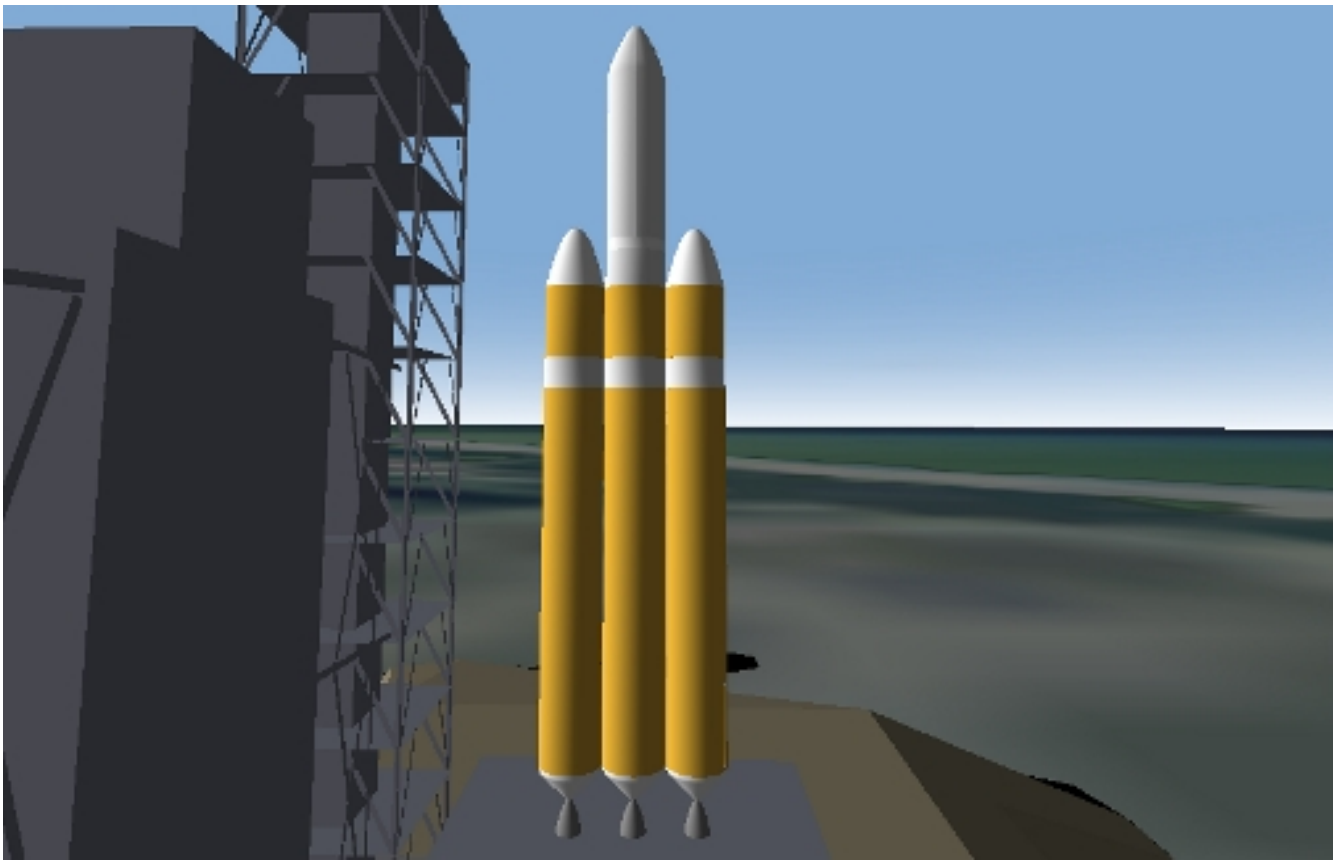


The Prometheus Launch Vehicle v0.7

By Zatnikitelman

Contents:

I. Legal

This addon is offered as-is without any warranty. The creator thereof accepts no liability whatsoever for your use of this addon.

