

Spacecraft Mission Visualiser v1.0

User Manual

CONTENTS

CONTENTS	1
INTRODUCTION	2
FEATURES AND FUNCTIONALITY	3
The help menu	3
Starting the application	3
Settings	3
Creating, opening, saving flight files	4
Text editor	4
Launching a PANGU server	5
Server log	5
Connecting to a PANGU server	5
Scanning commands	6
Executing commands	6
Supported commands	8
Playback interface	8
3D coordinates visualisation	9
Moving and resizing the interface elements	10
EXAMPLE USAGE INSTRUCTIONS	12

1. INTRODUCTION

The Spacecraft Mission Visualiser is an application that is meant to be an external helper to the PANGU suite of tools by making the production of PANGU spacecraft flight simulation files more intuitive and interactive. Its feature set focuses on users who are already familiar with the PANGU Viewer and the PANGU User Manual's Flight Files section.

While reading the manual please keep in mind that screenshots of the interface were taken on a Linux system. Some elements might appear slightly different on Microsoft Windows systems.

2. FEATURES AND FUNCTIONALITY

2.1. The help menu

The help menu can be accessed from the toolbar on top of the application. It contains a link to this manual and access to the 'About' window. The window has a brief description of the application as well as information on open source software used.

2.2. Starting the application

In the distribution for Microsoft Windows systems, the executable is located in the **usr/** directory with the title **SpaceMissionSimVis.exe**.

In the distribution for Linux systems, the executable is located in the **usr/bin/** directory with the title **SpaceMissionSimVis**.

After launching, a graphical user interface should appear resembling the one in **Figure 1**.

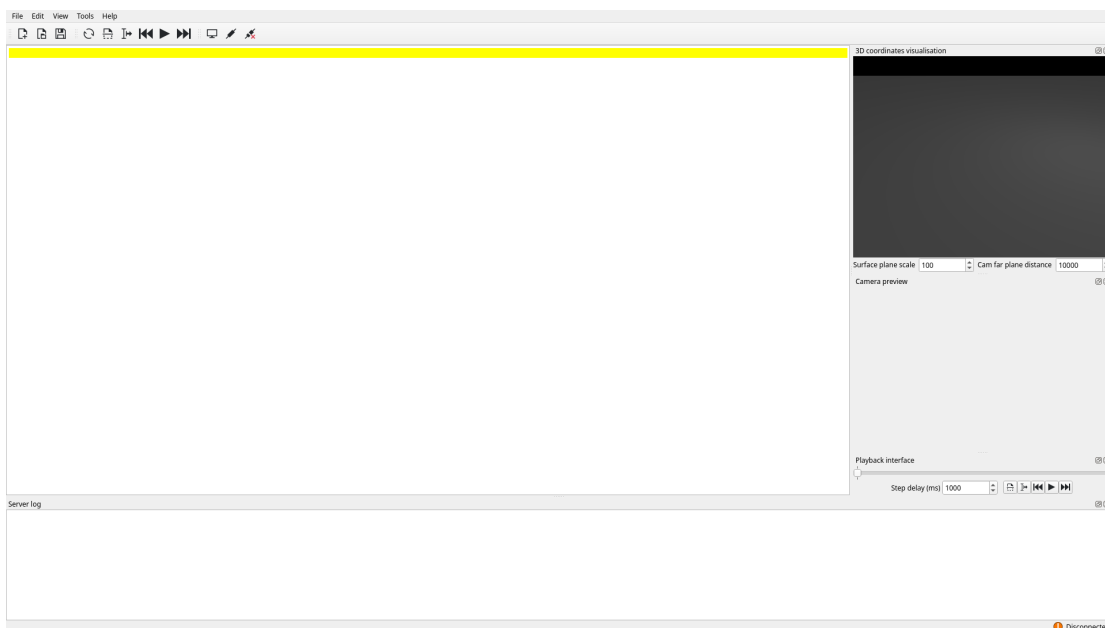


Figure 1: Example main window

2.3. Settings

The settings window can be accessed by navigating to the **Tools menu > Settings**. After opening, it should resemble the one in **Figure 2**.

