Multistage 2015

Release Documentation – May 07th, 2017

By Fred18

Multistage 2015

Release Documentation – May 07th, 2017

By Fred18

Introduction

The idea of Multistage2015 is to recreate Vinka's Multistage2 module and update it with a number of new features.

First Release: October 27th, 2015

This Release: May 7th, 2017

What is Multistage?

Multistage is a module that allows developers who are not capable or willing to code in C++ to develop multistage launchers programming a large set of features of their launcher through a standard windows .ini file, simply editable using Notepad. The file is divided in sections which contain the main characteristics and features. Some of the entries in the ini file are mandatory for the module to work and to avoid the notorious CTDs, while some others are optional and are part of a spectrum of features which can be used and customized at pleasure of the developer.

Of course the module is not only intended to be useful for addon developers not capable of coding, but also to all those who are willing to focus for example on space probes or other parts and can exploit Multistage215's functionalities to their best.

This release also includes a Multistage2015 MFD which can be very helpful for flight performance measuring and very funny to use.

Installation

Simply extract the content of the compressed file into Orbiter's main directory, letting it overwrite, if needed.

Compatibility

This addon is compiled against and compatible with Orbiter 2016 version.

A known issue is present with Attachment manager addon. It is strongly recommended to disable Attachment manager before using Multstage2015.

Required Addons

Stage.dll 2010 by BrianJ is required for default jettisoned items. It is included in this package with the permission of BrianJ.

Keys

[J] for jettison boosters / interstages / stages

[F] for jettison LES / Fairing

[P] for toggling the autopilot

[CTRL]+[SPACEBAR] for activating developer mode

[SPACEBAR] when developer mode is active to reset the vehicle and reload the .ini file.

Hangar Keys: see relevant Hangar Section

Release Policy

Multistage2015 is distributed as FREEWARE. Its code is distributed along with the dll. Nobody is authorized to exploit the module or the code or parts of them commercially directly or indirectly.

You CAN distribute the dll together with your addon but in this case you MUST:

- Include credit to the author in your addon documentation;
- Add to the addon documentation the official link of Orbit Hangar Mods for download and suggest to download the latest and updated version of the module.

You CAN use parts of the code of Multistage2015, but in this case you MUST:

- Give credits in your copyright header and in your documentation for the part you used.

- Let your project be open source and its code available for at least visualization by other users.

You CAN NOT use the entire code for making and distributing the very same module claiming it as your work entirely or partly.

You CAN NOT claim that Multistage2015 is an invention of yourself or a work made up by yourself, or anyhow let intend that it is not made and coded by the original author.

You install and use Multistage2015 at your own risk, author will not be responsible for any claim or damage subsequent to its use or for the use of part of it or of part of its code.

Be also aware that I am not a professional programmer so code may be difficult to read and may present issues that I am not aware of.

Credits

The biggest credit goes to Vinka for the idea and the implementation of the multistage module which has lasted for decades now. This is anyway a new module, since no code from Vinka was available or used.

Credit to Face for pointing me to the GetPrivateProfileString code which boosted a lot the implementation of the module.

Credit to Hlynkacg because the OrientForBurn function he published in the forum inspired the attitude control function used in the module.

Credit to rcraig42 for isp and pressure reference value inputs for stage engines.

Credit to Boogabooga for the SLS example scenarios and for the updated SLS ini file.

Credit to Ripley for the first review of this doc.

Credit to Interceptor for pointing out and solving the "height from ground" issue.

multistage2.dll replacement

A fast way to turn all of the original multistage2 launchers in multistage2015 vessels is to replace directly the original multistage2.dll file.

In order to do this, after the installation of Multistage2015:

1) Go to your orbiter_root\Modules folder
2) Find the Multistage2.dll and rename it to Multistage2_BACKUP.dll
3) Make a copy of the Multistage2015.dll file
4) Rename the copy into multistage2.dll

This will intercept all multistage2 vessels. If you want the original Vinka's Multistage2 back, just delete the multistage2.dll file you've created and rename the backup file to multistage2.dll.

Description and Limitations

Number Limitations:

Stages:	10
Engines for each Stage:	32
Boosters:	10
Engines for each Booster:	4
Interstages for each stage:	1
LES:	1
Payloads:	10
Meshes for each Payload:	5
Mach effects	10
Vent effects	10
Launch effects	No limitations
Ullage for each stage:	No limitations
Explosive Bolts for each stage / booster	1
Fairing Elements:	10
Particle Stream Definitions:	13
Particles for each engine group:	2
Exhaust Textures:	16
Exhaust for each engine group:	1

Other limitations, FROM VINKA DOCS

Boosters
Boosters can be attached only to the first stage, You can define up to 10 groups of boosters. A group can

contain up to 10 boosters (each booster in the group shares the same characteristics), use only different groups if you want to define different characteristics such as the thrust or the burning time. Boosters are automatically fired when the first stage engine level reaches 95% of the nominal thrust or after a programmable delay. Boosters are automatically jettisoned when fuel mass is falls to 0.

Stages
Only one stage can be fired at a time, when the fuel mass of the current stage reaches 0, the stage is jettisoned and the next stage (if any) is ignited. If the first stage is running out of fuel and boosters are still thrusting, the first stage separation will be delayed until all boosters are jettisoned. You can define up to 10 stages.

Interstages
An interstage can be defined between stages. The interstage is dropped after previous stage jettison and before next stage ignition. Up to 9 interstages can be defined.

Payloads
A maximum of 10 payloads can be defined. You can program the focus so that, at the defined payload jettison, the vessel focus goes to this payload giving you the control of the payload.
This change of focus is not very eye friendly because the image jumps to the new focus vessel with a different position and size. If a fairing is defined, the fairing must be jettisoned prior to payload jettison. During the payload separation phase, the last stage remains active. Therefore, you can change the attitude, rotation and the speed before jettisoning the payload. If you have kept enough fuel, you can even restart the stage main engine to change the orbit. The first payload that will be jettisoned must be placed on top of the payload stack.

Fairing
A fairing can protect the payloads. This fairing can be composed of 10 elements maximum.

ALL PARAMETERS – INI FILE

MISCELLANEOUS

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[MISC]	Miscellaneous Parameters	Optional	
COG=xxx	Center Of gravity Elevation in Meters, used to display the launcher at a certain altitude above ground	Optional	No
GNC_DEBUG=0/1	If set to 1 shows some debug information in the Debug String	Optional	No – Information displayed are different
TELEMETRY=0/1	If set to 1 each minute of flight saves Telemetry data into a Comma Separated Values txt file. See Telemetry paragraph	Optional	No – Saved data are partly different
FOCUS=xxx	Can be set between 1 and the number of payloads defined when the specified payload is jettisoned. The focus will change from the multistage vessel to the newly created payload.	Optional	No
THRUST_REAL_POS=0/1	If set to 1 thrust from the engines is applied where the engines are and in the direction specified with engine_dir. This is valid for both stages and boosters	Optional	**NEW! HOT!**
VERTICAL_ANGLE=xxx	[degrees] Set the angle from vertical with which the vehicle is sitting on the ground	Optional	**NEW!**
Pad_Module=abc	Module name for Launchpad Module. If not present a default ghost Launchpad will be used. This can be very useful for add-on developers willing to have personalized pad for their launchers	Optional	**NEW! HOT!**

TEXTURE

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[TEXTURE_LIST]	User must declare all the texture files that are used by engine exhausts, the list will be preloaded by the dll		
TEX_x=abc	x is the progressive texture number. Abc means the name of the texture file (without the .dds) that will be rendered as exhausts. Maximum of 16 textures allowed	Optional	No