The creator (Zatnikitelman) specifically allows this addon to be distributed freely for strictly non-commercial interests with the original creator to be conspicuously identified and credited. No payment may be recovered from distribution except to cover the costs of distribution (CD, DVD, etc.).

No part of this addon has been directly copied from any other source however inspiration has been derived from multiple sources. The only sources this addon specifically derives inspiration is from Boeing's Delta IV launch vehicle, Lockheed's Atlas V launch vehicle, and other common-core-technology vehicles.

Thanks are owed to Dr. Martin Schweiger for developing the Orbiter Space Flight Simulator for which this addon is specifically created; ar81 for creating the Orbiter Mesh Wizard which was useful in deriving various dimensions and correcting parts of meshes; the entire Orbiter community for at one point or another offering guidance and help without which this addon would not be possible; and Steven Glanville, the creator of Anim8or with which this addon was modeled.

II. Feedback

Please post any bug reports to my development forum on the Orbiter-Forum site on the topic that will be titled upon release of this addon: “PrometheusLV Bug Reports.” Other feedback can be posted in the generic PrometheusLV topic.

III. Description

The Prometheus Launch Vehicle is a heavy-lift vehicle designed to launch upwards of 40 metric tons into Low Earth Orbit. The design however is no bigger than Boeing's Delta IV Heavy. This achievement is possible due to the unique staging system employed.

The largest parts of the rocket are three Common Core System 1 (CCS1) boosters held together by frangible bolts. Each CCS is powered by an uprated RS-68 engine delivering 3.5MN of thrust with a specific impulse of 420s. During launch, the outer two CCS1s will provide the majority of thrust for the early stages of flight. Shortly after launch, the center CCS1 will throttle back to 55% thrust to conserve fuel. At 171 seconds, the center CCS1 will throttle up to 100% and at 176 seconds, the outer two CCS1s will separate along with the two-segment fairing.

When the outer CCS1s separate, they will reveal the second stage CCS2s. These utilize a modernized J-2 engine delivering 890KN of thrust with a specific impulse of 418s. While burning with the Center CCS1, the outer CCS2s will maintain 55% thrust until separation of the Center CCS1. The Center CCS2 is a bit different than the outer CCS2s. Early in the development process, it was realized that an engine delivering roughly 70% of the J-2 thrust while delivering almost one and a half more burntime would be ideal for the final stage of the launch vehicle. Thus, the RL-45 was selected delivering 600KN of thrust with a specific impulse of 435s.

Although the engines are different, the overall structure remains the same between the CCS2s allowing greater flexibility and reduced cost of production.

Options are being considered for an engine with higher ISP in the future.

Another craft is also featured with this launcher, it's called the Precision

Guidance Stage (PGS). This is designed to more precisely deliver payloads to the proper orbit and/or target (space station, other craft etc.) it makes use of a hypergolic engine delivering 40KN of thrust and an ISP of 330s. It uses a series of 8 thrusters grouped in pairs opposed across the x-z plane and angled 45 degrees from each axis facing outward to provide rotational control in all 3 axes and translation control in all 6 directions using only 4 thrusters per axis.

IV. Installation

The installation process assumes that you the end-user are familiar with basic computer aspects including the unpacking of .zip files and copying and pasting files and folders.

1. Extract the PrometheusLV_vx.zip file (x indicating the version number) into a temporary folder.
2. Copy and paste the entire contents of the extracted .zip file into the base Orbiter directory where you will be running the PrometheusLV. (Ex. C:\Orbiter\Orbiter06\Orbiter060929_base)
3. Launch the Orbiter Launch pad
4. Select a scenario that contains the Prometheus LV and run it.

