to the host platform's body orientation
TARGET	relative to the tracked target
UNKNOWN	the ABSOLUTE case

Table A.9: Available beam types

Type	Description
TABLE	see Section A.12.3.1 for details
RELATIVE	see Section A.12.3.2 for details
MONOPULSE	see Section A.12.3.3 for details
BILINEAR	see Section A.12.3.4 for details
NSMA	see Section A.12.3.5 for details
EZNEC	see Section A.12.3.6 for details
XFDTD	see Section A.12.3.7 for details

Table A.10: Available antenna pattern types

Value	Meaning
PEDESTAL	Pedestal antenna pattern model
GAUSS	Gaussian antenna pattern model
CSCSQ	Cosecant squared antenna pattern model
SINXX	Sin (x)/x antenna pattern model
OMNI	Omni directional antenna pattern model

Table A.11: Available antenna pattern algorithms

Value	Meaning
`Unknown`	Unknown polarization
`Horizontal`	Horizontal polarization
`Vertical`	Vertical polarization
`Circular`	Circular polarization
`HorzVert`	Horizontal transmit, Vertical receive polarization
`VertHorz`	Vertical transmit, Horizontal receive polarization
`LeftCirc`	Left-hand circular polarization
`RightCirc`	Right-hand circular polarization
`Linear`	Linear polarization

Table A.12: Available antenna polarization values

A.4 Gate

A.4.1 Gate Initialization

There are two keywords that are [**REQUIRED**] to be specified when defining a gate.

`GateID HostID ID`

`HostID`	Unique ID of the host beam for the gate.
`ID`	Unique ID assigned to the gate on creation. This ID is used to associate data with this gate. Must be a positive non-zero integer.

`GateType ID Type`

`ID`	Unique ID of the gate associated with this type.
`Type`	Gate type for this gate. Controls the method of processing when parsing the <**GateDataRAE**> command. See Table A.13 for available gate types.

For example:

```
GateID 101 200
GateType 200 "COVERAGE"
```

A COVERAGE type gate is drawn as a spherical wedge; the combination of vertical beam width and the elevation pointing value cannot exceed 90 degrees. The remaining gate types are drawn as a spherical patch that can be elevated to 90 degrees. The images below depict a COVERAGE gate type on the left and the remaining gate types on the right.

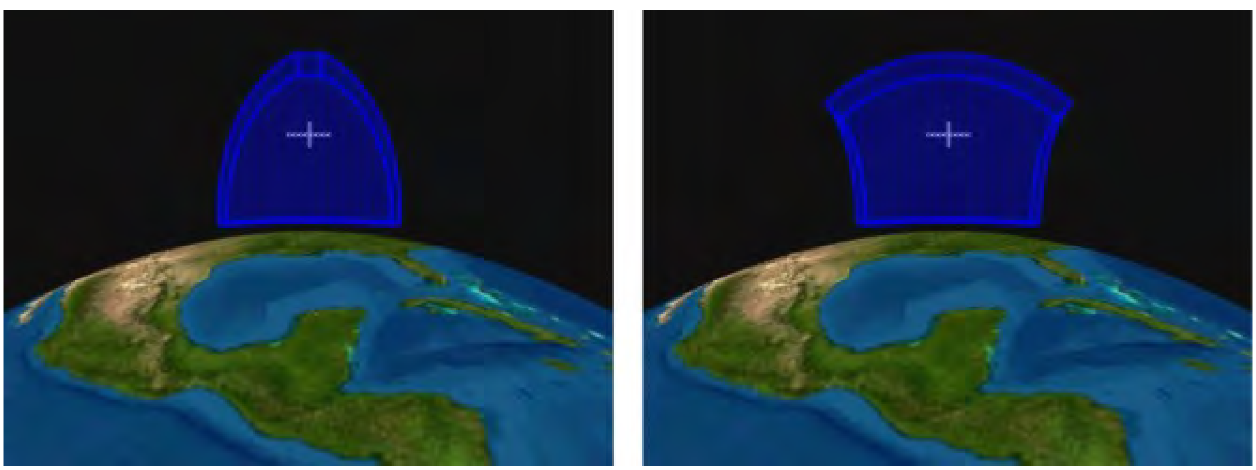


Figure A.1: Coverage Gate Types shown on the left.

NOTE: Coverage and Search Volumes would typically be drawn with a "COVERAGE" gate type while gates for tracking Radars would be represented using the other gate types.

The following [**OPTIONAL**] keywords can be used to further define a gate.

`GateDesc ID Desc`

`ID`	Unique ID of the gate referenced by this keyword.
`Desc`	Brief text description of the gate.

Example: **`GateDesc`** `200 "Range Gate"`

`GateInterpolate` `ID Interpolate`

`ID`	Unique ID of the gate referenced by this keyword.
`Interpolate`	If set to 1, interpolation is performed between discrete gate data points. If set to 0, no interpolation is performed.

Example: **`GateInterpolate`** `200 1`

`AzimOffset` `ID Offset`

`ID`	Unique ID of the gate referenced by this keyword.
`Offset`	Azimuth angle offset (in degrees) relative to True North. Positive offsets are clockwise.

Example: **`AzimOffset`** `200 5.0`

`ElevOffset` `ID Offset`

`ID`	Unique ID of the gate referenced by this keyword.
`Offset`	Elevation angle offset (in degrees) relative to the horizon. Positive offsets are upward.

Example: **`ElevOffset`** `200 5.0`

`RollOffset` `ID Offset`

`ID`	Unique ID of the gate referenced by this keyword.
`Offset`	Roll angle offset (in degrees). Positive offsets are to the right when looking from the host platform towards the gate.

Example: **`RollOffset`** `200 0.0`

Gate also supports the <**`OriginalID`**> keyword. See the documentation for information.

A.4.2 Gate Data

Data points for a gate can be defined after the gate is defined. There are three [**REQUIRED**] keywords used to specify gate data input.

`GateOnOffCmd` `ID Time OnOff`

`ID`	Unique ID of the gate referenced by this keyword.
`Time`	Time at which this data is valid.
`OnOff`	If set to 1, the gate is turned on. If set to 0, the gate is turned off.

Example: **`GateOnOffCmd`** `200 1.0 1`

`GateColorCmd` `ID Time Color`

`ID`	Unique ID of the gate referenced by this keyword.
`Time`	Time at which this data is valid.
`Color`	Color of the gate. See Section A.12.2 for information on accepted color formats.

Example: **GateColorCmd** `200 1.0 blue`

GateDataRAE `ID Time Az Elev AngWidth AngHeight StartPos EndPos CenterPos`

`ID`	Unique ID of the gate referenced by this keyword.
`Time`	Time at which this data is valid.
`Az`	Azimuth (in radians or degrees) of the gate, measured clockwise from north. Ignored if the gate is a `TARGET` type.
`Elev`	Elevation (in radians or degrees) of the gate. Positive values are measured above the horizon. Ignored if the gate is a `TARGET` type.
`AngWidth`	Angular width (in radians or degrees) of the gate. Set to less than or equal to 0 to use the host beam's horizontal beam width.
`AngHeight`	Angular height (in radians or degrees) of the gate. Set to less than or equal to 0 to use the host beam's vertical beam width.
`StartPos`	Start position (in meters) of the gate relative to the host or target platform.
`EndPos`	End position (in meters) of the gate relative to the host or target platform.
`CenterPos`	Center position (in meters) of the gate relative to the host or target platform. This value is optional. If not specified, a `CenterPos` will be generated halfway between `StartPos` and `EndPos`.

Example: **GateDataRAE** `200 1.0 2.1 0.15 0.5 0.5 10000. 11000. 10500.`
All angle units are radians unless the scenario keyword <**DegreeAngles**> is set.

The actual active state of a gate in SIMDIS at a given time is dependent on the presence of a valid host beam, a <**GateOnOffCmd**> command and a valid <**GateDataRAE**> data point command. All three are required to exist at the given time in order for the gate to be considered active and drawn.

If the host beam is a `TARGET` type beam, then the beam's <**BeamTargetIDCmd**> commands are used to set a valid target for the gate as well as the beam. If no valid target is found, the gate data is ignored.

If the gate is a `TARGET` type gate, the `StartPos`, `EndPos`, and `CenterPos` positions are relative to the target platform.

Type	Gate data parsed as
`BODY`	relative to the host platform's body orientation
`COVERAGE`	an absolute position, but drawn as a spherical wedge. See Figure A.1.
`TARGET`	relative to the tracked target
`User Specified`	an absolute position and marked as an **UNKNOWN** type.

Table A.13: Available gate types

A.5 Laser

A.5.1 Laser Initialization

There is one keyword that is [**REQUIRED**] to be specified when defining a laser.

`Laser HostID ID Name`

`HostID`	Unique ID of the host platform for the laser.
`ID`	Unique ID assigned to the laser on creation. This ID is used to associate data with this laser. Must be a positive non-zero integer.
`Name`	Name of the laser.

Example: `Laser 23 120 "Death Star"`

The following [**OPTIONAL**] keywords can be used to further define a laser.

`LaserCoordSystem ID System`

`ID`	Unique ID of the laser referenced by this keyword.
`System`	Type of coordinate system in which the laser's data points are represented. See Table A.1 for supported formats. The default is LLA. ECEF coordinate system for lasers is deprecated as of ASI version 22. Additionally, ECI and GTP coordinate systems are not supported for lasers.

Example: `LaserCoordSystem 120 LLA`

`LaserAzElBody ID Relative`

`ID`	Unique ID of the laser referenced by this keyword.
`Relative`	If set to 1, indicates orientation data is relative to the host platform's body coordinate system. If set to 0, not relative. 0 is the default.

Example: `LaserAzElBody 120 0`

Laser also supports the <`OriginalID`> keyword. See the documentation for information.

A.5.2 Laser Data

Data points for a laser can be defined after the laser is defined. There is one [**REQUIRED**] keyword used to specify laser data input.

`LaserOri ID Time Yaw Pitch Roll`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`Yaw`	Yaw or psi (in radians or degrees) of the laser's orientation.
`Pitch`	Pitch or theta (in radians or degrees) of the laser's orientation.
`Roll`	Roll or phi (in radians or degrees) of the laser's orientation.

Example: `LaserOri 120 1.0 60.0 25.0 0.0`
Orientation angle units are radians unless the scenario keyword <`DegreeAngles`> is set.

The following [**OPTIONAL**] keywords can be used to further define laser input data.

`LaserXYZOffset` `ID Time XOffset YOffset ZOffset`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`XOffset`	X position offset (in meters) from the host platform's origin. Defaults to 0.
`YOffset`	Y position offset (in meters) from the host platform's origin. Defaults to 0.
`ZOffset`	Z position offset (in meters) from the host platform's origin. Defaults to 0.

Example: `LaserXYZOffset 120 1.0 5.0 0.0 0.0`

`LaserMaxRange` `ID Time Range`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`Range`	Maximum range (in meters) to use when rendering the laser. Defaults to 1,000,000.

Example: `LaserMaxRange 120 1.0 50000.0`

`LaserWidth` `ID Time Width`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`Width`	Width (in pixels) to use when rendering the laser. Defaults to 1.

Example: `LaserWidth 120 1.0 10`

`LaserColor` `ID Time Color`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`Color`	Color of the laser. Defaults to red. See Section A.12.2 for information on accepted color formats.

Example: `LaserColor 120 1.0 0xFF00FF00`

`LaserOn` `ID Time OnOff`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`OnOff`	If set to 1, the laser is on. If set to 0, the laser is off. Defaults to 0.

Example: `LaserOn 120 1.0 1`

A.6 LOB

SIMDIS can render detections or lines of bearing. Because detections or lines of bearing often have a high frequency, it makes sense to group them together. A group of detections is known as a Lines of Bearing (LOB). A group of lines in a LOB are associated with a host platform, and are displayed along the history trail of the host platform, pointing in the appropriate direction.

A.6.1 LOB Initialization

There are three keywords that are [**REQUIRED**] to be specified when defining a LOB.

`LOBID HostID ID`

`HostID`	Unique ID of the host platform for the LOB.
`ID`	Unique ID assigned to the LOB on creation. This ID is used to associate data with this LOB. Must be a positive non-zero integer.

`LOBType ID Type`

`ID`	Unique ID of the LOB referenced by this keyword.
`Type`	Type indicating how LOB data should be interpreted. `ABSOLUTE` interprets angles as absolute angles relative to true north. `RELATIVE` interprets angles as relative to the host platform's body orientation.

`LOBDesc ID Desc`

`ID`	Unique ID of the LOB referenced by this keyword.
`Desc`	Brief text description or name of the LOB.

For example:

```
LOBID 11 16
LOBType 16 "ABSOLUTE"
LOBDesc 16 "Helo 1 Detections"
```

LOB also supports the <**`OriginalID`**> keyword. See the documentation for information.

A.6.2 LOB Data

Data points for a LOB can be defined after the LOB is defined. There is one [**REQUIRED**] keyword used to specify LOB data input.

`LOBData ID Time Yaw Pitch Range`

`ID`	Unique ID of the LOB referenced by this keyword.
`Time`	Time at which this data is valid.
`Yaw`	Yaw (in radians or degrees) of the LOB.
`Pitch`	Pitch (in radians or degrees) of the LOB.
`Range`	Range (in meters) to use when rendering the LOB.

Example: `LOBData 16 1.0 120.0 -7.0 10000.0`
`Yaw` and `Pitch` angle units are radians unless the scenario keyword <**`DegreeAngles`**> is set.

The following [**OPTIONAL**] keywords can be used to further define LOB input data.

LOBDrawStyle `ID Time StippleColor1 StippleColor2 StipplePattern1 StipplePattern2 Width`

`ID`	Unique ID of the laser referenced by this keyword.
`Time`	Time at which this data is valid.
`StippleColor1`	First stipple pattern color. See Section A.12.2 for accepted color formats.
`StippleColor2`	Second stipple pattern color. See Section A.12.2 for accepted color formats.
`StipplePattern1`	Hexadecimal bit pattern for the first stipple pattern.
`StipplePattern2`	Hexadecimal bit pattern for the second stipple pattern.
`Width`	Width (in pixels) to use when rendering the LOB.

Example: **LOBDrawStyle** `16 1.0 0xFF006AFF 0x00000000 0xFF00 0x00FF 2`

A.7 Projector

SIMDIS can project imagery (from a video camera for example) into the 3D scene in a manner similar to how a video projector can project an image onto a screen. In order to do so, SIMDIS needs the position of the projector, the orientation of the projector, the Field of View (FOV) of the projector, and the projector's imagery data.

A.7.1 Projector Initialization

There are two keywords that are [**REQUIRED**] to be specified when defining a projector.

`Projector HostID ID Name`

`HostID`	Unique ID of the host platform for the projector.
`ID`	Unique ID assigned to the projector on creation. This ID is used to associate data with this projector. Must be a positive non-zero integer.
`Name`	Name of the projector.

`ProjectorRasterFile ID File`

`ID`	Unique ID of the projector referenced by this keyword.
`File`	Filename of the image or video used by the projector. See below for supported formats.

For example:

```
Projector 15 300 "Projector 18"
ProjectorRasterFile 300 "projector01.png"
```

The following [**OPTIONAL**] keywords can be used to further define a projector.

`ProjectorInterpolateFOV ID Interpolate`

`ID`	Unique ID of the projector referenced by this keyword.
`Interpolate`	If set to 1, interpolation is performed between discrete projector data points. If set to 0, no interpolation is performed.

Example: `ProjectorInterpolateFOV 300 1`

Projector also supports the <`OriginalID`> keyword. See the documentation for information.

A.7.2 Projector Data

Data points for a projector can be defined after the projector is defined. There is one [**REQUIRED**] keyword used to specify projector data input.

`ProjectorFOV ID Time FOV`

`ID`	Unique ID of the projector referenced by this keyword.
`Time`	Time at which this data is valid.
`FOV`	Field of view (FOV) angle (in degrees) of the projector.

Example: `ProjectorFOV 300 1.0 60`

One [**OPTIONAL**] keyword can be used to further define projector input data.

ProjectorOn `ID Time OnOff`

`ID`	Unique ID of the projector referenced by this keyword.
`Time`	Time at which this data is valid.
`OnOff`	If set to 1, the projector is on. If set to 0, the projector is off. Defaults to 0.

Example: **ProjectorOn** `300 1.0 1`

A.8 Custom Rendering

SIMDIS can display custom rendered shapes as first-class entities known as Custom Renderings. See Section A.12.6 for information on Custom Rendering, available Renderers, and examples.

A.8.1 Custom Rendering Initialization

There is one keyword that is [**REQUIRED**] to be specified when defining a custom rendering.

`CustomRendering HostID ID Name Renderer`

`HostID`	Unique ID of the host platform for the custom rendering.
`ID`	Unique ID assigned to the custom rendering on creation. This ID is used to associate data with this custom rendering. Must be a positive non-zero integer.
`Name`	Name of the custom rendering.
`Renderer`	Name of the renderer used by this custom rendering.

Example: `CustomRendering 11 13 "Detections" "Error Ellipse"`

Custom rendering also supports the <**OriginalID**> keyword. See the documentation for information.

A.8.2 Custom Rendering Data

Data points for a custom rendering are defined used Data Tables. See Section A.12.6 for more information.

One [**OPTIONAL**] keyword can be used to further define custom rendering input data.

`CustomRenderingColor ID Time Color`

`ID`	Unique ID of the custom rendering referenced by this keyword.
`Time`	Time at which this data is valid.
`Color`	Color of the custom rendering. Defaults to yellow. See Section A.12.2 for accepted color formats.

Example: `CustomRenderingColor 13 1.0 0xAAFFFFFF`

A.9 Generic Data

Generic Data is a time-stamped text string for an entity or scenario, grouped by a tag name. There are reserved Generic Data commands that result in a change to the associated entity or scenario. Those commands are defined beginning in Section A.9.1.

There is one keyword that is [**REQUIRED**] when defining Generic Data.

```
GenericData ID Tag Data Time
```

`ID`	Unique ID of the entity (or 0 for the scenario) to associate with this Generic Data.
`Tag`	Tag for this Generic Data (as a character string).
`Data`	Data for this Generic Data (as a character string).
`Time`	Time at which this data is valid.

For example:

```
GenericData 601 "SHIPEVENT" "ship moves backward" 10.0
GenericData 602 "MISSILEEVENT" "bogey : turns on" 0.0
GenericData 602 "MISSILEEVENT" "bogey : looks for target" 2.0
GenericData 602 "MISSILEEVENT" "bogey : 2nd gate appears" 4.0
```

A <**GenericData**> command may appear before the definition of the entity specified by `ID`. `Time` is only valid if it falls within the beginning and ending time bounds of the entity referenced by `ID`.

To remove or clear custom Generic Data, use the `"ClearGenData"` keyword in the data field. For example, to remove any `"MISSILEEVENT"` Generic Data prior to time `6.0` from the entity specified by `ID 602`:

```
GenericData 602 "MISSILEEVENT" "ClearGenData" 6.0
```

Usage and behavior of `"ClearGenData"` for entity specific Generic Data commands are explained on a per-command basis in their sections below.

A.9.1 Reserved Generic Data Commands

Instead of relying on a purely textual representation, each of the following reserved Generic Data commands performs a graphics-based function within SIMDIS. Each command's tag is prefixed with `SIMDIS_`. For each of the following reserved Generic Data commands, the `Tag`, `Data`, function, and use of `"ClearGenData"` is defined. `Data` input parameters will be **red** when the parameter is required and **teal** when the parameter is optional.

The following commands are available to all entity types:

A.9.1.1 Call Sign

`SIMDIS_Callsign` changes the name for the referenced entity. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi` and `$(SIMDIS_DIR)/demos/SIMDIS/Users/NUWC/TorpInternal_good.asi`.

`Tag: SIMDIS_Callsign`

`Data: Name`

`Name` Name (as a character string) to set as the entity's call sign.

Examples:

```
GenericData 602 "SIMDIS_Callsign" "Missile" "10:01:05.0"
GenericData 602 "SIMDIS_Callsign" "Missile" "10:03:09.9"
GenericData 602 "SIMDIS_Callsign" "TSRM" "10:03:10.0"
GenericData 602 "SIMDIS_Callsign" "TSRM" "10:06:09.9"
GenericData 602 "SIMDIS_Callsign" "KW" "10:06:10.0"
```

Notice how the data is specified at a start and end time for a call sign. This should be done in order to preserve the correct call sign if the data is to be played backwards or is to loop back on itself.

A `"ClearGenData"` value will disable the setting allowing the user to regain control over the entity call sign.

Example: `GenericData 0 "SIMDIS_Callsign" "ClearGenData" "10:06:20.0"`

A.9.1.2 Animated Line

`SIMDIS_AnimatedLine` displays an animated line between two entities. An entity can have multiple concurrent instances of `SIMDIS_AnimatedLine`. `Data` parameters are prefixed with a variable name and an equals sign. `Data` parameters are delimited by a comma. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/animatedLine.asi` and `$(SIMDIS_DIR)/demos/SIMDIS/Examples/animatedLine/animatedLine.asi`.

`Tag: SIMDIS_AnimatedLine ID`

`ID` ID of this SIMDIS_AnimatedLine. Each new ID is considered a unique SIMDIS_AnimatedLine within the scope of the entity.

```
Data: entityid=EntityID,stipple=Stipple,color=Color,width=Width
 shiftsPerSecond=Shifts,numSegments=NumSegs
```

EntityID	Unique ID of the entity to which to draw the line.
Stipple	Two line stipple patterns separated by a colon.
Color	Two colors separated by a colon. Color format is 0xAABBGGRR.
Width	Line width (integer).
Shifts	Shifts per second of the line (float). Greater values result in a faster shifting line. Use negative values to shift in reverse.
NumSegs	This parameter is no longer used, and can be omitted.

Example:

```
GenericData 602 "SIMDIS_AnimatedLine 1" "entityid=101,stipple=0xFF00:0x00FF,
 color=0xFF14FF0E:0xFFFF4700,width=3,shiftsPerSecond=60.0,numSegments=1" "5"
```

A "ClearGenData" value will remove the line from the display.

Example: `GenericData 602 "SIMDIS_AnimatedLine 1" "ClearGenData" "15"`

A.9.1.3 Bearing Line

SIMDIS_BearingLine displays an animated line from an entity along a specified vector. An entity can have multiple concurrent instances of SIMDIS_BearingLine. Data parameters are prefixed with a variable name and an equals sign. Data parameters are delimited by a comma. Example usage can be found in $(SIMDIS_DIR)/demos/SIMDIS/import/bearingLine.asi.

Tag: SIMDIS_BearingLine ID

ID	ID of this SIMDIS_BearingLine. Each new ID is considered a unique SIMDIS_BearingLine within the scope of the entity.

Data: stipple=Stipple,color=Color,width=Width
shiftsPerSecond=Shifts,numSegments=NumSegs,rng=Range,az=Az,el=Elev

Stipple	Two line stipple patterns separated by a colon.
Color	Two colors separated by a colon. Color format is 0xAABBGGRR.
Width	Line width (integer).
Shifts	Shifts per second of the line (float). Greater values result in a faster shifting line. Use negative values to shift in reverse.
NumSegs	This parameter is now only a placeholder.
Range	Range (in meters) of the line.
Az	Azimuth (in degrees) of the animated line, relative to true north.
El	Elevation (in degrees) of the animated line, relative to the horizon.

Example:

```
GenericData 602 "SIMDIS_BearingLine 1" "stipple=0xFF00:0x00FF,color=0xFF14FF0E:
 0xFFFF4700,width=3,shiftsPerSecond=60.0,numSegments=1,rng=1000,az=220,el=3" "5"
```

A "ClearGenData" value will remove the line from the display.

Example: `GenericData 602 "SIMDIS_BearingLine 1" "ClearGenData" "15"`

A.9.2 Scenario Generic Data

In addition to the following scenario-level reserved Generic Data commands, plug-ins can store text data in a scenario by adding Generic Data with tags that are prefixed with `"PluginData_"`.

A.9.2.1 Info Text

`SIMDIS_InfoText` displays a text label at a fixed location on screen. The label is is automatically center justified. The screen placement of the label is fixed to 75 pixels up from the bottom, in the center of the screen. The horizontal screen justification is left. The vertical screen justification is bottom. `Data` parameters are delimited by a backtick (` ). Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi`.

Tag: `SIMDIS_InfoText`

Data: ``Label`Color`Font``

- `Label` Text to display in SIMDIS (as a character string).
- `Color` Color of the text in 0xAABBGGRR format. Defaults to `0xFFFFFFFF`.
- `Font` Font and font size to use for the label. Defaults to `arialbd.ttf 35`.

Examples:

```
GenericData 0 "SIMDIS_InfoText" "3rd Stage Deployed`0xFF00FF00`arialbd.ttf 20" "10:03:10.0"
GenericData 0 "SIMDIS_InfoText" "KW Deployed`0xFF00FF00`arialbd.ttf 20" "10:06:10.0"
```

A `"ClearGenData"` value will remove all previous text created by `SIMDIS_InfoText` from the display.

Example: `GenericData 0 "SIMDIS_InfoText" "ClearGenData" "10:06:20.0"`

A.9.2.2 Screen Text

`SIMDIS_ScreenText` displays a text label at a custom location on screen. Multiple concurrent instances of `SIMDIS_ScreenText` are allowed. The horizontal screen justification is left. The vertical screen justification is bottom. `Data` parameters are delimited by a backtick (` ). Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi` and `$(SIMDIS_DIR)/demos/SIMDIS/Users/NUWC/To`

Tag: `SIMDIS_ScreenText ID`

- `ID` ID of this SIMDIS_ScreenText. Each new ID is considered a unique SIMDIS_ScreenText within the scope of the scenario.

Data: ``Label`XPos`YPos`Just`Color`Font`Width``

- `Label` Text to display in SIMDIS (as a character string).
- `XPos` X position (in percentage of screen) of the text, starting from the left.
- `YPos` Y position (in percentage of screen) of the text, starting from the bottom.
- `Just` Justification (integer) of the text string. 0 for left, 1 for center, 2 for right.
- `Color` Color of the text in 0xAABBGGRR format. Defaults to `0xFFFFFFFF`.
- `Font` Font and font size to use for the label. Defaults to `arialbd.ttf 20`.
- `Width` Outline thickness (integer) for the text. 0 for no outline, 1 for a thin outline, >= 2 for a thick outline.

Examples:

```
GenericData 0 "SIMDIS_ScreenText" "Red Color Change`50`85`1`
 0xFF0000FF`arialbd.ttf 30" "10:04:10.0"
GenericData 0 "SIMDIS_ScreenText" "Orig Color Change`50`85`1`" "10:07:10.0"
GenericData 0 "SIMDIS_ScreenText 1" "Gen Torp Run`0`97`0`
 0xFF00FFFF`arialbd.ttf 11" "0"
GenericData 0 "SIMDIS_ScreenText 2" "Tactical Cycle Count: 1`0`94`0`
 0xFF00FFFF`arialbd.ttf 11" "0"
```

A `"ClearGenData"` value will remove all previous text associated with the `ID` created by `SIMDIS_ScreenText` from the display.

Example: `GenericData 0 "SIMDIS_ScreenText" "ClearGenData" "10:08:00.0"`
Example: `GenericData 0 "SIMDIS_ScreenText 1" "ClearGenData" "10"`

A.9.2.3 Wind Direction

`SIMDIS_WindDirection` changes the wind vane arrow and text shown in the lower right hand corner of the SIMDIS display. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/wake.asi`.

Tag: `SIMDIS_WindDirection`

Data: `Direction`

`Direction` Direction (in degrees) from which the wind is blowing, referenced to true north. For example, a value of 45 means that the wind is blowing from the northeast to the southwest.

Example: `GenericData 0 "SIMDIS_WindDirection" "45" "00:01:00.0"`

`"ClearGenData"` is not supported by `SIMDIS_WindDirection` and will be ignored if specified.

A.9.2.4 Wind Speed

`SIMDIS_WindSpeed` changes the wind speed text shown in the lower right hand corner of the SIMDIS display. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/wake.asi`.

Tag: `SIMDIS_WindSpeed`

Data: `Speed`

`Speed` Speed (in meters/sec) of the wind.

Example: `GenericData 0 "SIMDIS_WindSpeed" "2" "00:01:00.0"`

`"ClearGenData"` is not supported by `SIMDIS_WindSpeed` and will be ignored if specified.

A.9.3 Platform Generic Data

A.9.3.1 Alpha Volume

`SIMDIS_AlphaVolume` controls drawing a platform's icon with a second pass that draws back-facing polygons, which shows the model even the camera is inside it. This command has been used in conjunction with `SIMDIS_OverrideColor` (see Section A.9.3.9). If `SIMDIS_OverrideColor` is present, and the color specified is semi-transparent, turning `SIMDIS_AlphaVolume` on will cause the platform's icon to be rendered in a manner similar to SIMDIS beams. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/errorEllipse.asi` and `$(SIMDIS_DIR)/demos/SIMDIS/Examples/scaleXYZandAlphaVolume/scaleXYZandAlphaVolume.asi`.

Tag: `SIMDIS_AlphaVolume`

Data: `OnOff`

`OnOff` If set to 1, alpha volume is on. If set to 0, alpha volume is off.

Example: `GenericData 602 "SIMDIS_AlphaVolume" "1" "10:01:15.0"`

A `"ClearGenData"` value will turn the setting off on the platform, which is equivalent to specifying 0 for `OnOff`.

Example: `GenericData 602 "SIMDIS_AlphaVolume" "ClearGenData" "10:08:00.0"`

A.9.3.2 Change Icon

`SIMDIS_ChangeIcon` changes the 3D icon for a platform. Example usage can be found in
`$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi` and
`$(SIMDIS_DIR)/demos/SIMDIS/Users/SADM/simdis_pres_2008_0001.asi`.

Tag: `SIMDIS_ChangeIcon`

Data: `File`

`File` Name (as a character string) of the file to set as the platform's icon.

Examples:

```
GenericData 602 "SIMDIS_ChangeIcon" "sm-3" "10:01:05.0"
GenericData 602 "SIMDIS_ChangeIcon" "sm-3" "10:03:09.9"
GenericData 602 "SIMDIS_ChangeIcon" "sm-3_3rd_stage" "10:03:10.0"
GenericData 602 "SIMDIS_ChangeIcon" "sm-3_3rd_stage" "10:06:09.9"
GenericData 602 "SIMDIS_ChangeIcon" "sm-3_kw" "10:06:10.0"
```

Notice how the data is specified at a start and end time for an icon change. This should be done in order to preserve the correct icon if the data is to be played backwards or is to loop back on itself.

A `"ClearGenData"` value will disable the setting allowing the user to regain control over the platform icon.

Example: `GenericData 620 "SIMDIS_ChangeIcon" "ClearGenData" "10:06:20.0"`

A.9.3.3 Cull Face

SIMDIS_CullFace sets whether the specified Face for a platform should be a candidate for culling when rendered. The values available to the Face parameter are string representations of the OpenGL arguments passed to the glCullFace command. Data parameters are delimited by a backtick (`). Example usage can be found in $(SIMDIS_DIR)/demos/SIMDIS/import/errorEllipse.asi.

Tag: SIMDIS_CullFace

Data: OnOff`Face

OnOff	If set to 1, the Face has cull face on. If set to 0, the Face has cull face off.
Face	Face for which to set the cull face state. Options are GL_FRONT, GL_BACK, and GL_FRONT_AND_BACK.

Example: `GenericData 602 "SIMDIS_CullFace" "0`GL_FRONT" "10:01:05.0"`

"ClearGenData" is not supported by SIMDIS_CullFace and will be ignored if specified.

A.9.3.4 Cylinder

SIMDIS_Cylinder displays a cylinder near a platform. A platform can have multiple concurrent instances of SIMDIS_Cylinder. Data parameters are prefixed with a variable name and an equals sign. SIMDIS interpolates the cylinder values when the SIMDIS time is in between two cylinder Generic Data points. Data parameters are delimited by a comma. Example usage can be found in
$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi and
$(SIMDIS_DIR)/demos/SIMDIS/Examples/cylinder/cylinder.asi.

Tag: SIMDIS_Cylinder ID

ID	ID of this SIMDIS_Cylinder. Each new ID is considered a unique SIMDIS_Cylinder within the scope of the platform.