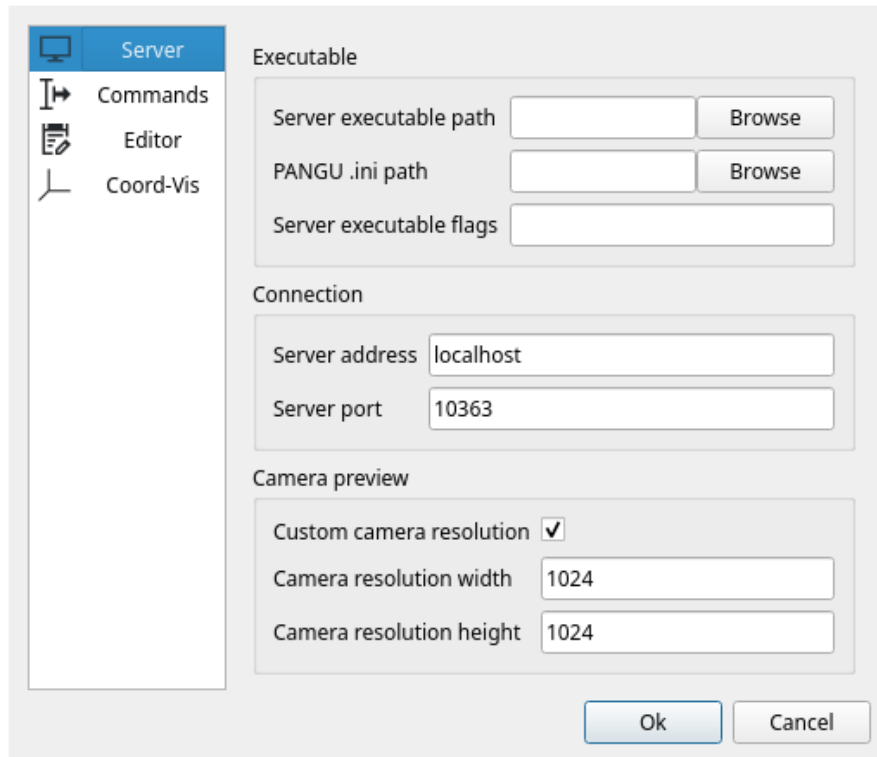


Figure 2: Settings window

Clicking 'Ok' saves the changes, 'Cancel' discards them. The changes can also be tried out without saving by keeping the settings window open and continuing to use the application. The panel on the left side of the window can be used to switch between the settings pages.

2.4. Creating, opening, saving flight files

Flight files can be newly created, opened and saved by doing one of the following:

- navigating to the **File menu** and selecting the appropriate action (including the option to **Open recent flight files**)
- clicking the appropriate icon, respectively, in the toolbar as seen in **Figure 3**



Figure 3: Highlighted file handling icons in the toolbar

2.5. Text editor

The application's text editor supports additional basic editing functionality like selecting, copying, cutting and pasting text, as well as undoing and redoing edits. It also supports the usual key bindings such as **Ctrl+Home** and **Ctrl+End** to jump to the start and end of the contents respectively.

2.6. Launching a PANGU server

For convenience, the user can specify a PANGU server executable path and additional flags in the settings window's Server page to then be able to launch the server right from the application. This is by no means necessary however, as the application can connect to any PANGU server using the server address and port specified in the same settings page. After setting the preferred path, flags and other settings like the path to the .ini file or custom camera resolution, the server can be launched by doing one of the following:

- navigating to the **Tools menu > Server > Start server**
- clicking the appropriate icon in the toolbar as seen in **Figure 4**

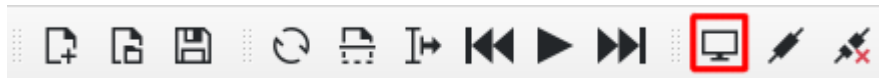


Figure 4: Highlighted 'Start server' icon in the toolbar

If the server fails to launch or crashes at any point, an error will be displayed to notify the user.

2.7. Server log

If a server is launched from the application, the Server log interface component will be populated with the outputs from the server. This can be helpful for debugging.

