

# gperlstk Documentation

Gregory Reed (reedgs@uah.edu)

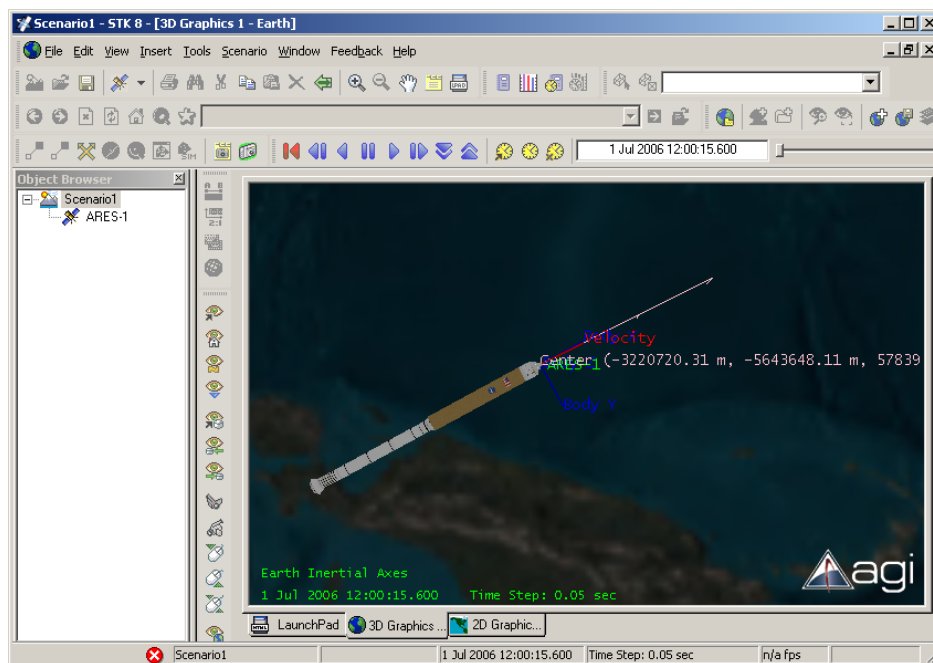
February 28, 2007

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Requirements/Initialization</b>	<b>2</b>
2.1	ActivePerl . . . . .	2
2.2	Use with STK . . . . .	2
<b>3</b>	<b>Running the Tool</b>	<b>3</b>
3.1	Navigation via the Command Prompt . . . . .	3
3.2	Command Line Arguments . . . . .	3

# 1 Overview

**gperlstk** is a tool used to automate the visualization of output from the orbit propagation simulations Maveric, POST, and OTIS. The tool communicates with AGI's STK product through AGI's Connect API, enabling the display of the vehicle on a two-dimensional projection and in a three-dimensional visualization. STK also provides the capability to display pertinent information about the vehicle, such as state vectors and coordinates. Combined with custom CAD models and associated articulation files, gperlstk helps provide an accurate depiction and a more rapid post-processing the simulation tools' output.



## 2 Requirements/Initialization

### 2.1 ActivePerl

gperlstk is programmed in Perl, an interpretive language designed initially to simplify parsing text files and extracting relevant data. Under Windows, a Perl distribution is required (though under most Linux implementations, a Perl distribution is included). The recommended distribution is ActiveState's free **ActivePerl**, available at <http://www.activestate.com/>.

ActivePerl requires administrative privileges to install, and it normally creates the directory **C:\Perl** upon installation.

### 2.2 Use with STK

gperlstk also requires the Perl libraries for STK's Connect API. Copy the files **Stk.pm** and **STKUtil.pm** (normally located in **C:\Program Files\AGI\STK 8\Connect\PERL\_LIB\**) to **C:\Perl\lib\**.

Note also that the STK installation that the tool will interface with must have access to the appropriate licenses. For STK 7 and below, these are the Connect license (CON) and, optionally, to the VO license (STKAVO) in order to view 3D visualizations. For STK 8, the STKIntegration license is required overall, and the STKProfessional license is needed if the 3D visualization capability is desired. Access to these licenses can be checked under STK's **Help->License Viewer** dialog.

## 3 Running the Tool

### 3.1 Navigation via the Command Prompt

The tool uses the Windows command prompt. The following describes how to use the prompt to navigate to the tool:

1. Press **Win+R** or Click **Start -> Run** to open the **Run...** dialog.
2. Type **cmd** into the input box. The command prompt should then appear.
3. At the prompt, type **X:**, where X is the hard drive letter where the tool resides.
4. Type **cd [PATH]**, where [PATH] is the full pathname to the tool.
5. Type **gperlstk.pl** with no command line arguments to view usage options.