Data: x=XOff,y=YOff,z=ZOff,px=PX,py=PY,z=PZ,l=Len,radiusNear=RadNear,
radiusFar=RadFar,colorNear=ColorNear,colorFar=ColorFar

XOff	X offset (in meters) of the cylinder from the platform's position.
YOff	Y offset (in meters) of the cylinder from the platform's position.
ZOff	Z offset (in meters) of the cylinder from the platform's position.
PX	X component of the cylinder's pointing vector.
PY	Y component of the cylinder's pointing vector.
PZ	Z component of the cylinder's pointing vector.
Len	Length of the cylinder (in meters).
RadNear	Radius of the cylinder at the start position defined by X, Y, and Z.
RadFar	Radius of the cylinder at the end position of the cylinder.
ColorNear	Color (in 0xAABBGGRR format) of the cylinder at the start position.
ColorFar	Color (in 0xAABBGGRR format) of the cylinder at the end position.

Example:

```
GenericData 602 "SIMDIS_Cylinder 1" "x=2.5,y=0,z=0,px=-1,py=0,pz=4,l=4,
 radiusNear=.2,radiusFar=0,colorNear=0xFF0000FF,colorFar=0xFFFFFFFF" "10:01:05.0"
```

A "ClearGenData" value will remove the cylinder from the display.

Example: `GenericData 602 "SIMDIS_Cylinder 1" "ClearGenData" "10:01:05.0"`

A.9.3.5 Draw Mode

SIMDIS_DrawMode changes the draw mode for a platform. Example usage can be found in $(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi.

Tag: SIMDIS_DrawMode

Data: Mode

Mode Draw mode (as a character string) for the platform. Options are solid, wire, and point.

Examples:
```
GenericData 602 "SIMDIS_DrawMode" "solid" "10:01:05.0"
GenericData 602 "SIMDIS_DrawMode" "solid" "10:03:09.9"
GenericData 602 "SIMDIS_DrawMode" "point" "10:03:10.0"
GenericData 602 "SIMDIS_DrawMode" "point" "10:06:09.9"
GenericData 602 "SIMDIS_DrawMode" "wire" "10:06:10.0"
```

Notice how the data is specified at a start and end time for draw mode. This should be done in order to preserve the correct draw mode if the data is to be played backwards or is to loop back on itself.

A "ClearGenData" value will disable the setting allowing the user to regain control over the platform draw mode.

Example: `GenericData 620 "SIMDIS_DrawMode" "ClearGenData" "10:06:20.0"`

A.9.3.6 Drop Track

SIMDIS_DropTrack is used to display the tracking status for a platform.

Tag: SIMDIS_DropTrack

Data: Drop

Drop If set to 1 (dropped), the platform will be drawn in point mode. If set to 0, the platform is drawn in solid mode.

Examples:
```
GenericData 602 "SIMDIS_DropTrack" "1" "10:01:05.0"
GenericData 602 "SIMDIS_DropTrack" "0" "10:03:09.9"
```

A "ClearGenData" value will disable the setting allowing the user to regain control over the platform draw mode (see Section A.9.3.5).

Example: `GenericData 602 "SIMDIS_DropTrack" "ClearGenData" "10:03:20.0"`

A.9.3.7 Dynamic Scale

SIMDIS_DynamicScale toggles the dynamic scaling preference for a platform. Example usage can be found in $(SIMDIS_DIR)/demos/SIMDIS/Users/SADM/simdis_pres_20008_0001.asi and

`$(SIMDIS_DIR)/demos/SIMDIS/Users/SADM/simdis_pres_60001_0001.asi`.

Tag: `SIMDIS_DynamicScale`

Data: `OnOff`

`OnOff` If set to 1, dynamic scaling is turned on. If set to 0, dynamic scaling is turned off.

Examples:
```
GenericData 602 "SIMDIS_DynamicScale" "0" "10:01:05.0"
GenericData 602 "SIMDIS_DynamicScale" "0" "10:03:09.9"
GenericData 602 "SIMDIS_DynamicScale" "1" "10:03:10.0"
GenericData 602 "SIMDIS_DynamicScale" "1" "10:06:09.9"
GenericData 602 "SIMDIS_DynamicScale" "0" "10:06:10.0"
```

Notice how the data is specified at a start and end time for dynamic scale state. This should be done in order to preserve the correct dynamic scale state if the data is to be played backwards or is to loop back on itself.

A `"ClearGenData"` value will disable the setting allowing the user to regain control over the platform dynamic scale toggle.

Example: `GenericData 602 "SIMDIS_DynamicScale" "ClearGenData" "10:06:20.0"`

A.9.3.8 Font Color

`SIMDIS_FontColor` changes a platform's label font color. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi`.

Tag: `SIMDIS_FontColor`

Data: `Color`

`Color` Color (in 0xAABBGGRR format) to set as the platform's label font color. Use `-1` to restore the font color to its original color.

Examples:
```
GenericData 602 "SIMDIS_FontColor" "-1" "10:01:05.0"
GenericData 602 "SIMDIS_FontColor" "-1" "10:03:09.9"
GenericData 602 "SIMDIS_FontColor" "0xFF00FF00" "10:03:10.0"
GenericData 602 "SIMDIS_FontColor" "0xFF00FF00" "10:06:09.9"
GenericData 602 "SIMDIS_FontColor" "-1" "10:06:10.0"
```

Notice how the data is specified at a start and end time for font color. This should be done in order to preserve the correct font color if the data is to be played backwards or is to loop back on itself.

A `"ClearGenData"` value will disable the setting allowing the user to regain control over the platform label font color.

Example: `GenericData 602 "SIMDIS_FontColor" "ClearGenData" "10:06:20.0"`

A.9.3.9 Override Color

`SIMDIS_OverrideColor` changes a platform's override color. The override color only applies to 3D models. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi` and

$(SIMDIS_DIR)/demos/SIMDIS/import/errorEllipse.asi.

Tag: SIMDIS_OverrideColor

Data: Color

Color Color (in 0xAABBGGRR format) to set as the platform's override color. Use -1 to restore the override color to its original color.

Examples:

```
GenericData 602 "SIMDIS_OverrideColor" "-1" "10:01:05.0"
GenericData 602 "SIMDIS_OverrideColor" "-1" "10:03:09.9"
GenericData 602 "SIMDIS_OverrideColor" "0xFF0000FF" "10:03:10.0"
GenericData 602 "SIMDIS_OverrideColor" "0xFF0000FF" "10:06:09.9"
GenericData 602 "SIMDIS_OverrideColor" "-1" "10:06:10.0"
```

Notice how the data is specified at a start and end time for override color. This should be done in order to preserve the correct override color if the data is to be played backwards or is to loop back on itself.

A "ClearGenData" value will disable the setting allowing the user to regain control over the platform override color.

Example: `GenericData 602 "SIMDIS_OverrideColor" "ClearGenData" "10:06:20.0"`

A.9.3.10 Polygon Mode

SIMDIS_PolygonMode sets the polygon mode for the specified Face for a platform. The values available to the Face and Mode parameters are string representations of the OpenGL arguments passed to the glPolygonMode command. Data parameters are delimited by a backtick (`). Example usage can be found in $(SIMDIS_DIR)/demos/SIMDIS/import/errorEllipse.asi.

Tag: SIMDIS_PolygonMode

Data: Face`Mode

Face Face for which to set the polygon mode. Options are GL_FRONT, GL_BACK, and GL_FRONT_AND_BACK.

Mode Polygon mode to set in the specified Face. Options are GL_POINT, GL_LINE, and GL_FILL.

Example: `GenericData 602 "SIMDIS_PolygonMode" "GL_BACK`GL_LINE" "70"`

"ClearGenData" is not supported by SIMDIS_PolygonMode and will be ignored if specified.

A.9.3.11 Polygon Stipple

SIMDIS_PolygonStipple sets whether the specified Face for a platform should be a candidate for culling when rendered. The values available to the Stipple parameter are string representations of the OpenGL arguments passed to the glPolygonStipple command. Data parameters are delimited by a backtick (`). Example usage can be found in $(SIMDIS_DIR)/demos/SIMDIS/import/errorEllipse.asi.

Tag: SIMDIS_PolygonStipple

Data: OnOff`Stipple

`OnOff`	If set to 1, the `Stipple` is displayed. If set to 0, the `Stipple` is not displayed.
`Stipple`	Stipple pattern to use. SIMDIS defines 9 stipple patterns, valid input is [1,9].

Example: `GenericData 602 "SIMDIS_PolygonStipple" "1`1" "55"`

`"ClearGenData"` is not supported by `SIMDIS_PolygonStipple` and will be ignored if specified.

A.9.3.12 Rocket Burn

`SIMDIS_RocketBurn` displays a cylinder near a platform. A platform can have multiple concurrent instances of `SIMDIS_RocketBurn`. `Data` parameters are prefixed with a variable name and an equals sign. SIMDIS interpolates the cylinder values when the SIMDIS time is in between two cylinder Generic Data points. `Data` parameters are delimited by a comma. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/Examples/rocketBurn/variousMissileRocketBurns.asi`.

Tag: `SIMDIS_RocketBurn ID`

`ID`	ID of this SIMDIS_RocketBurn. Each new ID is considered a unique SIMDIS_RocketBurn within the scope of the platform.

Data: `st=StartTime,c=Color,p=Pos,d=Dir,rn=RadNear,rf=RadFar,sc=Case`

`StartTime`	Start time (in seconds) of the rocket burn after the Generic Data start time.
`Color`	Color (in 0xAABBGGRR format) of the rocket burn.
`Pos`	Backtick (` ) delimited values (in meters) describing the rocket burn origin position with respect to the platform body (X`Y`Z).
`Dir`	Backtick (` ) delimited values (in meters) describing the rocket burn direction (X`Y`Z).
`RadNear`	Near radius (in meters) of the rocket burn.
`RadFar`	Far radius (in meters) of the rocket burn.
`Case`	Special case marker (see below).

Example:

```
GenericData 602 "SIMDIS_RocketBurn 1" "st=9,c=0xFF042EF9,p=-.28`.38`0,d=0`.6`0,rn=.12,
 rf=.01,sc=0" "1"
```

Due to the 72 character DCS Generic Data limit and to the desire to make a common case, below, easier to insert, a special case (`Case`) was added to the RocketBurn Generic Data parsing. For the non-special case (`sc=0`), SIMDIS interpolates between rocket burn states. Also, SIMDIS will use the values from a previously specified state if those values are not specified in a later state. For the special case, only one rocket burn state is specified. However, internally four states are added. They are:

- Two "zero size" states, one at Generic Data start time, and the other at "Generic Data start time + `StartTime` + .000000001".
- Two "specified state" states, one at "Generic Data start time + .000000001", and the other at "Generic Data start time + `StartTime`".

Therefore, if `sc=1`, then the meaning of the `StartTime` keyword changes from: "start time measured from the generic data start time" to the "duration that the rocket burn is on".

A "ClearGenData" value will remove the rocket burn from the display.

Example: `GenericData 602 "SIMDIS_RocketBurn 1" "ClearGenData" "15"`

A.9.3.13 Scale Level

`SIMDIS_ScaleLevel` changes a platform's dynamic scale level. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi`.

Tag: `SIMDIS_ScaleLevel`

Data: `Level`

`Level` Scale level (as a character string) for the platform.

Examples:
```
GenericData 602 "SIMDIS_ScaleLevel" "1" "10:01:05.0"
GenericData 602 "SIMDIS_ScaleLevel" "1" "10:03:09.9"
GenericData 602 "SIMDIS_ScaleLevel" "3" "10:03:10.0"
GenericData 602 "SIMDIS_ScaleLevel" "3" "10:06:09.9"
GenericData 602 "SIMDIS_ScaleLevel" "5" "10:06:10.0"
```

Notice how the data is specified at a start and end time for scale level. This should be done in order to preserve the correct scale level if the data is to be played backwards or is to loop back on itself.

A "ClearGenData" value will disable the setting allowing the user to regain control over the platform scale level.

Example: `GenericData 602 "SIMDIS_ScaleLevel" "ClearGenData" "10:06:20.0"`

A.9.3.14 Scale XYZ

`SIMDIS_ScaleXYZ` scales the platform in the X, Y, and Z planes. `Data` parameters are delimited by a back-tick (`). Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/errorEllipse.asi` and `$(SIMDIS_DIR)/demos/SIMDIS/Examples/scaleXYZandAlphaVolume/scaleXYZandAlphaVolume.asi`.

Tag: `SIMDIS_ScaleXYZ`

Data: ``X`Y`Z``

`X` X scale factor for the platform.
`Y` Y scale factor for the platform.
`Z` Z scale factor for the platform.

Example: ``GenericData 602 "SIMDIS_ScaleXYZ" "1000.0`1000.0`1000.0" "10:01:05.0"``

A "ClearGenData" value will disable the setting allowing the user to regain control over the platform scale XYZ.

Example: `GenericData 602 "SIMDIS_ScaleXYZ" "ClearGenData" "10:01:20.0"`

A.9.3.15 Track Color

SIMDIS_TrackColor changes a platform's track history color. Example usage can be found in
$(SIMDIS_DIR)/demos/SIMDIS/Users/SADM/simdis_pres_50001_0001.asi and
$(SIMDIS_DIR)/demos/SIMDIS/Users/SADM/simdis_pres_60001_0001.asi.

Tag: SIMDIS_TrackColor

Data: Color

Color — Color (in 0xAABBGGRR format) to set as the platform's track color. Use -1 to restore the track color to its original color.

Examples:

```
GenericData 602 "SIMDIS_TrackColor" "-1" "10:01:05.0"
GenericData 602 "SIMDIS_TrackColor" "-1" "10:03:09.9"
GenericData 602 "SIMDIS_TrackColor" "0xFF0000FF" "10:03:10.0"
GenericData 602 "SIMDIS_TrackColor" "0xFF0000FF" "10:06:09.9"
GenericData 602 "SIMDIS_TrackColor" "-1" "10:06:10.0"
```

Notice how the data is specified at a start and end time for track color. This should be done in order to preserve the correct track color if the data is to be played backwards or is to loop back on itself.

"ClearGenData" is not supported by SIMDIS_TrackColor and will be ignored if specified.

A.9.3.16 Vapor Trail

SIMDIS_VaporTrail displays a vapor trail near a platform. The vapor trail is drawn as a series of "vapor items" along the platform's track history. A platform can have multiple concurrent instances of SIMDIS_VaporTrail. Data parameters are prefixed with a variable name and an equals sign. Data parameters are delimited by a comma. Example usage can be found in
$(SIMDIS_DIR)/demos/SIMDIS/import/sampleGenericData2.asi and
$(SIMDIS_DIR)/demos/SIMDIS/Examples/vaporTrail/vaporTrail.asi.

Tag: SIMDIS_VaporTrail ID

ID — ID of this SIMDIS_VaporTrail. Each new ID is considered a unique SIMDIS_VaporTrail within the scope of the platform.

Data: initialRadius=InitRad,fadeTime=FadeTime,radiusExpansionRate=ExpRate,
numRadiFromPreviousSmoke=NumRadi,metersBehindCurrentPosition=BehindCurPos,
vaporTextureNames=TexNames

InitRad	Initial radius (in meters) of a vapor item.
FadeTime	Fade time (in seconds) of a vapor item.
ExpRate	Expansion rate (in meters per second) of a vapor item.
NumRadi	Number of InitRad spaces of distance between each vapor item.
BehindCurPos	Distance (in meters) behind the host platform to start the vapor trail.
TexNames	Colon delimited list of vapor texture filenames to use. The textures will be used in the order they are specified, one per vapor item, looping back to the start of the list once the end is reached.

Example:

```
GenericData 602 "SIMDIS_VaporTrail 1" "initialRadius=1,fadeTime=30,
 radiusExpansionRate=1,numRadiFromPreviousSmoke=2.5,metersBehindCurrentPosition=6,
 vaporTextureNames=vapor3.rgba:vapor4.rgba:vapor5.rgba:vapor6.rgba" "10:01:05.0"
```

"ClearGenData" is not supported by SIMDIS_VaporTrail and will be ignored if specified.

A.9.4 Beam Generic Data

A.9.4.1 HBW

`SIMDIS_HBW` changes a beam's horizontal beam width (HBW). Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/exampleBeamGate.asi`.

Tag: `SIMDIS_HBW`

Data: `Width`

`Width` Width (in degrees) of the beam's HBW. Valid range is `(0, 360]`.

Example: `GenericData 32 "SIMDIS_HBW" "5" "10:01:05.0"`

`"ClearGenData"` is not supported by `SIMDIS_HBW` and will be ignored if specified.

A.9.4.2 VBW

`SIMDIS_VBW` changes a beam's vertical beam width (VBW). Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/import/exampleBeamGate.asi`.

Tag: `SIMDIS_VBW`

Data: `Width`

`Width` Width (in degrees) of the beam's VBW. Valid range is `(0, 180]`.

Example: `GenericData 32 "SIMDIS_VBW" "5" "10:01:05.0"`

`"ClearGenData"` is not supported by `SIMDIS_VBW` and will be ignored if specified.

A.9.5 LOB Generic Data

A.9.5.1 Flash

`SIMDIS_Flash` toggles a LOB's flashing state. Example usage can be found in `$(SIMDIS_DIR)/demos/SIMDIS/Examples/LOB/LOBDemo.asi`.

Tag: `SIMDIS_Flash`

Data: `OnOff`

`OnOff` Use `On` to make the LOB start flashing. Use `Off` to make the LOB stop flashing.

Examples:

```
GenericData 16 "SIMDIS_Flash" "On" "14:00:10.0"
GenericData 16 "SIMDIS_Flash" "Off" "14:00:20.0"
```

`"ClearGenData"` is not supported by `SIMDIS_Flash` and will be ignored if specified.

A.10 Category Data

Category Data is considered any non-positional timestamped data that is used for filtering and display preferences, and can be associated with all entity types. Category Data is a critical part of entity matching used in Preference Rules (see Section 5.2.8), an automated method of entity preference application. Table A.14 contains common category names and values, but any category name and value can be used when defining Category Data. Category Data is intended for categories where each category has only a relatively small number of possible category values, such as the category Affinity with possible values of Friendly, Hostile, or Neutral. A category that can be any real number, such as Voltage, would not be a good use for Category Data. SIMDIS performance might decrease significantly if such values are stored as Category Data. To filter such a field, it is recommended to group values into bands, e.g. Low Voltage (0 - 100 V).

There is one keyword that is [**REQUIRED**] when defining Category Data.

```
CategoryData ID Time Name Value
```

`ID`	Unique ID of the entity to associate with this Category Data.
`Time`	Time at which this data is valid.
`Name`	Name of the category.
`Value`	Value within the category. `Value` may not be "Unlisted Value" or "No Value"; these are reserved values.

Specify `Time` as `-1` to specify a Category Data as the initial data point for the given `Name` and `Value`. If `-1` is used for `Time`, the Category Data will not be limited in live mode (see Section 3.11.4).

For example:

```
CategoryData 601 0 "Platform Type" "Aircraft"
CategoryData 601 4 "Platform Type" "Helicopter"
CategoryData 601 0 "Affinity" "Unknown"
CategoryData 601 4 "Affinity" "Friendly"
CategoryData 601 -1 "Data Source" "LATR"
```

NOTE: The use of special characters in the category name or value is not allowed. The following characters may not be used in the category name or value: **#, `,´,~**.

Category Name	Category Values
Data Format	"DIS" "GPX" "GCCS" "HGHS" "iNET" "JREAP" "LDP" "OTG" "PET" "PLM" "PMRF CSV" "SCORE CSV" "TLE" "UNKNOWN CSV" "WAM"
Data Source	"AMIE" "Format X" "LINK16" "MSN50" "Tewm"
Affinity	"Friendly" "Hostile" "Neutral"
Platform Type	"Unknown" "Surface Ship" "Submarine" "Aircraft" "Satellite" "Helicopter" "Missile" "Decoy" "Buoy" "Contact" "Reference Site" "Land Vehicle" "Land Site" "Torpedo"
Alert	"None"
Live Logging	"Marker" "Active" "Ignore" "Pending"

Table A.14: Common Category Names and Values

A.11 Data Tables

Two keywords are [**REQUIRED**] to be specified when defining a data table.

`DataTable OwnerID TableID Name ColumnDefs`

`OwnerID`	Unique ID of the entity (0 for the scenario) that owns the table.
`TableID`	Unique ID of the table. This must be a non-zero integer. This ID must be unique across *all* tables, not just tables associated with the owning entity or scenario.
`Name`	Unique name for the table. This name must be unique across all tables associated with the owning entity or scenario. If the name contains a space character, the name must be surrounded by double quotes.
`ColumnDefs`	One or more column name and storage type pairs defining the columns in the table. Each column name must be unique within this table. If a column name contains a space character, the column name must be surrounded by double quotes. The storage type defines how values are to be stored in memory. See Table A.15 for available storage types.

`DataTableRow TableID TimeStamp Values`

`TableID`	Unique ID of the table to which the row is being added.
`TimeStamp`	Time stamp for the row, stating when the data in the row is valid.
`Values`	One or more value to place in the row. If a column does not have a value for this `TimeStamp`, use NULL as the value. If a value contains a space character, the value must be surrounded by double quotes.

For example:

```
DataTable 11 1 "Engine1" "Voltage1" FLOAT64 "Fuel" FLOAT64 "Revolutions" FLOAT64
DataTableRow 1 0 5.0 90.0
DataTableRow 1 1 2.0 22.0
DataTableRow 1 2 3.0 NULL
```

Value	**Description**
`UINT8`	8 bit unsigned integer [0, 255]
`INT8`	8 bit signed integer [-128, 127]
`UINT16`	16 bit unsigned integer [0, 65535]
`INT16`	16 bit signed integer [-32768, 32767]
`UINT32`	32 bit unsigned integer [0, 4294967295]
`INT32`	32 bit signed integer [-2147483648, 2147483647]
`UINT64`	64 bit unsigned integer [0, 18446744073709551616]
`INT64`	64 bit signed integer [-9223372036854775808, 9223372036854775807]
`FLOAT32`	32 bit floating point number
`FLOAT64`	64 bit floating point number
`STRING`	ASCII character string

Table A.15: Available storage types for data table columns

One [**OPTIONAL**] keyword can be used to further define a data table.

`DataTableUnits` `TableID ColumnUnits`

`TableID`	Unique ID of the table referenced by this keyword.
`ColumnUnits`	One or more units for a column, delimited by a space character. If a column units description contains a space character, the column units description must be surrounded by double quotes. See Table A.16 for available column units.

Example: **`DataTableUnits`** `1 "Volts" "Liters" "Revolutions Per Minute"`

Seconds	Minutes
Hours	Radians
Degrees	Binary Angle Measurement
Meters	Kilometers
Yards	Miles
Feet	Inches
Nautical Miles	Centimeters
Millimeters	Kiloyards
Data Miles	Fathoms
Kilofeet	Meters per Second
Kilometers per Hour	Knots
Miles per Hour	Feet per Second
Mach	Kilometers per Second
Data Miles per Hour	Yards per Second
Meters per Second Squared	Kilometers per Second Squared
Yards per Second Squared	Miles per Second Squared
Feet per Second Squared	Inches per Second Squared
Knots per Second	Celsius
Fahrenheit	Kelvin
Rankine	Reaumur
Revolutions Per Minute	Radians Per Second
Degrees Per Second	Liters
Milliliters	Fluid Ounces
Cups	Pints
Quarts	Gallons
Teaspoons	Tablespoons
Millibar	Bars
Pounds Per Square Inch	Atmospheres
Torr	Pascals
Kilopascals	Megapascals
Volts	Millivolts
Microvolts	Kilovolts
Megavolts	Gigavolts
Unknown	

Table A.16: Available units for data table columns

A.12 Miscellaneous Formats and Data

The following sections are the "catchall" section for the ASI File Format. Here you will find information on various formats that the ASI importer supports.

A.12.1 Time Formats

SIMDIS assumes all times are referenced to the beginning of a calendar year. Time can be specified in the following formats:

Time Format	Example
`SS.SSS`	`28314045.4`
`"MM:SS.SSS"`	`"471900:45.4"`
`"HH:MM:SS.SSS"`	`"7865:00:45.4"`
`"DDD YYYY HH:MM:SS.SSS"`	`"328 2004 17:00:45.4"`
`"Month Day YYYY HH:MM:SS.SSS"`	`"Nov 23 2004 17:00:45.4"`

If time is not represented as a single value, then encompassing quotes are required. Furthermore, if < `ReferenceYear` > is not set, the Reference Year defaults to 1970.

The time formats have changed as of ASI file format version 9. Ordinal-formatted time values (DDD, day of the year 001-366) are now saved with a four-digit year. In Month-formatted time values, the four-digit year is now located after "Month Day", rather than at the end of the string. Old ASI time formats are still supported, but will not be saved out by any SIMDIS application that generates ASI format files. This change was made to allow for more precise definition of time values outside the current reference year. However, time values must still be positive, and must still occur after midnight January 1 of the reference year. Finally, the notation for Month values use the three character abbreviations: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, and Dec.

NOTE: For static entities, only a single data value is needed. Replace the time value with a -1.0. For example:

`PlatformData` 1 -1.0 1115.0 342.0 0.0 3.344 0.0 0.0 10.0 "Platform Type"

A.12.2 Color Formats

All colors are specified as either 0xAABBGGRR, where:

Value	Meaning
AA	Alpha component, range 00 to FF (0-255) in hexadecimal
BB	Blue component, range 00 to FF (0-255) in hexadecimal
GG	Green component, range 00 to FF (0-255) in hexadecimal
RR	Red component, range 00 to FF (0-255) in hexadecimal

Or the following character strings can be used. The hexadecimal numbers are the exact color mappings to the given color strings.

Value	Hexadecimal Representation
blue	0x80FF0000
red	0x800000FF
green	0x8000FF00
white	0x80FFFFFF
yellow	0x8000FFFF
purple	0x80FF00FF
magenta	0x808500FF
cyan	0x80FFFF00
black	0x80000000
brown	0x80105050
orange	0x800080FF
gray	0x80808080

A.12.3 ASCII Antenna Pattern File Formats

A.12.3.1 Table Antenna Pattern (`.aptf`)

This antenna file contains a series of sub-tables of antenna patterns based on their respective symmetry. The angular data can be to any desired degree of resolution and can be irregularly spaced. The data within each sub-table should be in increasing angular order, e.g., 0 to 180 degrees.
For this particular antenna pattern, the following four sub-tables define the pattern:

1: `[-180, 0] azimuth (H-plane)`
2: `[0, 180]  azimuth (H-plane)`
3: `[-90, 0]  elevation (E-plane)`
4: `[0, 90]   elevation (E-plane)`

The antenna symmetry value indicates the number of tables the user is going to provide. If the symmetry is 1, then the user will provide the [0, 180] azimuth table. The program will reuse this table for the other three tables. If the symmetry is 2, then the user will provide the [0, 180] azimuth table and the [0, 90] elevation table. The program will reuse these tables for the missing azimuth, and elevation tables. If the symmetry is 4, the user will provide all four azimuth and elevation tables.

For symmetric antenna patterns, Symmetry = 2, tables should cover the sectors [0, 180] for azimuth patterns and [0, 90] for elevation patterns. Internally, mirroring the data about the zero axes creates a 360-degree azimuth and a 180-degree elevation pattern. Sub-tables should have the following order:

1: `[0, 180]   azimuth (H-plane)`
2: `[0, 90]   elevation (E-plane)`

For asymmetric antenna patterns, Symmetry = 4, all four sub-tables are required. These four tables must be input sequentially and cover the sectors in the following order:

1: `[-180, 0]  azimuth (H-plane)`
2: `[0, 180]   azimuth (H-plane)`
3: `[-90, 0]   elevation (E-plane)`
4: `[0, 90]    elevation (E-plane)`

The first field in the antenna pattern file, "Angle units", represents the units for all angular data. Available formats:

```
0: degrees
1: radians
```

The pattern data is comprised of two columns. The first contains all angular data. The second column contains the antenna pattern gain in dB.

Below is a sample antenna pattern file for a symmetric pattern, i.e. Symmetry = 2. The asymmetric is similar, except the first azimuth pattern range is [-180, 0], the second table is the azimuth pattern from [0,180]. Elevation tables are 3 and 4 respectively with ranges [-90, 0] and [0, 90].

The file is delimited on white space (tabs and spaces) and can be found on the next page.

```
//Sample Data For a Symmetric Pattern of < AntennaPattern >
   table
0 2                        (Angle units (0:deg , 1:rad), Symmetry)
181                        (Azimuth Pattern Table Size)
0.000    - 62.400          (angle (deg), gain (dB))
1.000    - 63.200
2.000    - 64.600
3.000    - 63.000
.
.
.
177.000  - 22.000
178.000          - 10.000
179.000          - 3.200
180.000           0.000
91                         (Elevation Pattern Table Size)
0.000    - 54.000          (angle (deg), gain (dB))
1.000    - 52.000
2.000    - 48.800
3.000    - 50.500
.
.
.
87.000   - 33.200
88.000   - 7.200
89.000   - 1.200
90.000   0.000
```

Within SIMDIS linear interpolation is used to determine the gain value based on an input azimuth and elevation value. The input values are then normalized with the vertical and horizontal beam widths in order to return a weighted average antenna gain value.

Examples of the Table Antenna Pattern files can be found in the SIMDIS Distribution:

```
$(SIMDIS_DIR)/demos/SIMDIS/import/ant_sidelobes.aptf
$(SIMDIS_DIR)/demos/SIMDIS/import/ant_table_symmetry_1.aptf
$(SIMDIS_DIR)/demos/SIMDIS/import/ant_table_symmetry_4.aptf
```

A.12.3.2 Relative Table Antenna Pattern (`.aprf`)

This antenna file contains a series of 2D tables of antenna patterns based on azimuth (horizontal plane) and elevation (vertical plane) data. The angular data can be to any desired degree of resolution and can be irregularly spaced. It is required that the angles be input in the order of -180 to 180 for azimuth and -90 to 90 for elevation, with the main beam value located at 0.0 degrees. Angular units are in degrees. The corresponding gain data is referenced to the main beam gain. In other words, the maximum gain value should be 0.0, all remaining gain values will be relative to the main gain. Gain units are in dB.

The file is delimited on white space (tabs and spaces).

```
//   Sample Data For < AntennaPattern > anttable
181 91   // # of azimuth data points, # of elevation data
   points
// azimuth data:  angle (deg), gain (dB)
-180.0          -30.0000000001
-178.0          -30.3103568862
-176.0          -30.4808976808
.
.
-4.0            -8.8405020452
-2.0            -2.37823713123
0.0             0.0                    // main beam value
2.0             -2.37823713123
4.0             -8.8405020452
.
.
176.0           -30.4808976808
178.0           -30.3103568862
180.0           -30.0000000001

// elevation data:  angle (deg), gain (dB)
-90.0           -30.0000000001
-88.0           -30.3103568862
-86.0           -30.4808976808
.
.
-4.0            -8.8405020452
-2.0            -2.37823713123
0.0             0.0            // main beam value
2.0             -2.37823713123
4.0             -8.8405020452
.
.
86.0            -30.4808976808
88.0            -30.3103568862
90.0            -30.0000000001
```

Within SIMDIS linear interpolation is used to determine the gain value based on an input azimuth and elevation value. The input values are then normalized with the vertical and horizontal beam widths in order to return a weighted average antenna gain value.
An example of the Relative Table Antenna Pattern file can be found in the SIMDIS Distribution:

`$(SIMDIS_DIR)/demos/SIMDIS/import/umts.aprf`

A.12.3.3 Monopulse Antenna Pattern (.apmf)

This antenna file contains a series of monopulse antenna patterns based on frequency, azimuth and elevation data. The patterns are subdivided into two channels, the sum channel and the diff (difference or delta) channel. Both channels have the same format.

The first parameter should be the keyword <sum> or <diff>. This is followed by the frequency information. Frequencies are given in Hz and are in the following format:

start end step

Next is the azimuth data. All angular units are in degrees. It has the same format:

start end step **NOTE:** -180 deg < Angle(x) < 180 deg

Next is the elevation data. All angular units are in degrees. It has the same format:

start end step **NOTE:** -90 deg < Angle(x) < 90 deg

All data is required to be evenly spaced.