2.8. Connecting to a PANGU server

The application can be connected to a running PANGU server by specifying the address and port in the settings, then doing one of the following:

- navigating to the **Tools menu > Server > Connect to server**
- clicking the appropriate icon in the toolbar as seen in **Figure 5**

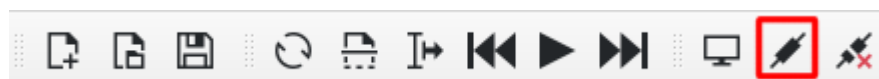


Figure 5: Highlighted 'Connect to server' icon in the toolbar

The application can be disconnected from the server it's currently connected to by doing one of the following:

- navigating to the **Tools menu > Server > Disconnect from server**
- clicking the appropriate icon in the toolbar as seen in **Figure 6**



Figure 6: Highlighted 'Disconnect from server' icon in the toolbar

The connection status is displayed on the right side of the status bar at the bottom of the window as seen in **Figure 7**.



Figure 7: Server status display

2.9. Scanning commands

To make use of the playback progress bar and 3D coordinates visualisation, the commands in the text editor must first be scanned to populate them with points of interest, which are commands that generate images. The editor can be scanned once by doing one of the following:

- navigating to the **Tools menu > Commands > Scan all commands**
- right-clicking inside the editor and selecting **Scan all commands** in the context menu
- clicking the appropriate icon in the toolbar as seen in **Figure 8**
- clicking the appropriate icon in the playback interface, which looks the same as the one in the toolbar

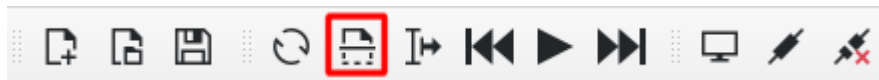


Figure 8: Highlighted 'Scan all commands' icon in the toolbar

Automatic scanning whenever edits are made can be toggled by doing one of the following:

- navigating to the **Tools menu > Commands > Toggle auto command scanning**
- clicking the appropriate icon in the toolbar as seen in **Figure 9**

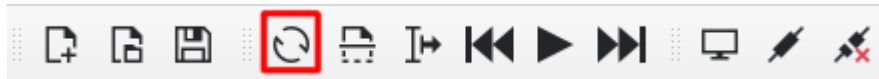


Figure 9: Highlighted 'Toggle auto command scanning' icon in the toolbar

2.10. Executing commands

A single command in the currently active line can be executed by doing one of the following:

- navigating to the **Tools menu > Commands > Execute line**
- right-clicking inside the editor and selecting **Execute line** in the context menu
- clicking the appropriate icon in the toolbar as seen in **Figure 10**
- clicking the appropriate icon in the playback interface, which looks the same as the one in the toolbar



Figure 10: Highlighted 'Execute line' icon in the toolbar

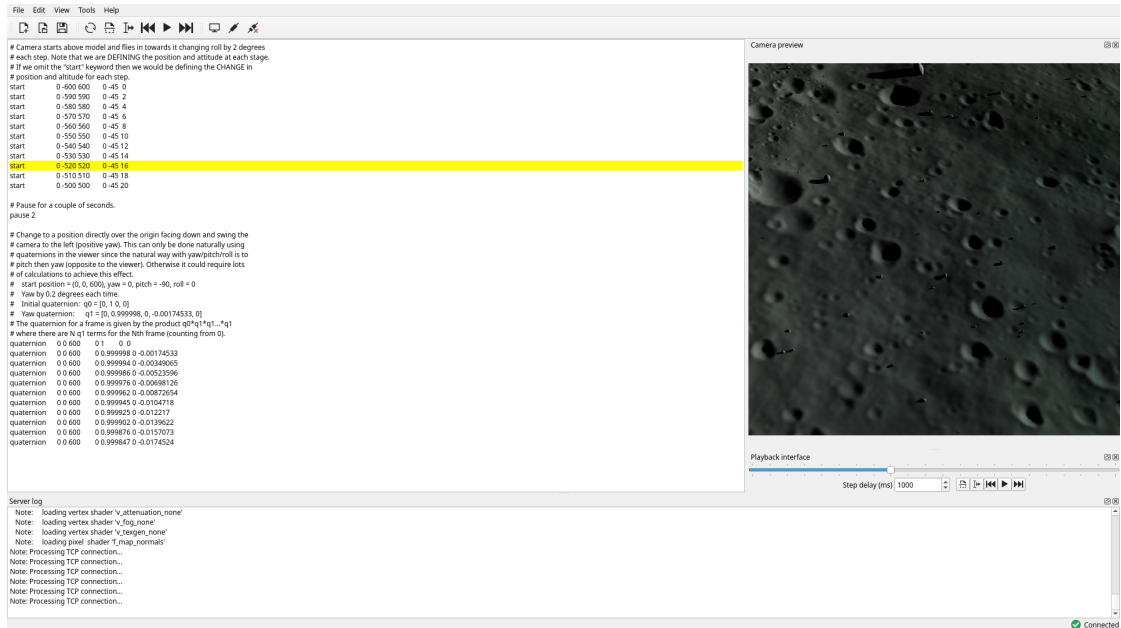


Figure 11: Example main window after executing a command that generates an image

A single command in the previous or next line can be executed by doing one of the following:

- navigating to the **Tools menu > Commands > Execute previous/next line**
- clicking the appropriate icon, respectively, in the toolbar as seen in **Figure 12**
- clicking the appropriate icon, respectively, in the playback interface, which looks the same as the one in the toolbar



Figure 12: Highlighted 'Execute previous/next line' icons in the toolbar

To emulate the simulation of the commands over time with little user input, command stepping can be used. It can be started and stopped by doing one of the following:

- navigating to the **Tools menu > Commands > Step through lines/Stop stepping**
- right-clicking inside the editor and selecting **Step through lines/Stop stepping** in the context menu
- clicking the appropriate icon in the toolbar as seen in **Figure 13**
- clicking the appropriate icon in the playback interface, which looks the same as the one in the toolbar

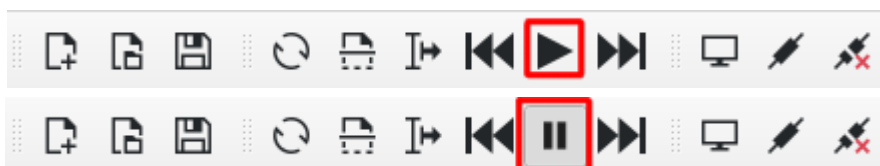


Figure 13: Highlighted 'Step through lines/Stop stepping' icons in the toolbar

The time delay between steps can be specified in the [Playback interface](#). If started at the end of the file, the stepping will start at the beginning of the file, which allows the user to easily restart the stepping after it is done by just clicking the same button again.

2.11. Supported commands

Command	Description	Parameters	Examples
start	position camera via euler angle and display the result	<ul style="list-style-type: none"> x, y, z - position of the camera a, b, c - yaw, pitch, roll respectively 	<pre>start 0 10 0 90 90 90 start 0.1 0 0 23.55 0 0</pre>
quaternion	position camera via quaternion and display the result	<ul style="list-style-type: none"> x,y,z - position of the camera q0, q1, q2, q3 - attitude quaternion with q0 being the scalar term 	<pre>quaternion 0 0 600 0 1 0 0 quaternion 0 0.5 0 0 0.9 0 0</pre>
update	display the camera result from the current position	N/A	update
pause	pause for an amount of seconds	s - seconds to pause for	<pre>pause 4 pause 32</pre>
set_time	sets the global relative to the global epoch (used for SPICE kernels)	t - the time to set	<pre>set_time 0.000000 set_time 45150.000000</pre>

When stepping through commands, unsupported commands will be ignored by default. To stop stepping and show an error message when an unsupported command is encountered instead, the appropriate setting can be enabled in the 'Commands' settings page.

2.12. Playback interface

The playback interface can be used to navigate the flight file without the editor itself, emulating a media player interface as seen in **Figure 14**. The interface includes:

- a playback progress bar
- a field to specify the minimum step delay between commands in milliseconds
- icons to scan all commands, execute the current, previous or next line as well as start/stop stepping through commands.

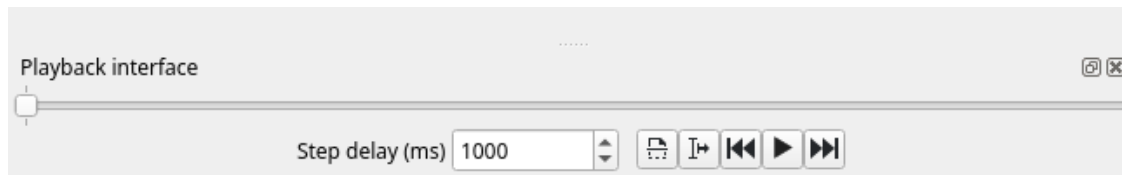


Figure 14: Playback interface

The playback progress bar's slider can be used to quickly move around the flight file using commands that generate images, once the text has been scanned to find any, as it affects the editor's currently active line position. The slider's position also updates according to the last executed image-generating command from the editor. When moving the slider, the commands being passed by the moving active line will get executed if possible. The time to send the command to the PANGU server and for it to then generate and send the image back is not instant, so some commands might be skipped while moving the slider around. This is done to avoid slowing down the application. However, letting the slider go after dragging it will always result in executing the last command to ensure accuracy.