When no command line arguments are specified the tool simply prints usage options and exits. To run the tool on an input file, run STK on the local machine, open a new scenario, and type **gperlstk.pl [FILE]**, where [FILE] is the full path to a Maverick, POST, or OTIS file.

To log the output of the tool to a text file, append **> [PATH]** to the end of the command, where [PATH] is the full pathname of the log file to generate.

### 3.2 Command Line Arguments

gperlstk supports several command line arguments to diagnose the tool's output, to allow for custom visualization parameters, and to extend the tool's usability. The command line parameters can also be combined and run together. The available options are as follows:

- **-t [TIMESTEP]** or **--timestep=[TIMESTEP]**

When this option is specified, gperlstk will send only the data residing at intervals of the given timestep in the input file. For example, if **--timestep=10** is specified on the command line, the tool will send data at times of 10 seconds, 20 seconds, 30 seconds, and so on, skipping the data in between and allowing STK to interpolate the rest.

This option is useful when the data file has numerous data points at very small intervals, since it will send a "rough sketch" of the simulation file to STK, enabling a quicker processing time for the visualization. The timestep can be lowered on subsequent runs to visualize the simulation file in more detail.

Note: The tool will ALWAYS send both the first data point and the last data point (or the point at time specified by **--limit**) in the file to ensure that the entire desired range is covered. If, for example, the input file ends at 103.5 seconds, and the specified timestep is 10 seconds, then the tool will send data at times of 100 seconds and 103.5 seconds.

- **-l [TIME]** or **--limit=[TIME]**

With the limit option given, the tool will stop sending data from the file at the specified time limit. If the simulation file contains 100 seconds' worth of data, **--limit=10** will send only the first ten seconds' worth of data to the visualization. This option can be combined with the timestep option to provide a "rough sketch" of, for example, the first ten seconds of the simulation.

- **-h [HOST]** or **--host=[HOST]**

By default, gperlstk attempts to connect to a STK session on the same machine from which it is run. However, with this option, it will instead attempt to connect to a remote STK session on another machine and run the visualization on that session. The HOST parameter can be in the form of an IP address or network hostname.

- **-e [PATH]** or **--ephemeris=[PATH]**  
**-a [PATH]** or **--attitude=[PATH]**

These options will produce an ephemeris or attitude file, respectively, from what is visualized (that is, if **--timestep** and/or **-limit** are used, the ephemeris or attitude file will reflect this). PATH can be an absolute or relative pathname, and it must include the filename to produce. The options can be combined to produce both an ephemeris file and an attitude file.

- **-i [PATH]** or **--initialize=[PATH]**

This option allows the user to customize the visualization. PATH should be an absolute or relative path to an existing text file. This file can contain custom Connect commands, one on each line, that will be sent to STK after the default visualization parameters have been provided and before the content from the simulation file is processed. Refer to the STK documentation for Connect for more details.

The option can be combined with **--verbose** to view the order in which STK sends commands from the initialization file and whether STK sends back an ACK or NAK for each command.

- **-m [PATH]** or **--model=[PATH]**

This option replaces the default satellite/launch vehicle model with one specified at the file located at PATH. If there are articulations associated with this model, they can be used accordingly. Refer to the STK documentation for more information on articulation files.

- **-v** or **--verbose**

With this option provided, the tool will output more information to the screen, including every individual Connect command and the response from STK (ACK/NAK).

Examples:

- **gperlstk.pl OTIS\Example.txt**

This command will use the data at OTIS\Example.txt as input to the visualization.

- **gperlstk.pl Maverick\STKShort.txt -m F:\Model\ares-1.mdl -l 20**

This command will use the first 20 seconds' worth of data from the file Maverick\STKShort.txt (with the path relative to the location of gperlstk.pl) as input, and the model at F:\Model\ares-1.mdl will be loaded.

- **gperlstk.pl F:\Data\mavdata.txt -v -l 10 -t 1 -e F:\Data\ephem.e > log.txt**

This command will use the first 10 seconds' worth of data from F:\Data\mavdata.txt, enforcing a timestep of 1 second. A verbose output will be logged to the file log.txt, and STK will generate an ephemeris file at the path F:\Data\ephem.e.

- **gperlstk.pl F:\Data\POST.txt -i F:\Init\init1.txt -v**

This command uses the data found at the path specified as input. Before the data is sent to STK, the Connect commands found at F:\Init\init1.txt are sent. The verbose parameter is used to ensure that the commands found in the file are valid.