Finally the remaining data consists of magnitude (dB) and phase (deg) pairs. Below is pseudo code that represents how the data is organized.

The file is delimited on white space (tabs and spaces).

```
// Sample data for < AntennaPattern > monopulse
sum // sum channel keyword (4 frequencies)
8800000000.0 9400000000.0 200000000.0 //freq bgn end step (Hz)
-36.0   36.0    0.5     // azim info:  bgn end step (deg) (144)
-36.0   36.0    0.5     // elev info:  bgn end step (deg) (144)
-27.520000      85.700000       // magnitude (dB) & phase (deg)
    pairs
-28.210000      86.400000
-29.280000      87.000000

diff    // difference channel keyword  (4 frequencies)
8800000000.0 9400000000.0 200000000.0 // freq bgn end step (Hz)
-36.0   36.0    0.5             // azim info:  bgn end step (
   deg) (144)
-36.0   36.0    0.5             // elev info:  bgn end step (
   deg) (144)
-41.670000      125.700000      // magnitude (dB) & phase (deg)
    pairs
-40.290000 120.200000
-39.130000 127.600000
.
.
.

// pseudo code for organization of magnitude and phase pairs
for (number of frequencies)  // above case would be 4
{
```

```
for (number of azimuths) // above case would be 144
{
for (number of elevations) // above case would be 144
{
magnitude(dB)   phase(deg)
}
}
}
```

In the above case, there are (4 x 144 x 144) 82944 magnitude and phase pairs for each channel.

Within SIMDIS bilinear interpolation is used to determine the gain value based on an input azimuth and elevation value. Unlike the table antenna pattern file format, the input values are not normalized with the vertical and horizontal beam widths and the resulting gain is not weighted.

An example of the Monopulse Antenna Pattern file can be found in the SIMDIS Distribution:

`$(SIMDIS_DIR)/demos/SIMDIS/import/monopulse.apmf`

A.12.3.4 Bilinear Antenna Pattern (`.apbf`)

This antenna file contains a series of bilinear antenna patterns based on frequency, azimuth and elevation data.

The first parameter should be the keyword <bilinear>. This is followed by the frequency information. Frequencies are given in Hz and are in the following format:

start end step

Next is the azimuth data. All angular units are in degrees. It has the same format:

start end step **NOTE:** -180 deg < Angle(x) < 180 deg

Next is the elevation data. All angular units are in degrees. It has the same format:

start end step **NOTE:** -90 deg < Angle(x) < 90 deg

All data is required to be evenly spaced.

Finally the remaining data consists of magnitude (dB) and phase (deg) pairs. Below is pseudo code that represents how the data is organized.

The file is delimited on white space (tabs and spaces).

```
// Sample data for < AntennaPattern > bilinear
bilinear // (4 frequencies)
8800000000.0 9400000000.0 200000000.0 //freq bgn end step (Hz)
-36.0    36.0    0.5     // azim info:  bgn end step (deg) (144)
-36.0    36.0    0.5     // elev info:  bgn end step (deg) (144)
-27.520000               // magnitude (dB)
-28.210000
-29.280000
```

```
// pseudo code for organization of magnitude data
for (number of frequencies) // above case would be 4
{
for (number of azimuths) // above case would be 144
{
for (number of elevations) // above case would be 144
{
Gain (dB)
}
}
}
```

In the above case, there are (4 x 144 x 144) 82944 gain values.

Within SIMDIS bilinear interpolation is used to determine the gain value based on an input azimuth and elevation value. Unlike the table antenna pattern file format, the input values are not normalized with the vertical and horizontal beam widths and the resulting gain is not weighted.

An example of the Bilinear Antenna Pattern file can be found in the SIMDIS Distribution:

`$(SIMDIS_DIR)/demos/SIMDIS/import/bilinear.apbf`

A.12.3.5 National Spectrum Manager's Association (NSMA) Antenna Pattern (`.nsm`)

This antenna pattern file contains an interpretation of the NSMA standard format for the electronic transfer of antenna patterns.

File Naming Convention:

The filename should be a case insensitive, unique, six digit ID code assigned by the manufacturer with the file extension of `".nsm"`. (e.g. 24032g.nsm) 1st digit: [0-9, 1 digit Frequency Code Number] + 2nd-5th digits: [0000-ZZZ Z, 4 digit alpha-numeric unique Manufacturer ID Number assigned by manufacturer] +6th digit: [A|C|G|M; 1 digit Manufacturer Code which is registered with the NSMA] + Extension: [`.nsm`, a standard file extension for this version (v1) of the NSMA standard]

File Structure:

```
[Antenna Manufacturer] + CRLF
[Antenna Model number] + CRLF
[Comment] + CRLF
[FCC ID number] + CRLF
[reverse pattern ID number] + CRLF
[date of data] + CRLF
[Manufacturer ID Number (see file naming convention)] + CRLF
[frequency range] + CRLF
[mid-band gain] + CRLF
[Half-power beam width] + CRLF
[polarization (char 7) + chr$(32) + data count (char 7) + chr$(32) + CRLF]
```

```
[angle(1) (char 7) + chr$(32) + relative gain in dB(char 7) + chr$(32) + CRLF]
...
[angle(data count) (char 7) + chr$(32) + relative gain in dB (char 7) + chr$(32) +
CRLF]
...
[polarization (char 7) + chr$(32) + data count (char 7) + chr$(32) + CRLF]
[angle(1) (char 7) + chr$(32) + relative gain in dB(char 7) + chr$(32) + CRLF]
...
[angle(data count) (char 7) + chr$(32) + relative gain in dB (char 7) + chr$(32) + CRLF]
```

The first ten lines must exist and end with both a carriage return and line feed.

Dates should be formatted (12/31/[19]96 or 31 DEC [19]96 or [19]96.12.31).

Mid-band Gain should be in dBi (relative to and isotropic radiator).

Frequency Range must be in Megahertz. (6525.0 - 6875.0)

Place "NONE" on any rows which otherwise would be blank.

All values should be left justified in their character fields.

Polarization must be in the set [HH|HV|VV|VH|ELHH|ELHV|ELVV|ELVH]

-180 deg < Angle(x) < 180 deg for [HH|HV|VV|VH]

-90 deg < Angle(x) < 90 deg for [ELHH|ELHV|ELVV|ELVH]

Angle (1) < Angle (2) < ... < Angle (data count)

Relative Gain in dB < ~0 including sign (Most values will be non-positive. Exceptions may include antennas which have slightly depressed main lobes at 0 degrees. E.g. some panel antennas) (relative to the mid-band gain given on line 9 above)

Details from NSMA WG-16 publication:

The following are detailed explanations of each of the data lines.

`(Antenna Manufacturer) (30 data)`

This is the name under which the data was filed with the FCC. There will be no abbreviations.

`(Full model number) (30 data)`

This is the full model number as used when the data was filed with the FCC. Modifiers to the model number such as dashes or exceptions are to be included.

`(FCC ID number) (16 data)`

This is the ID number issued by the Common Carrier Branch of the FCC. For services which do not issue ID numbers, insert the word (none) in upper case.

`(Reverse pattern ID number) (16 data)`

This lists the reverse pattern FCC ID number. The reverse pattern is generally obtained by inserting the feed in a opposite manner in order to reverse the pattern.

`(date of data) (16 data)`

This date is the date referenced on the published pattern

`(manufacturer ID number) (4 data)`

This is the reference number assigned by the antenna manufacturer. This 4 digit alpha-numeric should be included with all antennas models and should also be used to name the antenna pattern filename.

`(frequency range) (16 data)`

This is to identify the full frequency range for which this pattern is valid and agrees with the range as specified in the printed pattern. The frequency is in Megahertz.

`(mid-band gain) (16 data)`

This is the gain of the antenna at mid-band. The gain is in gain above an isotropic radiator (dBi).

`(half power beam width) (16 data)`

This is the included angle centered on the main beam of the antenna and defines the angle where the antenna response falls -3 dB.

```
(polarization) (data count) (7 data, 1 space) (7 data, 1 space)
```

The data is preceded by an indication of the polarization of the data. The commonly accepted polarization designators for linear polarization are to be used:

HH	Horizontal polarized port response to a horizontally polarized signal in the horizontal direction.
HV	Horizontal polarized port response to a vertically polarized signal in the horizontal direction.
VV	Vertical polarized port response to a vertically polarized signal in the horizontal direction
VH	Vertical polarized port response to a horizontally polarized signal in the horizontal direction
ELHH	Horizontal polarized port response to a horizontally polarized signal in the vertical direction
ELHV	Horizontal polarized port response to a vertically polarized signal in the vertical direction
ELVV	Vertical polarized port response to a vertically polarized signal in the vertical direction
ELVH	Vertical polarized port response to a horizontally polarized signal in the vertical direction
HH	Horizontal polarized port response to a horizontally polarized signal in the horizontal direction.
HV	Horizontal polarized port response to a vertically polarized signal in the horizontal direction.
VV	Vertical polarized port response to a vertically polarized signal in the horizontal direction
VH	Vertical polarized port response to a horizontally polarized signal in the horizontal direction
ELHH	Horizontal polarized port response to a horizontally polarized signal in the vertical direction
ELHV	Horizontal polarized port response to a vertically polarized signal in the vertical direction
ELVV	Vertical polarized port response to a vertically polarized signal in the vertical direction
ELVH	Vertical polarized port response to a horizontally polarized signal in the vertical direction

The data count will be the number of data points to follow.

All eight responses should be included. If different polarizations have identical responses, they are to be duplicated in order that a full set of data be listed.

```
(angle) (response) (7 data, 1 space) (7 data, 1 space)
```

A full complement of data will show the antenna response in the horizontal direction for a 'horizontal cut' and in the vertical direction for a 'vertical cut'.

The data is presented in two columns. The angle of observation is listed first followed by the antenna response.

For the horizontal direction, the angle of observation starts from -180 degrees (defined as the left side of the antenna) and decrease in angle to the main beam, 0 degrees, and then increase to +180 degrees. The full data will cover the 360 degrees of the antenna.

For the vertical direction, the angle of observation starts from -5 (-90) degrees (defined as the antenna response below the main beam) and decrease in angle to the main beam, 0 degrees, and then increase to +5 (+90) degrees. The full data will cover the 10 (180) degrees centered about the main beam.

The antenna response is listed as dB down from the main lobe response and is shown as negative.

As a minimum, the data points are the breakpoints. That is, those points which define a change in the slope of the data or an adequate number of points to define a non-linear line. It is acceptable to include periodical points (e.g. every 1 degree or more) between the breakpoints.

Within SIMDIS, linear interpolation is used to determine the gain value based on an input azimuth and elevation value. Unlike the table antenna pattern file format, the input values are not normalized with the vertical and horizontal beam widths and the resulting gain is not weighted. Furthermore, the mid band gain and half power beam widths found in the NSMA file are used in place of the values found in the ASI file for determining the gain response of the antenna. In other words, modifying the main lobe gain value in the Prefs Tool Beam Tab will have no effect on the gain value returned from this type of antenna pattern.

An example of the NSMA Antenna Pattern file can be found in the SIMDIS Distribution:

```
$(SIMDIS_DIR)/demos/SIMDIS/import/24032g.nsm
```

A.12.3.6 EZNEC Antenna Pattern (`.txt`)

This antenna pattern file contains the 3D far field antenna gain patterns as saved from the antenna analysis program EZNEC (http://eznec.com). SIMDIS only supports a subset of the possible output file configuration options, specifically:

Far field antenna patterns must be organized in azimuth slices.

SIMDIS will accept either the counter-clockwise (CCW) from X-axis or the compass bearing angle conventions.

SIMDIS will accept either the text based or comma delimited far field patterns, although the comma delimited patterns are preferred.

SIMDIS expects gain values referenced to dB.

SIMDIS only supports gain data saved in the V (vertical), H (horizontal) and Tot (total) polarization format.

Examples of both formats are presented below.

Text file version:

```
EZNEC Demo ver. 5.0

Cardioid                                8/31/2007
    8:15:04 PM
```

```
--------------- FAR FIELD PATTERN DATA ---------------

Frequency = 299.793 MHz

Reference = 0 dBi

Azimuth Pattern    Elevation angle = -90 deg.
Deg       V dB       H dB       Tot dB
0       -99.99     -99.99     -99.99
5       -99.99     -99.99     -99.99
...
```

Comma delimited version:

```
"EZNEC Demo ver. 5.0"
"Cardioid","08-31-2007","20:15:33"
"--------------- FAR FIELD PATTERN DATA ---------------"
"Frequency (MHz):",299.793
"Reference (dBi):",0
"Azimuth Pattern    Elevation angle (deg):",-90
"Deg","V dB","H dB","Tot dB"
0,-99.99,-99.99,-99.99
5,-99.99,-99.99,-99.99
...
```

Within SIMDIS linear interpolation is used to determine the gain value based on an input azimuth and elevation value. Unlike the table antenna pattern file format, the input values are not normalized with the vertical and horizontal beam widths and the resulting gain is not weighted.

Examples of the EZNEC Antenna Pattern files can be found in the SIMDIS Distribution:

```
$(SIMDIS_DIR)/demos/SIMDIS/import/cardioid_az.txt
$(SIMDIS_DIR)/demos/SIMDIS/import/cardioid_az_cd.txt
```

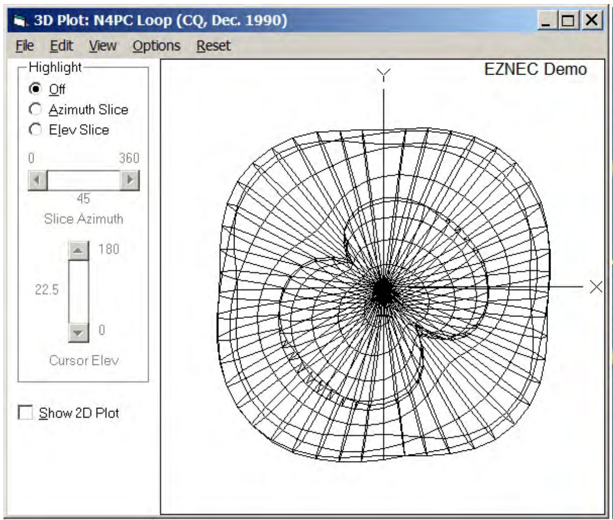
EZNEC angle conventions are defined as follows: Compass Bearing - Zero is in the direction of the +y axis (at the top of the 2D azimuth plot display). Angles increase as you go clockwise from zero.

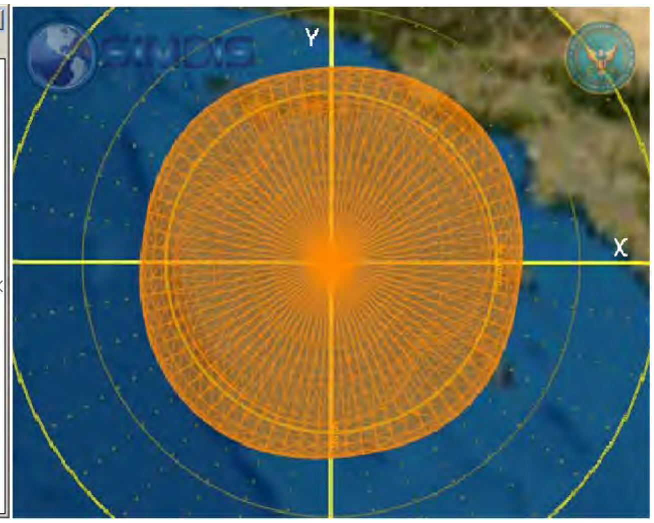
CCW From X Axis - Zero is in the direction of the +x axis (to the right of the 2D azimuth plot display). Angles increase as you go counter-clockwise from zero. This is the convention commonly used in mathematics and physics.

In SIMDIS the antenna patterns are oriented along the X-axis with a clockwise angle convention. The angle convention conversions from EZNEC to SIMDIS are:

Compass Bearing: Angle - 90, to orient antenna pattern along X axis.

CCW From X Axis: 360 - Angle to orient antenna pattern in clockwise direction.





A.12.3.7 XFDTD Antenna Pattern (.uan)

This antenna pattern file contains the 3D far zone antenna gain patterns. This file format is shared between Remcom's "XFDTD" and "Wireless InSite" software packages. The file format consists of two parts, a delimited parameters section, and a section containing all of the angle data.

The "parameters" section typically looks like the following:

```
begin_<parameters>
format free
phi_min 0
phi_max 360
phi_inc 5
theta_min 0
theta_max 180
theta_inc 5
complex
mag_phase
pattern gain
magnitude dB
maximum_gain 0
phase degrees
direction degrees
polarization theta_phi
end_<parameters>
```

The parameters are self-explanatory, for example, the value after "phi_min" will represent the minimum phi slice to save along with the file. Also, note that some of the parameters will directly affect the data values. For instance, if you change the "magnitude" parameter to "linear", the gain will be stored as a linear value instead of a dB gain. Similarly, the "phase" parameter can be changed to "radians". After the "parameters" section will follow a data section. This section is not delimited by anything other than the "end of file" marker. There will be up to 6 columns of data in this section, in the following order:

Theta-angle
Phi-angle
Theta-gain(db)
Phi-gain(db)
Theta-phase(degrees)
Phi-phase(degrees)

The data section has the following format:

```
0 0   4.300829   -311.0333   -44.63837   130.9943
0 5   4.234599   -38.08733   -44.63837   135.3616
```

Within SIMDIS linear interpolation is used to determine the gain value based on an input azimuth and elevation value. Unlike the table antenna pattern file format, the input values are not normalized with the vertical and horizontal beam widths and the resulting gain is not weighted.

An Example of the XFDTD Antenna Pattern file can be found in the SIMDIS Distribution:

```
$(SIMDIS_DIR)/demos/SIMDIS/import/dipoleXFDTD.uan
```

A.12.4 ASCII Radar Cross-Section File Formats

A.12.4.1 Lookup Table (LUT) Radar Cross-Section Format

This RCS file has the ability to contain multiple sub-tables of Radar cross sections associated to an azimuth value. The sub-tables are organized into hierarchical containers stored under specified polarity, frequency and elevation values. A given polarity can have one or more frequencies associated with it. A given frequency can have one or more elevation values, and an elevation can have one or more data pairings of azimuth and RCS values. The azimuthal data can be to any desired degree of resolution and can be irregularly spaced. If a requested polarity is not found in the file -300 dB is returned. Frequency selection is based on a nearest neighbor selection. Elevation and azimuth values are interpolated, if the data allows.

The Table Type field indicates how the RCS data is to be interpreted.

Table Type 0 applies a distribution function to the acquired RCS value. The acquired RCS value is determined via LUT method used for Table Type 1. The following distribution functions are supported:

Mean: Adds the scintillation modulation value to the LUT value.

Gaussian: A Gaussian distribution times the scintillation modulation value is added to the LUT value.

Rayleigh: The Root Sum Square (RSS) of two Gaussian distributions times the scintillation modulation value is added to the LUT value.

Log Normal: The log of the Rayleigh distribution.

Table Type 1 returns the RCS value as found in the table, based on given polarity, nearest frequency and interpolated elevation and azimuth values. The scintillation modulation value is not used.

Table Type 2 uses a symmetric RCS pattern (only 0-180° of azimuth data is required) about the azimuth axis. The acquired RCS value is determined via method used for Table Type 1.

Supported polarizations include 1: Horizontal, 2: Vertical, 3: Circular, 4: HV (Horizontal XMT, Vertical RCV), 5: VH (Vertical XMT, Horizontal RCV), 6: Left Circular, 7: Right Circular, 8: Linear.

An example LUT RCS file is shown below:

```
0                   RCS Lookup Table File Designation
RCS Test Case Data (RCS File Description)
1                   Table Type (0=distribution , 1=LUT, 2=symmetric)
0                  Func (0=Mean, 1=Gaussian , 2=Rayleigh , 3=Log Normal)
0.0                 Scintillation Modulation (dBsm)
2                   Number of Tables
8000.0              Frequency Table 1 (MHz)
0.0                 Elev (deg)
0                   Polarity (1=Horizontal , 2=Vertical , 3=Circular)
360                 Number of Aspect Data Elements
0         0         Angle Units (0=deg, 1=rad) RCS Units (0=sq m, 1=dBsm)
0.02      8.02
1.00      7.88
...
358.00    4.76
358.98    8.35
8000.0              Frequency Table 2 (MHz)
10.0                Elev (deg)
0                   Polarity (1=Horizontal , 2=Vertical , 3=Circular , etc)
360                 Number of Aspect Data Elements
0         0         Angle Units (0=deg, 1=rad) RCS Units (0=sq m, 1=dBsm)
0.02      8.02
1.00      7.88
...
358.00    4.76
358.98    8.35
```

An example of the Lookup Table RCS file can be found in the SIMDIS Distribution:

`$(SIMDIS_DIR)/demos/SIMDIS/import/AspectRCS.rcs`

A.12.4.2 Ship Air Defence Model (SADM) Radar Cross-Section Format Version 1

Version 1 of the SADM RCS data file specifies the total RCS of an entity as a function azimuth, elevation, frequency and polarization. The RCS is treated as a point source.

For this form of the RCS file the comments are followed by an &RCS keyword, and then keywords specifying the number of azimuth values in the file (`RCS_N_AZ`), the number of elevation values in the file (`RCS_N_EL`), the number of frequencies in the file (`RCS_N_FREQ`), and the elevations in degrees for which RCS data is supplied (`RCS_EL`). Elevation values must be listed in ascending order. Also, note that spaces are required on either side of the equal signs in these keywords.

These initial values are followed by sets of RCS tables for each frequency / polarization combination. There will be `RCS_N_FREQ` × `RCS_N_EL` total RCS tables. Each of these tables is preceded by `RCS_FREQ` and `RCS_POL` keywords to specify the frequency and polarization that the following table applies to.

There is also an `RCS_ON_AXIS` keyword to specify the RCS in dBsm of a ASM Seeker, if the seeker is pointing at a Radar. This value is used for ASMs, and it is ignored for ships, aircraft, and background targets. Since the effect of the RCS_ON_AXIS value is calculated within the SADM it is not available in SIMDIS and is ignored.

The RCS table itself begins with an "RCS_TABLE =" keyword, and from 1 to 360 azimuth / RCS value entries. The azimuth angles must be monotonically increasing, with one angle value per line. The angle values may occur at irregular intervals, but they must increase monotonically.

An example SADM RCS file is shown below:

NOTE: The header comments are required for proper file parsing. SIMDIS does not use the RCS_FREQ_INTERPOLATE value. SIMDIS always employs a nearest neighbor selection for frequency.

```
%**********************************************************
% This is a sample RCS file.  Comments prefaced by a "%"
% symbol can be included before the "RCS" keyword to
% document the contents of this file.  To be safe, DO
% NOT include the keyword "&+RCS" anywhere within these
% comments.
%
% The initial parameters that go in this file are:
%
% RCS_FREQ_INTERPOLATE = T/F
% RCS_N_AZ =
% RCS_N_EL =
% RCS_N_FREQ =
% RCS_EL =   el1   el2   ...
%
% They are followed by the following values, repeated for
% each frequency / polarization combination.
%
% RCS_FREQ = freq1
% RCS_POL = H
% RCS_ON_AXIS = on_axis_rcs @ freq1 & H pol
% RCS_TABLE =
%    ...
% RCS_FREQ = freq1
% RCS_POL = V
% RCS_ON_AXIS = on_axis_rcs @ freq1 & V pol
% RCS_TABLE =
%    ...
% RCS_FREQ = freq2
% RCS_POL = H
% RCS_ON_AXIS = on_axis_rcs @ freq2 & H pol
% RCS_TABLE =
%    ...
% RCS_FREQ = freq2
% RCS_POL = V
% RCS_ON_AXIS = on_axis_rcs @ freq2 & V pol
```

```
% RCS_TABLE =
%     ...
% etc
% /               (terminate with a slash)
%
&RCS
RCS_FREQ_INTERPOLATE = T
RCS_N_AZ = 73
RCS_N_EL = 3
RCS_N_FREQ = 3
RCS_EL = 0 30 60
RCS_FREQ = 9
RCS_POL = V
RCS_ON_AXIS = -100
RCS_TABLE =
0.0000 44.0000 42.1000 40.1000
5.0000 38.0000 35.8000 33.8000
.
.
.
355.0000 35.0000 32.8000 30.8000
360.0000 41.0000 39.1000 37.1000
```

NOTE: SIMDIS does not support SADM version 2 (high resolution I/Q) and version 3 (multi-scatter) RCS file formats.

A.12.4.3 XPatch (SAIC/Demaco) Radar Cross-Section Format

The XPatch RCS file contains multiple calculations of Radar cross sections in decibels (dB), with each RCS value based on a measured frequency, elevation angle, azimuth angle and four XMT/RCV (VV, HV, VH, HH) polarizations. Frequencies are typically listed in 10 MHz increments. Elevations are in 1° increments and can range from +90° to -90°. Azimuths are in 2° increments and can range from 0° to 358°.

An example XPatch RCS file is shown below:

```
#     Horizontal Incidence ---------------------------------|
#                                                           |
#                                                           |
#                                                           |
#     Vertical Incidence ---------------|                   |
#                                       |                   |
#                                       |                   |
#                              **************
   **************
f(GHz)   inc-EL    inc-AZ          VV       HV        VH       HH
8.500    90.000    -0.000          7.001    -75.792   -75.335   7.204
8.540    90.000    -0.000          6.757    -84.899   -82.041   7.003
```

8.580	90.000	−0.000	6.366	−75.364	−75.434	6.662
8.620	90.000	−0.000	6.526	−85.044	−80.745	6.821
8.660	90.000	−0.000	7.074	−76.336	−77.998	7.314
8.700	90.000	−0.000	6.921	−78.402	−76.152	7.146

An example of the XPatch RCS file can be found in the SIMDIS Distribution:

```
$(SIMDIS_DIR)/demos/SIMDIS/import/XPATCH.rcs
```

A.12.5 True Type Fonts

When specifying a font in the ASI file format, both the font filename and size are required. Most integer values are valid, below are some examples:

```
// Courier New Bold font, with a point size of 15
"courb.ttf 15"
```

```
// Times New Roman Italic font, with a point size of 32
"timesi.ttf 32"
```

The following True Type Fonts and Font Families are available in SIMDIS.

Andale Mono, Version 2.0	
andalemo.ttf	Andale Mono
Arial, Version 2.76	
arial.ttf	Arial
arialbd.ttf	Arial Bold
arialbi.ttf	Arial Bold Italic
ariali.ttf	Arial Italic
Arial Black, Version 2.35	
ariblk.ttf	Arial Black
Comic Sans MS, Version 2.10	
comic.ttf	Comic Sans MS
comicbd.ttf	Comic Sans MS Bold
Courier New, Version 2.76	
cour.ttf	Courier New
courbd.ttf	Courier New Bold
courbi.ttf	Courier New Bold Italic
couri.ttf	Courier New Italic
Georgia, Version 2.05	
georgia.ttf	Georgia
georgiab.ttf	Georgia Bold
georgiai.ttf	Georgia Italic
georgiaz.ttf	Georgia Bold Italic
Impact, Version 2.35	

continued on next page

impact.ttf	Impact
Times, Version 2.76	
times.ttf	Times
timesbd.ttf	Times Bold
timesbi.ttf	Times Bold Italic
timesi.ttf	Times Italic
Trebuchet MS, Version 1.15	
trebuc.ttf	Trebuchet MS
trebucbd.ttf	Trebuchet MS Bold
trebucbd.ttf	Trebuchet MS Bold Italic
trebucit.ttf	Trebuchet MS Italic
Verdana, Version 2.35	
verdana.ttf	Verdana
verdanab.ttf	Verdana Bold
verdanai.ttf	Verdana Italic
verdanaz.ttf	Verdana Bold Italic
Webdings	
webdings.ttf	Webdings

A.12.6 Custom Renderers

When creating a **Custom Rendering** object, you must list a **Custom Renderer** to use when drawing the rendering. Renderers are defined by SIMDIS extensions, and can be used in ASI files or Plug-in API contexts.

A Renderer defines the shape or shapes to display, as well as any additional information needed to display the shape. Most Renderers expect the Custom Rendering entity to have a data table with the same name as the Renderer. The number of columns will vary between Renderers, depending on the information requirements. The order in which columns are listed in the ASI file or API calls does not matter, but column names must match exactly. Most columns support units. Some columns are required, and will prevent the Custom Rendering from being displayed if they are missing or have no value for the given time.

At present there is only one SIMDIS extension which defines Custom Renderers: CustomUncertainties.

A.12.6.1 CustomUncertainties Extension

The CustomUncertainties extension allows for shapes associated with the positional uncertainties of various entities. A data table associated with the Custom Rendering controls the size and orientation of the shape. Some shapes are co-located with their host **Platform** while others have independent location, as defined in the data table.

The CustomUncertainties extension supports numerous preferences. Most shapes support a subset of the avaiable preferences. The preferences can be configured on the "Custom" tab in Prefs Tool when a Custom Rendering is selected. Avaiable preferences include:

- **Outline:** Draws only the shape's outline.
- **Center Axis:** Draws a plus sign at the center of the shape. Only works with outline shapes or semitransparent shapes. Only the current shape gets a center axis, the historical shapes do not get a center axis.
- **Time Persistence:** Sets how long to display the Custom Rendering from the point at or before the current scenario time.
- **Second History:** Sets length of history in seconds.
- **Points History:** Sets length of history in points.
- **History Use Override Color:** Toogles use of the history override color.
- **Override Color:** Sets the history override color. To reduce screen clutter use a history override color with a low alpha value.

Renderer Definitions

The following tables enumerate all renderers defined in the CustomUncertainties extension. It is important to note that the Renderer name does not necessarily match the Data Table name. In the following tables, a Renderer's name is presented as the title of its table.

Custom Ellipse

Description	Displays an ellipse.
Data Table Name	"Custom Ellipse"
Required Columns	Latitude, Longitude, Heading, X Size, Y Size
Optional Columns	Altitude, Pitch, Roll

Custom Ellipse Relative

Description	Displays an ellipse at a host platform's position.
Data Table Name	"Custom Ellipse Relative"
Required Columns	Heading, X Size, Y Size
Optional Columns	Pitch, Roll

Custom Ellipsoid

Description	Displays an ellipsoid.
Data Table Name	"Custom Ellipsoid"
Required Columns	Latitude, Longitude, Heading, X Size, Y Size
Optional Columns	Altitude, Pitch, Roll, Z Size

Custom Line

Description	Displays a line with optional uncertainty using the host platform's location as the starting point for the line.
Data Table Name	"Custom Line"
Required Columns	Azimuth, Elevation, Range
Optional Columns	Horizontal Uncertainty, Vertical Uncertainty

Custom Line Absolute

Description	Displays a line with optional uncertainty using a start point defined by the Latitude, Longitude, and (optionally) Altitude columns.
Data Table Name	"Custom Line"
Required Columns	Latitude, Longitude, Azimuth, Elevation, Range
Optional Columns	Altitude, Horizontal Uncertainty, Vertical Uncertainty

Custom Line Target

Description	Displays a line between a host platform and a target platform with optional uncertainty. Specifying a blank target name turns the line off. If the Original ID column is specified then the Name and Original ID must match the target platform.
Data Table Name	"Custom Line Target"
Required Columns	Name
Optional Columns	Original ID, Horizontal Uncertainty, Vertical Uncertainty

Custom Rectangle

Description	Displays a rectangle.
Data Table Name	"Custom Rectangle"
Required Columns	Latitude, Longitude, Heading, X Size, Y Size
Optional Columns	Altitude, Pitch, Roll

Various Shapes

Description	Displays either an ellipse or rectangle, as defined by the Shape column.
Data Table Name	"Various Shapes"
Required Columns	Shape, Latitude, Longitude, Heading, X Size, Y Size
Optional Columns	Altitude, Pitch, Roll

Column Definitions

The following table defines the data table columns used by various Renderers. If a column is optional, its default value will be used when it is not present in the data table.