Figure 15: Playback interface with points of interest and some progress in the flight file

2.13. 3D coordinates visualisation

The 3D coordinates visualisation is another way of viewing the progress of the flight file. The visualisation displays the expected resulting camera positions according to the arguments in the commands that generate images. This creates a 3D view of the expected 'flight path' the camera would have as seen in **Figure 16**. The camera of the 3D visualisation itself can be controlled by right-clicking and dragging to orbit around the active point and scrolling to zoom in/out. Fields to change the surface plane scale and the 3D camera's far plane distance are also provided. To modify the colours and point sizes of the visualisation, the appropriate settings can be adjusted in the 'Coord-Vis' settings page.

Please keep in mind that this is an experimental feature, as such it can have some unexpected behaviour and may change drastically in future versions.

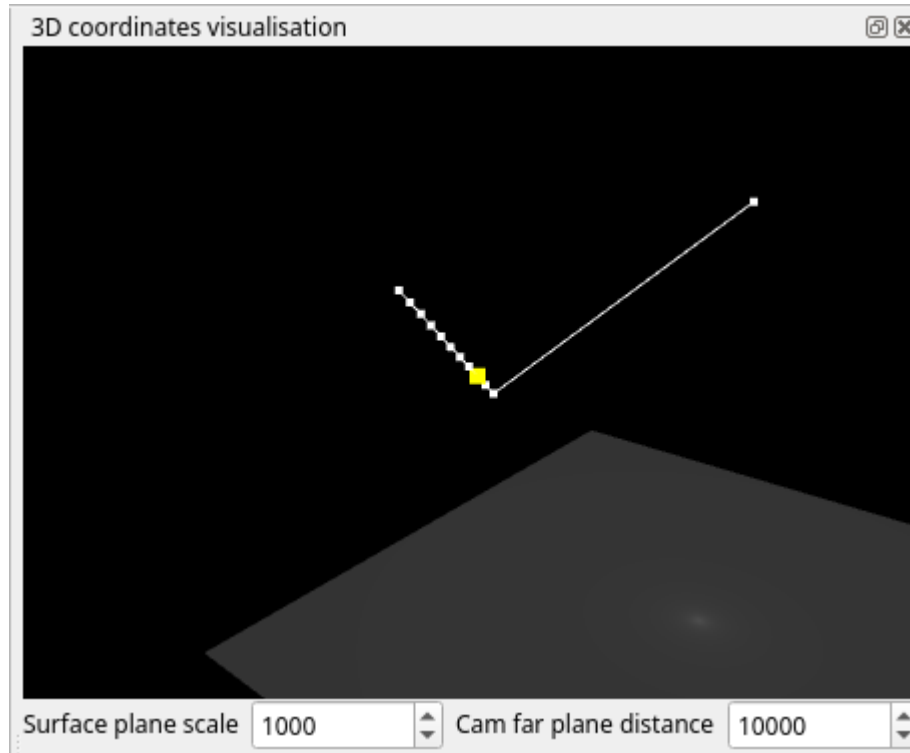


Figure 16: Playback interface with points of interest

2.14. Moving and resizing the interface elements

Some interface elements can be moved around, resized as well as completely taken out of the main window or closed.

Resizable elements include:

- 3D coordinates visualisation
- Camera preview
- Playback interface
- Server log

They can be resized by clicking and dragging their appropriate edges.

Movable elements include:

- 3D coordinates visualisation
- Camera preview
- Playback interface
- Server log
- toolbars

They can be moved around by clicking and dragging the area containing the component's name or, in the case of toolbars, the leftmost edge of the element. While moving an element, it can be attached and detached from the main window by moving it to and away from its edges or by clicking the appropriate button in the top-right corner of the element.

The positions, sizes and visibility of the interface elements will all persist between sessions.



Figure 17: Buttons visible on some of the movable/resizable elements. The leftmost one attaches/detaches the element to the main window, the rightmost one closes it.