V. Launch

The Prometheus can be considered a realistic launch vehicle in that you can't just light the engines point it roughly skyward and eventually get into orbit. That's what a guidance profile is for. Unfortunately during the development of Prometheus, I discovered a bug in Multistage2's guidance system which will put the final stage into an unrecoverable pitching-spin when the guidance ends. For this reason, I'm not including a guidance file so it is highly recommended that you download the LaunchMFD by Enjo. The rough procedure for launching into orbit using LaunchMFD is decribed below:

1. Bring LaunchMFD up in one of the MFD displays with Prometheus as

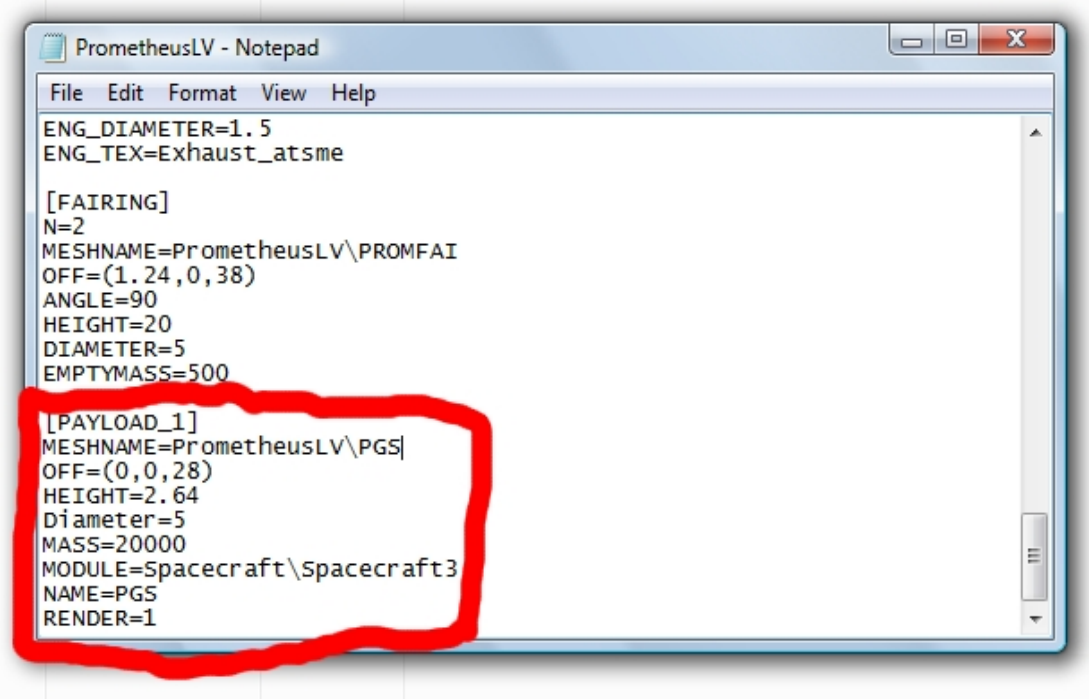
the focus vessel
2. Set your desired orbit by changing the parameters in the MFD.
3. Tap the numpad + button to briefly engage the "1st stage" engine*
4. As you rise off the pad, begin rolling so that the launch compass heading indicator is pointed toward the top of the screen.
5. Begin a slow pitch to about 70 degrees for most of the first part of the flight.
6. As you clear 30Km, begin following the pitch indicator on LaunchMFD.
7. Once you reach CCS1 separation, jettison the fairing by pressing "f" on your keyboard, this happens exactly 176 seconds into the flight.
8. Continue to follow LaunchMFD's pitch profile until you're either in orbit, or you run out of fuel. If you run out of fuel, it will be assumed that your payload is capable of inserting itself into orbit as will be explained in the Payload section of this document.
9. After you've achieved orbit, you have essentially two options depending on the payload you've configured as will be explained in the payload section. Your first option is to separate the Precision Guidance Stage from the "1st stage" and begin fine-tuning your orbit based on mission requirements. Your second is to use the fuel left in the "1st stage" to do rough orbital corrections, but as Multistage2 doesn't have translation thrusters, you will not have the precision you do with the Precision Guidance Stage solo. That being said however, the "1st stage" contains a much higher thrust engine, and higher fuel load than the PGS so may be useful early on-orbit operations.