PARTICLE STREAM

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[PARTICLESTREAM_x]	Particle Stream Definition, x is the progressive number of the particle definition, from 1 to 13	Section is optional, parameters are mandatory if particle is defined	No
Name=abc	Name of particle stream, to be used later on	Mandatory	No
Srcsize=xxx	Size of the Particle	Mandatory	No
Srcrate=xxx	Rate of Particle Production	Mandatory	No
V0=xxx	Initial velocity of the emitted particles	Mandatory	No
Srcspread=xxx	Spread of the emitted particles	Mandatory	No
Lifetime=xxx	Life of the particles in seconds	Mandatory	No
Growthrate=xxx	Growth rate of the particles	Mandatory	No
Atmslowdown=xxx	Atmospheric slow down of the particle	Mandatory	No
Ltype=EMISSIVE/DIFFUSE	Ltype of the particle	Mandatory (if not specified a warning in the log file will be produced and the emissive type will be applied)	No
Levelmap=LVL_FLAT / LVL_LIN / LVL_SQRT / LVL_PLIN	Level Map of the particle	Mandatory (if not specified a warning in the log file will be produced and LVL_LIN will be applied	No
Lmin=xxx	Minimum ref Level	Mandatory	No
Lmax=xxx	Maximum ref Level	Mandatory	No
Atmsmap= ATM_FLAT / ATM_PLIN / ATM_PLOG	Atmospheric Map	Mandatory (if not specified a warning in the log file will be produced and ATM_PLIN will be applied)	No
Amin=xxx	Atmo min ref level	Mandatory	No
Amax=xxx	Atmo max ref level	Mandatory	No
Tex=abc	Texture name of the particle	Mandatory(if not specified a warning in the log file will be produced and contrail3 will be applied)	No
GrowFactor_size=xxx	Growing Particles factor relevant to Size (see dedicated chapter)	Optional	**NEW! HOT!**
GrowFactor_rate=xxx	Growing Particles factor relevant to Growthrate (see dedicated chapter)	Optional	**NEW! HOT!**

BOOSTERS

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[BOOSTER_x]	Booster Section, x is the number from 1 to 10	Optional	
N=xxx	Number of boosters in the defined boosters' group	Mandatory	No
Meshname=abc	Name of mesh of the boosters (remember that then the meshes abc_1, abc_2… abc_N will be searched)	Mandatory	No
Off=(xx,yy,zz)	[meters] Offset of the first booster mesh	Mandatory	No
Height=xxx	[meters] Height of the booster	Optional	No
Angle=xxx	[degrees] rotation of the first mesh in the group around Z axis	Optional (default value is 0)	No
Diameter=xxx	[meters] Diameter of the booster	Optional	No
Thrust=xxx	[Newton] Thrust of the single booster of the group (will be multiplied by N for total boosters group thrust)	Mandatory	No
Emptymass=xxx	[Kg] empty mass of the single booster of the group	Mandatory	No
Fuelmass=xxx	[Kg] fuel mass of the single booster of the group	Mandatory	No
Burntime=xxx	[seconds] Burn time of the boosters of the group	Mandatory	No
Burndelay=xxx	[seconds] Delay of booster group ignition	Optional	No
Speed=(xx,yy,zz)	[m/s] Separation speed of the boosters of the group (of the first booster in the group, the others will be symmetrical about Z axis)	Optional (default is (3,0,0))	No (default value added)
Rot_speed=(xx,yy,zz)	[deg/s] Separation rotational speed of the boosters in the group (of the first booster in the group, the others will be symmetrical about Z axis)	Optional (default is (0,-0.1,0))	No (default value added)
Module=abc	Module name of the jettisoned booster	Optional (default is "stage")	No
Eng_x=(xx,yy,zz)	Position of the engine of the booster, x is the	Mandatory	No

	progressive number, from 1 to 4 at maximum		
Eng_diameter=xxx	[meters] diameter of the exhaust of the engine	Optional (default is half of booster diameter)	No (default value added)
Eng_tex=abc	Name of the texture of the engine	Optional	No
Eng_dir=(xx,yy,zz)	Engine direction, exhaust will be rotated and this vector will be used in case of thrust_real_pos=1 *Note: only 1 parameter for all the engines is required. The direction will be applied to all the engines of the booster group*	Optional	**NEW!**
Eng_pstream1=abc	Engine first particle stream name	Optional	No
Eng_pstream2=abc	Engine second particle stream name	Optional	No
Curve_x=(xx,yy)	Thrust curve for booster, x is the progressive number, maximum 10 points are allowed. Xx=time of power Yy=thrust level	Optional	No
Expbolts_pos=(xx,yy,zz)	Explosive Bolts position	Optional	**NEW!**
Expbolts_pstream=abc	Name of the explosive bolts particle stream	Optional (mandatory if expbolts_pos is defined)	**NEW!**
Expbolts_anticipation=xxx	[seconds] Time of explosive bolts effect before booster cutoff	Optional (mandatory if expbolts_pos is defined)	**NEW!**

STAGES

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[STAGE_x]	Stage section, x is the stage number from 1 to 10, maximum 10 allowed, no holes in the sequence allowed, stage_1 is mandatory		
Meshname=abc	Name of the mesh	Mandatory	No
Off=(xx,yy,zz)	Offset of the mesh	Mandatory	No
Height=xxx	[meters] Height of the stage	Mandatory	No
Diameter=xxx	[meters] Diameter of the stage	Mandatory	No
Thrust=xxx	[N] Thrust of the stage	Mandatory	No
Emptymass=xxx	[kg] empty mass of the stage	Mandatory	No
Fuelmass=xxx	[kg] fuel mass of the stage	Mandatory	No
Burntime=xxx	[sec] time of burning of the stage	Mandatory	No
Ignite_delay=xxx	[sec] delay before stage ignition (not to be used with first stage)	Optional	No
Speed=(xx,yy,zz)	[m/s] speed of separation of stage once jettisoned	Optional	No
Rot_speed=(xx,yy,zz)	[deg/s] rotational speed of separation of stage once jettisoned	Optional	No
Module=abc	Name of the module of the stage once jettisoned	Optional (default is "stage")	No
Pitchthrust=xxx	[N] pitch thrust	Optional	No
Yawthrust=xxx	[N] yaw thrust	Optional	No
Rollthrust=xxx	[N] roll thrust	Optional	No
Linearthrust=xxx	[N] translational thrust. If present will add translational thrusters to the stage.	Optional	**NEW!**
Linear_isp=xxx	[sec] Useful for users willing to have a customized isp for linear thrusters as well, in order not to alterate the total dV of a stage. Note: this parameter is not present in the DMD, but only in the ini file	Optional	**NEW!**
Eng_x=(xx,yy,zz)	Engine position, x is the progressive number	Optional (default is (0,0,-height*0.5))	**Yes!**

Eng_x=(xx,yy,zz,tt)	from 1 to 32, maximum 32 allowed. New fourth optional parameter has been added. It's a scale parameter, in order to scale the exhaust of the specified engine, example ENG_3=(0,0,-10,0.7) will produce an exhaust 30% smaller than the others		
Eng_diameter=xxx	[meters] diameter of the exhaust of the engines	Optional (default is half of stage diameter)	No
Eng_dir=(xx,yy,zz)	Direction of the engine, exhaust will be rendered in the opposite direction, thrust will be applied in this direction if thrust_real_pos=1 is defined *Note: only 1 parameter for all the engines is required. The direction will be applied to all the engines of the booster group*	Optional	**NEW! HOT!**
Eng_tex=abc	Name of the texture of the exhaust	Optional	No
Eng_pstream1=abc	Name of the first particle stream engines	Optional	No
Eng_pstream2=abc	Name of the second particle stream of the engines	Optional	No
Reignitable=0/1	If set to 0 once the stage has been turned off it cannot be turned on again	Optional (default is 1)	**NEW! HOT!**
Battery=xx	[Hours] battery life in hours, if present battery will discharge while mission is proceeding and if battery gets to 0 the stage will be dead (no engines or rcs). If not present batteries will simply never discharge	Optional	**NEW! HOT!**
Ullage_thrust=xx	[Newton] ullage thrust in Newton, if present ullage are added	Optional	**NEW!**

Ullage_anticipation=xx	[sec] time of ullage ignition before engine ignition	Optional	**NEW!**
Ullage_overlap=xx	[sec] time of ullage remaining ignited once stage has ignited	Optional	**NEW!**
Ullage_N=xx	Number of ullage engines	Optional (mandatory if ullage is defined)	**NEW!**
Ullage_angle=xx	[degrees] angle of first ullage engine from the vertical	Optional (default is 0)	**NEW!**
Ullage_diameter=xx	[meters] diameter of ullage exhausts	Optional (mandatory if ullage is defined)	**NEW!**
Ullage_length=xx	[meters] length of ullage exhausts	Optional (mandatory if ullage is defined)	**NEW!**
Ullage_dir=(xx,yy,zz)	Direction of first ullage engine (the others will be symmetrical with Z axis)	Optional (mandatory if ullage is defined)	**NEW!**
Ullage_pos=(xx,yy,zz)	Position of first ullage engine (the others will be symmetrical with Z axis)	Optional (mandatory if ullage is defined)	**NEW!**
Ullage_tex=abc	Name of the texture for the ullage exhausts	Optional	**NEW!**
Ullage_rectfactor=xx	Scale of asymmetry of distribution of ullage exhausts. Applies only if ullage_N is even, the higher than 1 the more the exhausts are compressed laterally	Optional (default is 1, square distribution)	**NEW!**
Expbolts_pos=(xx,yy,zz)	Position of explosive bolts	Optional	**NEW!**
Expbolts_pstream=abc	Name of explosive bolts particle stream	Optional	**NEW!**
Expbolts_anticipation=xx	[sec] Time of anticipation of explosive bolts before stage cutoff	Optional	**NEW!**
Particles_packed_to_engine=xx	This command allows to pack all the particles definition of a stage into a single particle stream for a single engine. To get more details on its use check the dedicated paragraph below.	Optional	**NEW!**