Column Name	**Description**	**Units**	**Default Value**
Latitude	The latitude of the center of the shape	Angle	
Longitude	The longitude of the center of the shape	Angle	
X Size	The size of the shape in the X direction	Distance	
Y Size	The size of the shape in the Y direction	Distance	
Azimuth	The azimuth angle of the shape, with zero pointing North and increasing clockwise	Angle	
Elevation	The elevation angle of the shape, with zero pointing horizontally and positive values pitching upwards	Angle	
Heading	The heading of the shape, with zero pointing North and increasing clockwise	Angle	

continued on next page

Range	The length of the line	Distance	
Shape	A text string with the shape to display. Valid text strings are “Ellipse” and “Rectangle”.	N/A	
Name	A text string specifying the name of a platform.	N/A	
Original ID	An original ID of a target platform targeted by a line.	N/A	
Altitude	The altitude of the center of the shape	Distance	40 meters
Z Size	The size of the shape in the Z direction	Distance	1.0 meters
Pitch	The pitch of the shape, with zero pointing horizontally and positive values pitching upwards	Angle	0.0 degrees
Roll	The roll of the shape, with zero pointing up and positive values rolling clockwise	Angle	0.0 degrees
Horizontal Uncertainty	The horizontal uncertainty of the line	Angle	0.0 degrees
Vertical Uncertainty	The vertical uncertainty of the line	Angle	0.0 degrees

ASI Example

A working example ASI file can be found in the SIMDIS directory under *demos/SIMDIS/Examples/CustomRender.asi*. Below is a snippet of an ASI file that defines a custom ellipsoid object:

```
# Create the Custom Rendering, assume there is a platform with an ID of 1
CustomRendering 1 2 "Uncertainty" "Custom Ellipsoid"
# Map the Custom Rendering to the data source with OriginalID 0
OriginalID 2 0
# Create a Data Table (with ID 1) for our Custom Rendering (with ID 0) and list the
    columns that will be used paired with their data types
DataTable 2 1 "Custom Ellipsoid" "Latitude" FLOAT64 "Longitude" FLOAT64 "Altitude"
    FLOAT64 "Heading" FLOAT64 "Pitch" FLOAT64 "X Size" FLOAT64 "Y Size" FLOAT64
    "Z Size" FLOAT64
# Declare the units to be used for each column
DataTableUnits 1 "Degrees" "Degrees" "Meters" "Degrees" "Degrees" "Meters" "Meters"
    "Meters"
# Add the first row to our Data Table
DataTableRow 1 "0"   0.01 0.01 1000 0   45   1000 2000 30000
# Declare the color to use for our Custom Rendering
CustomRenderingColor 2 "0" Green
```

Plug-in API Example

Below is a snippet of C++ code that uses the SIMDIS Plug-in API to create a custom ellipsoid object:

```
// Assume 1 is the Unique ID for a platform
PIData::PICustomRenderingHeader header("Uncertainty", "Custom Ellipsoid");
uniqueId_ = PIData::createCustomRendering(1, &header);
StrVec columnNames;
UTILS::vVariableType_t columnTypes;
UTILS::vUnitType unitTypes;

columnNames.push_back("Latitude");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_DEGREES);

columnNames.push_back("Longitude");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_DEGREES);

columnNames.push_back("Altitude");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_METERS);

columnNames.push_back("Heading");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_DEGREES);

columnNames.push_back("Pitch");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_DEGREES);

columnNames.push_back("X Size");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_METERS);

columnNames.push_back("Y Size");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_METERS);

columnNames.push_back("Z Size");
columnTypes.push_back(UTILS::vtFLOAT64);
unitTypes.push_back(UTILS::CU_METERS);

// Create our Data Table using our column lists
PIData::addDataTable(uniqueId_, "Custom Ellipsoid", &columnNames, &columnTypes,
    &tableId_);
// Declare the units to use with each of our columns
PIData::setDataTableColumnUnits(tableId_, &unitTypes);
```

```
// Add the first row to our Data Table
DataTableRow row;
row.setTimeStamp(0.0);
row.setCell<double>(0, 0.01);
row.setCell<double>(1, 0.01);
row.setCell<double>(2, 1000.0);
row.setCell<double>(3, 0.0);
row.setCell<double>(4, 45.0);
row.setCell<double>(5, 1000.0);
row.setCell<double>(6, 2000.0);
row.setCell<double>(7, 30000.0);
PIData::addDataTableRow(tableId_, &row);

// Set the color (Green) for our Custom Rendering
PIData::setColor(uniqueId_, PIRGBA(0, 255, 0, 255), 0);
```

A.13 Version History

ASI Version 24

Added **OriginalID**, **GenericData**, **CategoryData**, and **DataTable** keywords support to **Projector** entities.

ASI Version 23

CustomRendering and **CustomRenderingColor** keywords were added. These keywords specify a new class of entity, Custom Renderings. Custom Renderings display user-defined areas, and can be hosted by a platform or the scenario itself.

ASI Version 22

LaserRefLLA keyword has been deprecated. The **LaserCoordSystem** keyword value **ECEF** has been marked as deprecated and is no longer supported in SIMDIS version 10.

ProjectorInterpolateFOV keyword is added. The keyword specifies if field of view of the projector specified by the unique ID should be interpolated between data points.

ASI Version 21

RollOffset and **GateRollOffset** keywords were added. These keywords are used for specifying a roll offset for both beam and gates that affect the azimuth and elevation planes.

ASI Version 20

LOB, **LOBID**, **LOBType**, **LOBDesc**, **LOBData** and **LOBDrawStyle** keywords were added. These keywords are used for specifying a new class of entities, Lines of Bearings (LOB). LOBs are collections of lines or detections associated with a platform that are drawn along the host platform's history trail.

ASI Version 19

`SIMDIS_HBW` and `SIMDIS_VBW` keywords were added. These are used for specifying two new supported types of SIMDIS specific generic data for controlling a sensor's horizontal and vertical beam widths.

Laser, **LaserCoordSystem, LaserAzElBody, LaserOri, LaserXYZOffset, LaserMaxRange, LaserWidth, LaserColor,** and **LaserOn** keywords were added. These are used to define a laser entity within SIMDIS.

ASI Version 18

The **DataTable** keyword was added. This keyword is used to specify time dependent data for a platform, beam or gate entity.

GateAzimOffset and **GateElevOffset** keywords were added. These keywords are used to offset the angular position of a gate.

The **BeamType** keyword was added. This keyword is used to specify the type of processing to apply to the BeamDateRAE keyword. As of version 18, both BeamType and GateType are now required arguments.

BeamDateRAE and **GateDataRAE** keywords were added. These keywords are used to specify the positions of beams and gates respectively. These commands also replace the following commands: **BeamData, BeamDataBody, BeamDataTarget, GateData, TargetGateData** and **BodyGateData**.

BeamColorCmd and **GateColorCmd** keywords were added. These keywords are used to specify color changes for beams and gates respectively. Note, color is no longer specified as part of the beam and gate data keywords. Color must be specified separately using these new keywords.

ASI Version 17

Projector, ProjectorRasterFile, ProjectorFOV, and **ProjectorOn** keywords were added. These are used to define a projector of still images or digital video.

EZNEC and **XFDTD** keywords were added. These are used for specifying two new supported types of antenna pattern file formats.

`SIMDIS_BearingLine` and `SIMDIS_RocketBurn` keywords were added. These are used for specifying two new supported types of SIMDIS specific generic data.

ASI Version 16

Added the Platform keyword **PlatformOriOffset** to specify a rotation offset relative to a platform's body coordinates.

Added the Gate keyword **BodyGateData** to allow azimuth and elevation offset values to be used with the body orientation of a host platform when drawing a gate.

ASI Version 15

The scenario keyword **RefLLA** was added. This is used to specify the reference latitude, longitude, and altitude. The following scenario keywords **RefLat**, **RefLon** and **RefAlt** were deprecated.

Added support for two new coordinate systems, Earth-Centered Inertial (ECI) and Generic Tangent Plane (GTP).

Added "TRUE" as an argument accepted by the **MagneticVariance** keyword.

Added the Scenario keyword **VerticalDatum** to specify the zero surface to which elevations or heights are referenced.

Added the Scenario keyword **TangentPlaneOffset** to specify the x and y translation positions and a rotation angle to offset the Generic Tangent Plane (GTP) coordinate system.

Added the Scenario keyword **ReferenceTimeECI** to specify the time at which the Earth-Centered Inertial reference frame is defined.

Replaced the Platform keyword **PlatformInterpolateOri** with **PlatformInterpolate** as interpolation is now controlled for all data, not just orientation. If the **PlatformInterpolateOri** keyword is encountered in older ASI files it will now behave as if it were the newer **PlatformInterpolate** keyword.

Replaced the Beam keyword **BeamInterpolatePos** with **BeamInterpolate** to be consistent with the Platform version. If the **BeamInterpolatePos** keyword is encountered in older ASI files, it will behave as if it were the **BeamInterpolate** version.

Replaced the Gate keyword **GateInterpolatePos** with **GateInterpolate** to be consistent with the Platform and Beam versions. If the **GateInterpolatePos** keyword is encountered in older ASI files, it will behave as if it were the **GateInterpolate** version.

Added the Platform keyword **PlatformUsesQuaternion** to indicate the use of a quaternion for specifying orientation instead of yaw, pitch and roll or Euler angles.

Added support for platform independent coordinate systems which allows more than one system to be used in an ASI file. As such the following Platform keywords were added: **PlatformRefLLA, PlatformCoordSystem, PlatformMagneticVariance, PlatformVerticalDatum, PlatformReferenceTimeECI,** and **PlatformTangentPlaneOffset**. These keywords mimic their Scenario equivalents with the exception that they are applied at the Platform level.

Added the Beam keyword **AntennaPolarity** to specify a beam's polarization.

Added support for two new antenna pattern file formats, bilinear and the National Spectrum Managers Association (NSMA).

Added support for a symmetrical RCS look up table. When the table setting is 2, the azimuth look up value is clamped between 0-180.

ASI Version 14

Added the Beam keyword **BeamInterpolatePos** to control the interpolation of position between discrete data points.

Added the Gate keyword **GateInterpolatePos** to control the interpolation of position between discrete data points.

ASI Version 13

Added the Scenario keywords **ITConfigFile** and **RuleFile** to control the association of an imagery and terrain configuration file and a preference rules file to the data file.

ASI Version 12

Added the Platform keyword **PlatformInterpolateOri** to control the interpolation of orientation between discrete data points.

ASI Version 11

Added the generic data keyword `SIMDIS_DrawMode`

ASI Version 10

Added the **OriginalID** keyword to Platforms, Beams and Gates.

Added the **RefAlt** keyword to scenario initialization

ASI Version 9

Changed time format strings; removed the weekday restriction.

Added support for category data to Platforms, Beams and Gates.

Changed Platform keywords **PlatformType** and **PlatformFHN** to become macros for creating Platform level category data.

Added the Beam keyword **BeamMissileOffset** for automatically offsetting a missile seeker beam to the nose of the missile.

ASI Version 8

Added the Beam keyword **BeamTargetIDCmd** as a replacement for the **BeamTargetCmd** keyword. BeamTargetCmd is now deprecated.

Changed `SIMDIS_AnimatedLine` generic data to use an entity ID instead of a name.

ASI Version 7

To avoid frequent confusion, generic data was changed to be a "property" of entities or of the scenario, rather than an object by itself.

Deprecated: **GenericObjID, EventObjID, GenericObjRefID**, and **EventObjRefID**.

ASI Version 6

Added the Scenario keyword **MagneticVariance** to specify the magnetic variance (declination) between magnetic north and true north.

The Platform keyword **PlatformRCS** was added to handle the association of RCS files to a platform.

Added the Platform keyword **PlatformAttachedGOG** to provide the ability to save GOG files attached to platforms.

The Beam keyword **AREPSPattern** was added to handle the association of AREPS ASCII output text files to a beam.

The Gate keyword **GateType** was added to indicate the type gate to be drawn.

Appendix B

GOG Commands

What are the different commands possible in a GOG file and how do you use them?

It is relatively simple for you to create your own GOG files. All you need to know are the various commands in the GOG file format. This Appendix details each of the commands and then gives several examples for them. Do not be alarmed by the selection of commands. Often times, when creating a simple GOG file only a few commands are needed.

B.1 GOG Structure Commands

B.1.1 start

Notes: Beginning command for a GOG object
Required Arguments: None
Optional Arguments: None
Required within start/end block: No
Supported: All

B.1.2 end

Notes: Ending command for a GOG object
Required Arguments: None
Optional Arguments: None
Required within start/end block: No
Supported: All

B.1.3 comment

Notes: One word is allowed; connect multiple words with a non-space,
i.e."_" or other punctuation
Required Arguments: None
Optional Arguments: None
Required within start/end block: No

Supported: All
See Also: C++ style usage, and shell script style usage. These commands are for SIMDIS use only.

B.1.4 off

Notes: Turns object off, place directly after the start command
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.1.5 version

Notes: Specifies the version of the GOG specification which this GOG was developed for.
Required Arguments: (1) Version number
Optional Arguments: None
Required within start/end block: No
Supported: Version 2, SIMDIS only

B.2 GOG Type Commands

B.2.1 annotation

Notes: 3D text. Annotations can also be used with Range Tool.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Single words all. Multiple words, SIMDIS only

B.2.2 arc

Notes: `centerxy` or `centerll`, `radius`, `anglestart` and `angledeg` commands are drawn as an arc.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.3 circle

Notes: `centerxy` or `centerll`, and `radius` commands are drawn as a circle.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.4 cone

Notes: `centerxy` or `centerll`, `radius`, and `height` commands create a cone. Cones are always filled.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.2.5 cylinder

Notes: `centerxy`, `centerll`, `radius`, and `height` commands create a cylinder.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.2.6 ellipse

Notes: `centerxy` or `centerll`, `majoraxis`, and `minoraxis` commands are drawn as an ellipse.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.7 ellipsoid

Notes: `centerxy`, `centerll`, `radius` or `diameter`, `majoraxis`, and `minoraxis` commands create an ellipsoid. You can use the keyword `height` to set the ellipsoid height if radius or diameter is not available.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.2.8 hemisphere

Notes: `centerxy`, `centerll`, `radius` commands create a hemisphere (flat portion towards the Earth)
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.2.9 latlonaltbox

Notes: Create a 3D box (without 6 arguments, the result is a 2D rectangle)
Required Arguments: (5) North lat, south lat, east lon, west lon, and altitude bounds
Optional Arguments: (1) Altitude for the top of the box
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.2.10 line

Notes: Position commands are drawn as a line. Requires at least two position commands
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.11 linesegs

Notes: Pairs of positions drawn as line segments. Requires an even number of position commands.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.12 points

Notes: Position commands are drawn as points.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.13 poly or polygon

Notes: Position commands are drawn as a polygon. Requires at least three position commands.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.2.14 sphere

Notes: centerxy, centerll, and radius commands create a sphere
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.3 GOG Position Commands

B.3.1 centerll or centerlatlon or centerlla

Notes: Used to specify the starting offset for arcs, circles or ellipses.
Required Arguments: (2) Latitude, longitude
Optional Arguments: (1) Altitude in feet, defaults to 0.0 if not set
Required within start/end block: Yes
Supported: All

B.3.2 centerxy or centerxyz

Notes: Used to specify the starting offset for arcs, circles or ellipses.
Required Arguments: (2) X and Y offset positions in yards
Optional Arguments: (1) Z offset in feet, defaults to 0.0 if not set
Required within start/end block: Yes
Supported: All

B.3.3 ll or latlon or lla

Notes: Static Geodetic position command. Latitude and Longitude can be in the following notations: DMS, DM.D, D.D
Required Arguments: (2) Latitude, longitude
Optional Arguments: (1) Altitude in feet, defaults to 0.0 if not set
Required within start/end block: Yes
Supported: All

B.3.4 mgrs

Notes: Behaves similar to the ll/lla command. It takes a single MGRS coordinate (that decodes to latitude and longitude in one token), with an optional altitude entry.
Required Arguments: (1) MGRS position string.
Optional Arguments: (1) Altitude
Required within start/end block: Yes
Supported: All

B.3.5 ref or referencepoint

Notes: Reference position point, used to statically fix a relative GOG object to a point
Required Arguments: (2) Latitude, longitude
Optional Arguments: (1) Altitude in feet, defaults to 0.0 if not set
Required within start/end block: Yes
Supported: All

B.3.6 xy or xyz

Notes: Relative offset position command.
Required Arguments: (2) X and Y offset positions in yards
Optional Arguments: (1) Z offset in feet, defaults to 0.0 if not set
Required within start/end block: Yes
Supported: All

B.4 GOG Unit Commands

B.4.1 altitudeunits

Notes: Sets the units to be used for all altitude (z) quantities
Required Arguments: (1) Distance Units: mm,cm,in,ft,yd,m,fm,kf,kyd,km,sm,nm,dm
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.4.2 angleunits

Notes: Sets the units to be used for angular quantities
Required Arguments: (1) Angle units: deg,rad,bam,mil
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.4.3 rangeunits

Notes: Sets the units to be used for planar (x,y) quantities
Required Arguments: (1) Distance Units: mm,cm,in,ft,yd,m,fm,kf,kyd,km,sm,nm,dm
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.4.4 timeunits

Notes: Specifies the unit to be used for time quantities
Required Arguments: (1) Time units: sec,min,hour
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.4.5 verticaldatum

Notes: Sets the Earth model for altitude values
Required Arguments: (1) Vertical datum: WGS84, EGM1984, EGM1996, EGM2008
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5 GOG Modifier Commands

B.5.1 altitudemode

Notes: Sets the manner in which altitude is interpreted for the GOG
Required Arguments: (1) Choice from:
`clampToGround` - ignore altitude, place points on terrain
`relativeToGround` - add altitude to terrain
`absolute` - use altitude as relative to vertical datum
(ignore terrain)
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.2 angledeg

Notes: Sets the clockwise angle an arc will extend. Use this command for arcs or cylinders that cross 360.
Required Arguments: (1) Angle value in degrees.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.3 angleend

Notes: Sets the absolute ending angle for an arc or cylinder. Neither arcs nor cylinders will cross 0 or 360 with this command.
Required Arguments: (1) Angle value in degrees.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.4 anglestart

Notes: Sets the absolute starting angle for an arc or cylinder.
Required Arguments: (1) Angle value in degrees.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.5 diameter

Notes: Sets the diameter for arc, circle, cylinder, ellipse, sphere, hemisphere and ellipsoid objects.
Required Arguments: (1) Distance value in specified units.
Optional Arguments: None
Required within start/end block: Yes

Supported: Version 2, SIMDIS only

B.5.6 depthbuffer

Notes: Enables the Open GL depth buffer
Required Arguments: (1) True or false
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.7 extrude

Notes: Cause the GOG to extend towards the Earth. Points become lines,
lines become rectangles, circles become cylinders, etc.
Required Arguments: (1) True or false
Optional Arguments: (1) A delta altitude extrusion limit, based on
current altitude unit setting within start end block
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.8 fillcolor

Notes: Only six colors in the original format, however, a 2nd value
will override the original. This keyword will only work with:
2D GOG types: arc, circle, ellipse, and polygon;
Extruded 2D GOG types: line, linesegs, arc, circle, ellipse, and polygon;
3D GOG types: sphere, ellipsoid, hemisphere, cylinder, cone, and latlonaltbox.
Required Arguments: (1) Fill color
Optional Arguments: (1) Hexadecimal color in the format 0xAABBGGRR
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.9 filled

Notes: Displays GOGs as a solid instead of wireframe. This keyword will only work with:
2D GOG types: arc, circle, ellipse, and poly;
3D GOG types: sphere, ellipsoid, hemisphere, cylinder, and latlonaltbox.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS Only

B.5.10 fontname

Required Arguments: (1) Font name
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.11 fontsize

Required Arguments: (1) Font pixel size
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.12 height

Notes: Sets the height value to extrude a cylinder upwards.
It is also used to set the height of ellipsoids and cones.
Required Arguments: (1) Value in feet, unless unit is specified by the
keyword 'altitudeunits'.
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.13 linecolor

Notes: Sets the line color for GOG shapes, and the text color of annotation GOGs.
Only six colors in the original format, however, a 2nd value
will override the original.
Required Arguments: (1) Line color (used as fill color in version 1)
Optional Arguments: (1) Hexadecimal color in the format 0xAABBGGRR
Required within start/end block: Yes
Supported: Original colors, all. Hexadecimal colors, SIMDIS only

B.5.14 lineprojection

Notes: Set the projection to use when rendering GOG lines or points
Required Arguments: (1) Line projection: greatCircle, rhumbline

Optional Arguments: None
Required within start/end block: Yes
Supported: SIMDIS only

B.5.15 linestyle

Notes: Sets the line style for a line object.
Required Arguments: (1) Line style: solid, dash, dot
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.16 linewidth

Notes: Sets the line width for a line object.
Required Arguments: (1) Line width
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.17 majoraxis

Notes: Sets the major axis for an ellipse.
Required Arguments: (1) Length of major axis in yards.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.18 minoraxis

Notes: Sets the minor axis for an ellipse
Required Arguments: (1) Length of minor axis in yards.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.19 orient

Notes: Rotate GOG by the given amount(s)
Required Arguments: (1) Change in heading in degrees
Optional Arguments: (2) Changes in pitch and roll in degrees
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.20 outline

Notes: Causes the GOG to be drawn with an outline along all edges
Required Arguments: (1) True or false

Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.21 pointsize

Notes: Sets the point size for a point object.
Required Arguments: (1) Point size
Optional Arguments: None
Required within start/end block: Yes
Supported: SIMDIS only

B.5.22 priority

Notes: Affects text priority for SIMDIS Text Deconfliction feature. Positive values indicate relative priority, with 0 being the lowest priority and higher values less likely to be hidden. Negative values indicate that the text should never be hidden. Only applies to text, and only applies when Text Deconfliction is enabled. Default value is 0.
Required Arguments: (1) Positive priority integer value
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.23 radius

Notes: Sets the radius for arc, circle, cone, cylinder, ellipse, sphere, hemisphere and ellipsoid objects.
Required Arguments: (1) Distance value in yards.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.24 rotate

Notes: Rotates a relative GOG object based on an attached platform's course.
Required Arguments: None
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.25 scale

Notes: Multiply size of GOG in each dimension
Required Arguments: (3) x, y, and z scales
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.26 semimajoraxis

Notes: Sets the semimajor axis for an ellipse. Equivalent to majoraxis with the argument doubled
Required Arguments: (1) Half length of major axis in yards.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.27 semiminoraxis

Notes: Sets the semiminor axis for an ellipse. Equivalent to minoraxis with the argument doubled
Required Arguments: (1) Half length of minor axis in yards.
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.28 tessellate

Notes: Cause lines within the GOG to be broken up into many points, useful for terrain following and GOGs that span large distances

Required Arguments: (1) True or false
Optional Arguments: None
Required within start/end block: Yes
Supported: All

B.5.29 textoutlinecolor

Notes: Set the outline color of annotation objects.
Required Arguments: (1) Outline color
Optional Arguments: (1) Hexadecimal color in the format 0xAABBGGRR
Required within start/end block: Yes
Supported: Version2, SIMDIS only

B.5.30 textoutlinethickness

Notes: Set the thickness of the outline for annotation objects.
Required Arguments: (1) Outline thickness (None, thin, or thick)
Optional Arguments: None
Required within start/end block: Yes
Supported: Version 2, SIMDIS only

B.5.31 3d

Notes: SIMDIS specific command. Allows specific modifications to GOG objects.
Required Arguments: Yes, see below
Supported: SIMDIS only.

B.5.32 3d billboard

Notes: Text object will always be perpendicular to the viewer. This is only available for absolute GOG files. Billboards will not work with GOG
Required Arguments: None
Optional Arguments: None
Supported: SIMDIS only.

B.5.33 3d follow cpr

Notes: Course, pitch and roll offsets used from an attached platform for a relative GOG object.
Required Arguments: (1) Combination of cpr flags, i.e., c, p, r, cp, cr, pr, cpr
Optional Arguments: None
Required within start/end block: Yes
Supported: SIMDIS only.

B.5.34 3d name

Notes: Sets a name, similar to a call sign, for a GOG object.
Required Arguments: (1) A character string.
Optional Arguments: None
Supported: SIMDIS only.

B.5.35 3d offsetalt

Notes: Altitude offset for an attached GOG object.
Required Arguments: (1) Altitude offset in feet.
Optional Arguments: None
Supported: SIMDIS only.

B.5.36 3d offsetcourse

Notes: Course offset for an attached GOG object.
Required Arguments: (1) Angle offset in degrees.
Optional Arguments: None
Supported: SIMDIS only.

B.5.37 3d offsetpitch

Notes: Pitch offset for an attached GOG object.
Required Arguments: (1) Angle offset in degrees.
Optional Arguments: None
Supported: SIMDIS only.

B.5.38 3d offsetroll

Notes: Roll offset for an attached GOG object.
Required Arguments: (1) Angle offset in degrees.
Optional Arguments: None
Supported: SIMDIS only.

Appendix C

Command Line and Installation Supplement

C.1 Creating a Batch File

Running SIMDIS through the desktop shortcut or typing **simdis** in the command prompt (if **PATH** is configured) will load the default settings. A batch file with command line arguments allows flexibility in running SIMDIS to load custom settings file, preference rules, terrain, and etc.

You can combine multiple arguments to satisfy your requirement such as multiple terrain and/or bookmarks. To get the list of arguments, open the command prompt (terminal in Linux platforms) and type **simdis10 –help**.

To create a batch file, open a text editor (e.g. Notepad) and save a file with a **.bat** file extension. The following subsections are examples of how to run SIMDIS from a batch file.

C.1.1 Start SIMDIS

To simply start SIMDIS with no arguments, type the following in your text editor:

```
@echo off
start simdis
```

NOTES:

- The line **@echo off** is used to prevent command line outputs in the command prompt.
- The command **start** is used to avoid running an extra command window. Omit this if you want the extra window specifically to see the errors (it is also logged in the console window in SIMDIS).

C.1.2 Start SIMDIS with Default Settings

To start SIMDIS with the default settings, add the **–clearSettings** argument:

```
@echo off
start simdis --clearSettings
```

NOTE: This argument will cause your settings to be reset to the default. A backup of your settings file will be saved as **SIMDIS.ini.back**. This backup file is useful in many cases, such as when **–clearSettings** solved a problem. You can then post your backup settings file to the JIRA Help Desk, to assist SIMDIS developers in reproducing the problem.

C.1.3 Start SIMDIS with an ASI File

To automatically load an ASI file when SIMDIS starts, add the **–file** argument and specify the file name of the ASI file:

```
@echo off
start simdis --file "test.asi"
```

NOTE: In this example, assume that the **.asi** file is in the same directory with the batch file.

C.1.4 Start SIMDIS with a Custom Toolbar

To start SIMDIS with a custom toolbar:

```
@echo off
start simdis --Custom_Toolbar_Plugin:configFile "myToolbar.xml"
```

NOTE: The **"myToolbar.xml"** represents the directory of the generated custom toolbar. Please see Section 4.2.2.1 for guidance to generate a custom toolbar.

C.1.5 Start SIMDIS with specified SIMDIS User Directory

To start SIMDIS with a specified SIMDIS User Directory (`SIMDIS_USER_DIR`) use the Windows **setlocal** line for environment variable:

```
@echo off
setlocal
set SIMDIS_USER_DIR=C:\textbackslash SIMDIS
start simdis
endlocal
```

NOTES:

- The lines **setlocal** and **endlocal** allow you to set any environment variables not only limited to SIMDIS.
- Make sure that there are no spaces in setting the environment variable other than after the command **set**.

C.2 Full Installer Command Line Parameters

Option	Argument	Description
-help		Display the list of valid options.
-version		Display the product information.
-unattendedmodeui	`<unattendedmodeui>`	Unattended Mode UI Default: none Allowed: none minimalWithDialogs
-optionfile	`<optionfile>`	Installation option file. Default
-debuglevel	`<debuglevel>`	Debug information level of verbosity. Default: 2 Allowed: 0, 1, 2, 3, 4
-mode	`<mode>`	Installation mode. Default: win32 Allowed: win32 unattended
-installer-language	`<installer-language>`	Language Selection. Default: en Allowed: sq, ar, es, _AR, pt, _BR, bg, ca, hr, cs, da, nl, en, et, fi, fr, de, el, he, hu, it, ja, ko, lv, lt, no, pl, pt, ro, ru, sr, zh, _CN, sk, sl, es, sv, th, zh,_TW, tr, va, cy
-prefix	`<prefix>`	Installation Directory. Default: C:\Program Files (x86)\SIMDIS
-setpath	`<setpath>`	Default: true Allowed: true, false
-createDesktopIcons	`<createDesktopIcons>`	Create SIMDIS Desktop Icons Default: 0

C.3 Extensions Distribution and Dependency

Name	Distribution	Dependency	Description	Comments
BookmarkEvents	Full/Core	FileSearch GogTool PrefsTool Views	Adds bookmark support. Includes event support for demos (i.e. BMD Demo).	
Console	Full/Core		Info, warning , and error message display in a console popup.	
CPA	Full		Closest Point of Approach (CPA) calculations and GUI.	
DataScripts	Full		Data scripts GUI and execution	Useful for processing live TM data for MDA.
DatumConvert	Full/Core		Provides Utils datum conversion stuff to calculations that interacts with MagVar and VD.	May become obsolete, shift to simCore/simUtil solution.
DISCN	Full/Core	FileSearch	Loads DISCN files needed for live configuration setups.	
Ephemeris	Full		Provides ephemeris data for label calculations and advanced platform lighting calculations.	Based on US Naval Observatory Solar-Lunar Almanac Core (SLAC); Limited Distribution.
Export	Full/Core		Responsible for providing import and export functionality, integrated through plug-ins.	
FileSearch	Full/Core		Controls file search options such as recursion depth.	
GogTool	Full/Core		Loads Generalized Overlay Graphics (GOG)	
HotKeys	Full/Core		GUI for manipulating hot keys.	
Hud	Full/Core		Draws compass, wind, corner text, classification, and logos.	
LabelContent	Full/Core		Provides implementation of text display for labels and legends.	
Legend	Full/Core	HUD	Movable legends on the HUD based on input from Prefs Tool.	
LiveDebugging	PMRF		Not Distributed (except PMRF) - Various developer tools for live mode debugging.	
MapEditor	Full/Core		Replacement for Imagery and Terrain tool.	
Marker	Full		Create platform markers.	

Name	Distribution	Dependency	Description	Comments
MediaPlayer	Full/Core		Synchronizes TMD, LST, PST, and media files with SIMDIS.	
MiniMap	Full		Draws a small version of the earth on the HUD.	
MouseStatusBar	Full/Core		Adds mouse controls to the status bar.	
OceanModel	Full/Core		Simple Ocean, and no ocean.	Defaults to no ocean if not installed. Triton is part of the Triton_SilverLining Distribution and not in the Full or Core Build.
PlanetariumAzEl	Full		Adds planetarium display capability where beams, gates and platforms are projected on a dome	
PlatformExpiration	Full		Swapping between queued and immediate mode deletion.	Used for controlling how live data is removed, defaults to immediate deletion if not installed
PluginManager	Full/Core		Plug-in Manager allows for starting, stopping, etc. of plug-ins; else there is no GUI.	
Prefs Tool	Full/Core		Edit settings for entities in Prefs Tool.	
ReferenceGrid	Full		Draws a geodetic grid on the earth.	
RFPropagation	Full		RF Propagation integration used to load and display RF propagation data.	
RTStandardGroups	Full/Core		Default list of range tool calculations; including slant range, composite angle, etc.	
RangeTool	Full/Core	RTStandardGroups	Associate two entities and add range calculations. Can save and load calculations from file and command line.	
RemoteControl	Full		Share entity prefs with another machine running the same data feed.	
Ruler	Full		Adds a mouse tool for calculating range and bearing where you click.	
Selection	Full		Manages legacy Current Selection capability for Plug-in API.	
Settings	Full/Core		GUI for viewing and editing all persistent settings in SIMDIS.	