VI Payloads

Because I do not know how to code in C++ yet, the payloads you launch with Prometheus have to use the sometimes complex Multistage2/Spacecraft3 payload description system. What follows is how to put your own custom payload on Prometheus in two different ways.

The first method is just using Multistage2's own payload system to place a payload on top of the "1st stage" of the launcher.
Open PrometheusLV.ini under *orbiter_directory*\config\Multistage in your

favorite text editor (ex. Notepad).
Scroll down to the [PAYLOAD_x] section (default is [PAYLOAD_1]) as shown:



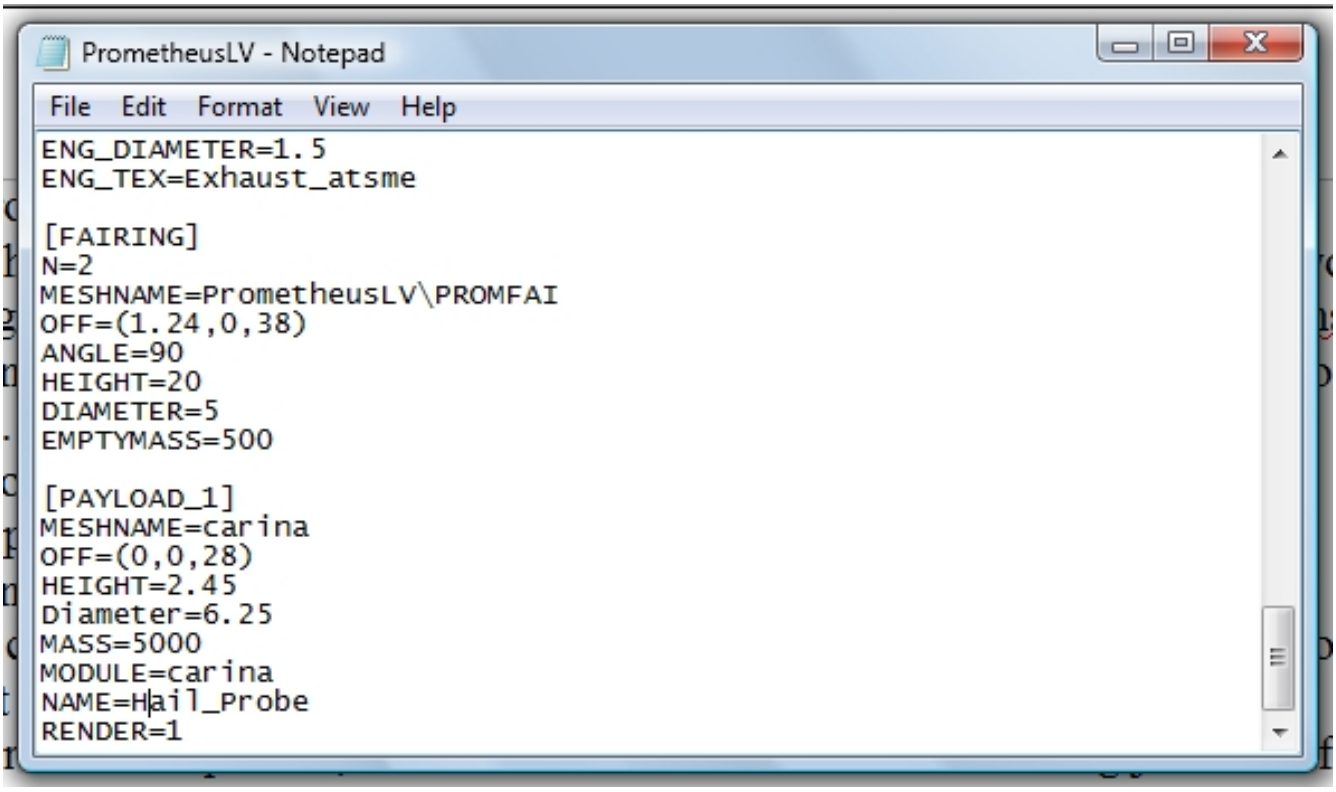
Let's look at each section:

[PAYLOAD_1]	The payload number designator up to 9
MESHNAME=	The name of the mesh of your payload
OFF=(x,y,z)	The position the mesh of your payload should be
HEIGHT=	The approximate height of the payload
DIAMETER=	The approximate diameter of the payload
MASS=	The mass of the payload in kilograms
MODULE=	The config file the payload will use on jettison
NAME=	The name the payload should take on jettison
RENDER=	Render the mesh before payload jettison

The example above is for the included PGS which itself carries a payload (more on this later)

Today, we'll configure a simple carina satellite payload
Start by finding the meshname of the probe which should be carina.msh
Enter probe after the = after MESHNAME
The next value is the offset of the payload relative to the rocket. The only

value we'll be changing will be the Z-axis. The reason is because the rocket flies “forward” along the Z-axis even though it appears vertical on the launch pad. Go ahead and make it (0,0,28) indicating 28 meters from the center of the rocket.
The height value really doesn't matter a whole lot, but change it to 2.45 (you can get this from the config file, read the z-dimension under “crosssections”).
Carina is 3.51 meters across the y-axis so go ahead and set the diameter to 3.51.
Carina has a mass of 5,000Kg so enter this for mass
Carina's module is simply carina
You can name it anything you want, but I'm naming it Hail_Probe (the probe must be hailed by all those wishing luck like our Carina)
The render is optional, but it looks much better after the fairing jettisons if there's something there. All this does is tell Spacecraft3 to show the payload's mesh before it is jettisoned and becomes its own vessel. Set to 1.
Your completed payload section should look like this:



As stated before, Multistage2 doesn't allow translation thrusters on stages so I've created a “carrier” of sorts called the Precision Guidance Stage. As it sounds, it allows more precise maneuvers than can be accomplished with the

stage alone.

Because this is a Spacecraft3 derived vessel, we'll be using Spacecraft3's payload system along with Multistage2's.

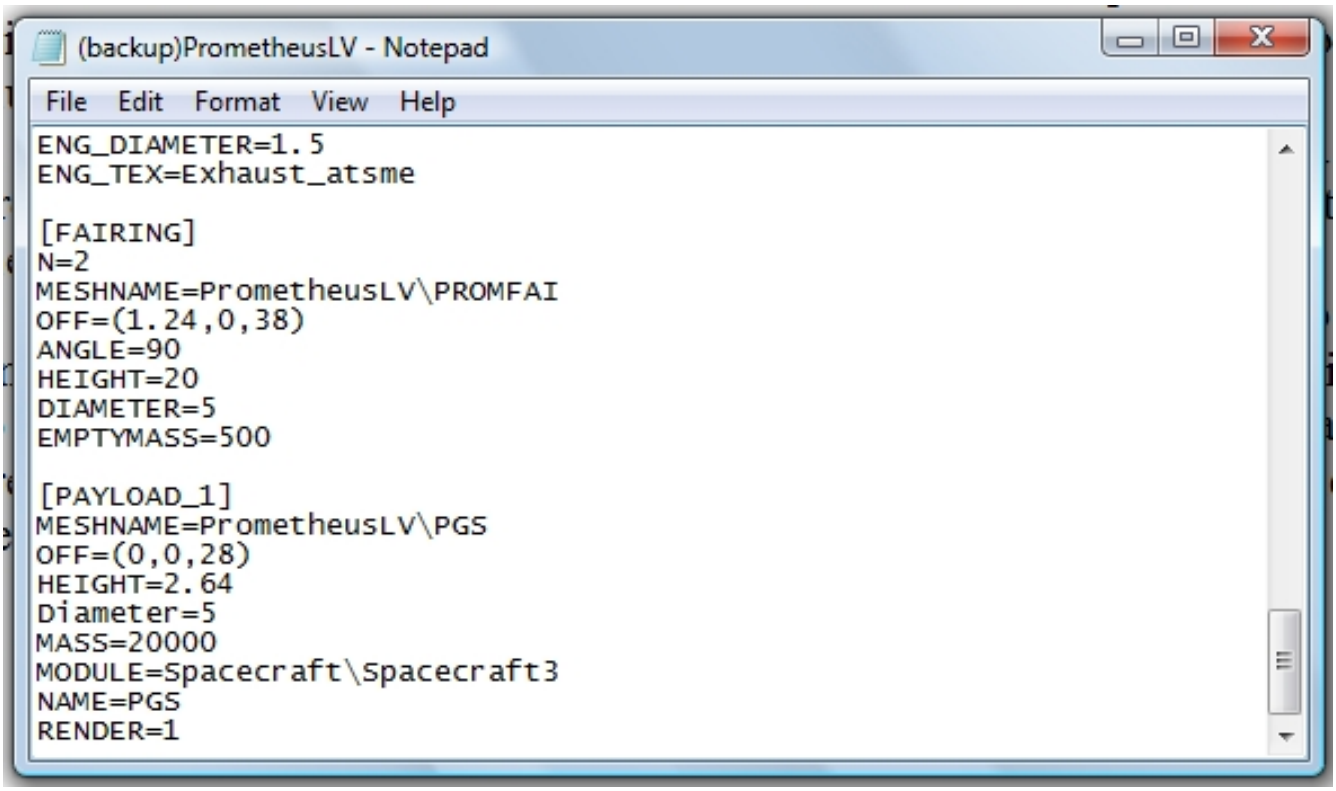
First, we need to define the payload we are launching in the PGS.ini file. Open PGS.ini located here: *orbiter_base*\config\spacecraft\PGS.ini. Scroll down to the [PAYLOAD_x] section (default is [PAYLOAD_0]). The payload description section is nearly identical to Multistage2's except two paramaters are missing, but two different ones are present.