INTERSTAGES

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[SEPARATION_XY] [ADAPTER]	Interstage definition, x is lower stage, y is upper stage It is also possible to have a payload adapter, defining x as last stage and y as last stage+1 or by setting the following parameters in a section named Adapter: meshname, off, height, diameter, emptymass		
Meshname=abc	Meshname	Mandatory	No
Off=(xx,yy,zz)	[meters] mesh offset	Mandatory	No
Height=xxx	[meters] height	Optional	No
Diameter=xxx	[meters] diameter	Optional	No
Emptymass=xxx	[kg] empty mass	Mandatory	No
Separation_delay=xxx	[sec] time between staging and interstage separation	Optional	No
Speed=(xx,yy,zz)	[m/s] translation separation speed	Optional	No
Rot_speed=(xx,yy,zz)	[deg/s] rotation separation speed	Optional	No
Module=abc	Module name of the jettisoned interstage	Optional (default is "stage")	No

LAUNCH ESCAPE SYSTEM

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[LES]	Launch Escape System Section	Optional	**NEW!**
Meshname=abc	Name of the mesh	Mandatory if les is defined	**NEW!**
Off=(xx,yy,zz)	[meters] offset of the mesh	Mandatory if les is defined	**NEW!**
Height=xx	[meters] height of the Les	Mandatory if les is defined	**NEW!**
Diameter=xx	[meters] diameter of Les	Mandatory if les is defined	**NEW!**
Emptymass=xx	[kg] emptymass of the Les	Mandatory if les is defined	**NEW!**
Speed=(xx,yy,zz)	[m/s] translation speed of jettisoned les	Optional	**NEW!**
Rot_speed=(xx,yy,zz)	[deg/s] rotation speed of jettisoned les	Optional	**NEW!**
Module=abc	Name of the module of the jettisoned les	Optional (default is "stage")	**NEW!**

FAIRING

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[FAIRING]	Fairing section	Optional	
N=xx	Number of fairing meshes	Mandatory	No
Meshname=abc	Name of meshes of fairing (remember that abc_1…abc_N will be searched)	Mandatory	No
Off=(xx,yy,zz)	[meters] Offset of first mesh	Mandatory	No
Height=xx	[meters] Height of fairing	Mandatory	No
Angle=xx	[degrees] Rotation angle of first fairing mesh	Optional (Default is 0)	No
Diameter=xx	[meters] Diameter of fairing	Mandatory	No
Emptymass=xx	[kg] empty mass of fairing	Mandatory	No
Speed=(xx,yy,zz)	[m/s] translation speed of jettisoned fairing	Optional	No
Rot_speed=(xx,yy,zz)	[deg/s] rotation speed of jettisoned fairing	Optional	No
Module=abc	Name of module of jettisoned fairing	Optional (default is "stage")	No

PAYLOADS

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[PAYLOAD_x]	Payload section, x is the progressive number from 1 to 10, maximum 10 allowed		
Meshname=abc	Meshname. Up to 5 meshes allowed	Mandatory	No
Off=(xx,yy,zz)	[meters] mesh offset, up to 5, one for each mesh	Mandatory	No
Height=xx	[meters] payload height	Mandatory	No
Diameter=xx	[meters] payload diameter	Mandatory	No
Mass=xx	[kg] payload mass	Mandatory	No
Module=abc	Name of module of jettisoned payload	Mandatory	No
Name=abc	Name of jettisoned payload	Mandatory	No
Speed=(xx,yy,zz)	[m/s] translation speed of jettisoned payload	Optional	No
Rot_speed=(xx,yy,zz)	[deg/s] rotation speed of jettisoned payload	Optional	No
Rotation=(xx,yy,zz)	[degrees] rotation of payload around the three axis	Optional	**NEW! HOT!**
Render=0/1	If set to 1 renders the payload mesh even if fairings are on	Optional	No
Live=0/1	If set to 1 creates a live payload instead of a dead mesh! If a payload is live the mass parameter can be avoided and the vehicle will load the live payload mass. If there is the need of manual overriding the live payload mass value, just keep the mass parameter	Optional	**NEW! HOT!**

EFFECTS

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[FX_MACH]	Section of Mach 1 effect	Optional	**NEW!**
Pstream=abc	Name of the particle stream effect	Mandatory	**NEW!**
Mach_min=xx	Minimum mach number for effect trigger	Mandatory	**NEW!**
Mach_max=xx	Maximum mach number for effect cutoff	Mandatory	**NEW!**
Off_x=(xx,yy,zz)	[meters] position of mach effect. X is the progressive number, maximum 10 allowed. Note: all of the mach effects will share the particle definition, the direction and the other parameters, it will be only replicated x times in the position desired by the user	Mandatory	**NEW!**
Dir=(xx,yy,zz)	Direction of mach effect exhaust stream	Mandatory	**NEW!**

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[FX_VENT]	Section for venting effect (before launch)	Optional	**NEW!**
Pstream=abc	Name of particle stream to be used	Mandatory	**NEW!**
Off_x=(xx,yy,zz)	[meters] position of exhaust number x. x is the progressive number from 1 to 10, maximum 10 allowed	Mandatory	**NEW!**
Dir_x=(xx,yy,zz)	Direction of exhaust number x	Mandatory	**NEW!**
Time_fin_x=xxx	Second of the countdown when vent number x will be turned off. Check will be performed up to MET=+5 sec.	Optional (default value is 0)	**NEW!**

Section / Parameter	Explanation	Mandatory/Optional	Changed from Vinka?
[FX_LAUNCH]	Section for fire effect at launch. It will create a circular symmetrical array of particles	Optional	**NEW! HOT!**
N = xxx	Number of particles in the effect (can be up to whatever is needed)	Mandatory	**NEW!**
Height=xxx	[meters] Height of the effects from the ground	Opational (default value is 0)	**NEW!**
Distance=xxx	[meters] Distance of the effects from the center of the array	Optional (default value is 0)	**NEW!**
Angle=xxx	[deg] Reference angle of the first effect	Optional (default value is 0)	**NEW!**
Pstream1=abc	Name of the first particle	Mandatory	**NEW!**
Pstream2=abc	Name of the second particle	Mandatory	**NEW!**
CutoffAltitude = xxx	[meters] height of the vehicle at which the effects stop, if 0 or not set the effects will stop as soon as the vehicle detaches from the ramp. Otherwise the effects level will be linearly interpolated up to the cutoff altitude	Optional (default value is 0)	**NEW!**

~~Section / Parameter~~	~~Explanation~~	~~Mandatory/Optional~~	~~Changed from Vinka?~~
~~[SOUND]~~			
~~Main_thrust~~	~~Specify the sound file to be played for main thrust~~	~~Optional~~	~~No~~
~~Hover_thrust~~	~~Specify the sound file to be played for hover thrust~~	~~Optional~~	~~No~~
~~RCS_Thrust_attack~~	~~Specify the sound fil to be played for RCS thrust~~	~~Optional~~	~~No~~
~~RCS_Thrust_sustain~~	~~Specify the sound fil to be played for RCS thrust~~	~~Optional~~	~~No~~
~~Air_conditionning~~	~~Specify the sound file to be played for air conditionning~~	~~Optional~~	~~No~~
~~Cockpit_ambience_1 .. 9~~	~~Specify the sound file to be played for cockpit ambience~~	~~Optional~~	~~No~~