Name	Distribution	Dependency	Description	Comments
SkyModel	Full/Core		Swap between different sky models based on desired CPU/GPU performance and graphics	Defaults to no sky if not installed. SilverLining is part of the Triton_SilverLining Distribution and not in the Full or Core Build.
SstFormat	Full		Implements the Plug-in API SST format for screen text.	
Stereoscopic	Full		Enables stereoscopic functionality and parameter manipulation.	
TimeEditor	Full/Core		Time Editor GUI for manipulating time.	
Toast	Full/Core		Displays toast popups for certain notification text.	
UnitEditor	Full		Change per-context units and precision and formats.	
VersionCheck	Full/Core		Checks your SIMDIS version and informs user when the software is out of date.	
Views	Full/Core		Views dialog and Eye Positions dialog.	
WindowGeometry	Full/Core		Adds geometry command line options.	

C.4 Environment Variables

SIMDIS uses a number of environment variables for proper operation

Variable	Description
`$(SIMDIS_DIR)`	The SIMDIS installation Directory.
`$(SIMDIS_HOME)`	The location of the SIMDIS home directory containing the user's preference and rules files. This setting cannot be changed. Windows: `%APPDATA%\Naval Research Laboratory\home` Linux: `~/config/Naval Research Laboratory/home`
`$(SIMDIS_LICENSE_########)`	The SIMDIS license key based on computer ID number.
`$(SIMDIS_TERRAIN)`	The location of the SIMDIS terrain texture and elevation sets. This variable can be changed to use a different directory e.g. a different drive.
`$(SIMDIS_USER_DIR)`	A user-defined directory for the storage of the user data, e.g. ASI files, GOG files, 3D models, etc. This variable is to be created and set by you. The data stored in this directory will be protected from future SIMDIS updates, installs, etc.

This Page is Intentionally Left Blank

Appendix D

Terrain and Imagery Supplement

D.1 Supported Terrain Formats

D.1.1 Tile Map Service (TMS)

The Tile Map Service specification describes a standard for organizing and accessing cartographic maps of geo-referenced data. The TMS driver provides access to OSGeo Tile Map Service repositories. This could be an online repository, (e.g. http://readymap.org/readymap/tiles/1.0.0/22/), or point to a repository on the local file system.

Properties:

- TMS Type - set to Google to invert the Y-axis of the tile index
- Format - file format, e.g. **.png**, **.jpg**, only used if no valid TMS file found
- Tile Size - tile size in pixels of the imagery, only used if no valid TMS file found

D.1.2 Web Mapping Service (WMS)

A specification developed by the Open Geospatial Consortium. The WMS standard provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. The WMS driver provides access to an OGC Web Mapping Service data source (e.g. http://onearth.jpl.nasa.gov/wms.cgi).

Properties:

- Layers - list of layers to use from the WMS, comma separated
- Format - file format, e.g. **png**, **jpg**, defaults to **png**

- Tile Size - override the default tile size in pixels of the imagery, defaults to 256
- Spatial Reference System - expected to be an **EPSG** code, defaults to **EPSG:4326**
- Style - WMS styles to render

D.1.3 Geospatial Data Abstraction Layer (GDAL)

GDAL is a translator library for raster geospatial data formats. It can interpret a large number of data formats. The GDAL driver lets you access raster or elevation data sources through the GDAL abstraction layer, opening up a wide range of file format and web service support. This is the driver to use when reading local files in formats such as **PNG**, **GeoTIFF**, **ECW**, and **MrSID**.

Properties:

- Extensions - semi-colon delimited list of extensions to query if the URL options points to a directory
- Tile Size - override the default tile size in pixels of the imagery, defaults to 256
- Max Data Level - max level where this data source will return data, will be computed if not set
- Interpolation - interpolation method used for sampling data source
- Interpolate Imagery - if set, will interpolate imagery

D.1.4 Map Box Tiles (mbtiles)

The MBTiles specification developed by MapBox provides an efficient mechanism for storing web maps in a SQLite database.The MBTiles driver reads imagery tiles from the MBTiles SQLite-based packaging format.

Properties:

- Format - file format, e.g. **png**, **jpg**

D.1.5 DB Driver

The DB driver is a custom implemented osgEarth plug-in to handle the original SIMDIS 9 terrain type files.

D.2 Convenience Drivers

D.2.1 Tile Package

The Tile Package driver loads the ESRI 10.1 version of their compact cache. The driver does not support the .tpk archive, bundle files must be extracted into a directory and the directory layout must contain a **Conf.xml** file.

D.2.2 ReadyMap

SIMDIS also has some convenience image layers for basic terrain image data, using the built-in osgEarth drivers. ReadyMap Tiles is a pre-tiled, worldwide collection of satellite imagery and elevation data that is available from Pelican Mapping's online server.

D.2.3 Weather Maps

The Weather Maps section provides access to WMS feeds for current weather imagery. This functionality is presently using online weather data provided by the Iowa State University and the National Oceanic and Atmospheric Administration (NOAA) nowCOAST. Various layer options include visible and infrared satellite imagery, National Weather Service warnings, and data from the NEXRAD radar network.

D.2.4 USGS HRO

The United States Geological Survey (USGS) High-Resolution Orthoimagery (HRO) section includes access to WMS feeds of 1m and 1ft resolution color imagery for selected areas in the United States.

D.3 Earth File Reference

For more information about **.earth** file, please refer to the osgEarth Documentation.

This Page is Intentionally Left Blank

Appendix E

Range Tool

E.1 Distance Calculations

E.1.1 Ground Range | Dist

The **Ground Range | Dist** calculation is the length of the geodesic line that connects two entities. If the association's Earth model option is set to **WGS-84**, then the calculation uses Sodano's fast, iterative approach to the inverse geodesic problem as described in `"Direct and Inverse Solutions in Geodesics Technical Report 7", 1963 pp 15-27.`

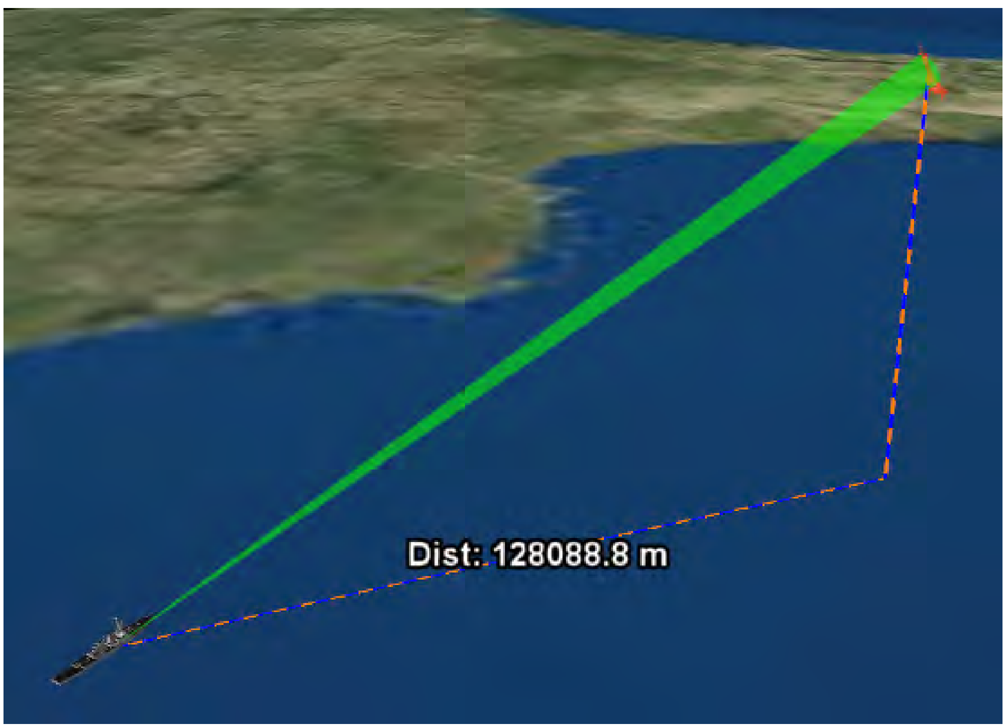


Figure E.1: Ground Range

E.1.2 Slant Range | Rng

Slant Range | Rng is the calculated distance along the slant line between two entities. The slant range calculation measures the shortest distance between two entities, even if it cuts through the Earth. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

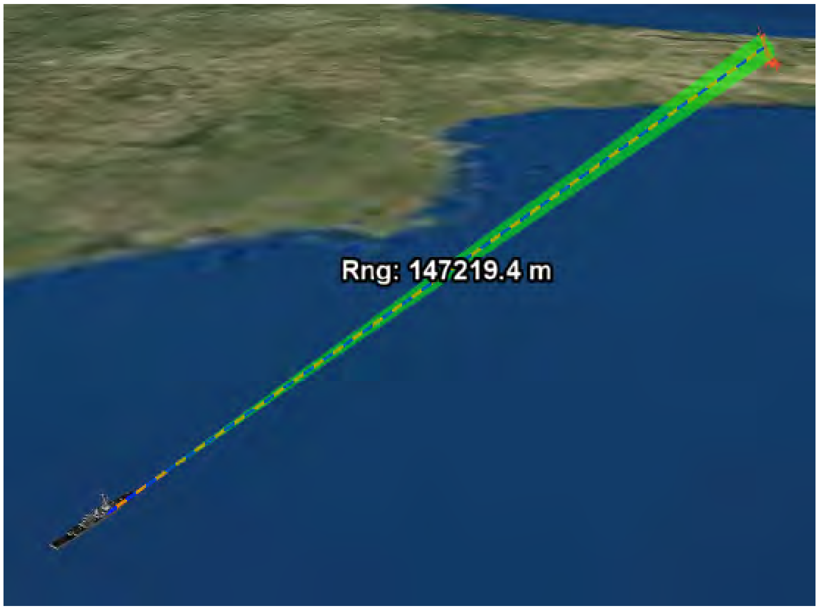


Figure E.2: Slant Range Calculation

E.1.3 Altitude | Alt

The **Altitude | Alt** calculation is the vertical difference between the altitude values of two entities. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

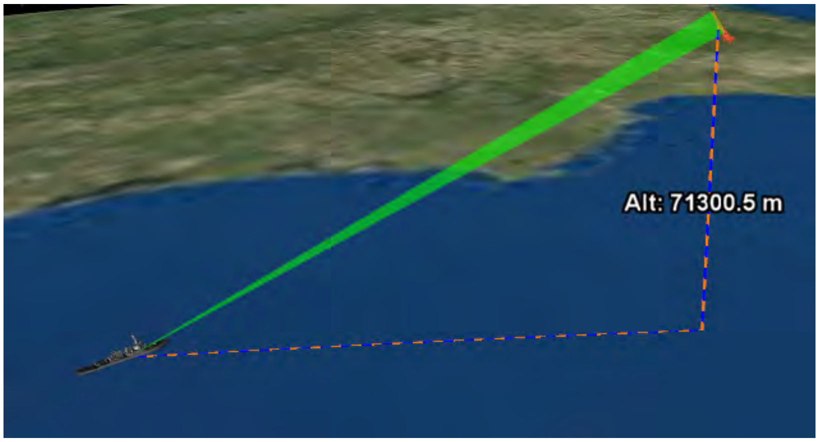


Figure E.3: Altitude Calculation

E.1.4 Beam Ground Range | Dist(B)

The **Beam Ground Range | Dist(B)** calculation is similar to the ground range calculation. However, instead of the beam's origin being used in the calculation, a point along the beam's pointing vector, normal to the other entity, is used. One of the entities must be a beam for this calculation to be valid.

Figure E.4: Beam Ground Range Calculation

E.1.5 Beam Slant Range | Rng(B)

The **Beam Slant Range | Rng(B)** calculation is similar to the slant range calculation. However, instead of the beam's origin being used in the calculation, a point along the beam's pointing vector, normal to the other entity, is used. One of the entities must be a beam for this calculation to be valid.

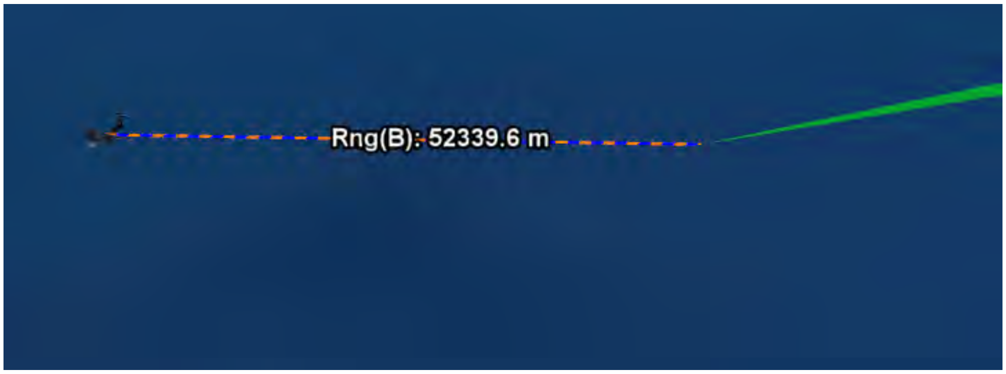


Figure E.5: Beam Slant Range Calculation

E.1.6 Beam Altitude | Alt(B)

The **Beam Altitude | Alt(B)** calculation is similar to the altitude calculation. However, instead of the beam's origin being used in the calculation, a point along the beam's pointing vector, normal to the other entity, is used. One of the entities must be a beam for this calculation to be valid.

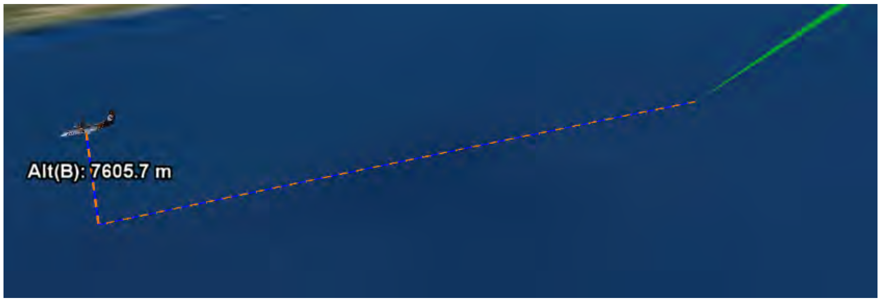


Figure E.6: Beam Altitude Calculation

E.1.7 Down Range | DR

The **Down Range** (**DR**) calculation is the range subcomponent of the horizontal planar range to the end entity that is parallel to the yaw value of the begin entity. The horizontal planar range is determined using a local tangent plane at the begin entity. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

Figure E.7: Down Range Calculation

E.1.8 Cross Range | CR

The **Cross Range | CR** calculation is the range subcomponent of the horizontal planar range to the end entity that is perpendicular to the yaw value of the begin entity. The horizontal planar range is determined using a local tangent plane at the begin entity. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

Figure E.8: Cross Range Calculation

E.1.9 Down Value | DV

The **Down Value | DV** calculation is the shortest distance between a plane tangent to the Earth and the end entity. The origin of the tangent plane is located at the beginning entity's position and altitude. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

Figure E.9: Down Value Calculation

E.1.10 Geodesic Down Range | DR(g)

The **Geodesic Down Range | DR(g)** calculation is the range subcomponent of the horizontal geodesic range between the begin and end entities that is parallel to the yaw value of the begin entity. The horizontal geodesic range is determined using Sodano's fast, iterative approach to the inverse geodesic problem as described in `"Direct and Inverse Solutions in Geodesics Technical Report 7", 1963 pp 15-27`. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

Figure E.10: Geodesic Down Range Calculation

E.1.11 Geodesic Cross Range | CR(g)

The **Geodesic Cross Range | CR(g)** calculation is the range subcomponent of the horizontal geodesic range between the begin and end entities that is perpendicular to the yaw value of the begin entity. The horizontal geodesic range is determined using Sodano's fast, iterative approach to the inverse geodesic problem as described in `"Direct and Inverse Solutions in Geodesics Technical Report 7", 1963 pp 15-27`. When non-platform entities are selected, the calculation uses the centroid point for gates and the origin point for beams, lasers, and LOBs.

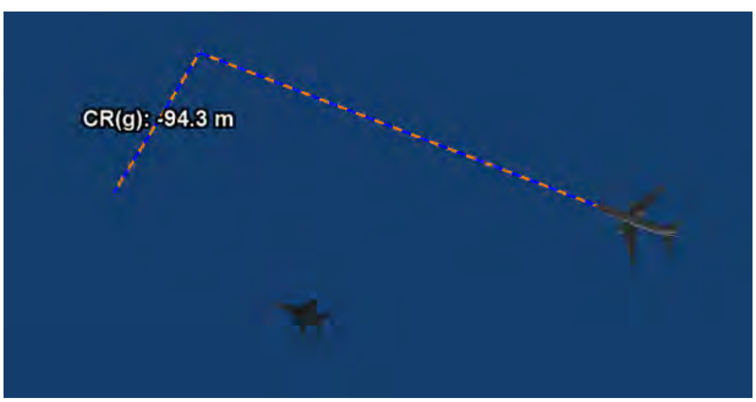


Figure E.11: Geodesic Cross Range Calculation

E.1.12 Velocity Azimuth Down Range | DR(v)

The **Velocity Azimuth Down Range | DR(v)** calculation is the range subcomponent of the horizontal planar range to the end entity that is parallel to the azimuth component of the velocity vector of the begin entity. The horizontal planar range is determined using a local tangent plane at the begin entity.

Figure E.12: Velocity Azimuth Down Range Calculation

E.1.13 Velocity Azimuth Cross Range | CR(v)

The **Velocity Azimuth Cross Range | CR(v)** calculation is the range subcomponent of the horizontal planar range to the end entity that is perpendicular to the azimuth component of the velocity vector of the begin entity. The horizontal planar range is determined using a local tangent plane at the begin entity.

Figure E.13: Velocity Azimuth Cross Range Calculation

E.1.14 Velocity Azimuth Geodesic Down Range | DR(gv)

The **Velocity Azimuth Geodesic Down Range | DR(gv)** calculation is the range subcomponent of the horizontal geodesic range between the begin and end entities that is parallel to the azimuth component of the velocity vector of the begin entity. The horizontal geodesic range is determined using Sodano's fast, iterative approach to the inverse geodesic problem as described in `"Direct and Inverse Solutions in Geodesics Technical Rep`

Figure E.14: Velocity Azimuth Geodesic Down Range Calculation

E.1.15 Velocity Azimuth Geodesic Cross Range | CR(gv)

The **Velocity Azimuth Geodesic Cross Range | CR(gv)** calculation is the range subcomponent of the horizontal geodesic range between the begin and end entities that is perpendicular to the azimuth component of the velocity vector of the begin entity. The horizontal geodesic range is determined using Sodano's fast, iterative approach to the inverse geodesic problem as described in `"Direct and Inverse Solutions in Geodesics Technical Report 7", 1963 pp 15-27.`

Figure E.15: Velocity Azimuth Geodesic Cross Range Calculation

E.2 True Angles

E.2.1 Magnetic Azimuth | Az(M)

The **Magnetic Azimuth | Az(M)** calculation is the horizontal component angle between two entities referenced to Magnetic North. The first vector points from the begin entity north along a plane that is tangent to the surface of the Earth at the same altitude as the begin entity. The second vector points from the beginning entity toward the end entity along this same tangent plane.

This calculation is not supported between beams, gates, lasers, or LOBs that do not share the same host platform. When the calculation is supported, the computed angle will be the difference between Magnetic North and the azimuth angle of the beam, gate, laser, or LOB.

Figure E.16: Magnetic Azimuth Calculation

E.2.2 True Azimuth | Az(T)

The **True Azimuth | Az(T)** calculation is the horizontal component angle between two entities referenced to True North. The first vector points from the begin entity north along a plane that is tangent to the surface of the Earth at the same altitude as the begin entity. The second vector points from the begin entity toward the end entity along this same tangent plane.

This calculation is not supported between beams, gates, lasers, or LOBs that do not share the same host platform. When the calculation is supported, the computed angle will be the difference between True North and the azimuth angle of the beam, gate, laser, or LOB.

Figure E.17: True Azimuth Calculation

E.2.3 True Elevation | El

The **True Elevation | El** calculation is the vertical component angle between two entities referenced to the horizon. The first vector points from the begin entity to the end entity. The second vector lies in a plane tangent to the Earth's surface, located at the begin entity and pointing towards the end entity.

This calculation is not supported between beams, gates, lasers, or LOBs that do not share the same host platform. When the calculation is supported, the computed angle will be the difference between the horizon and the elevation angle of the beam, gate, laser, or LOB.

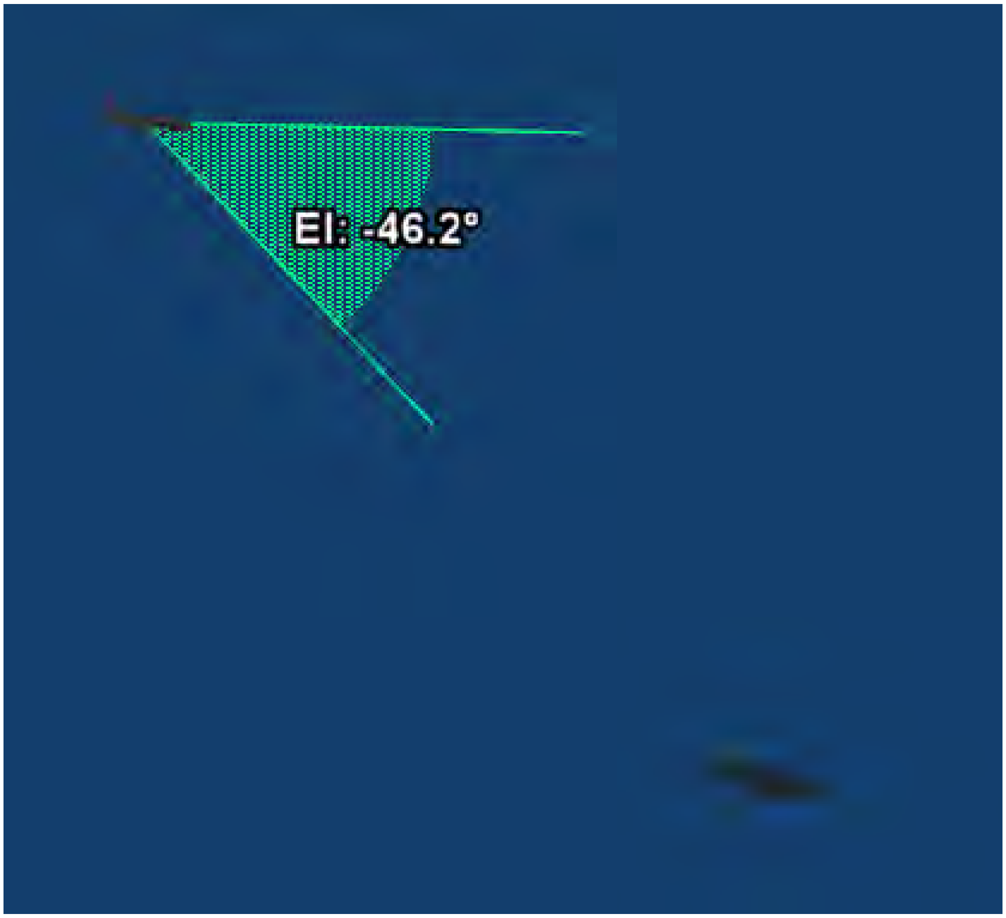


Figure E.18: True Elevation Calculation

E.2.4 True Composite Angle | Cmp(T)

The **True Composite Angle | Cmp(T)** calculation is the angle between two vectors that define a plane between the entities. The first vector originates at the begin entity and points north along a plane that is tangent to the Earth at the same altitude as the begin entity. The second vector originates at the begin entity and points directly towards the end entity, this vector may cut through the Earth.

This calculation is not supported between beams, gates, lasers, or LOBs that do not share the same host platform. When the calculation is supported, the computed angle will be the difference between True North and the pointing vector (azimuth and elevation) angle of the beam, gate, laser, or LOB.

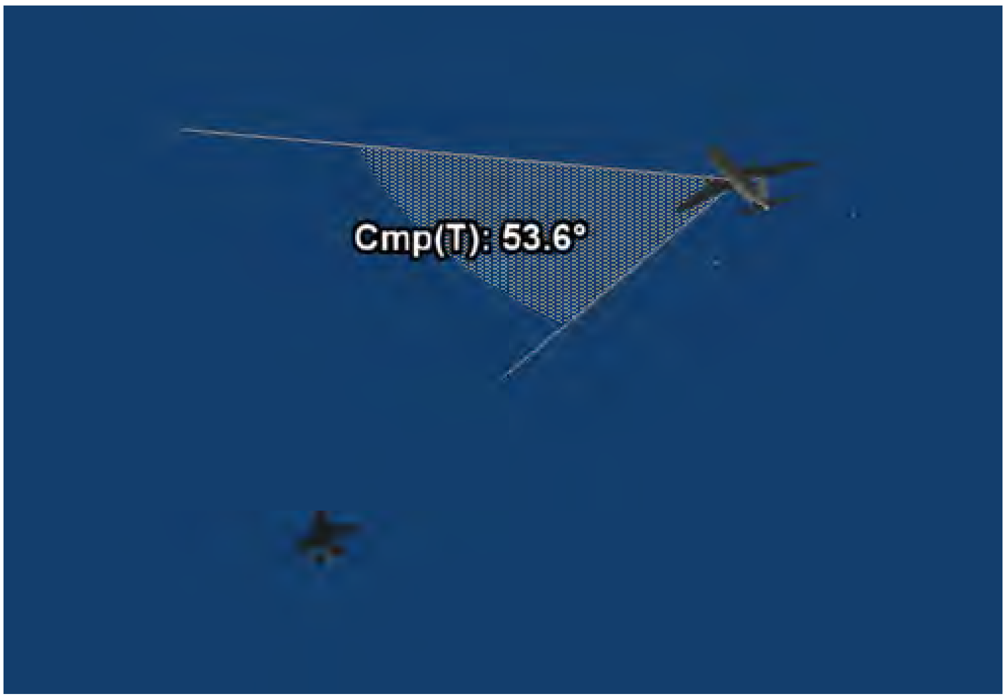


Figure E.19: True Composite Angle Calculation

E.3 Orientation Relative Angles

E.3.1 Orientation Relative Azimuth | Az(r)

The **Orientation Relative Azimuth Calculation | Az(r)** is measured along a tangent plane that is rotated to align with the orientation vector of the begin entity. The angle is a measurement between the heading axis of the plane and a point along the plane tangent to the end entity's location. This angle is measured in a positive sense according to the right-hand rule about the down vector perpendicular to the orientation plane. A relative azimuth of 0 degrees specifies a location directly in front of the entity and a relative azimuth of 180 degrees specifies a location directly behind the entity.

This calculation is not supported when an end entity beam, gate, laser, or LOB does not share the same host platform. When the calculation is supported, the computed angle will be the difference between the azimuth angle of the begin entity and the azimuth angle of the beam, gate, laser, or LOB.

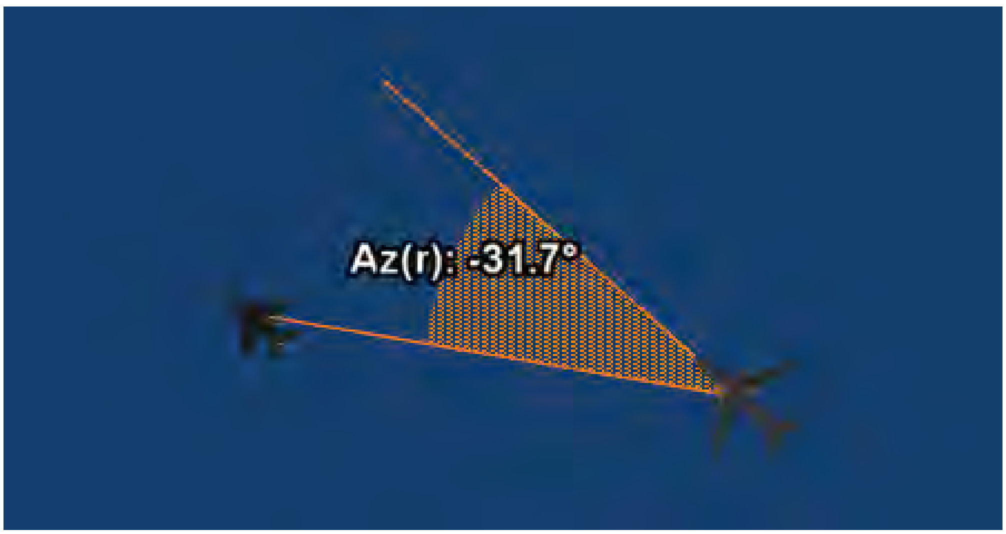


Figure E.20: Orientation Relative Azimuth Calculation

E.3.2 Orientation Relative Elevation | El(r)

The **Orientation Relative Elevation | El(r)** calculation is measured along a tangent plane that is rotated to align with the orientation vector of the begin entity. The angle is measured between the plane and a point along the plane tangent to the end entity's location. The relative elevation angle is positive for entities above the plane and negative for entities below the plane.

This calculation is not supported when an end entity beam, gate, laser, or LOB does not share the same host platform. When the calculation is supported, the computed angle will be the difference between the elevation angle of the begin entity and the elevation angle of the beam, gate, laser, or LOB.

Figure E.21: Orientation Relative Elevation Calculation

E.3.3 Orientation Relative Composite Angle | Cmp(r)

The **Orientation Relative Composite Angle | Cmp(r)** calculation is measured along a tangent plane that is defined by the orientation vector of the begin entity and a position vector between the two entities. This angle is measured in a positive sense according to the right-hand rule about the down vector perpendicular to the defined plane.

This calculation is not supported when an end entity beam, gate, laser, or LOB does not share the same host platform. When the calculation is supported, the computed angle will be the difference between the orientation angle of the begin entity and the pointing (azimuth and elevation) angle of the beam, gate, laser, or LOB.

Figure E.22: Orientation Relative Composite Angle Calculation

E.3.4 Aspect Angle | Asp(r)

The **Aspect Angle Calculation | Asp(r)** is measured between the line of sight of the begin entity to the end entity and the line extending through the longitudinal axis of the end entity. This angle is between 0 and 180 degrees. An aspect angle of 0 degrees occurs when the end entity oriented along the line of sight between the two entities and is pointing at the start entity. An aspect angle of 180 degrees occurs when the end entity is oriented along the line of sight between the two entities and is pointing away from the start entity. This calculation is only valid for begin and end entities that are platforms.

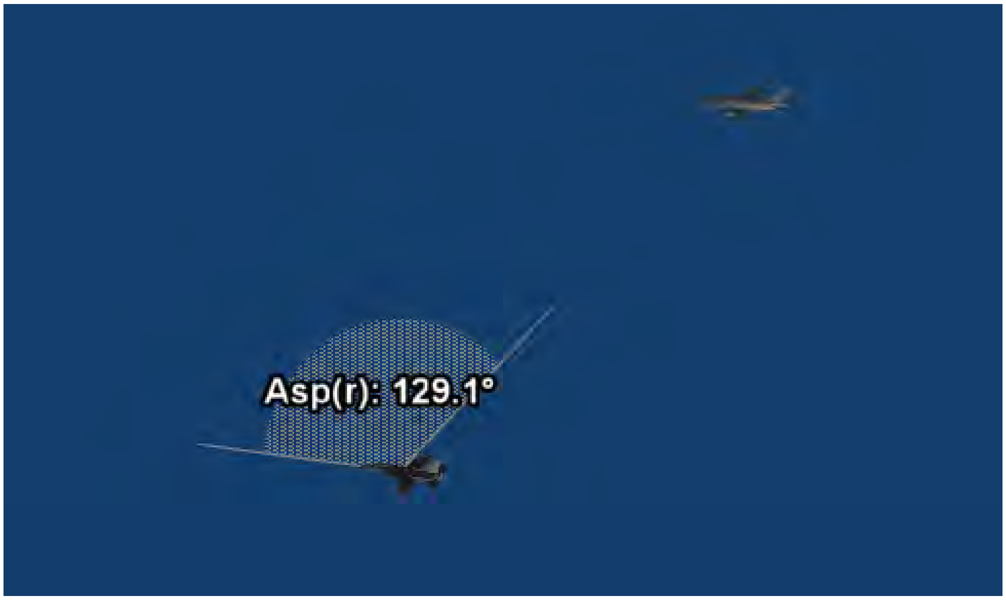


Figure E.23: Aspect Angle Calculation

E.4 Velocity Vector Relative Angles

E.4.1 Velocity Relative Azimuth | Az(v)

The **Velocity Relative Azimuth | Az(v)** calculation is measured along a tangent plane that is rotated to align with the velocity vector of the begin entity. The angle is a measurement between the heading axis of the plane and a point along the plane tangent to the end entity's location. This angle is measured in a positive sense according to the right-hand rule about the down vector perpendicular to the orientation plane. A relative azimuth of 0 degrees specifies a location directly in front of the entity and a relative azimuth of 180 degrees specifies a location directly behind the entity.