Fill out the parameters like before for the Carina satellite. (again, this won't fit completely in the Prometheus fairing, but it won't affect anything other than appearance.)

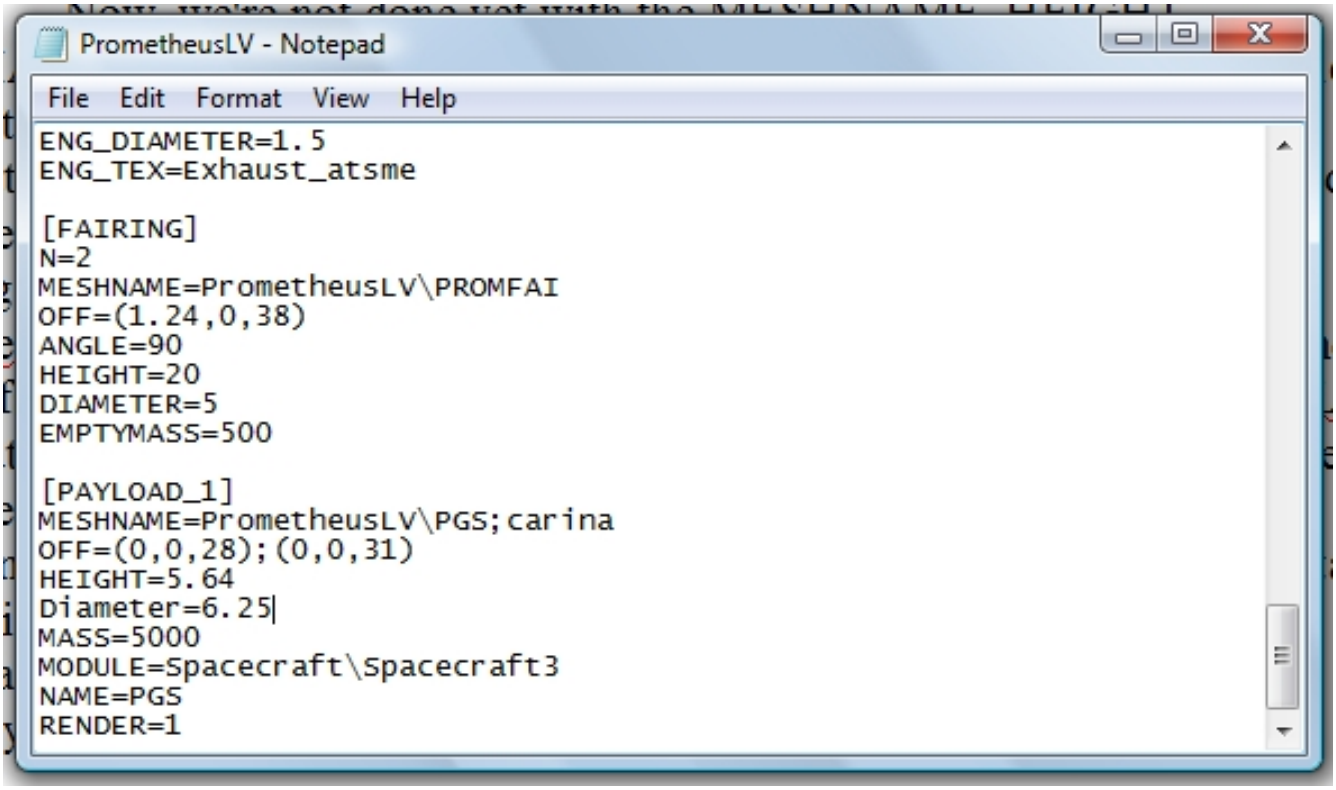
Notice that there are two new parameters at the bottom of the payload description. These are SPEED and ROT_SPEED respectively. Speed simply refers to how fast the payload will fly away from the PGS after jettison and Rot_speed refers to how fast the payload will be rotating. If you're using this to deliver supplies to a station and need a stationary payload for a robotic arm to grab onto, then I recommend you leave these at 0. Since Carina is a satellite that needs to free itself of the carrier craft, set the speed to 0.5. This will push the satellite away from the PGS without affecting the orbit too much.

