

HOW TO CUSTOMIZE THE ARIANE 6 ROCKET

Here you will find all the advice and basic steps for adapting and customizing an Ariane 6 rocket. This tutorial is limited to explaining how to change and modify the rocket's fairing, and how to install your satellite or payload.

Even a beginner can do it !



All you need is a drawing application that can open and save textures, which in Orbiter are in **DDS** (**S**urface **D**irect **D**raw) format. This format divides the image into 4x4 pixel blocks (4x4x3 bytes) and compresses each block into 8 bytes. One million pixels become 500,000 bytes (500 KB). If two **DDS** images have the same number of pixels, they have the same file size. The size of a **DDS** file is a multiple of 8 bytes.



TABLE OF CONTENTS

I - GENERAL INFORMATION.....	page 2
II - LET'S GO (the tutorial).....	page 4
III - WHY MAKE IT SIMPLE?.....	page 8
IV - HOW TO CHANGE THE PAYLOAD.....	page 9
V - SYLDA and DISPENSER.....	page 10
VI - MODIFY THE ROCKET SPECIFICATIONS.....	page 11

I - GENERAL INFORMATION

Two "generic" Ariane 6 rockets are available as templates for customizing your rocket :

- a **6.2** version equipped with **two boosters** and a **short fairing**
- a **6.4** version equipped with **four boosters** and a **long fairing**

You can use either of these two versions of the Ariane 6 rocket to **customize** it (the fairing) and install the **payload** or satellite of your choice.

I'll explain how to do this in this short tutorial. Don't worry, it's very easy.

But before we get into the details, let's quickly review (for beginners) the files required to have this rocket in Orbiter..

A) FILES REQUIRED TO HAVE AN ARIANE 6 ROCKET

Here is the list of files we will modify to create our new custom rocket :

- 1) A **scenario** file (e.g.: *Ariane 6.2 (Generic).scn*) with a **SCN** extension
- 2) Two **configuration** files:
 - One file with a **CFG** extension (e.g. : *A62.cfg*)
 - One file with an **INI** extension (e.g. : *A62.ini*)
- 3) Several **meshes** files (e.g. : *A6_GEN_Fairing_1.msh*) with an **MSH** extension
- 4) One or more **texture** files (e.g. : *A6_GEN_ext.dds*) with a **DDS** extension
- 5) A **guidance** file (optional) (e.g.: *A62_guidance.txt*) with a **TXT** extension
- 6) A **Lua** file (very optional) (e.g. : *Ariane 6.2.lua*) with a **LUA** extension

Now let's take a quick look at all these files...

They are all editable with the Windows® **Notepad** application except for the **DDS** files (the textures).

(1) The scenario file "scn"

This seems obvious : to use your "MultiStage" rocket, you need a file to launch your scenario. This file is located in your <ORBITER-2024>\Scenario folder (or a subfolder of it).

Example : In the **Ariane 6.2 (Generic).scn** file we find the following section :

```
Ariane6:Kourou_Rockets\Ariane6\A62
STATUS Landed Earth
POS -52.7604642 5.2711303
HEADING 341
AFCMODE 7
FUEL 1
CONFIG_FILE Config\Kourou_Rockets\Ariane6\A62.ini
GUIDANCE_FILE Config\Kourou_Rockets\Ariane6\A62_guidance.txt
CONFIGURATION 0
END
```

4 lines are important to understand (in red) :

- Declare the **name** (of your choice) of your rocket (*which will appear in the window opened by pressing the F3 key*) as well as the name (without ".cfg") and the path of your rocket's **CFG** file.
- Declare the name and the path of the **INI** file
- Declare the name and the path of the "guidance" file (*optional*)
- Declare the rocket's status (*here **CONFIGURATION 0** → rocket on the ground in "full" mode*) (*required*)

(2a) The configuration file "cfg"

The content of this file is very easy to understand. It allows you to define your rocket as a "Multistage" spacecraft.

Example : In the *A62.cfg* file we find the following section :

```
ClassName = A62
Module = multistage2015
Size = 100 ; This line is not mandatory
```

This file must be located in your <ORBITER-2024>\Config folder (or a subfolder of it).

The name (for example, "A62" here) must be the same as your **INI** file.

(2b) The configuration file "ini"

This file is the most important : it's where you'll build and declare all the elements of your rocket (stages, payloads, characteristics, meshes, etc.)...

It would be too long to detail all the elements here. Refer to the documentation for the "*multistage2015*" module.

This file must be located in your <ORBITER-2024>\Config folder (or a subfolder of it).

The name (and path, if applicable) must be found in the SCN file.

In our example, we have the following file and path : `Config\Kourou_Rockets\Ariane6\A62.ini`

(3) The mesh file(s) "msh"

This is (are) the **3D** file(s) of your rocket's components (stages, fairing, payloads, etc.).

This file must be located in the <ORBITER-2024>\Meshes folder (or a subfolder of it).

The name (and the path) - e.g. "`...\Texture\Kourou_Rockets\Ariane6`" - must be in the **INI** file.

It is important to know that the name(s) of the *texture file(s)* associated with your **3D** object are declared at the end of this file.

Example : in the `A6_GEN_Fairing_1.msh` file we can see at the end the following section :

```
TEXTURES 2 ;<----- this means that there are 2 declared textures.
Kourou_Rockets\Ariane6\A6_GEN_ext.dds ;<----- declared texture number 1
Kourou_Rockets\Ariane6\A6_GEN_int.dds ;<----- declared texture number 2
```

(4) The texture file(s) "dds"

These files provide the beautiful textures for your rocket components.