This calculation requires that the begin entity is a platform with valid velocity data. It does not support between beams, gates, lasers, or LOBs that do not share the same host platform.

Figure E.24: Velocity Relative Azimuth Calculation

E.4.2 Velocity Relative Elevation | El(v)

The **Velocity Relative Elevation | El(v)** calculation is measured along a tangent plane that is rotated to align with the velocity vector of the begin entity. The angle is measured between the plane and a point along the plane tangent to the end entity's location. The relative elevation angle is positive for entities above the plane and negative for entities below the plane.

This calculation requires that the begin entity is a platform with valid velocity data. It does not support between beams, gates, lasers, or LOBs that do not share the same host platform.

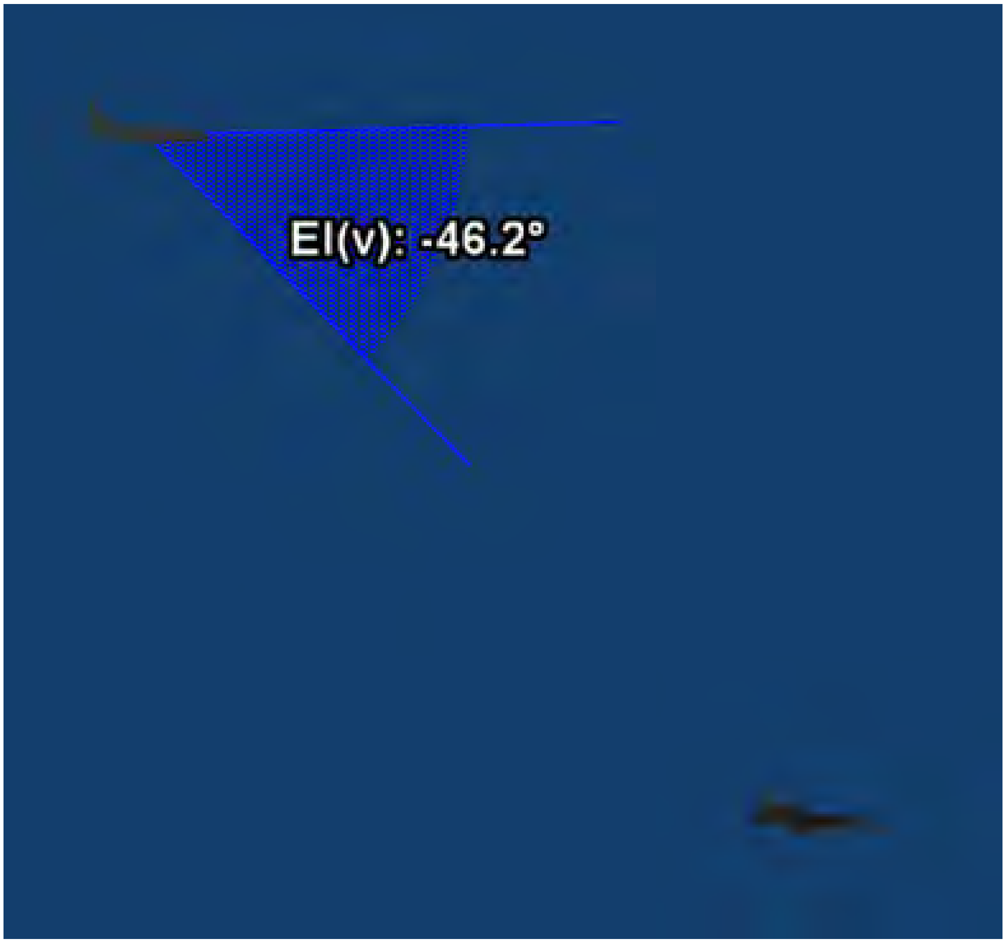


Figure E.25: Velocity Relative Elevation Calculation

E.4.3 Velocity Relative Composite Angle | Cmp(v)

The **Velocity Relative Composite Angle | Cmp(v)** is measured along a tangent plane that is defined by the inertial velocity vector of the begin entity and a position vector between the two entities. This angle is measured in a positive sense according to the right-hand rule about the down vector perpendicular to the defined plane.

This calculation requires that the begin entity is a platform with valid velocity data. It does not support between beams, gates, lasers, or LOBs that do not share the same host platform.

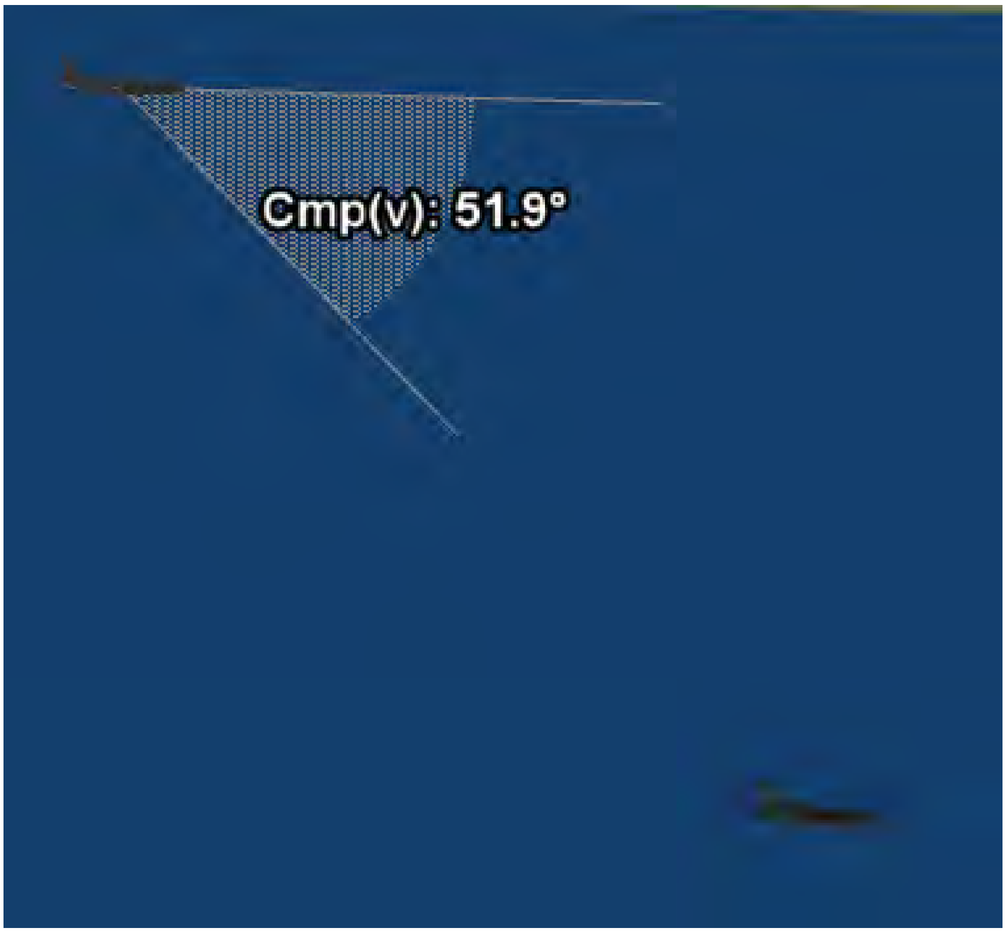


Figure E.26: Velocity Relative Composite Angle Calculation

E.5 Velocities

E.5.1 Closing Velocity | V(c)

The **Closing Velocity | V(c)** calculation is the velocity at which the begin entity and end entity are moving towards each other. Closing velocity is positive when the distance between two entities is decreasing (i.e. they are moving towards each other) and negative when the distance between two entities is increasing (i.e. they are moving away from each other). This calculation is only valid for begin and end entities that are platforms.

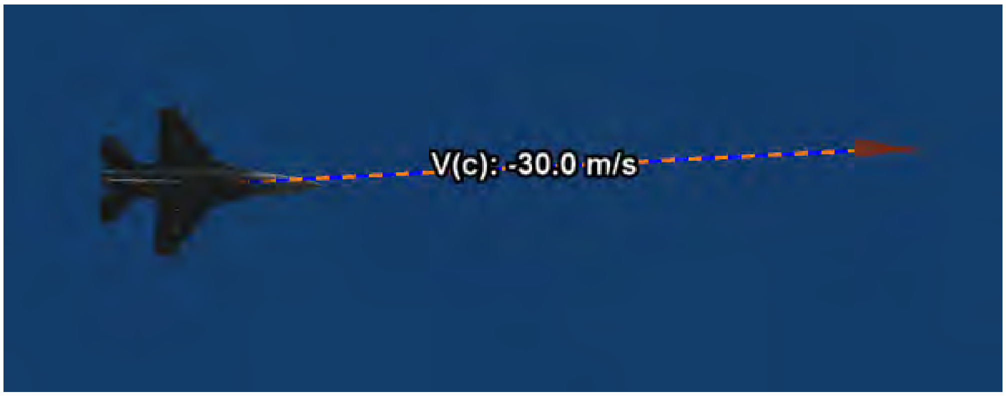


Figure E.27: Closing Velocity Calculation

E.5.2 Separation Velocity | V(s)

The **Separation Velocity | V(s)** calculation is the velocity at which the begin entity and end entity are moving away from each other. Separation velocity is positive when the distance between two entities is increasing (i.e. they are moving away from each other) and negative when the distance between two entities is decreasing (i.e. they are moving toward each other). This calculation is only valid for begin and end entities that are platforms.

Figure E.28: Separation Velocity Calculation

E.5.3 Velocity Delta | V(d)

The **Velocity Delta | V(d)** calculation is the difference in velocity magnitude between two entities. This calculation is only valid for begin and end entities that are platforms.

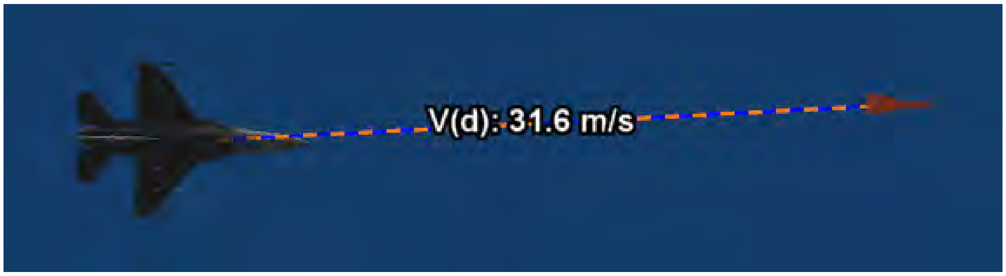


Figure E.29: Velocity Delta Calculation

E.6 Radio Frequency (RF)

E.6.1 Antenna Gain | Gain

The **Antenna Gain | Gain** calculation returns the antenna pattern gain in dB relative to the end entity. Antenna gain is a measure of an antenna's ability to detect entities, through the ability to amplify incoming Radar returns. The begin entity must be a beam. Furthermore the beam must have either an antenna pattern or an antenna pattern algorithm associated with it. Without either of these, this calculation is unavailable.

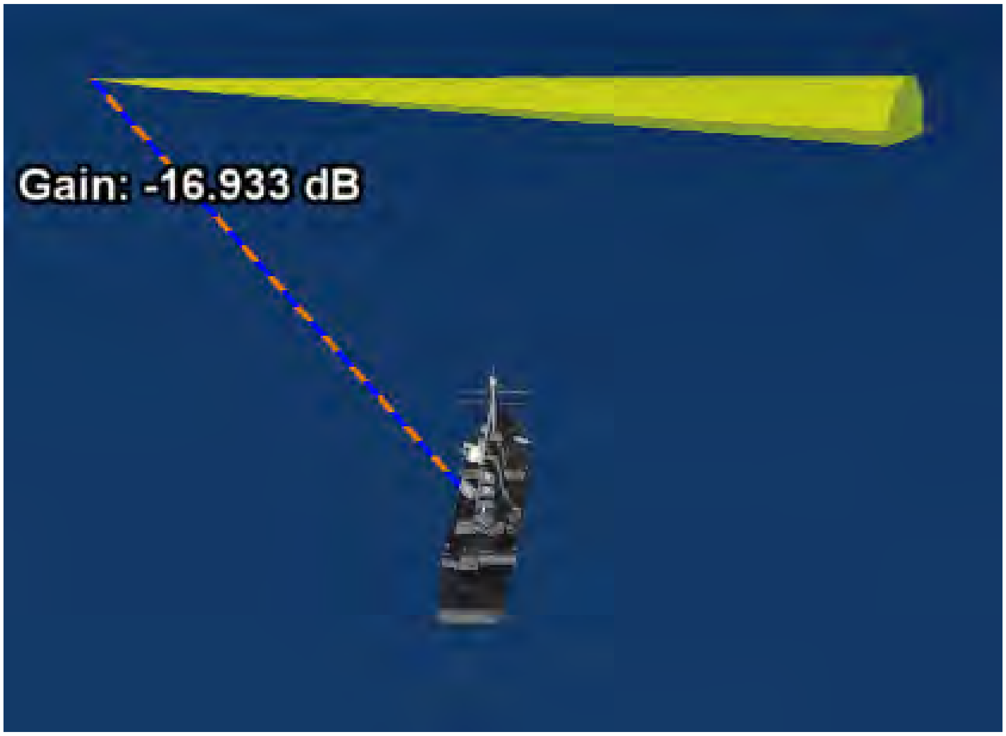


Figure E.30: Antenna Gain Calculation

E.6.2 Received Power | Pwr

The **Received Power | Pwr** calculation returns the two-way power density received at an antenna in dBW. The begin entity must be a beam and must have either an antenna pattern file or an antenna pattern algorithm associated with it. This calculation also takes RCS into consideration. If the end entity does not have an RCS file associated with it, then a target RCS of 1 square meter is assumed. If propagation data is available, the Radar range equation utilizing propagation factor is used from `L.V. Blake (Radar Range Performance Analysis, Lexington Books, D.C. Heath and Co., Lexington, MA, 1980)`. If propagation data is unavailable, the free space Radar range equation is used.

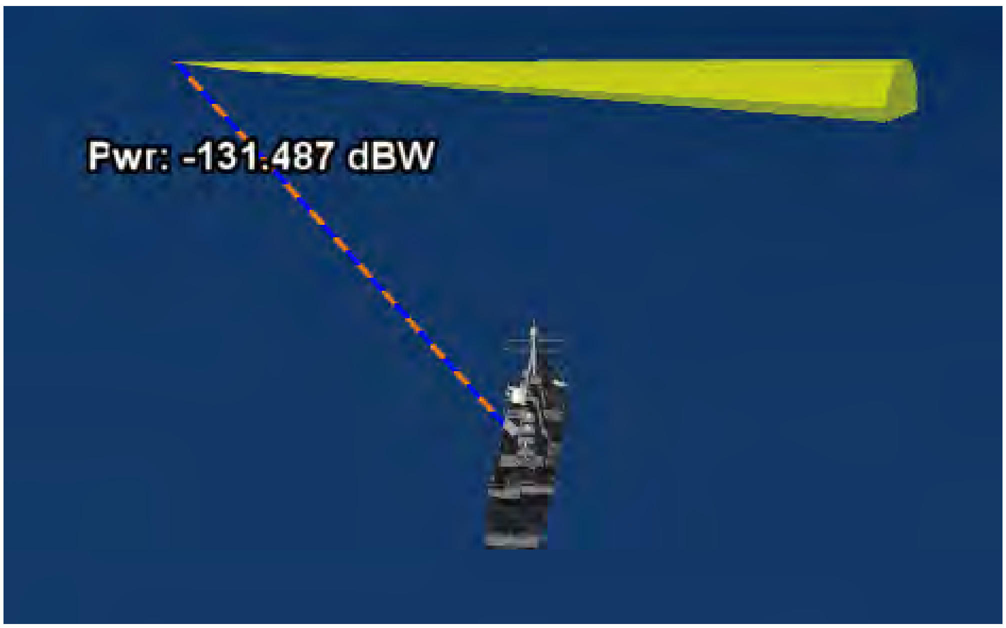


Figure E.31: Received Power Calculation

E.6.3 One-Way Power | Pwr(1)

The **One-Way Power | Pwr(1)** calculation returns the one-way power density received at an antenna in dBw. The begin entity must be a beam and must have either an antenna pattern file or an antenna pattern algorithm associated with it. If propagation data is available, the Radar range equation utilizing propagation factor is used from `L.V. Blake (Radar Range Performance Analysis, Lexington Books, D.C. Heath and Co., Lexington, MA, 1980)`. If propagation data is unavailable, the free space Radar range equation is used.

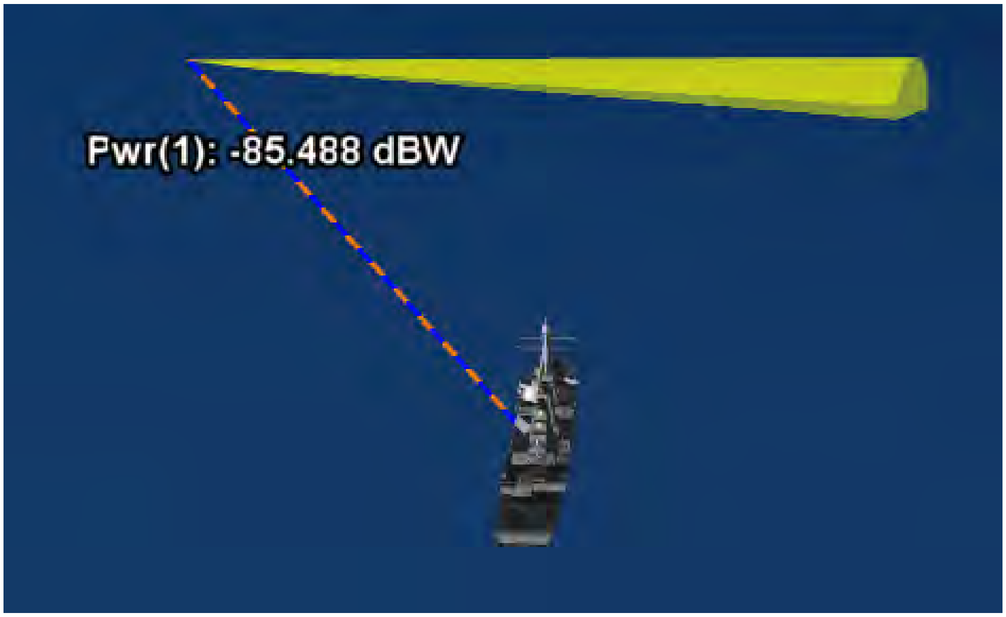


Figure E.32: One-Way Power Calculation

E.6.4 Radar Cross Section | RCS

The **Radar Cross Section | RCS** calculation returns the RCS of a target as seen by begin entity sensor in dBsm. The end entity must have an RCS file associated with it and the begin entity must be a beam, otherwise, this calculation is unavailable. RCS files are organized into hierarchical containers stored under polarity, frequency, and elevation. A given polarity can have one or more frequencies associated with it. A given frequency can have one or more elevation values, and an elevation can have one or more data pairings of azimuth and RCS values. Polarity and frequency values used in the RCS pattern calculation are obtained from the begin entity beam. If the beam's polarity is not found in the RCS file, -300 dB is returned. If an Unknown polarization is specified by the beam, the RCS pattern will use the first polarization encountered in the data structure based on the following order:

1. Horizontal
2. Vertical
3. Circular
4. HV (Horizontal XMT, Vertical RCV)
5. VH (Vertical XMT, Horizontal RCV)
6. Left Circular
7. Right Circular
8. Linear

Frequency uses a nearest neighbor lookup. Azimuth and elevation values are interpolated.

Figure E.33: Radar Cross Section (RCS) Calculation

E.6.5 Radio Horizon | Hor(r)

The **Radio Horizon | Hor(r)** calculation determines whether the entity is above or below the radio horizon of the begin entity. If terrain altitude sets are loaded, terrain masking will be taken into account. This calculation does not consider atmospheric effects.

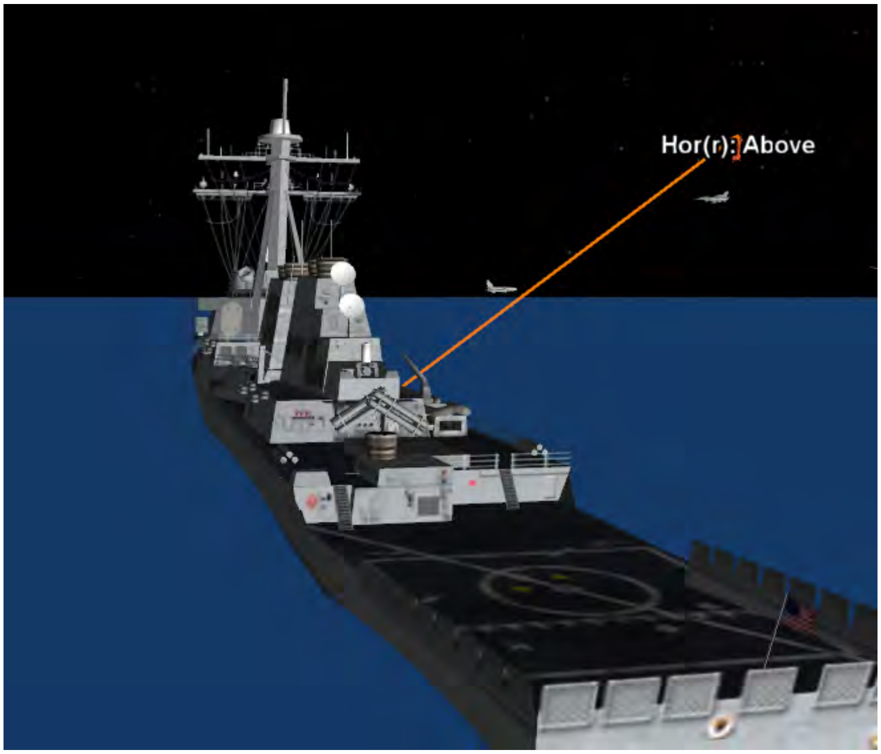


Figure E.34: Radar Horizon Calculation

NOTE: The Line of Sight (LOS) and RF Horizon calculations used in the Range Tool are performed in a 2D plane defined by the locations of the two selected entities. Platform orientation is not considered as the calculation computes a line-of-sight angle based on the difference in altitude of the two entities. The result of this calculation may not agree with the Data Browser's altitude read-out due to differences in height between the selected platforms.

E.6.6 Optical Horizon | Hor(o)

The **Optical Horizon | Hor(o)** calculation determines whether the entity is above or below the optical horizon of the begin entity. If terrain altitude sets are loaded, terrain masking will be taken into account.

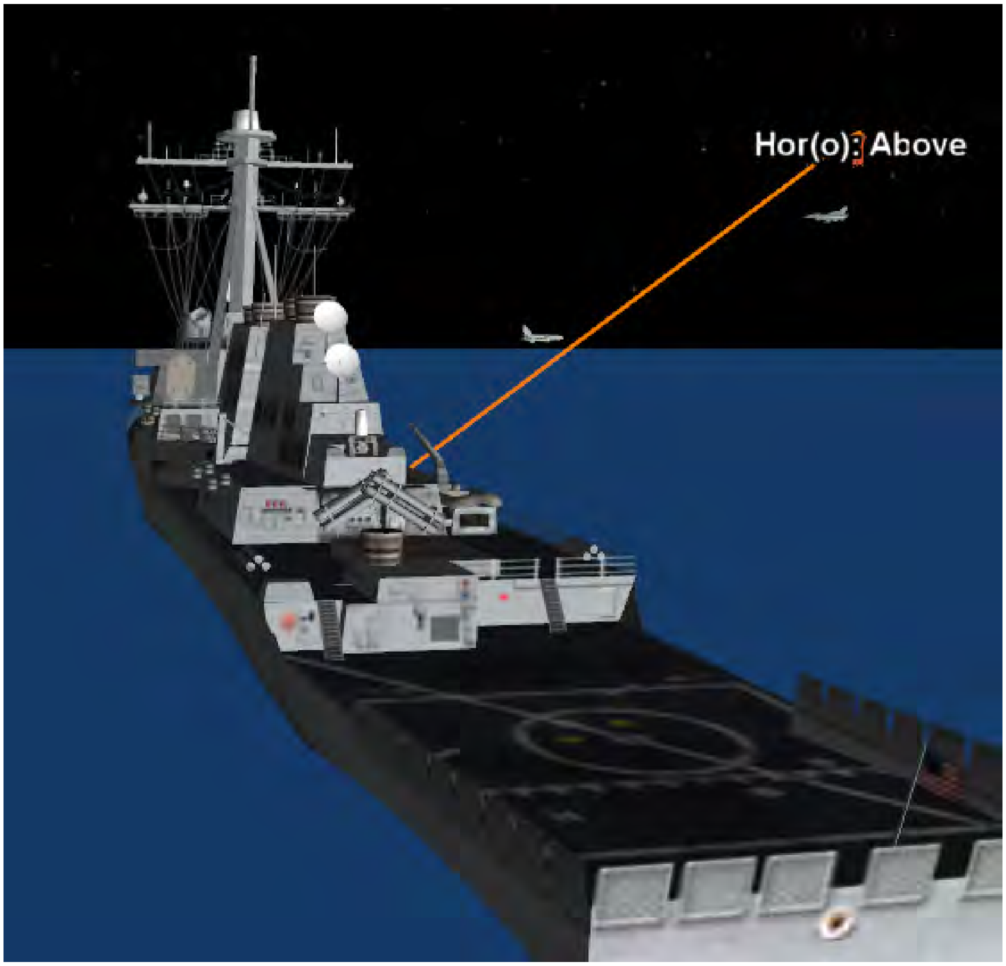


Figure E.35: Optical Horizon Calculation

NOTE: The Line of Sight (LOS) and Optical Horizon calculations used in the Range Tool are performed in a 2D plane defined by the locations of the two selected entities. Platform orientation is not considered as the calculation computes a line-of-sight angle based on the difference in altitude of the two entities. The result of this calculation may not agree with the Data Browser's altitude read-out due to differences in height between the selected platforms.

E.7 Propagation

E.7.1 Probability of Detection | POD

The **Probability of Detection | POD** calculation determines the ability of a Radar to detect targets based on specified Radar system parameters such as transmitter power, frequency, pulse length, noise factor, and system loss. This calculation requires propagation loss data, if the data is not found, the calculation will not be available.

SIMDIS natively supports the Advanced Refractive Effects Prediction System (AREPS). AREPS allows for the calculation of a detection threshold based upon coherent integration, incoherent integration, or no integration (simple). A simple type calculation is normally used for rotating, pulsed Radar that uses non-coherent pulse integration to increase its sensitivity. The signal- to-noise ratio required for a given probability of detection and false alarm rate is known as either the visibility factor or the detectability factor. Shown in Figure E.36 is the RF Propagation Tool dialog.

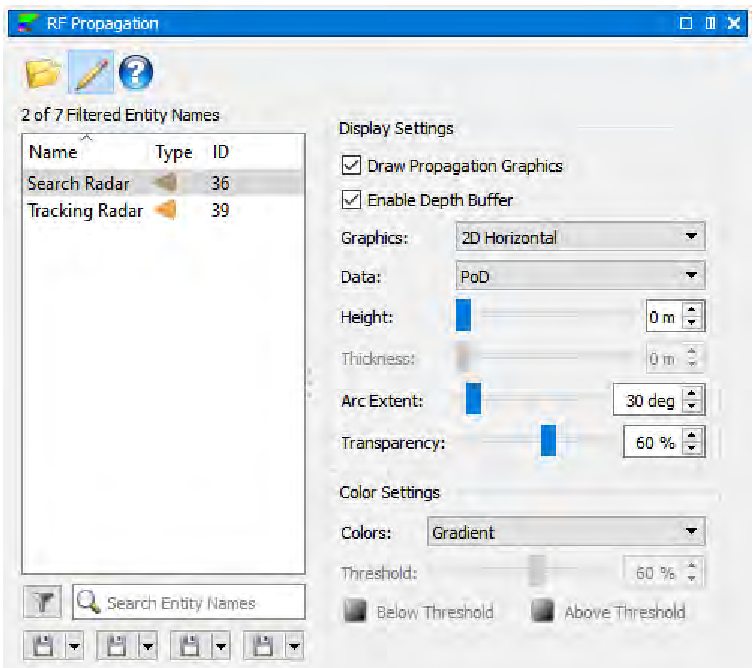


Figure E.36: RF Propagation Tool - PoD

Figure E.37: Probability of Detection (POD) Calculation

E.7.2 Propagation Loss | Loss

The **Propagation Loss | Loss** calculation is the ratio, expressed in dB, of the effective radiated power transmitted in the direction of maximum radiation of the antenna pattern to the power received at any point by an omni-directional antenna. This calculation determines the propagation loss between the Radar (beam) and another entity. This calculation requires propagation loss data, if the data is not found, the calculation will not be available.

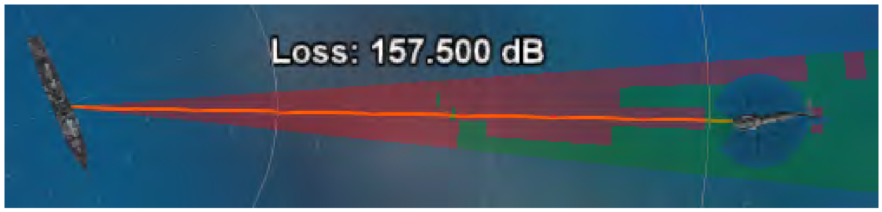


Figure E.38: Propagation Loss Calculation

E.7.3 Pattern Propagation Factor | PPF

The **Pattern Propagation Factor | PPF** calculation is the ratio, expressed in dB, of the actual field strength at a point to the field strength that would occur at the same range in free-space in the direction of maximum radiation. This calculation determines the Pattern Propagation Factor between a Radar (beam) and another entity. This calculation requires propagation pattern factor (PPF) data, if the data is not found, the calculation will not be available.

Figure E.39: Pattern Propagation Factor (PPF) Calculation

E.7.4 Signal to Noise | SNR

The **Signal-to-Noise | SNR** calculation is the ratio, expressed in dB, of the signal received at the input of the Radar receiver to the noise generated within the receiver itself. This calculation determines the SNR between a Radar (beam) and another entity. This calculation also takes RCS into consideration. If the end entity does not have a RCS file associated with it, then a target RCS of 1 square meter is assumed. This calculation requires propagation pattern factor (PPF) data, if the data is not found, the calculation will not be available.

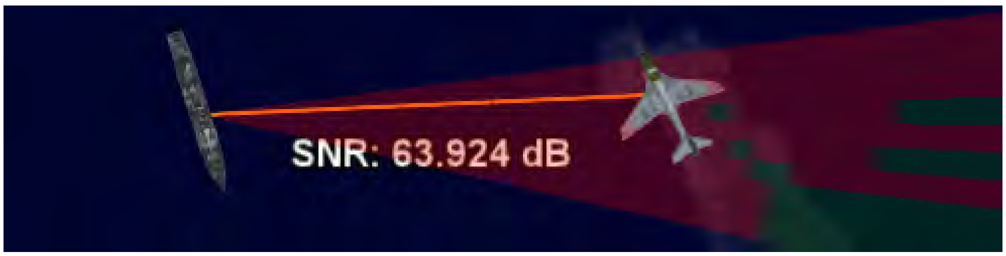


Figure E.40: Signal-to-Noise Ratio (SNR) Calculation

E.7.5 Clutter-to-Noise | CNR

The **Clutter-to-Noise** (**CNR**) calculation is the ratio, expressed in dB, of the clutter power received at the input of the Radar receiver to the noise generated within the receiver itself. This calculation determines the CNR for area clutter between a Radar (beam) and another entity. This calculation requires clutter to noise (CNR) data, if the data is not found, the calculation will not be available.