No Longer Supported

ALL PARAMETERS – GUIDANCE FILE

Command	Explanation	Changed from Vinka?
engine(xx,yy,zz)	Set Engine Level from xx to yy in time zz	No
Roll(xx,yy,zz,tt,kk)	Vinka Roll Program	No
Pitch(xx,yy,tt)	Vinka Pitch step, from xx, to yy in time tt	No
Fairing() **Fairing(xx)**	Fairing() jettison fairing Fairing(xx) jettison fairing when altitude xx in km is reached. For example 10=fairing(80) will start checking at MET=10 seconds for altitude and will jettison fairing when altitude reaches 80km	**YES!**
Les() **Les(xx)**	The same as fairing, but with Les	**NEW!**
Disable(pitch|Roll|Jettison)	Disables pitch control, roll control or autojettison	No
Playsound(abc)	Plays soundfile abc	No
Target(xx)	Check for ApA. If the value xx (in km) is reached cutoff the engines	No
Jettison()	Simulate the [j] key press	No
Aoa(xx) **Aoa(xx,tt)**	Set attitude in order to get the desired xx aoa If parameter tt is present, the command last for time tt otherwise for 60 seconds. Time not applicable if loaded by MFD	**YES!**
Attitude(xx,yy,zz) **Attitude(xx,yy,zz,tt)**	Set Attitude of xx pitch, yy heading, zz roll If parameter tt is present, the command lasts for time tt otherwise for 60 seconds. Time not applicable if loaded by MFD	**YES!**
Spin(xx) **Spin(xx,tt)**	Spin the vehicle with the specified xx roll velocity If parameter tt is present, the command lasts for time tt otherwise for 60 seconds. Time not	**YES!**

	applicable if loaded by MFD	
Inverse()	Switches the flight mode from upright to upside down or viceversa *Note: its use in the guidance program will not be saved in the scenario file. Therefore if you use it, close the scenario and open it again the rocket will start to roll back to the original mode.*	**NEW!**
Engineout() **Engineout(xx)**	Turn off engine number xx. If no engine is specified it turns off the last engine set in the ini file	**NEW!**
Orbit(PeA,ApA) **Orbit(PeA,ApA,Inc)** **Orbit(PeA,ApA,Inc,Mode)** **Orbit(PeA,ApA,Inc,Mode,GravTurnPitch)** **Orbit(PeA,ApA,Inc,Mode,GravTurnPitch,Abside)**	The complete orbit autopilot! All these forms of calls are valid. PeA and ApA are in km Inclination is in degrees, positive to launch northward, negative to launch southward. If not specified or less than launch latitude azimuth=90° will be used Mode =1 for upright, -1 for upside down GravTurnPitch is the initial gravity turn pitch in case automatic value is not satisfactory Abside is the target altitude of cutoff in case automatic value is not satisfactory (note: this, may lead to less precision of the autopilot)	**NEW! HOT!!!**
Defap(xx)	Activates the Default Autopilots, input an integer number to choose which to activate (it's enough to count them in the HUD from left to right to get the right number)	**NEW!**
Glimit(xx)	When triggered it will spend 250 seconds checking if the vehicle is accelerating more than xx*G. If so it trhottles down the vehicle (warning, it will not throttle it back by itself)	**NEW!**

Destroy()	When triggered it simply deletes the Multistage2015 vessel from the scenario. It can be useful to avoid leaving the last stage of the rocket around when having long journeys	**NEW!**
Explode()	When called it manually triggers the rocket explosion, even on the launchpad	**NEW! HOT!**

ALL PARAMETERS – SCENARIO FILE

ITEM	Explanation	Note	Changed from Vinka?
CONFIG_FILE abc	Vehicle config file, path relative to orbiter root	Mandatory input by user	No
CONFIGURATION 1/0	Flag for vehicle configuration: 0 for landed, 1 for flying. For first scenario setup just put Configuration 0 will set the vehicle to its initial status	Mandatory input by user	No
FAILURE_PROB xx	Probability of failure of the vehicle	Optional	**NEW!**
GUIDANCE_FILE abc	Guidance file to load on startup	Optional	No
TELEMETRY_FILE abc	Reference telemetry file to be viewed in the MFD	Optional	**NEW!**
COMPLEX	If present, it activates the complex flight that: 1) randomizes engine performances. 2) Will create aerodynamic forces which will try to turn the rocket out of the gravity turn. If the wind angle gets too wide while drag is still significant (>500kN) the vehicle will be destroyed	Optional	**NEW! HOT!**
GROWING_PARTICLES	If present, it activates the growing particle effects (see dedicated chapter)	Optional	**NEW! HOT!**
ALT_STEPS xxx,yyy,zzz,ttt	[meters] Altitude Steps manual setting (See notes of the following section)	Optional	**NEW!**
PEG_PITCH_LIMIT xxx	[degress] Peg Pitch Limit manual setting (See notes of the following section)	Optional	**NEW!**
PEG_MC_INTERVAL xxx	[seconds] Peg Major Cycle Interval (See notes of the following section)	Optional	**NEW!**
HANGAR	If present creates the Multistage2015 Hangar	Optional	**NEW!**

	with the rocket inside (see relevant paragraph)		
CRAWLER	If present creates the Multistage2015 Crawler with its Launchpad underneath the rocket (see relevant paragraph)	Optional	**NEW!**
CAMERA	If present creates the Multistage2015 Camera Vessel (see relevant paragraph)	Optional	**NEW!**
DENY_IGNITION	If this is present the stage won't reignite (it's used in automatic when a not reignitable stage is used)	Optional (automatically saved)	**NEW!**
RAMP	If present it avoids the creation of a new Launchpad underneath the rocket	Optional (automatically saved)	
ATTMSPAD	Used if the rocket is attached to the pad brought by the Crawler	Optional (automatically saved)	
GNC_RUN 1/0	If 1 autopilot is running, otherwise autopilot is off	Optional (automatically saved)	No
MET xxx	Current MET	Optional (automatically saved)	No
BATTERY xxx	Battery residual charge for the current stage	Optional (automatically saved)	**NEW!**
STAGE_IGNITION_TIME xxx	MET for current stage ignition	Optional (automatically saved)	No
GNC_AUTO_JETTISON 1/0	Flag for automatic jettison	Optional (automatically saved)	No
CURRENT_BOOSTER / STAGE / INTERSTAGE / PAYLOAD xx	Number of current booster, stage, interstage or payload	Optional (automatically saved)	No
STAGE_STATE xx	Flag for current stage state: 1 stage shutdown, 2 stage ignited, 3 stage waiting for ignition	Optional (automatically saved)	No
FAIRING 1/0	Flag for fairing status	Optioanl (automatically saved)	No

SOME MORE NOTES

The New Orbit Guidance Program

One of the main new features of Multistage2015 is without any doubt the new Orbit call for the guidance program, but how exactly does it work?

It takes the vehicle from ground to final orbit in full sequence, only requiring as little as target perigee, target apogee and target inclination as inputs.

The first thing to know it's that the Orbit call divides the ascent in 5 phases, based on altitudes:

The Null Phase, from 0 to first altitude Step:

Nothing happens, the rocket simply climbs and clears the tower.

The Roll Phase, from first altitude Step to second altitude Step:

Roll program, note that for Orbit program if the roll isn't over at the moment of reaching the second altitude step it's not a big deal, everything will work anyway. The Roll program will turn the vehicle to the desired azimuth and will leave it at the Gravity Turn Initial Pitch.

Initial Pitch Phase: from second altitude Step to third Altitude Step:

In this phase the launch vehicle turns to the gravity turn initial pitch. In case the initial pitch is small it is advisable to raise the third altitude step in order to have a smoother and more realistic initial turn.

Gravity Turn Phase: from third altitude Step to fourth Altitude Step:

Here gravity turn takes place, it's an open loop phase, where the rocket is kept with proper roll and heading and the only relevance for pitch is the aoa.

PEG Phase: from fourth altitude Step upward:

This is the key step, that will take the rocket from the fourth altitude step and bring it to its target orbit

For Earth, default Altitude Steps are: 100m, 350m, 1.400m, 35.000m . For other bodies they are scaled on a planet mass proportion.

Altitude Steps can be managed through scenario file and/or the MFD.

In order to have an exact solution of the PEG equations the rocket always points toward a precise abside.

If in the orbit call no abside (or 0) is specified, the abside is chosen via a default procedure as follows: if the target perigee is above 80 km then the target abside will be the perigee, otherwise it will be the apogee. I digged around orbiter apis but could not find an appropriate function to get the "almost zero drag" altitude, so the autocheck works smoothly only for Earth, if launching from another body significantly different from Earth it could be better to specify the abside in the orbit call (last parameter).