Now that you've defined the payload the PGS will carry, we need to tell Prometheus that it's carrying a Spacecraft3 payload and that it has multiple meshes.

Open up PrometheusLV.ini as before and go to the payload section. This time, because you're using my PGS and the Prometheus is actually designed to carry the PGS, the MESHNAME, OFF, MODULE, and NAME parameters are set. If your file doesn't look like this already, go ahead and make the necessary changes: (image on next page)



Now, we're not done yet with the MESHNAME, HEIGHT, DIAMETER, or MASS parameters. When you modified the PGS.ini file, you entered an offset relative to the PGS craft. With Multistage, it won't automatically read the Spacecraft3 payload system and put your payload mesh on top of the PGS mesh so we have to do that manually.
Again using the Carina example, we add a semi-colon, and the new meshname of “carina” (no quotes). The offset value is a tad trickier. Find the offset used relative to the PGS and add that to the value in PrometheusLV.ini, but don't just increase the number, add a semi-colon and the offset value of the new mesh.
Simply update the remaining values so HEIGHT is now equal to the total height of PGS + your payload, DIAMETER is now equal to the size of the PGS, and MASS is equal to the mass of the PGS + your payload. Example below: (on next page)



There is a caveat to using this system however. Unfortunately, mesh files can not be the same name. So if you're launching three Carinas, then you have to use different mesh files. Because of this, I've added a Carina1.msh to the addon to provide an example of how to use multiple meshes/payloads. Unless stated otherwise in addon documentation, all you need to do is copy and paste the mesh file of the payloads you want in the same place and name it something unique. For example, you could have the Carina.msh, Carina1.msh, Carina2.msh up to a total of 5 meshes used.

VII. Future

Short Term Goals:

- Add textures to remainder of rocket
- Change engines and fairings from spline-based profile, to linear-profile
- Add more rockets to Prometheus line (like Delta IV)

Long Term Goals:

- Build .dll version of Prometheus

VIII. Changelog

090120 v0.7:

First release of PrometheusLV