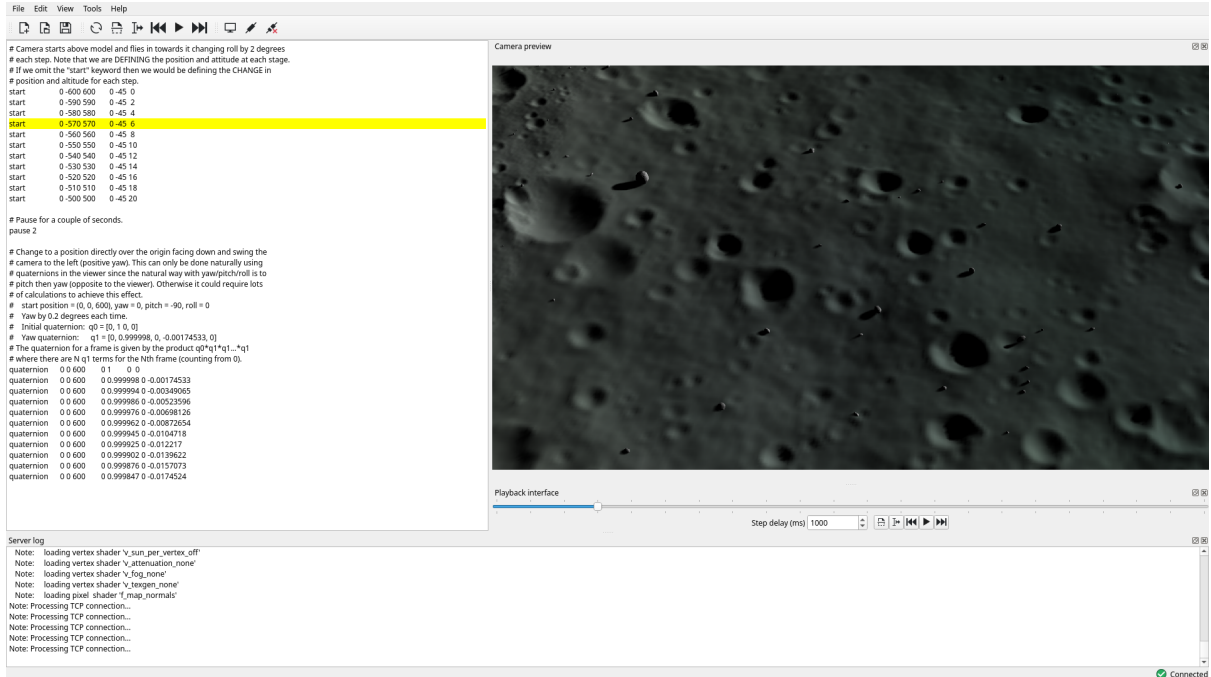


Figure 18: The main window after resizing and removing some interface elements

3. EXAMPLE USAGE INSTRUCTIONS

- 1) Navigate to `example_project` and run `make_model.sh` or `make_model.bat`, depending on your system
- 2) Open the application
- 3) Navigate to settings
- 4) Set the server executable path by clicking 'Browse' next to the input field and navigating to `example_project/` and selecting `view.sh` or `view.bat`, depending on your system. You may need to change the file filter in the browsing window from Executables (*.exe) to Bash scripts (*.sh) or to Batch scripts (*.bat)
- 5) Set the path to the `pangu.ini` file by clicking 'Browse' next to the input field and navigating to your own `pangu.ini` file, which should have been generated by your PANGU installation. This should provide appropriate paths to shaders and other usual configuration that the PANGU Viewer requires.
- 6) Set the server executable flags field to `-server`
- 7) Set the server address field to `localhost`
- 8) Set the server port field to `10363`
- 9) (Optional) Explore other settings and customise them to your liking
- 10) Save the settings by clicking 'Ok'
- 11) Start the server by clicking the 'Start server' icon
- 12) Connect to the server by clicking the 'Connect to server' icon
- 13) Open the example flight file by clicking the 'Open file' icon, navigating to `example_project` and selecting `flight.fli`
- 14) Scan all commands by clicking the 'Scan all commands' icon
- 15) Move the progress bar slider. The application should now execute commands in the flight file and display generated images as the slider is moved around
- 16) Explore any other features highlighted in the FEATURES AND FUNCTIONALITY section
- 17) Close the application
- 18) Close the server if it did not close by itself