If on the contrary an abside is specified in the call the guidance program will target the abside and the relevant vertical speed for cutoff conditions. Usually this is useful for particular vehicles, like space shuttle was, which needs to reach cutoff with a significant upward velocity at a certain altitude. Note that this usually leads the autopilot to be slightly less precise on the final reached absides' altitude.

By default the PEG major cycle has a 0.1 seconds interval (for Apollo was 10 seconds...), but also this value is customizable through scenario and/or the MFD.
Between one interval and the other the pitch is calculated as a linear variation of the parameters and the time, (please see also the original NASA document relevant to PEG). The estimation cycle of PEG has been a bit revisited and is based on DeltaV prediction that can be computed quite exactly in orbiter. Please note that since the frequency is 10Hz, linear estimation is done only for 1 tenth of seconds, 10% of the overall time.

The last point is that in order to avoid out unrealistic or unexpected behavior of the autopilot, the range of pitch at which PEG can turn the vehicle is limited. The value is called PEG Pitch Limit and by default is

35 degrees (plus or minus). This value can be modified as well via the MFD or via the proper input in the scenario file.

Computational Evaluation of Gravity Turn

What is the gravity turn? It is called gravity turn the procedure of maneuvering the vehicle in order to maintain the direction of thrust aligned with the velocity vector. Why is it called "gravity turn"? because when you lift off for examples from Earth, you pitch down a few degrees your velocity vector will start to "fall" down towards Earth's surface. In order to avoid excessive aerodynamic stress on the vehicle the vehicle itself shall be pitched down to follow the velocity vector triggering the velocity vector to fall even more and so on. The entire curve described in this phase is characterized by the first disturbance to the vertical attitude, the gravity turn initial pitch. A high disturbance will result in a more rapid gravity turn with a low apex of the described curve of trajectory. A small disturbance will result in a much slower rate of turn, letting the apex higher. For a rocket climbing to orbit the sooner it is possible to pitch down the better it is, but with planets with consistent atmosphere such as the Earth it cannot be done too early to avoid the increase of aerodynamic stress.

For this reason a particular attention has been paid to the gravity turn: when the scenario opens up the guidance computer of the ms2015 module takes all the parameters and tries to simulate the gravity turn until it finds the minimum pitch value that leads the rocket to have an altitude of at least the fourth altitude step, from which the PEG algorithm will take over. It then adds a 5% just for safety (if the rocket does not reach the proper altitude PEG will never start and the gravity turn will lead the rocket to crash back on earth at full thrust). BUT the computational evaluation of gravity turn is good but not perfect. It cannot know if for example engines are being throttled down for maxQ reasons, and each step is so dependent on the other that even if the results are quite good they cannot be taken as gold, also because actually the best conditions at which the rocket is left at PEG takeover depends also on the final orbit we want to achieve: for a low orbit of 160x160 km we will want to be almost horizontal for maximum tangential acceleration. For an high orbit of 300x300 km we will want to be still pointing up and gaining some more vertical speed to get to 300km. Therefore the computational calculated value usually works, but it is more a "suggested" value, that can be later optimized by the user. In the orbit call of the guidance program users can set this at their pleasure, in order to find the optimal value. I've made hundreds of tests about this, but since all the rockets are different and the firing sequences are different it cannot be made completely closed to the user inputs, and by specifying which is the "suggested" value the user can have an idea of which value shall be used for that particular orbit. If the user does not input anything the guidance computer will use the suggested value.

Complex Flight

When the complex flight option is activated (either by scenario file or via the MFD) an estimation is made for each engine of the rocket (boosters included) which will set the parameters (amplitude and period) of thrust oscillation during the whole powered flight. You can monitor this oscillation for each engine through the MFD –THR screen where you can check the performance together with the thrust and the throttle. This is a very nice parameter for randomizing the flights a bit, in order not to have the very same situation for the same scenario, but always a bit of difference.

If Complex Flight is active also an additional aerodynamic force will be applied to the vehicle, which will try to take its nose away from the gravity turn, as it happens in reality. If the angle with the wind (AOA and Slip Angle) gets too wide while drag is still high (above 500 kN) the aerodynamic force will prevail and the vehicle will loose attitude control and destroy itself.

Failures

Failure probability can be set only via scenario file. If higher than 0 a probability estimation is calculated at the beginning of the simulation. If a failure happens then some conditions are checked: if the first stage is running when the failure happens there is a fixed probability of 25% that the rocket will actually explode… so sit there and cross your fingers. In the other cases (not explosion or not the first stage running) one engine will be shut down and it will not be possible to restart it anymore. If Orbit program is activated, it will try to reach the orbit with the new situation.

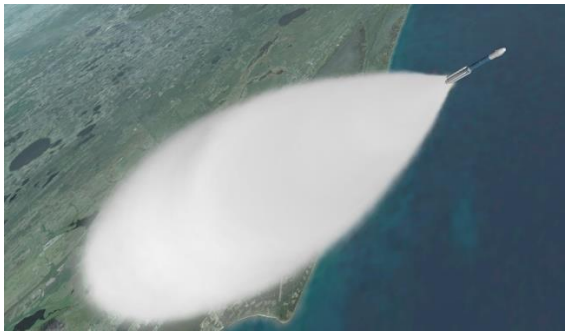
Telemetry

If Telemetry=1 is set in the ini file, then each minute a telemetry file .txt with comma separated values will be saved, in particular the following parameters will included in the file:

Met – Altitude – Speed – Pitch – Thrust – Mass - Vertical Speed- Acceleration

Note that the file can anyway be saved and loaded manually through the MFD at anytime.

Growing Particles Effect



A peculiar effect was implemented in Multistage2015 which is the growing particles effect. Basically size and growthrate of particle streams can change with the altitude changing of the vehicle.

Two parameters can be set in the ini file, as seen in .ini details above but in order for the effect to be triggered the word GROWING_PARTICLES must also be present in the scenario file in the vehicle section, so by default the effect is off. This is because the particle effect may affect graphic performance and it should be avoided to have it triggered by mistake.

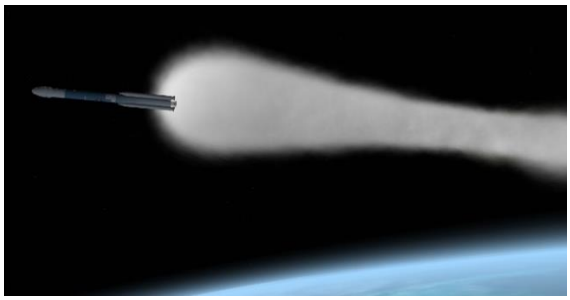
The formulae governing the effect are:

$$G.R._{CurrAlt} = G.R._{SeaLvl} + G.R.Factor * \\ * \log_{10}\left(\frac{Pressure_{SeaLevel}}{\text{Pressure}_{\text{CurrentAltitude}}}\right)$$

And

$$Size_{CurrAlt} = Size_{SeaLvl} + S.Factor * \\ * \log_{10}\left(\frac{Pressure_{SeaLevel}}{Pressure_{CurrAltitude}}\right)$$

Both can be set positive or negative so particles may decrease or increase in both size or growthrate with increasing altitude.



The position of emitted particles is also updated at size variation: in order to keep the particle ideally "tangent" to the engine emitting it if particle size increases then the particle stream is generated half of the size farther in engine direction, and this happens for the whole ascent.

If the effect is triggered particles updating happens with 1hz frequency, so once per second. In order to avoid visual interruptions of the particles flow a minimum time of overlap of the "old" and the "new" particle stream had to be placed. Currently the overlap time is set to 0.1s.

Particles packaging

A new parameter for stages have been added which is the Particles_packed_to_engine. This call allows to save a lot of graphic performances by merging the particle streams of a stage into a single stream. The call explicitly refers to the engine to which the unified stream shall be referenced to.

For example a definition like this:

[…]
eng_1=(-1.8347,3.3135,-32.408)
eng_2=(-1.8347,-3.3135,-32.408)
eng_3=(1.8347,-3.3135,-32.408)
eng_4=(1.8347,3.3135,-32.408)
eng_5=(0.,0,-32.408)
eng_diameter=2.7
ENG_PSTREAM1=growingcont
particles_packed_to_engine=5
[…]

Assigns a single particle stream (named growingcont) to engine n.5 (as can be seen it is the center engine of the array.

However, engine arrays may not have a center engine. For example the Space Shuttle had 3 main engine so no center engine could be picked.