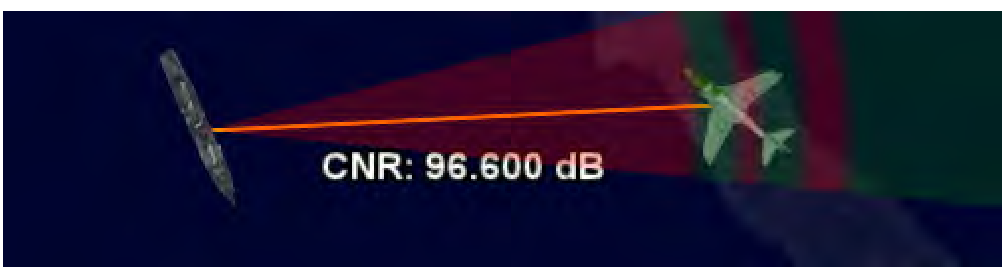


Figure E.41: Clutter-to-Noise Ratio (CNR) Calculation

E.8 Units

SIMDIS allows you to change the unit format for Range Tool display. You can configure the display unit output of calculations or match the global settings.

E.8.1 Time Format

Seconds	Shown as seconds since the beginning of the reference year.
Minutes	Shown as minutes and seconds since the beginning of the hour.
Hours	Shown as hours, minutes and seconds since the beginning of the day.
Ordinal	The number of days since the beginning of the year, where 1 is the first day of the year and 365 is the last day of a non-leap year, 366 during a leap year. Example: 242 21:30:10.000
Month Day Year	Shown as week day month, month day, hours, minutes, seconds and year. Example: Wed Aug 30 21:30:10.000 2006
Date Time Group	Military time shown as days, hours, minutes, seconds, time zone, month and year. Example: 302130:10.000 Z Aug06

E.8.2 Geodesic Units

Degrees	Formats latitude and longitude to a degrees decimal notation.
Degrees Minutes	Formats latitude and longitude to a degrees minutes decimal notation
Degrees Minutes Seconds	Formats latitude and longitude to a degrees, minutes, and seconds notation

E.8.3 Distance Units

Yards (yds)
Meters (m)
Kiloyards (kyds)
Kilometers (km)
Miles
Nautical Miles (NM or nmi)
Millimeters (mm)
Centimeters (cm)
Inches (in)
Feet (ft)
Fathoms (ftm)
Kilofeet (kft)
Data Miles

E.8.4 Altitude Units

Feet (ft)
Yards (yds)
Meters (m)
Kilofeet (kft)
Millimeters (mm)
Centimeters (cm)
Inches (in)
Fathoms (ftm)
Kiloyards (kyds)
Kilometers (km)
Miles
Data Miles
Nautical Miles (NM or nmi)

E.8.5 Angle Units

Degrees	Degrees unit
Radians	Radians unit
Angular Mil	NATO angular mil unit, where an angle is between 0 and 3600
Milliradians	One thousandth of a radian angular unit
Binary Angular Measure (BAM)	Where an angle is between 0 and 1

E.8.6 Speed Units

Meters per Second (m/s or mps)
Miles per Hour (Mph)
Knots (kts)
Feet per Second (ft/s)
Yards per Second (yds/s)
Kilometers per Second (km/s)
Kilometers per Hour (km/h or kmph)
Data Miles per Hour

E.8.7 Coordinate System

LLA	geodesic Degrees Decimal
ECEF	Geocentric Earth Centered Earth Fixed
X-East	Tangent plane where X is East and Y is North
ENU	East North Up (Scaled Flat Earth)
NED	North East Down (Scaled Flat Earth)
NWU	North West Up (Scaled Flat Earth)
Generic	Generic tangent plane
ECI	Geocentric Earth Centered Inertial

E.8.8 Magnetic Variance

True Angles	No variance, yaw referenced to true north. This setting is the default SIMDIS representation.
Magnetic Angles	Variance based on World Magnetic Model, selection denoted in the label display by the "YM" prefix.
User Defined	User specified variance, selection denoted in the label display by the "YU" prefix.

E.8.9 Vertical Datum

Height Above Ellipsoid	Surface referenced to the WGS-84 ellipsoid.
Mean Sea Level	Mean sea level surface based on the NGA EGM.
User Defined	User specified vertical datum, in meters.

This Page is Intentionally Left Blank

Appendix F

Bookmarks Output Format

F.1 Execute Action Event

Execute a hot key action by name.

Tag: `__SIMDIS::EXECUTE_ACTION`
Format: <Action>

Action	(required)	Action to execute, using the name from Hot Keys dialog.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::EXECUTE_ACTION" "\"Center Origin\""
Bookmark "001 2000 00:00:01" "__SIMDIS::EXECUTE_ACTION" "\"Full Screen\""
```

F.2 Eye Position Event

Transition the view to a saved eye position.

Tag: `__SIMDIS::VIEW_CHANGE`
Format: <Eye Pos> [Duration] [Viewport]

Eye Pos	(required)	Name of saved eye position.
Duration	(optional)	Duration in seconds for transition.
Viewport	(optional)	Name of viewport to modify.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::VIEW_CHANGE" "\"OVH\" 0"
Bookmark "001 2000 00:00:01" "__SIMDIS::VIEW_CHANGE" "\"F-16_OVH\" 2.5 \"target\""
```

F.3 GOG Event

Set visibility of a loaded GOG file.

Tag: `__SIMDIS::GOG_CHANGE`
Format: <GOG Name> [Platform] [Show Flag]

GOG Name	(required)	Name portion of a GOG to change visibility.
Platform	(optional)	Name of host platform for attached GOG, or empty string if none.
Show Flag	(optional)	Show (1) or hide (0) the GOG. If not specified, defaults to 0.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::GOG_CHANGE" "\"us_states.gog\" \"\" 1"
Bookmark "001 2000 00:00:01" "__SIMDIS::GOG_CHANGE" "\"circle.rxy\" \"aegis\" 0"
```

F.4 Highlight Platform Event

Display highlight for a given platform.

Tag: `__SIMDIS::HIGHLIGHT_CHANGE`
Format: <Platform> [Color] [Duration] [Wall Clock] [Always Show]

Platform	(required)	Name of the platform to use.
Color	(optional)	Specified in hex as 0xAABBGGRR.
Duration	(optional)	Seconds that the highlight will remain on-screen.
Wall Clock	(optional)	Use data time (0) or wall clock (1) for interpretation of duration.
Always Show	(optional)	Use (0) or ignore (1) the duration field.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::HIGHLIGHT_CHANGE"
"\"Radar Site 2\" 0xff00ffff 5 0 0"
```

F.5 Inset Viewport Visibility Event

Set visibility of an inset viewport.

Tag: `__SIMDIS::INSET_CHANGE`
Format: <Inset> [Show Flag]

Inset	(required)	Name of the inset to modify.
Show Flag	(optional)	Show (1) or hide (0) the inset. If not specified, defaults to 0.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::INSET_CHANGE" "\"target\" 0"
Bookmark "001 2000 00:00:01" "__SIMDIS::INSET_CHANGE" "\"target\" 1"
```

F.6 Preference Rule: Load/Append Event

Loads preference rules, either replacing or appending to existing rules.

Tag: `__SIMDIS::PREF_RULE_CHANGE`
Format: <Filename> <Load Type>

Filename	(required)	Rule file to load or append.
Load Type	(required)	Use "Load" to load the rule file, clearing out old values. Use "Append" to append the rule file to existing rules.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::PREF_RULE_CHANGE" "\"DisplayLabels.rul\" Load"
Bookmark "001 2000 00:00:01" "__SIMDIS::PREF_RULE_CHANGE" "\"MoreRules.rul\" Append"
```

F.7 Preference Rule: Clear Event

Clears all the loaded preference rules.

Tag: `__SIMDIS::PREF_RULE_CHANGE`
Format: <Clear>

Clear	(required)	The value "Clear"

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::PREF_RULE_CHANGE" "Clear"
```

F.8 Screen Image Event

Display an image on the HUD for a given duration.

Tag: `__SIMDIS::IMAGE_CHANGE`
Format: <Image> <X> <Y> <Percent> <Width> <Height> <Duration> <Wall Clock> [Always Show] [H-Align] [V-Align] [Orig Size]

Image	(required)	Filename of the image to display.
X	(required)	X position of the image from left side of screen.
Y	(required)	Y position of the image from bottom side of screen.
Percent	(required)	Use pixel (0) or percent (1) values for position and width and height. Percentage is from 0-100.
Width	(required)	Width of the image.
Height	(required)	Height of the image.
Duration	(required)	Seconds that the image will remain on-screen.
Wall Clock	(required)	Use data time (0) or wall clock (1) for interpretation of duration.
Always Show	(optional)	Use (0) or ignore (1) the duration field.
H-Align	(optional)	L|C|R - horizontal alignment setting.
V-Align	(optional)	B|C|T - vertical alignment setting.
Orig Size	(optional)	Use provided width and height values (0) or force original native image size (1).

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::IMAGE_CHANGE"
"\"compass.png\" 90 5 1 10 10 5 1 0 L B 0"
Bookmark "001 2000 00:00:01" "__SIMDIS::IMAGE_CHANGE"
"\"windVane.png\" 0 5 1 10 10 10 1 0 L B 0"
```

F.9 Screen Text Event

Display text on the HUD for a given duration.

Tag: `__SIMDIS::SCREEN_TEXT_CHANGE`
Format: <Text> [Font] [Size] [Color] [X] [Y] [Duration] [Wall Clock] [H-Align] [V-Align]

Text	(required)	Text to display on the HUD.
Font	(optional)	Name of the font to use.
Size	(optional)	Point size of the text in the given font.
Color	(optional)	Specified in hex as 0xAABBGGRR.
X	(optional)	X position of the text in percentage (0-100) from left side of screen.
Y	(optional)	Y position of the text in percentage (0-100) from bottom side of screen.
Duration	(optional)	Seconds that the text will remain on-screen.
Wall Clock	(optional)	Use data time (0) or wall clock (1) for interpretation of duration.
H-Align	(optional)	L|C|R - horizontal alignment setting.
V-Align	(optional)	B|C|T - vertical alignment setting.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::SCREEN_TEXT_CHANGE"
"\"Bookmark Demo\" \"impact.ttf\" 28 0xff00ffff 50 8 0 0 C T"
Bookmark "001 2000 00:00:01" "__SIMDIS::SCREEN_TEXT_CHANGE"
"\"F-16 Tops off at Tanker\" \"arial.ttf\" 20 0xffffffff 50 90 5 1 C T"
```

F.10 Settings File Event

Load a settings file.

Tag: `__SIMDIS::PREF_FILE_CHANGE`
Format: <Filename>

Filename	(required)	Name of the settings file to load.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::PREF_FILE_CHANGE" "\"BMD_Demo.ini\""
```

F.11 Time: Change Rate Event

Change the step-time or real-time rate multiplier.

Tag: `__SIMDIS::TIME`
Format: <Rate:> <Value>

Rate:	(required)	Use the string "Rate:" here to indicate the change rate sub-function.
Value	(required)	Time step or real-time rate to set.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::TIME" "Rate: 5.5"
Bookmark "001 2000 00:00:01" "__SIMDIS::TIME" "Rate: 1.0"
```

F.12 Time: Jump To Event

Jump to the specified scenario time.

Tag: `__SIMDIS::TIME`
Format: <Jump To:> <JDay> <Year> <Time>

Jump To:	(required)	Use the string "Jump To:" here to indicate the jump to sub-function.
JDay	(required)	Julian day for the scenario time.
Year	(required)	Year of validity for the Julian day.
Time	(required)	Scenario time string in format HH:MM:SS[.sss]

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::TIME" "Jump To: 001 2000 00:00:00"
Bookmark "001 2000 00:00:01" "__SIMDIS::TIME" "Jump To: 001 2000 00:00:05"
```

F.13 Time: Pause Event

Stop playing at the current time for the specified number of seconds. The "Pause" is always specified in "Wall" time.

Tag: `__SIMDIS::TIME`
Format: <Pause:> <Seconds>

Pause:	(required)	Use the string "Pause:" here to indicate the pause sub-function.
Seconds	(required)	Positive whole-number seconds to pause playback before resuming.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::TIME" "Pause: 2"
Bookmark "001 2000 00:00:01" "__SIMDIS::TIME" "Pause: 5"
```

F.14 Time: Play Event

Start playing at the current time.

Tag: `__SIMDIS::TIME`
Format: <Play>

Play	(required)	Use the string "Play" here to indicate the play sub-function.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::TIME" "Play"
```

F.15 Time: Real-Time Event

Toggle the real-time vs step-time flag.

Tag: `__SIMDIS::TIME`
Format: <Real Time:> <Flag>

Real Time:	(required)	Use the string "Real Time:" here to indicate the real-time sub-function.
Flag	(required)	If "True", then set to real-time; if "False" then set to step-time.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::TIME" "Real Time: True"
Bookmark "001 2000 00:00:01" "__SIMDIS::TIME" "Real Time: False"
```

F.16 Time: Stop Event

Stop playing at the current time.

Tag: `__SIMDIS::TIME`
Format: <Stop>

Stop	(required)	Use the string "Stop" here to indicate the stop sub-function.

Examples:

```
Bookmark "001 2000 00:00:00" "__SIMDIS::TIME" "Stop"
```

Appendix G

SIMDIS Regular Expression

Regular expressions (**regex** or **regexp**) are used to identify the text of interest such as particular characters, words, or patterns of characters. The matched text returned by the regular expression is then used by SIMDIS to filter the list of entities in **Prefs Tool** and **Data Browser**. Only the matched name of the entities will be shown in the list. To learn more about Regular Expressions refer to https://regex101.com/.

NOTE: Use escape character or backslash (\) with special characters to treat them as characters instead of the regular expression symbols (e.g. `\(, \), \[, \]`).

G.0.1 Examples

[abc]	Matches single character of **a**, **b**, or **c**. Examples: ▪ **[x]** with **[xray, regex, apple]** will match **[xray, regex]**. ▪ **[mlk]** with **[missile, milk, key, pod]** will match **[missile, milk, key]**.
[^abc]	Matches a character except **a**, **b**, or **c**. Examples: ▪ **[^x]** with **[Falcon, Fox, Wolf]** will match **[Falcon, Wolf]**. ▪ **[^abc]** with **[Falcon, Fox, Wolf]** will match **[Fox, Wolf]**.
[a-c]	Matches a character in the range of **a** to **c**. Examples: ▪ **[x-z]** with **[Mix, Mod, Maize]** will match **[Mix, Maize]**. ▪ **[0-2]** with **[F-18, F-22, F-35]** will match **[F-18, F-22]**. ▪ **[A-Z]** with **[centipede, 1337, AA-12]** will match **[AA-12]**.

.	The period (**.**) matches any single character. Examples: ▪ **j.l** with **[jack, jill, julian]** will match **[jill, julian]**. ▪ **F.1** with **[F-16, F-18, F-35]** will match **[F-16, F-18]**.
\s	Matches any whitespace character. Example: ▪ \s with **[Black Bear, Brown Fox, Silver_Wolf]** will match **[Black Bear, Brown Fox]**.
\S	Matches any non-whitespace character. Example: ▪ **X\S** with **[X 1, X 2, X(1)]** will match **[X(1)]**.
\d	Matches any digit or a number (**[0-9]**). Example: ▪ **P\d** with **[P3, P52, Pax1, PS4]** will match **[P3, P52]**.
\D	Matches any non-digit (**[^0-9]**). Example: ▪ **P\D** with **[P3, P52, Pax1, PS4]** will match **[Pax1, PS4]**.
\w	Matches any word character (**[A-Za-z0-9]**). Example: ▪ **P\w** with **[P-51, P-47, Proctor, Point]** will match **[Proctor, Point]**.
\W	Matches any non-word character (**[^A-Za-z0-9]**). Example: ▪ **P\W** with **[P-51, P-47, Proctor, Point]** will match **[P-51, P-47]**.
(...)	Match enclosed character set. Examples: ▪ **(SBN)** with **[SSN-21, SSN-754, SSBN-736, SSGN-726]** will match **[SSBN-736]**. ▪ **(F-16)** with **[F-15 1, F-16 1, F-16 2, F-18 1]** will match **[F-16 1, F-16 2]**.
a?	Matches zero or one of **a**. Examples: ▪ **C?at** with **[Cat, Bat, Can, Ban]** will match **[Cat, Bat]**. ▪ **DDG?** with **[DD-991, DDG-1000, DDG-114, CVN-76]** will match **[DD-991, DDG-1000, DDG-114]**.
a*	Matches zero or more of **a**. Examples: ▪ **Pas*** with **[Pass, Past, Patton, Post]** will match **[Pass, Past, Patton]**. ▪ **S*N** with **[SSBN, SSN, SSK, CVN-78]** will match **[SSBN, SSN, CVN-78]**.

^	Matches start of the string. Examples: ▪ **^IFF** with **[IFF 136, GIFF-2, IFFx1, 28 IFF]** will match **[IFF 136, IFFx1]**. ▪ **^F-1** with **[F-16, F-18, F-22, F-35]** will match **[F-16, F-18]**.
$	Matches end of the string. Examples: ▪ **Abrams$** with **[M1A2 Abrams, M1A3 Abrams, M142 HIMARS]** will match **[M1A2 Abrams, M1A3 Abrams]**. ▪ **18$** with **[F/A-18, EA-18, E-3, EP-3]** will match **[F/A-18, EA-18]**.
a|b	Matches either **a** or **b**. Examples: ▪ **EA|EF** with **[F/A-18, EA-18G, EF-11A, EC-1]** will match **[EA-18G, EF-11A]**. ▪ **CG|DG** with **[DDG-921, CG-52, DD-321, CFS-6]** will match **[DDG-921, CG-52]**.
a+	Matches one or more of **a**. Examples: ▪ **Pas+** with **[Pass, Past, Patton, Post]** will match **[Pass, Past]**. ▪ **S+N** with **[SSN-774, SSN-23, SSGN-726]** will match **[SSN-774, SSN-23]**.
a{3}	Matches three repetitions of **a**. Examples: ▪ **D{2}** with **[DDG-72, DS-12, MDXD-13, MDD-13X]** will match **[DDG-72, MDD-13X]**. ▪ **0{3}** with **[DDG-1000, X 700, B-50, HAL 9000]** will match **[DDG-1000, HAL 9000]**.
a{3,}	Matches three or more repetitions of **a**. Examples: ▪ **S{2,}** with **[SSS-800, SSGN-726, AGSS-555, SNS-1]** will match **[SSS-800, SSGN-726, AGSS-555]**. ▪ **m{1,}** with **[120 mm, 4.13 in, AGM-114, 130m]** will match **[120 mm, 130m]**.
a{3,6}	Matches three to six repetitions of **a**. Examples: ▪ **m{2,3}** with **[120 mm, 4.13 in, 105-mmm, 130m]** will match **[120 mm, 105-mmm]**. ▪ **0{1,2}** with **[DDG-1000, X 700, B-50, HAL 9000]** will match **[X 700, B-50]**.

Appendix H

Third Party Components

SIMDIS includes various third party components with the toolset. Each component included in SIMDIS is listed below along with the license it is distributed with and a website for more information.

Audio File Library (v0.3.6)
License: LGPL (Lesser GNU Public License)
Website: https://audiofile.68k.org/
Boost (v1.69.0)
License: Boost Software License Version 1.0
Website: http://www.boost.org/
cURL Library (v7.68.0)
License: MIT/X Derivate License
Website: https://curl.haxx.se/
Expat (v2.2.7)
License: Expat License ('As-Is')
Website: http://expat.sourceforge.net/
FFMPEG (v4.1.4)
License: LGPL
Website: http://ffmpeg.org/legal.html
FOX GUI Toolkit (v1.6.50)
License: LGPL
Website: http://www.fox-toolkit.org
FreeType (v2.10.1)
License: FreeType Project License (BSD style license)
Website: http://freetype.sourceforge.net/index2.html
FTGL (v2.1.2)
License: LGPL
Website: http://homepages.paradise.net.nz/henryj/code/index.html#FTGL
GDAL (v2.4.4)
License: MIT/X11 License
Website: http://www.gdal.org/
GeographicLib (v1.18)
License: MIT/X11 License
Website: https://geographiclib.sourceforge.io/
GIFLIB (v5.2.1)
License: Open Source License
Website: http://giflib.sourceforge.net/
GLEW (v1.8.0)

License: Modified BSD License
Website: http://glew.sourceforge.net/
HDF5 (v1.10.5)
License: BSD-style License
Website: https://www.hdfgroup.org/
JPEG (v9d)
License: IJG JPEG License (BSD style license)
Website: http://www.ijg.org/
MrSID (v9.5.4.4709)
License: LizardTech License
Website: https://www.lizardtech.com
OpenAL (v1.15.1)
License: LGPL
Website: https://www.openal.org/
OpenSceneGraph (v3.6.4)
License: LGPL
Website: http://www.openscenegraph.org/
OpenSSL (v1.1.1d)
License: OpenSSL License
Website: https://www.openssl.org/source/license.html
osgEarth (v3.0)
License: LGPL
Website: http://osgearth.org/
osgQt (v145)
License: LGPL
Website: https://github.com/openscenegraph/osgQt
PCRE (v8.43)
License: BSD-style License
Website: https://www.pcre.org/
PNG (v1.6.37)
License: Open Source License
Website: http://www.libpng.org/pub/png/libpng.html

Proj4 (v5.2.0)
License: MIT License
Website: http://proj4.org/
Protocol Buffers (v2.6.0)
License: New BSD License
Website: https://code.google.com/p/protobuf/
Python (v3.8.2)
License: Python License
Website: http://docs.python.org/license.html
Qt (v5.9.8)
License: LGPL
Website: https://www.qt.io/
SunDog SilverLining (v5.052)
License: Commercial License
Website: http://sundog-soft.com/features/real-time-3d-clouds/
SunDog Triton (v4.24)
License: Commercial License
Website: http://sundog-soft.com/features/ocean-and-water-rendering-with-triton/
SQLite3 (v3.31.1)
License: Open Source
Website: https://www.sqlite.org/copyright.html
SWIG (v4.0.0)
License: GPL License
Website: http://www.swig.org/
TBB (v4.4.2015)
License: Apache 2.0
Website: https://www.threadingbuildingblocks.org
TIFF (v4.1.0)
License: TIFF Library License
Website: http://www.libtiff.org
Wild Magic Library (v1.08)
License: Boost Software License Version 1.0
Website: http://www.geometrictools.com/

ZeroMQ (v4.0.9)
License: LGPL
Website: http://zeromq.org/

ZLib (v1.2.11)
License: 'As-Is'
Website: https://zlib.net/

This Page is Intentionally Left Blank

Appendix I

Acronym Glossary

2D	Two-Dimension
3D	Three-Dimension
AAR	After Action Report
AGP	Accelerated Graphic Port
ANSI	American National Standards Institute
API	Application Programming Interface
APM	Advance Propagation Model
AREPS	Advanced Refractive Effects Prediction System
ASCII	American Standard Code for Information Interchange
ASI	ASCII Scenario Input File Format
AUV	Autonomous Underwater Vehicle
AVI	Audio Video Interleave
BMD	Ballistic Missile Defense
CD	Compact Disc
CentOS	Community Enterprise Operating System
CFMWC	Canadian Forces Maritime Warfare Centre
CG	Guided-Missile Cruiser
CGIAR	Consultative Group for International Agricultural Research
CNR	Clutter-to-Noise Ratio
COMOPTEVFOR	Commander Operational Test and Evaluation Force
CONUS	Contiguous United States
COTS	Commercial off-the-Shelf
CPA	Closest Point of Approach
CPU	Central Processing Unit
CSCSQ	Cosecant Squared Antenna Pattern Model
CTEIP	Central Test and Evaluation Investment Program
DB	Database
DCM	Direction Cosine Matrix
DCS	Data Client Server
DDF	Data Descriptive Files
DDG	Guided-Missile Destroyer
DDR	Double Data Rate
DED	Digital Elevation Data
DIR	Directory
DIS	Distributed Interactive Simulation

DISCN	SIMDIS Data Initialization Scenario
DLL	Dynamic Link Library
DMAP	Digital Mapping Charting and Geodesy Analysis Program
DOD	Department of Defense
DOF	Degrees of Freedom
DT/OT	Developmental and Operational Testing
DVD	Digital Versatile Disk
E	East
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
EGM	Earth Gravitational Model
ENC	Electronic Navigational Charts
ENC	Electronic Nautical Chart
ENU	East-North-Up
EOSDIS	Earth Observing System Data and Information System
EPSG	European Petroleum Survey Group
ESRI	Environmental Systems Research Institute
EW	Electronic Warfare
FAQ	Frequently Asked Question
FEDR	Flat Earth Direct Ray
FFVC	Fleet Forces Virginia Capes
FOV	Field of View
FOX	Free Objects for X
FPS	Frames-Per-Second
FSB	Front Side Bus
FST	Fleet Synthetic Testing
GARS	Global Area Reference System
GB	Gigabytes
GCC	GNU Compiler Collection
GDAL	Geospatial Data Abstraction Library
GEBCO	General Bathymetric Chart of the Oceans
GEODAS	Geophysical Data System
GIBS	Global Imagery Browse Services
GIS	Geographic Information System
GLCF	Global Land Cover Facility
GLSL	OpenGL Shading Language
GMST	Greenwich Meridian Sidereal Time
GNU	GNU Not Unix
GOG	Generalized Overlay Graphics
GOTS	Government off the Shelf
GPS	Global Positioning System
GPU	Graphics Processing Unit
GTP	Generic Tangent Plane
GUI	Graphical User Interface
HAE	Height Above Ellipsoid
HDD	Hard Disk Drive
HDF	Hierarchical Data Format
HLA	High-Level Architecture
HRO	High Resolution Orthoimagery
HUD	Heads-Up Display
Hz	Hertz
I/O	Input/Output
IA	Information Assurance

ID	Identification number
IEEE	Institute of Electrical and Electronics Engineer
ITU-R	International Telecommunication Union Radiocommunication
J7	Joint Force Development
JFCOM	Joint Forces Command
JMETC	Joint Mission Environment Test Capability
JPEG	Joint Photographic Experts Group
JPL	Jet Propulsion Laboratory
JVTSE	Joint Training Special Event
KML	Keyhole Markup Language
L2	Level 2 Cache
LAN	Local Area Network
LATR	Large Area Tracking Range
LDTS	LATR Display Terminal System
LLA	Latitude Longitude Altitude
LOB	Line of Bearing
LOD `(Icon)`	Level of Detail
LOD `(RF)`	Limit of Detection
LTS	Long Term Support
M&S	Modeling and Simulation
MAC	Media Access Control
MB	Megabytes
MBTiles	Map Box Tiles
MDA	Missile Defense Agency
MGRS	Military Grid Reference System
MHz	Megahertz
MODIS	Moderate Resolution Imaging Spectroradiometer
MSL	Mean Sea-Level
N	North
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NAWC	Naval Air Warfare Center
NCTS	New Cooperative Tracking System
NE	North-East
NED	North-East-Down
NED `(terrain)`	National Elevation Database
NEXRAD	Next Generation Radar
NGDC	National Geophysical Data Center
NGP	National Geospatial Program
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defense Command
NOS	National Ocean Service
NRL	United States Naval Research Laboratory
NSMA	National Spectrum Managers Association
NTADS	Naval Tracking and Display System
NTF	National Transfer Format
NUWC	Naval Undersea Warfare Center
NW	North-West
NWAS	Naval Warfare Assessment Station
NWSC	Naval Warfare Systems Center
NWU	North-West-Up
OCONUS	Outside [the] Contiguous United States

OGC	Open Geospatial Consortium
OGDI	Open Geographic Database Interface
ONR	Office of Naval Research
OpenGL	Open Graphics Library
OS	Operating System
OSG	Open Scene Graph
OT	Operational Test
OTG-D	Over-The-Horizon Targeting GOLD Revision D
OTH	Over-the-Horizon
PC	Personal Computer
PET	Performance Evaluation Tool
PIGUI	Plug-in GUI
PMRF	Pacific Missile Range Facility
PNG	Portable Network Graphics
POD	Probability of Detection
POSIX	Portable Operating System Interface
PPF	Pattern Propagation Factor
Radar/RADAR	Radio Detection and Ranging
RAM	Random Access Memory
RCS	Radar Cross-Section
RF	Radio Frequency
RGB	Red Green Blue
RGBA	Red Green Blue Alpha
RHEL	Red Hat Enterprise Linux
RIMPAC	Rim of the Pacific
ROC	Range Operations Center
S	South
SADM	Ship Air Defence Model
SCORE	Southern California Offshore Range
SDBF	Simulator Database Facility
SDK	Software Development Kit
SE	South-East
SHB	Secure Host Baseline
SIMDIS	Simulation Display
SNR	Signal to Noise Ratio
Sonar/SONAR	Sound Navigation and Ranging
SQL	Structured Query Language
SR	Service Release
SRTM	Shuttle Radar Topography Mission
SSD	Solid State Drive
SW	South-West
TADIXS	Tactical Data Information Exchange System
TALO	Time after lift-off
TCP	Transmission Control Protocol
TCS	Time Client/Server
TDP	Tactical Data Processor
TENA	Test and Training Enabling Architecture
TEWD	Tactical Electronic Warfare Division
TIFF	Tagged Image File Format
TIGER	Topologically Integrated Geographic Encoding and Referencing
TLE	Two-Line Element
TMS	Tile Map Service
TOI	Target of Interest

TRMC	Test Resource Management Center
TS	Time Server
TSPI	Time Space and Position Information
TTL	Time To Live
TV	Television
UDP	User Datagram Protocol
UK	United Kingdom
US/U.S.	United States
USGS	United States Geological Survey
USS	United States Ship
UTC	Coordinated Universal Time
UTM	Universal Transverse Mercator
VC	Visual C++
VM	Virtual Machine
W	West
WGS	World Geodetic System
WGS84	World Geodetic System 1984
WMM	World Magnetic Modeling
WMS	Web Map Service
WMTS	Web Map Tile Service
WMV	Windows Media Video
WSMR	White Sands Missile Range
WVS	World Vector Shoreline
XEAST	X-East Tangent Plane
XML	Extensible Markup Language
YPG	Yuma Proving Grounds
ZIP	Zip File

This Page is Intentionally Left Blank

Appendix J

Version History

J.1 SIMDIS 10.0 Service Release 7 (SR7)

Released: March 2020
Users: 26,900+

New Features

- SIMDIS-3296: Environment extension is now part of the SIMDIS core install.
- SIMDIS-3312: Precision for editing Platform 3D Model Offset Settings in Prefs Tool increased from 2 to 3.
- SIMDIS-3372: Command line argument –noredirect added to disable stdout/stderr redirection to console and log file. Potentially rarely useful for debugging errors in plug-in development. Combine with –showconsole for best effect.
- Command line option –frameless will cause the SIMDIS window to display without normal window framing.
- Custom Rendering now supports displaying a center axis for most shapes.
- Earth File Map layers now automatically refresh if they contain a 'refresh' key set to a numeric value representing the minutes between refresh. This is particularly useful for live weather image layers.
- Entity lists supports centering on an inactive entity by changing the scenario time to the closest time when the entity is active.
- GOG Export from the Data Browser's TSPI display has been replaced by the GOG Export by the Scenario Editor.
- GeoFilter enable state is no longer global and shared between plug-ins. Each plug-in now maintains its own enable state for each GeoFilter.
- HTTP Client implementation for osgEarth can now be changed in SIMDIS settings using the Imagery / HTTP Client setting. Choose between cURL, which is more mainstream and thoroughly tested, or Windows WinInet, which allows for use of PKI features on sites that require PKI.

- Highlight Shape is a new preference that applies to platforms. This lets you change the shape from the typical glowing animated circle to another non-animated shape.
- Map Scale now has settings to control colors and fonts.
- Piccolo CSV Loader and Static Sites CSV Loader are two new Python-based CSV loader plug-in is now distributed with SIMDIS.
- Preference rules can now be copied to the clipboard.
- Prefs Tool entity list provides right-click menu option to view rules that match the selected entity.
- Prefs Tool now includes an easier-to-access append button for preference rules on the main toolbar.
- Prefs Tool rules list now allows editing individual rule values in place with double click.
- Prefs Tool rules list right-click menu option added to show all entities that match selected rule(s).
- Prefs Tool supports managing Platform Time Ticks.
- Python-based plug-ins can now be loaded in SIMDIS. Python Plug-ins use the Python-wrapped Plug-in API modules and have the same capabilities as C++ plug-ins. Refer to the User Manual section on Python Plug-ins for more information.
- Python Label Code Editor has new examples for course and pitch.
- Python Labels can now access a global dictionary (GlobalCache) to store values between executions. The key must be globally unique. The dictionary is automatically cleared when a new scenario is loaded.
- Right-clicking on an entity in Prefs Tool or Data Browser while Range Tool is open now lets you add the entity to Range Tool.
- SIMDIS_AnimatedLines can now be drawn from any entity type to any entity type (no longer limited to only platform entities).
- SIMDIS_BearingLines can now be drawn from any entity (no longer limited to platforms).
- Scenario Editor Plug-in now supports exporting a platform’s TSPI points as a GOG file.
- Scenario Editor Plug-in supports centering on an inactive entity by changing the scenario time to the closest time when the entity is active.
- Velocity Vector Line Color is now configurable in Prefs Tool. It can be set using the new PlatformVelocityVectorColor rule.
- World Magnetic Model coefficients updated to the latest 2020 release.

J.2 SIMDIS 10.0 Service Release 6 (SR6)

Released: September 2019
Users: 25,700+

New Features

- SIMDIS-2590: Set Eye LLA dialog now includes the option to use MGRS coordinates.
- SIMDIS-3172: The error popup for failed-to-load plug-ins now lets you retry loading, in addition to providing details about how to re-enable the plug-in.
- Action "Toggle Standard Bars" added to control the display of the standard bars without including the time slider.
- Appending ASI files or HDF5 files will now load all Preference Rule files if the ASI files or HDF5 files are not from a Live Logging recording.
- CDS Overlay files can now be loaded in SIMDIS.
- Chat Display plug-in now supports version 2 of the Chat Server log format.
- Command line option -nosplash added to hide the SIMDIS splash screen on start-up.
- Custom Rendering ellipses and rectangles now have an "outline" preference.
- Custom Rendering options have been moved to Prefs Tool. The Configuration dialog is no longer available.
- Custom ToolBar Plug-in can execute actions from .xml files when being loaded.
- Custom ToolBar Plug-in now accepts multiple config files through command line.
- Exporting a scenario as an ASI file to a directory is now supported. The directory will include the support files like the View File, Pref Rule File, etc.
- Fence Selection of Platforms added to Prefs Tool and Data Browser.
- GOG Tool provides sorting options for shapes
- Media menu adds new Playlist Management dialog to control mute state of individual playlists.
- Plug-ins developed by any version of MSVC 2015-2019 can be loaded in a SIMDIS from any other MSVC 2015-2019. MSVC 2015 can load up to 2019, and 2019 can load back to 2015.
- Prefs Tool Undo and Redo have been added as new global hot key actions.
- Prefs Tool has a new toolbar button to toggle whether to prefer manual preference changes over preference rules.
- Python in Data Scripts and Labels now uses Python 3 syntax. This may have a significant impact on your scripts. Please update for compatibility.
- Scenario level Custom Rendering entities now affect the scenario time range.

- Set Eye LLA dialog now includes the option to use GARS coordinates.
- Setting "Case Sensitive Category Data" under the Scenario folder was added to control if the Category Data is case sensitive. Category Data now defaults to case insensitive.
- Shapefiles can now change color properties in GOG Tool.
- Text Decluttering can now be done with callouts. Toggle callout usage and appearance settings in the Settings Dialog under Main Window / Display.
- Triton Ocean and SilverLining Sky Models and support files have been moved to an external installer to reduce the size footprint of the default SIMDIS installation.

J.3 SIMDIS 10.0 Service Release 5 (SR5)

Released: March 2019
Users: 24,500+

New Features

- SIMDIS-3067: Lines of Bearing (LOBs) are now pickable in SIMDIS using the mouse. Hover near either endpoint of the LOB to select it.
- AC3D files (*.ac) are now supported in SIMDIS.
- Bookmark Dialog's Time Events GUI now has a clarified interface for toggling Real Time playback.
- Bookmark Extension's execution will pause when scenario playback is paused, and will now resume when playback continues.
- Bookmarks can now be executed in live mode.
- Data Scripts can now create target beams. BeamRef now includes a setTarget() function.
- Data Tables in the Data Browser now supports a precision value for column types of `simData::VT_FLOAT` and `simData::VT_DOUBLE` as defined by the Generic Units.
- GOG saving now shows success or failure in a toast popup.
- HUD items can now be moved around in HUD Editor mode. Select HUD Editor from the HUD menu. Presets can be configured and bound to hot keys.
- Label prefs now let you choose between using the actual data values or modified display values in the platform label.
- Legend Overlay can now show more than one column of entity legends.
- Live Logging configuration GUI can now be displayed with the hot key action 'Show Live Logging'.
- MP terrain engine is no longer supported.
- Models with Degree-of-Freedom (DOF) nodes with animation settings can now automatically animate using the 'Animate DOF Nodes' preference for platforms.
- Moon scale and image are now exposed as settings.
- Projector FOV can now be overridden in Prefs Tool and using rules.
- Remote Control now supports a Command channel. When connected, user Data Flush operations on the server will also flush clients on the Command channel.
- Time Stamped Imagery example updated to include altitude components for GOES and NEXRAD layers.
- Vapor Trail now implements Wake processing.
- Vapor Trail now respects the Expire Mode Track History preference.
- World Magnetic Model (WMM) updated to latest NOAA release, WMM2015v2.

J.4 SIMDIS 10.0 Service Release 4 (SR4)

Released: September 2018
Users: 23,600+

New Features

- SIMDIS-2619: Custom Toolbar now remembers the last file(s) loaded.
- SIMDIS-2944: PISIMDIS::geodeticToScreenXY() now returns whether the requested point is behind the camera, outside of the view frustum, or inside the view frustum.
- SIMDIS-2947: Viewports Dialog can now set Horizontal Field of View (FOV-X) for the main view and inset viewports.
- SIMDIS-2977: Plug-in API PIButton instances now support a setIcon() method.
- Bookmarks now support an XML format with the extension “*.bml”.
- Command line argument –clearSettings now creates a backup of the settings file before clearing it.
- Display menu now provides options to save the current window layout to a file.
- Drag-and-drop allows appending scenario, rule, and range tool configuration files by using the CTRL key.
- Dynamic scaling can be set to be on by default via settings at Settings / Main Window / Platform Dynamic Scaling.
- Earth file for new global ocean layer mask added.
- Example ASI files have been added to demonstrate PST, LST, and Video Icons.
- File Reader Plug-in now supports merging ASI files with different reference years.
- File menu now has “Append" option to append data files.
- Filtering of Duplicate Color and On/Off commands can now be controlled using the Scenario / Reject Duplicate Data settings value.
- Geo-Filters can now be edited with the Geo-Filter Service Plug-in. The plug-in is useful for users not using the Geo-Alert Plug-in or the Read SCORE Plug-in.
- HDF5 Reader Plug-in now supports merging HDF5 Streaming files with different reference years.
- Hover Text can now be toggled with a new action, usable as a hot key or as part of a custom toolbar.
- Layout settings are now saved to a separate settings file, layout.ini. Use the new command line option –layout to load a specific layout file.
- Live Logger’s GUI now has a submenu for configuring auto start/halt times of the recording.
- MediaPlayer app now displays the last-loaded filename in the title bar and the status bar. Mousing over the filename in the status bar will show a tooltip with the full file path.

- Mesa accelerated drivers no longer complain on startup about being software renderers, so long as they support modern OpenGL.
- ModelViewer replaces Model Editor as a GL3 compatible application for viewing 3D models.
- OpenGL Core Profile is now supported in SIMDIS, increasing overall compatibility with graphics cards and drivers developed in the last decade.
- Prefs Tool has a new setting for LOS Altitude Offset on the Platform tab under EO and EM Setting.
- Prefs Tool preference rule list now has a search option to filter the list of displayed rules.
- Prefs Tool's Filter and Entity tabs can now be accessed directly with new Hot Key Actions (unbound by default).
- Prefs Tool's category filter no longer shows entity counts in Exclude mode.
- Projector frustum can now be visualized with new preference in Prefs Tool's Projector tab can now toggle visualization of the Projector's frustum.
- Projectors can now be configured to project onto platforms in the scene in addition to the terrain. Platforms can currently only be selected in the Plug-in API.
- Range Tool now allows users to specify the RF Radius Scalar and the Optical Radius Scalar via the Range Tool Configuration dialog.
- Reference Grid has a new setting for toggling the visibility of Geodetic grid lines. The setting is available at Reference Grid / Grid Lines Visible.
- Reference Grid now includes a setting to toggle visibility of graticule labels in the geodetic reference grid.
- Remote API Plug-in now supports disconnecting a client.
- SIMDIS supports zipped HDF5 files via a file extension "*.hdfz".
- SST text now defaults to a thin outline if no explicit outlining is specified.
- Settings dialog adds button to save "deltas," only settings that are different from their default values.
- Time Editor provides setting to enlarge font size of time display.

J.5 SIMDIS 10.0 Service Release 3 (SR3)

Released: March 2018
Users: 22,300+

New Features

- SIMDIS-2657: Bookmarks Extension's Add Bookmark action now opens the Bookmarks dialog with the Comment edit frame shown and sets the keyboard focus to the text edit. Pressing Ctrl-Enter will instantly create a comment bookmark event with the entered text.
- SIMDIS-2863: Gates can now be centered on using Prefs Tool and Data Browser.
- SIMDIS-2934: Bookmarks now have their own Units Context and the Time Rate spinner for the Time Event now supports 4 digits after the decimal point.
- Bookmark GUI now indicates when a bookmark has been modified but changes have not been saved.
- Bookmark file paths in events are now saved relative to the TOC file.
- Bookmarks now include a Platform Highlight event.
- Bookmarks now include an Execute Action event that allows activation of arbitrary hot-key-able actions during bookmark playbacks.
- Category Data breadcrumbs are now shown in Data Browser and Prefs Tool.
- Category Filter GUI has been revamped from the ground up to provide an easier way to filter entities. The new widget is intended to improve usability and understanding of filtering state compared to the old checkbox tree GUI.
- Category Filter GUIs now show an entity count indicating how many entities match the given filter.
- DCS Plug-in now supports an Append mode.
- Data Table data limiting now uses half as much memory as before when limiting tables by size.
- Dragging and dropping multiple data files will result in loading the first file and appending the remaining files.
- GOG Tool now sorts GOGs in ascending order.
- Layout saving and restoring now displays a Toast window on success.
- Live Logging now supports logging to an HDF5 file for better performance and smaller files. Live Logging picks the file type based on the extension. Use a file extension of .asi for an ASI file and .hdf5 for a HDF5 file.
- Loading scenario files is now a two-step process if appending to a current scenario is an option.
- MGRS Grid colors can now be dynamically adjusted through Settings.
- Main View Field Of View is now editable in the Viewports dialog. The View / FOV setting has been removed. FOV for main view is now managed by a view file.

- Map Editor's Load Layer dialog now includes the ability to load Tile Package image layers.
- Map Scale graphic has been added to the HUD menu and is displayed in the lower right corner. Settings control the units display. Map Scale is off by default.
- Mouse cursor can now be set to hide after a certain number of seconds of user inactivity. To toggle this setting on/off or change the inactivity duration, go to Settings / Main Window / Mouse / Hide Cursor Duration.
- Mouse navigation modes are no longer shown by default on the status bar. Use the 'Main Window / Mouse / Nav Modes on Status Bar' setting to show them again. Changing this requires a restart to take effect.
- Mouse selection now defaults to the new Dynamic Selection Algorithm. This can be changed with the Main Window / Mouse / Picking Algorithm setting (SIMDIS restart required after change).
- Multiple groups of timed imagery is now supported. At most one image from each group will be shown depending on time. Timed image layers with no specified group will be considered in the same group.
- Platform lighting is now on by default.
- Platform models now load in the background. Changing icon models will no longer have a potentially drastic impact on frame rate.
- Prefs Tool has a new setting for Icon Alignment on the Platform under Image Icon Settings.
- Quick Find Platform window is now available with the default hot key of Ctrl-F.
- Range Tool calculation abbreviations were updated to be shorter and more consistent. The abbreviations are now used for both the 3D Display and the Column Heading in the table display.
- Range Tool now provides the option to display magnetic azimuth.
- Range Tool now supports additional text display options. Additionally, displaying graphics and text are optional.
- Range Tool now supports showing a Pairing Line instead of the calculation graphics to reduce screen clutter.
- Range Tool now supports swapping the Begin Entity and the End Entity via a new button.
- Range Tool now supports templates for defining and applying a set of calculations to selected associations.
- Range Tool now supports using the mouse to select platforms from the 3D display when defining an association.
- Reference Grid button on toolbar now includes a submenu to choose the specific grid type to display. Corresponding hot key actions have been added.
- Reference Grid extension is the new name for the Lat/Lon Grid extension. Settings previously found under Lat/Lon Grid are now found under Reference Grid.

- Remote Control Client now has an option to control local rule processes. Local rule processing is off by default.
- Ruler Extension allows creating multiple Ruler lines by holding Ctrl while clicking (in Mouse Rotation mode) or dragging (in No Rotation mode).
- Ruler Extension now allows deletion of an existing Ruler line by right clicking on the line.
- Ruler Extension now allows editing of an existing Ruler line by clicking on (in Mouse Rotation Mode) or dragging (in No Rotation mode) the endpoint.
- Ruler Extension now includes a Ruler List window for viewing all created Ruler lines as well as deleting selected lines.
- Ruler Extension now supports toggling the platform lock state when the Alt key is pressed.
- Scenario Editor Plug-in has a new fence select functionality. Click and drag on the 3D display to draw a box. Data points of selected/filtered platforms within the box on mouse release are deleted.
- Scenario Editor Plug-in now supports velocity filtering on a calculated velocity instead of the provided velocity.
- Snapshot Plug-in now supports millisecond (`%MSEC%`) variable in filenames.
- Time stamped imagery demo added.
- UTM Grid text now scales down as you zoom in.
- Video Recorder Plug-in can now be set to play simulation clock when recording starts.
- Video Recorder Plug-in has a new functionality that allows starting and stopping recording automatically based on scenario time.
- Video Recorder Plug-in now supports millisecond (`%MSEC%`) variable in filenames.
- View files now show a Toast window when successfully saved.
- Viewports dialog allows editing FOV for all insets.
- World Imagery mbtiles files now replace the older ETOPO2 files.

J.6 SIMDIS 10.0 Service Release 2 (SR2)

Released: September 30, 2017
Users: 21,000+

New Features

- SIMDIS-2653: SIMDIS SDK DLLs and SOs renamed to include version data.
- SIMDIS-2655: Marker Tool GUI adds several new quality-of-life features. Ctrl-Enter is a new shortcut for adding a marker. Ctrl-/ is a new shortcut for manually querying elevation at the marker position. Opening the GUI sets focus on the Name field, ready for user input.
- SIMDIS-2772: Map Editor now allows setting the vertical datum used for each loaded elevation layer.
- ASI Loader is no longer distributed. Please use the File Reader Plug-in, which supersedes ASI Loader.
- Beam rendering has been significantly improved.
- Custom Toolbar Plug-in now supports loading GOG, Preference Rule and View files via a DISCN file.
- DIS Plug-in now adds entity damage state to category data.
- Data Browser now supports selecting multiple entities. Properties specific to a type of object will only be available when selecting multiple entities of the same type. In the Generic Data and Category Data tabs, only currently active data points will be visible.
- Data Browser's Current Data lists now support multi-select and include the entity name of the current data.
- Data Script commands executeAction(), doesActionExist(), and addCategoryData() were added.
- Data Scripts can now support preference rules with a specified preference rule version.
- Drag-and-Drop now works with many more file types when dropping on the main SIMDIS GUI.
- Entity lists now use entity type iconography instead of a character.
- File Reader Plug-in now supports appending files via a hot key.
- File locations of files in the standard directories referenced by a scenario are now saved to an XML file for later reuse. This caching of the file locations can significantly improve the performance of file searching.
- Gate rendering has been significantly improved.
- Hook Text is a new field grouping in the Labels tab of Prefs Tool for managing display fields in the Hook Track window.
- Hook Track Extension provides capability to "hook" entities, which pops up a distinct dialog with information about the hooked entity. Dialog contents are defined by the Label / Hook Text display fields.
- Image format for a layer can now be set in the webMaps.config file.

- Improved readability of viewer statistics display.
- JDS ASCII Overlay format is now supported.
- KML files now support image overlays. Note that loading a .kml with an image overlay and saving it as a .gog will not capture the overlay.
- LOB Groups now support flashing via Generic Data. See User Manual Appendix A for details and the file `$(SIMDIS DIR)/demos/SIMDIS/Examples/LOB/LOBDemo.asi` for an example.
- MGRS Grid values now show on the side of the SIMDIS view when sufficiently zoomed in when the MGRS grid is enabled.
- MGRS Grids now become more finely detailed when zooming in.
- MP terrain engine is now automatically enabled for older video cards and drivers, improving system compatibility.
- MP terrain engine is now preferred when detected OpenGL GLSL version is less than 3.3.
- Marker Tool has a new option to update existing markers with new values.
- Marker Tool now includes the ability to use any model or icon as the icon for a marker.
- Media Timeline Display has been added and can be accessed from Media / Show Media Timeline.
- Multiple SIMDIS extension dialogs now feature Drag-and-Drop file loading capability.
- osgEarth packager (command line and GUI tool) removed from SIMDIS distribution. No replacement has been added back at this time.
- Overhead mode now uses a true orthographic camera, fixing many display discrepancies seen in non-orthographic overhead mode.
- Overhead mode performance is improved.
- Platforms now glow when the mouse hovers over them to give selection feedback. This feature can be toggled with the setting Main Window / Mouse / Highlight.
- Plug-in API command PISIMDIS::setPreference() now respects hex values for colors and stipples when prefixed with 0x.
- Preference Rules with invalid regular expressions display red in the Prefs Tool's rule list.
- Preference Rules with invalid regular expressions now cause warnings on file load.
- Prefs Tool is a new extension that replaces both Super Form and Preference Rules GUI. New features include: undo and redo; saving and loading entity preferences to file; instant rule creation; automatic rule compression.
- Prefs Tool now allows applying the currently selected preference value with a new "Apply Current Value" button, so preferences can be updated and rules created without changing the value.
- Prefs Tool rules list now shows whole names (such as Platforms, Beams, etc.) in the Entity Type column where appropriate.

- Prefs Tool's Entity Label tab Character Limit preference is now located in the Name and Alias section, so that it is co-located with an entity's name.
- Prefs Tool's rules GUI now has the ability to add and edit comments.
- Range Tool Plug-in's Begin and End Entity lists now feature filter configuration buttons like those found in Data Browser and Prefs Tool.
- Range Tool colors for the Azimuth, Elevation, Composite angle types can now be selected independently.
- Range Tool now allows adding Begin and End Entities by right clicking on the entities in SIMDIS and selecting the appropriate option from the "Range Tool" sub menu.
- Range Tool now supports Entity State filtering.
- Read-only text fields now show with a grey background to give a stronger indication that they cannot be edited.
- Record Video action from simVideo Plug-in now correctly shows toggle state in Custom Toolbar Plug-in.
- Splitters now have small dots to make their handles more easily visible.
- Super Form and Preference Rule GUIs are now deprecated in favor of Prefs Tool. You can re-enable them for now by adding –enable-superform and –enable-prefrules to your SIMDIS command line.
- Terrain DB files with time information can be loaded as a timed image layer, which will automatically show or hide depending on current time.
- Timed image layer support, such as for weather, has been added.
- Users constantly creating Markers only in the Western hemisphere can now select the "Force West" option and omit the leading minus sign when entering a location. The expectation is the feature is turned on and left on. The same feature is available for the Southern hemisphere.
- Windows that are docked and stacked can now be closed by middle clicking their tab.

J.7 SIMDIS 10.0 Service Release 1 (SR1)

Released: March 28, 2017
Users: 20,000+

New Features

- ASI Append option now compares entities by Name and Original ID and merges matching entities.
- ASI Loader Plug-in supports appending data to existing entities.
- Articulation GUI has been added for displaying and manipulating entity articulations.
- Bookmark demos can now be signaled to close SIMDIS on completion with the new –bookmarkquit command line argument.
- Bookmark execution settings are now cached. Changes to settings from Bookmarks no longer make permanent changes to your environment. Exiting from Execute mode will restore "normal" settings.
- Bookmarks Filter GUI now provides a "Toggle All" option.
- Bookmarks can now execute immediately on load when you specify the –execute command line argument to SIMDIS.
- Bookmarks created outside the valid time range are now shown in red.
- Bookmarks list now always uses time as a secondary sort criteria and execution priority as a tertiary sort criteria, providing a more consistent sorting of items.
- Bookmarks now support previewing real-time edits and previewing multiple events.
- Cache policy and max age settings have been added to the Quick Add map layers.
- Command line arguments listed in simdis –help are now categorized for easier reading, browsing, and searching.
- Country Borders are now available, and can be accessed from Display/Country Borders or the Ctrl-P hot key. This is similar to the Point Map display in SIMDIS 9.
- Data Browser window configuration buttons are now maintained per entity type.
- Detailed Frame Rate view now provides more detailed timing statistics.
- Dock widgets can now be closed with typical system shortcuts like Ctrl-F4.
- Dock widgets now maximize and restore if the title bar is double-clicked while floating.
- Dock widgets now show focus by coloring the title bar based on system settings.
- Entity Tree now sets the text color to gray in cases where an Entity Name is shown because the Entity Alias is empty.
- Environment Settings is a new extension available at Tools/Environment. This extension allows easy modification of the sky and ocean settings, and includes easily accessible environment presets such as Fair, Overcast, and Stormy.

- File search list of known locations has been reordered. See user manual for details.
- GIS Navigation mode is an alternate mouse configuration similar to other GIS applications. To enable this, set the Main Window/Mouse/Navigation setting to GIS Mode.
- GOG Tool now includes the ability to flash GOGs in the 3D display for quick identification. This ability can be accessed by right clicking on a GOG in the GOG Tool list and clicking "Flash".
- Label, Hover, and Legend now show no name if the entity should show the alias name and the alias is blank.
- Legends now have the option to display an icon for its entry.
- Live Logger Plug-in now includes a small screen indicator in the lower-right corner to notify users that it is recording.
- MAKO Version 2 chat logs now have message overlay import support.
- MGRS Grids are now supported. To toggle between Lat/Lon and MGRS grids, change the setting 'Lat Lon Grid/Use MGRS Grid'.
- Main View Swap now supported via the Shift-V hot key.
- Map Editor convenience drivers can now be configured via a configuration file.
- Map Editor now includes support for accessing and displaying several free ArcGIS base maps.
- Map Editor now provides a field for changing the Minimum Valid Value for elevation terrain sets.
- Map Editor's Elevation Details pane now shows more data about the tile source.
- Map Editor's Load Layer GUI now remembers the last selected layer driver type, instead of always defaulting to DB.
- Media Timeline now displays loaded media files in a Gantt chart display.
- Plug-in Manager now features a Clean button that removes invalid entries from your start-up plugins.ini configuration file.
- Preference Rules GUI now combines related buttons into drop-downs in the toolbar.
- Preference Rules GUI now has a Replace button to replace the selected rule(s) and a Comment button to toggle the displaying of comments.
- Python label code now supports calling some format methods for common types. See the file Format.i for details.
- Python label code now supports calling some of the Plug-in API methods. See the files PICommon.i, PIData.i, and PISIMDIS.i for details.
- RF Propagation GUI now supports 3D Texture, RAE, 2D Vertical, 2D Tee, and 3D Points graphics modes. The graphics modes are separated into categories based on efficiency, and not all displays will perform well with all data sets.

- Range Tool now calculates the optimal number of segments for a line and ignores the value from the Settings dialog.
- Ruler Angle now follows the Magnetic Variance setting. The Mouse Cursor Readout now follows the Vertical Datum setting.
- SIMDIS-2486: Command line argument –maximized added for showing the main window as maximized. Unlike –fullscreen, maximized windows show decorations.
- SIMDIS-2498: Time slider can now be repositioned to different edges of the main view, and can invert its appearance. Right-click on the slider to move it.
- SIMDIS-2553: Entity Type Filter for Super Form and Data Browser is now persisted.
- Scenario Editor Plug-in now supports detecting stationary platforms via deltas in ground distance and orientation angles.
- Scenario Editor Plug-in now supports merging of platforms when the Name and the Original ID match.
- Scenario/Reject NaN setting has been added to allow users to control if Data Tables accept or reject NaN values. The setting defaults to false.
- Terrain Engine is now shown in the Help/About window.
- Undock Main View button in the mouse status bar can now be toggled on and off to undock and dock the main SIMDIS 3D display.
- View files with invalid values for position due to hand editing now print a warning to the console.
- Views tool now provides enhanced precision for placing insets.

J.8 SIMDIS 10.0

Released: September 30, 2016
Users: 19,000+

New Features

- ASI version has been increased to 22.
- About GUI's Copy-to-Clipboard functionality now includes the SIMDIS build number, the command line used to start SIMDIS 10, and plug-in information.
- Bookmark demos for several scenarios have been added as replacements for SPY scripts.
- Bookmarks now support a small subset of the options in a SIMDIS 9 Preference file.
- Calculator can now be started from SIMDIS 10.
- Collapse/Expand All buttons added to Preference Rules and Super Form menus.
- Console window now has a button to open the logs directory in a file explorer.
- Core build no longer includes SilverLining, Triton, and several other shared libraries, in order to reduce size of the distribution. These files can still be found in the full SIMDIS installer.
- Cylinders can now be created in ASI files as GenericData and attached to a platform. Controllable features include: Position (relative to its platform), Direction, Length, Color, and Radius. Color and Radius can be set independently for the two circular faces. Cylinders can be transformed over time by declaring different values at each time step.
- Data Browser and Super Form now share entity filtering.
- Display menu provides a new 'Restore Layout' option to restore previously loaded window layout.
- Eye Position information on saved positions is now available via the Selected tab of the Eye Position dialog.
- File Reader Plug-in now supports Append via a check box on the File Open Dialog.
- File Writer Plug-in now supports projectors.
- GOG Tool's Edit GUI is now slimmer than before.
- Gate outline visibility can now be toggled, to improve visibility in some use cases.
- Graphics requirements for SIMDIS 10 are now checked on start-up and a GUI is shown if minimum requirements for OpenGL support are not met.
- HDF5 Import/Export now supports projectors.
- Incremental compilation of shaders is now accessible through various advanced settings under the Incremental Compilation group. Incremental compilation is currently experimental in SIMDIS and is disabled by default. When enabled, it can help smooth frame rates in many cases.
- Installer now creates icons in the All Users location when run as administrator.

- Label Deconfliction feature now respects priority on the labels. Use Super Form or Preference Rules Labels tab to set a priority on your labels.
- Legend leader lines are now drawn correctly.
- Legend text color can now be set to use the corresponding entity's label text color by checking 'Use Entity Label Color' in Settings.
- Legends can now be controlled using the Legend Manager found in Tools/Legend Manager. All settings found in the Settings dialog are present in Legend Manager.
- Legends can now be moved around the screen by clicking and dragging. They can be returned to default position by a right click menu option or by hot key command.
- Legends can now be sorted by entity latitude or entity affinity.
- Live Logging now supports Projectors.
- MGRS coordinates are now supported in GOG files.
- Map Editor's WMS Get Capabilities layer display has been improved, and additional USGS and Ready Map data sources were added.
- Marker placement with the mouse has been updated to improve efficiency.
- Markers now default to the current eye position.
- Ocean Models can now be cycled by pressing the U hot key.
- Ocean settings for Triton have been enhanced to include sea state, choppiness, rendering quality, and wind direction. Wind is pulled from the scenario by default.
- Ocean transparency level can now be changed using the Ocean Model/Opacity setting.
- Outline and shadow placement options have been added to labels in Super Form and Preference Rules.
- Platform right click menu now includes an 'Attach GOG' option.
- Plug-in Manager can now show both file and error state of plug-in.
- Plug-in Manager now hides invalid plug-ins from list based on settings.
- Plug-ins now alert using System pop-ups when they cannot be started because dependencies are missing. This popup only shows when manually loading a plug-in.
- Precision control fields have been added for all the units for the Label, Hover, and Legend text.
- Preference Rules GUI added 'Force All Rules' action to force application of all preference rules.
- Projector FOV interpolation serialization to added to FileReaderPlugin, FileWriterPlugin, and HDF5Plugin.
- Projector FOV is now interpolated by default. This setting can be changed in Super Form.
- Projector alpha values can now be changed in Super Form and Preference Rules.
- Projector raster files can now be changed using Super Form and Preference Rules.

- Range Tool GUI has been slightly improved for usability.
- Range Tool now has 4 color wells to control the color of entity pair calculations: A color well for each line stipple, one for pie slice fill color, and one for text color.
- Range Tool text labels now include a drop shadow to make them easier to read.
- Remote Control now supports changes to Eye Positions and Insets.
- Remote Control now supports enabling/disabling the 'Entity Preference' channel and the 'View' channel.
- SIMDIS-677: GOG annotations can now be assigned a priority with the new 'priority' keyword.
- SIMDIS-2096: The precision of the Time Step spinner of Time Editor now follows the precision of the HUD time precision up to 6 digits.
- SIMDIS-2317: Platform scale can now be increased and decreased by using * and / hot keys, respectively.
- SIMDIS-2362: Plug-ins missing proper initialization statements now give better load errors in SIMDIS.
- SIMDIS-2369: Terrain Toggle button can now be found under the Display menu, in the toolbar, and bound by default to hot key Shift I.
- SIMDIS-2423: Circle highlight now draws an animated circle, instead of making the model glow.
- SIMDIS-2433: DTED extensions dt0, dt1, and dt2 have been added to the Quick-Add Elevation GUI.
- SIMDIS-2463: Data Browser and Preference Rules user interfaces now use radius to specify the local grid size. The preference rule still uses diameter, but the Preference Rules user interface automatically converts between diameter and radius.
- SIMDIS-2467: Label alignment for entities can now be changed in Super Form and Preference Rules.
- SIMDIS-2474: Track history mode is now persisted.
- SIMDIS-2479: Entity preferences for units now default to the value of 'Entity Defaults' units context. The default values will be overwritten by Super Form and Preference Rules.
- Scenario properties tab added to Scenario Data dialog. Contains information such as source, classification, and reference year.
- SilverLining advanced sky model settings have been exposed through the SIMDIS Settings GUI. Users can now swap between cloud cover presets, enable precipitation, and modify other model related settings. Scenario wind controls cloud formation.
- Sky model lighting can now be adjusted by a number of hours using the 'Sky Model/Hours Offset' setting.
- Super Form preference tabs checkboxes are now single click, and entries are less verbose.

- Terrain avoidance offset can now be specified in Settings at 'Main Window/Mouse/Terrain Avoidance Distance.' When terrain avoidance is enabled, this is the minimum distance from surface permitted for the eye.
- Terrain configurations that fail to load from the command line or from settings now display a Toast notification on start-up.
- Time Editor buttons have been resized back up to 32x32 pixels.
- Track draw mode (point, line, ribbon, bridge, off) can now be individually controlled and activated by global actions and bound to hot keys.
- Various image settings for a selected layer are now shown based on driver type.
- Version information can now be viewed using the command line.
- Weather maps in Map Editor can now be changed by editing the weatherMaps.config file in `SIMDIS_DIR/data/sdTerrain`.