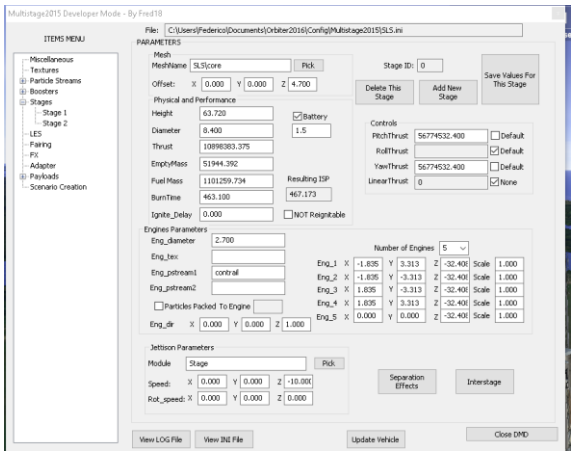
Therefore there is the possibility to set the single particle stream generation at the center of the engine arrays. This can be achieved by setting the particles_packed_to_engine call to a negative number. A call like the one in the example above can be set also to

Particles_packed_to_engine= -2

This will lead to have level of particles driven by engine 2 but particles generated from the center of the stage engines array.

Developer Mode Dialog (DMD)

By pressing [CTRL]+[SPACEBAR] the Developer Mode Dialog (or DMD) will open and the rocket will enter in the Developing Mode. This will also be highlighted by a large red text that will show up on the screen. This is still experimental but the first results of the tests are very promising.



Each section of the ini file will be displayed in the dialog and users will have the ability to create and customize all the parameters of their launchers in a purely intuitive way.

It will also be possible to update the vehicle to see live the changes made, together with the possibility for example to hide the entire or part of the fairing in order to check payload position inside of it.

While in the developer mode the vehicle can also be reloaded and updated by just pressing [SPACEBAR].

Together with the various sections of the ini file there is an additional section that allows to create in a few clicks a launch scenario and save it. If no saving name is specified the scenario will be save inside the Multistage2015 scenarios folder with the name of the launcher and the system MJD as reference.

Every time the DMD is open by the user a backup of the ini file is automatically created in the Config\Multistage2015\Backups folder, in order to let every mistake to be "recoverable" ☺ .

A direct ini file and saver and a direct log viewer are also available inside the DMD.

Some notes on the DMD:

1) Once closed it cannot be opened again during the same Orbiter session. If you need it again you have to close and reopen Orbiter.
2) When the user edits any parameter in each window, to have it saved to the ini file the

relevant save button in the section must be pressed. If not, any view switch of the sections will make the user to lose his changes.

3) If deleting an item, it is mandatory to go to the root of the relevant section and press its "Save" button, otherwise the changes will be lost.
4) Some of the parameters are NOT entirely comprised in the DMD. i.e. Stage engines can be up to 32, vent and mach effects up to 10 etc. Only the first are available in the DMD, if more are needed just edit the ini file!
5) In the boosters and fairing section if you pick the mesh name using the "pick" button remember that the name of the file shall be stripped of the "_X" number part!
6) When resetting the rocket with a live payload some weird behavior may occur. It is strongly suggested to set everything relevant to the payload with the "dead" version of it, and once happy change it to "live"
7) The DMD is still experimental. It seems to work quite well but the suggestion is anyway to recheck the ini file for the definitive version of your addons.

What if I have a live payload and I need to add specific parameters for it in the scenario file?

In this case simply load the scenario with the live payload as it is. Once loaded close the simulation and get the CurrentState.scn file. It will contain the launcher and the payload attached to it, so you can modify directly the payload sections a you wish, rename the CurrentState.scn to whatever you please and launch it again to enjoy the ride!

Log File

Please note that in the orbiter.log file a LOT of information will be noted during your flights. From mesh names, to engine positions, stage ignition, even failures calculation results. So if anything does not behave in the way you thought, please first give a check to the orbiter.log file, that can be a big help.

clbkGeneric Calls

For addon developers, in order to get useful data from the multistage rocket (i.e. to sync custom pad animations or to allow live payload autopilots to sync the targets) some clbkGeneric responses are implemented, in particular:

msgid=VMSG_USER prm=1 to obtain the MET (double)

msgid=VMSG_USER prm=2 to obtain the Autopilot status (bool, active or inactive)

msgid=VMSG_USER prm=3 to obtain Orbit Program autopilot targets (VECTOR3, x=target apogee, y=target perigee, z=target inclination)

Default Exhaust and Contrail particles

Following you can find the default "Exhaust" and "Contrail" particle streams definitions:

Contrail:
size: 8
rate: 5
v0: 150
spread: 0.3
lifetime: 8
atmslowdown: 3
ltype: Emissive
levelmap: lvl_psqrt
lmin: 0
lmax: 0.5
atmsmap: atm_plog
amin: 1e-6
amax: 0.1
texture: Contrail1

Exhaust:
size: 4
rate: 20
v0: 150
spread: 0.1
lifetime: 0.3
atmslowdown: 2
ltype: emissive
levelmap: lvl_psqrt
lmin: 0
lmax: 0.5
atmsmap: atm_plog
amin: 1e-6
amax: 0.1
Texture: Contrail3

Multistage 2015 MFD

Together with the vehicle module, an MFD mode module is provided.

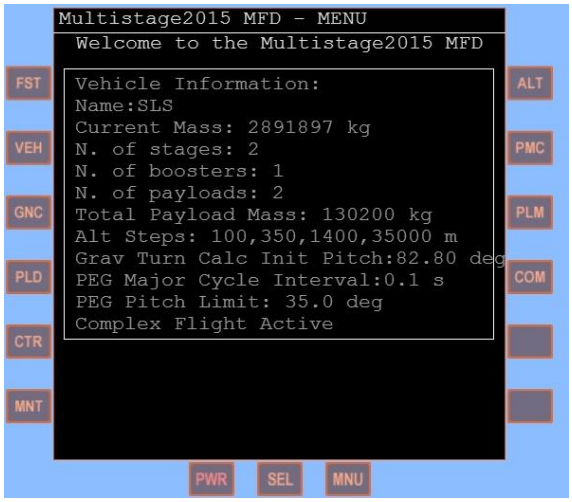
To activate the MFD mode, once installed the addon in orbiter directory, open orbiter, and in the Launchpad go to Modules tab – Miscellaneous and activate Multistage2015_MFD.

The Multistage2015_MFD has a number of screens for setting, monitoring and interact with any multistage2015 vehicle.

The left column of buttons is alwas the same for every screen and it's basically the menu of the MFD. The right column of buttons represent the screen specific commands.

Let's see the details and the commands screen by screen.

Flight Settings Screen - FST



Shows some general vehicle information and gives the possibility to manually set the altitude steps, the Peg Major Cycle Interval, the Peg Pitch Limit and toggles the Complex flight on/off. It also shows the calculated gravity turn initial pitch, for reference if a manual input wants to be set in the guidance program.

Recapping:

[ALT] – Sets the altitude steps, input must be XXX,YYY,ZZZ,TTT

[PMC] – Sets the Peg Major Cycle interval (input in seconds)

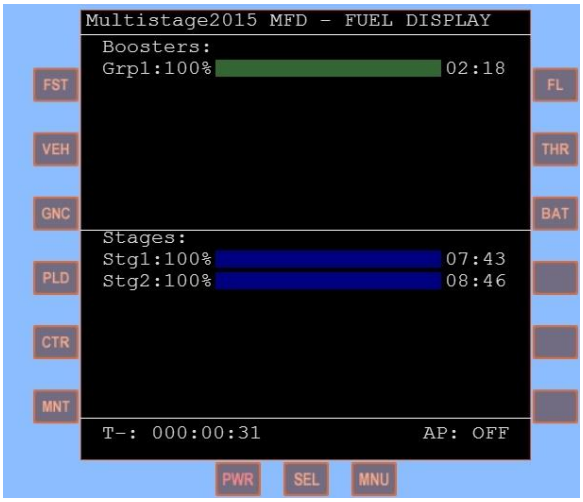
[PLM] – Sets Peg Pitch Limit (input in degrees)

[COM] – Toggle Complex Flight on / off

Vehicle Screen – VEH

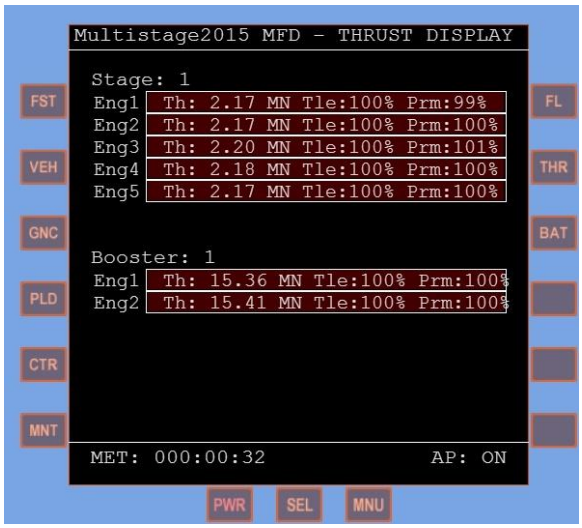
This screen contains actually 3 screens inside, that can be chose via the right column buttons:

The Fuel Screen FL:



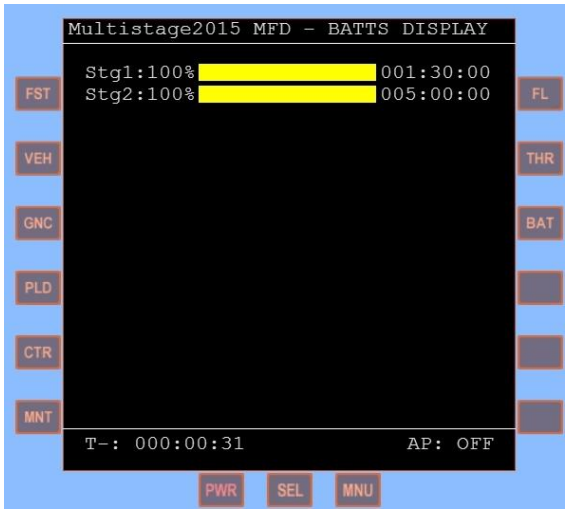
Which shows fuel percentage and time remaining for all the tanks in the rocket.

The Thrust Screen – THR:



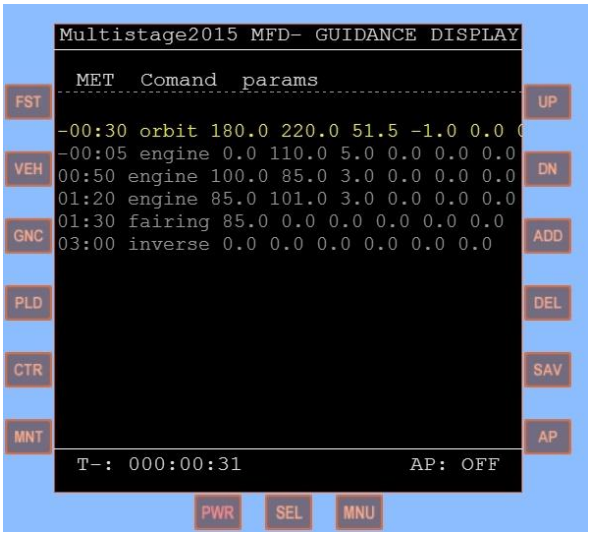
Which shows engines thrust, throttle and performance

And Batteries Screen - BAT



Which shows batteries status of each stage.

Guidance Screen - GNC



This is a very important screen because it gives the opportunity to both check and interact with the guidance program.

You can move across the preset commands through the [UP] and [DN] buttons, delete a specific command through the [DEL] button or add a new one with the [ADD] button.

If you want to keep the guidance file once prepared you can simply press [SAV] and a guidance file will be saved in orbiter_root\config\Multistage2015\Guidance\ directory with the following syntax:

Name of vehicle_ sysMJD_GNC.txt

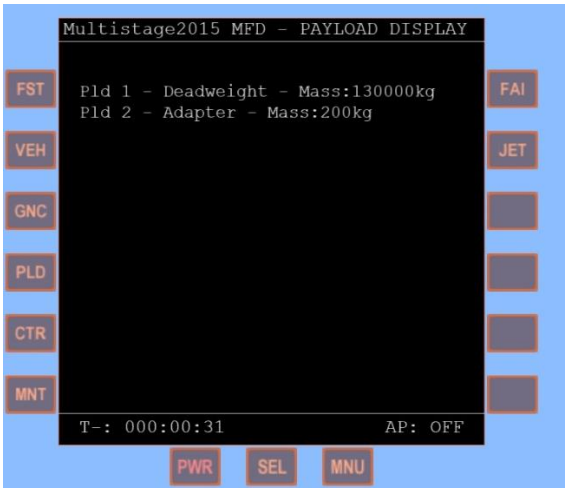
The [AP] button tolgges the autopilot on or off.

Please note that before the launch, if you add a command with negative time it will set the Countdown to that time, i.e. adding

-30=orbit(180,220,51.5,1)

Will set the mission timer to T-30 if nothing has been set with a previous time.

Payload Screen – PLD



This screen shows the information of the payloads onboard the vehicle.

[FAI] – Simulates keypress [F] so fairing or les jettison

[JET] – Simulates keypress [J] so booster, stage, interstage or payload jettison

Control Screen – CTR

This screen gives the opportunity to check if the vehicle is pointing in the direction which is given by the autopilot and with what error.

Moreover, with the right column keys gives the opportunity to toggle particular attitude controls if something is not working properly and the pilot needs to take over, or if the user simply wants to test himself in following the proper attitude.

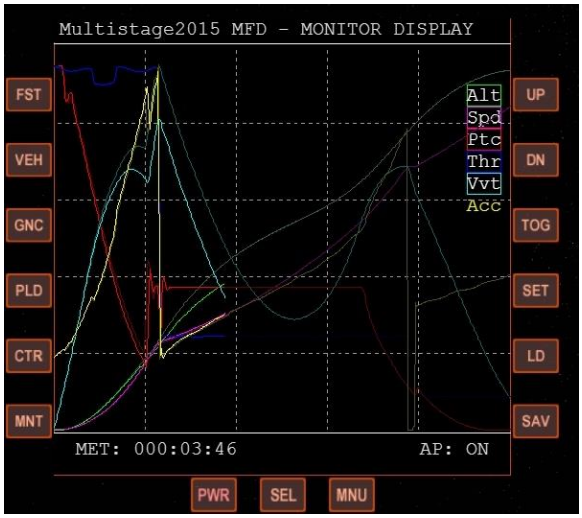
[ATT] – Toggle Autopilot Controls for all the three axis

[PIT] – Toggle Autopilot Control for pitch

[YAW] – Toggle Autopilot Control for yaw

[ROL] – Toggle Autopilot Control for roll

Monitor Screen - MNT



The monitor screen shows livetime graph of most of the parameters of telemetry.

It also can show with dimmed colors a preset telemetry values in order to have a reference to check.

The scale range is by default automatic but can be set manually throught the appropriate button.

In particular the user can move through the times shown with the [UP] and [DN] buttons.

Each of the graphs can be turned off by pressing the [TOG] button

It is possible to manually set the range of each graph by pressing the [SET] button. Min,max format is required for this

[LD] – will load a telemetry file from the HD, please note that the path to the file shall be relative to orbiter_root path

[SAV] – will save a telemetry file in orbiter_root\Config\Multistage2015\Telemetry directory with the following syntax

Name of vehicle_ sysMJD_TLM.txt

External Reference Vessel

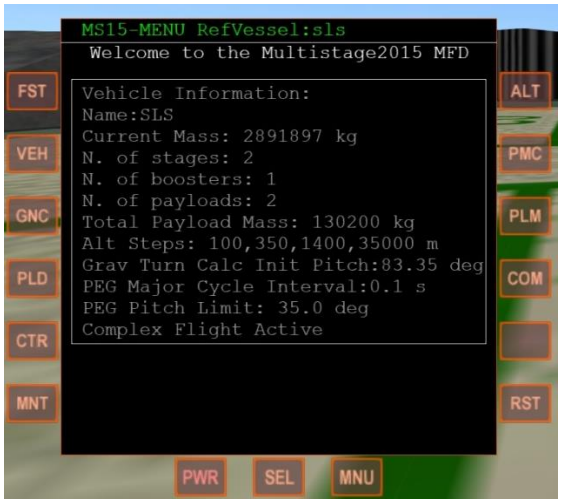
If we are sitting inside a non Multistage2015 vessel we can point the MFD to a Multistage2015 vessel inside the scenario and operate with it exactly as if we were in the Multistage launcher.

This comes particularly in handy if we are launching "live" payloads and we want to enjoy the ride from the flight deck of our payload and interact with the launcher at the same time.

Also it gives option to simulate a Mission Control Center from the ground.

In order to operate this, while we are sitting in the non multistage2015 vehicle we must open the Multistage2015_MFD and click the [SEL] button on top left. A dialog will open and we have to input the name of the vehicle we want to stick the MFD to. Only names of Multistage2015 vessels currently present in the scenario will be accepted.

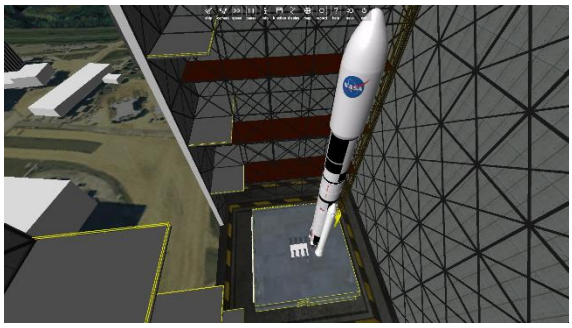
The fact that the MFD is not set to the current focus vessel has a graphical highlight because all the screen titles will become green with the reference vessel name shown.



It is possible to reset the external reference clicking the [RST] button in the Flight Settings Screen, in the lower right corner.

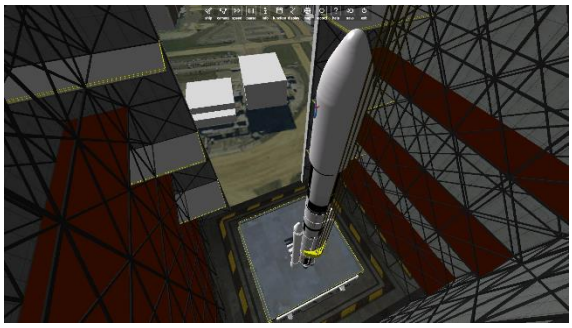
HANGAR

If the word "HANGAR" is present in the scenario the new Hangar designed for giving a proper "birth" to the new launchers designed by users will be loaded and the rocket will be placed hooked on its crane.



The Hangar will be placed exactly at the coordinates of the launcher.

Every kind of activity is possible while here, from testing to updating the vehicle, or whatever is liked.



Below a list of keys of the hangar. IMPORTANT NOTE: the modules are set so you can use those keys DIRECTLY from the Multistage vessel without changing the active vessel to the hangar, so you can move around and do anything you may need remaining "inside" the Multistage vessel.

[CTRL]+[L] = Toggle Hangar Lights

[SHIFT]+[UP] = Crane Up

[SHIFT]+[DOWN] = Crane Down

[SHIFT]+[LEFT] = Crane Left

[SHIFT]+[RIGHT] = Crane Right

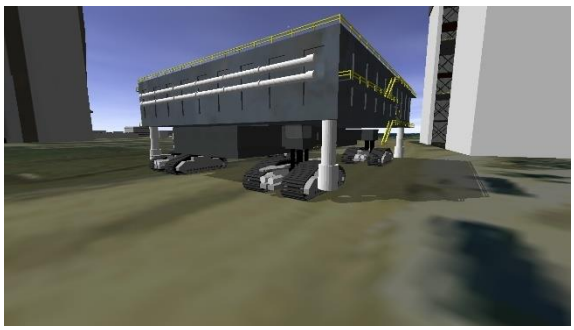
[CTRL]+[SHIFT]+[UP] = Crane Forward

[CTRL]+[SHIFT]+[DOWN] = Crane Backward

[CTRL]+[D] = Drop the rocket and attach it to the pad if present (Note: attaching to the pad works only if the ctrl D command is thrown within the Multistage vessel, if done from the Hangar vessel it just drops the rocket.)

CRAWLER

If you put the word "CRAWLER" in your scenario file you'll find a crawler just below the launcher at the opening of the scenario. By default the crawler will have a Launchpad attached to it and its lifter will be fully raised. You can drop the rocket on the Launchpad, take it to your preferred launch site, drop the launch pad and launch the rocket. This is a bit of an eye candy part of the addon, but I think it's quite entertaining.



Control Keys for the crawler are:

[UP] = start moving forward

[DOWN] = start moving backward

[LEFT] = start rotating left

[RIGHT] = start rotating right

If any movement is on progress and any of the keys above is pressed the movement will stop

Tracks animations will stop if time is accelerated.

[CTRL]+[SHIFT]+[UP] = start raising the crawler (and if attached the Launchpad)

[CTRL]+[SHIFT]+[DOWN] = start lowering the crawler.

[CTRL]+[D] = Drops the Launchpad, works only if the crawler is lowered

[CTRL]+[G] = reattaching the Launchpad, works only if the crawler and the Launchpad are within 5 meters from each other.

[CTRL]+[L] = toggle crawler's lights

When the rocket is dropped on the Launchpad the height of the rocket can be adjusted by pressing (always inside the multistage vessel) :

[SHIFT]+[UP]/[DOWN] = move the rocket up or down on the pad.

Crawler Config file

You can control speed, angular speed and which mesh or which module to load for the launch pad changing the relevant values:

Speed=xxx – Movement speed of the crawler [m/s]

AngSpeed=xxx – Angular speed of the crawler [deg/s]

Pad_Mesh = abc – Mesh name of the pad to be loaded by the crawler

Pad_module = abc – Module of the pad loaded by the crawler (if Pad_module is present, Pad_Mesh is ignored).

If no Pad is wanted to be loaded by the crawler for any reason write in the cfg file:

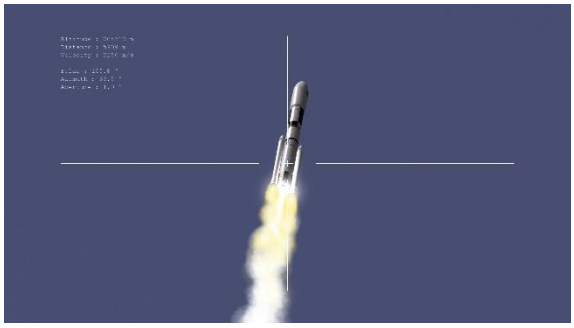
Pad_module=NOPAD

If the pad is not wanted to be attached to the Crawler the word NOATTPAD should be specified in the crawler section of the scenario file (once saved). This is anyway saved automatically if the pad is detached manually.

If users are willing to build their own custom pad to be moved around by the crawler it will be useful to know that the "touch" point of the crawler for the pad is at 9.405 m from the ground when lifter is lowered and the lifter has an excursion of 2 meters. For best usage it is strongly suggested to name the custom pad vessel "MS_Pad". The ID of the attachment point between crawler and Launchpad is "PadCrawl".

CAMERA Vessel

If you put the word "CAMERA" in your scenario file (section of Multistage2015), when you open the simulation you'll find among the list of the ships available also the "MS_Camera" vessel. Jump in to see your rocket from there.



By default the Camera vessel will be created at +0.01 deg Latitude and +0.03 deg Longitude. If other coordinates are preferred, instead of the simple word "CAMERA" write in the scenario file:

CAMERA xxx yyy where xxx is the delta latitude and yyy the delta longitude.

Toggle the HUD ([H] or [CTRL]+[H] keys) to see also the aiming and some information about camera and the vehicle distance.

[CTRL]+[R] = input new reference vessel to follow

[ARROW UP] / [ARROW DOWN] = if vehicle close enough (camera aperture > 0.1°) zoom one step in or out. Maximum 4 steps

SLS Heavy Lifter Example

Together with the package an example is provided. It is actually quite a realistic example, since it is based on the early design of SLS, and all the proper calculation relevant to mass, fuel and engine performance has been made. This has been initially developed by me, and finalized with all the final implementations by boogabooga, consider it a present for having downloaded the Multistage2015 module ☺

Thanks

Thanks to all the members of the OF community and to the Iltalian FOI community who helped me with their answers about the most difficult implementations of the module.

A particular thanks to some beta testers who patiently tried the module and pointed me towards the various glitches present during developing. To name just some of them (please those who are not included don't get disappointed): boogabooga, Interceptor, crisbeta, romanasul. 4throck, Longjap.

Thanks to the patience and support of my ~~girlfriend~~ wife (yep, we got married in the meantime!) who has been asking me about this project as if it was a real life one, understanding my passion for it.

A big thank you goes to my dad, who put me on his legs on a Sunday morning when I was just a kid and with a piece of paper and a pencil started to explain me "how does a rocket engine work".

And as usual the biggest thanks goes to Dr. Martin Schweiger for creating Orbiter!

Have fun up there! Fred18