This file (or these files) must be located in the <ORBITER-2024>\Textures folder (or a subfolder of it).

The name (and the path, if applicable.) – here, "`A6_GEN_ext.dds`" for example – must be in the **MSH** file.

(5) The Guidance file "txt"

This file is optional. It allows you to create an *autopilot* for your rocket.

It would be too lengthy to detail all the elements here. Refer to the "*multistage2015*" module's documentation.

(6) The LUA file "lua"

Okay, maybe you shouldn't worry too much about this. Just know that it's used (among other things) to display text on your screen. Its declaration is located in the **SCN** file in the **BEGIN_ENVIRONMENT (...)** **END_ENVIRONMENT** section.

Let's recap the structure and operation

- The **SCN** scenario file will retrieve your two rocket configuration files – CFG and INI – (and optionally the autopilot file).
- The **CFG** file declares your rocket as a multistage spacecraft and is linked to the **INI** file.
- The **INI** file is the core of your rocket and specifies which **3D** elements (*MSH files*) to use.
- The **MSH** files are the **3D** elements of your rocket and specifies the names and paths of its **DDS** texture files.
- The **DDS** files (linked to the **MSH** file) are loaded.
- The (*optional*) **TXT** guidance file allows you to pilot your rocket automatically.
- The **LUA** file (*also optional*) allows you (among other things) to display text on your screen

Now that everything is clear in your mind, let's move on to the next step :

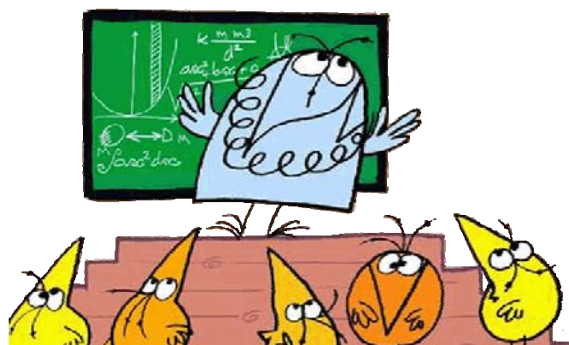
how to customize your beautiful future rocket...

Our goal is the following :

- We will keep the basic rocket design, without changing it.
- We will change the **fairing** to a customized one.
- We will adapt a different **payload** (payload or satellite).
- And of course, we will develop a new and customized scenario.

To do this, we'll go through the files we just briefly examined one by one, but... in reverse order !...

Yes, you'll see, it makes perfect sense...





II - LET'S GO.... (the tutorial)

In this chapter, we assume you're going to install a **short** fairing.
For a **long** fairing, the procedure is exactly the same ; I'll explain how to do it at the end.

A) DDS FILE or how to modify the texture

Since you bought an Ariane 6 rocket to launch your satellite, you want a **fairing** that displays your colors as well as those of your satellite.

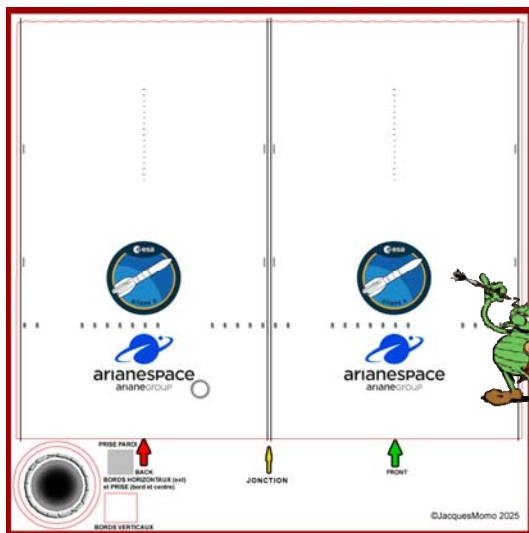
- 1) Go to the <your_Orbiter-2024>\Textures\Kourou_Rockets\Ariane6 folder
- 2) Make a copy of the **A6_GEN_ext.dds** file and rename it as you wish.
Let's assume that this new file will be named **A6_foo_ext.dds**
- 3) Open this file using your preferred drawing application (Photoshop, Gimp, etc...)

Now all you have to do is modify (redesign) the texture of your fairing.

This file corresponds to the texture applied to the outside of the fairing.

Note: The A6_GEN_int.dds file is used for the inside of the headgear.

Open the **A6_foo_ext.dds** copy you made of the **A6_GEN_ext.dds** file with your drawing software.
Here's what you'll see :



You'll find some guidelines to make things easier. Delete the two logos and replace them with your own.
You can also — why not ? — change the background color of the fairing. Let your imagination run wild...
When you're finished, don't forget to save your new texture in **DDS** format.



*If you would like to receive the source file (**Photoshop** format with layers), you can ask me for it.
I will be happy to send you all the files you require.*

For your convenience, we'll assume you'll keep this texture in the same folder.
However, this isn't mandatory. Do as you wish...

If you want to change the textures of the boosters or stages, the principle is the same.

The first step is complete; now we'll work on the 3D file, that is, the meshe.msh file....

B) MSH FILES or how to adapt the fairing's 3D elements

Yes, because we do not modify the form, but we adapt it.

1) Step 1: Make a copy of the files.

The fairing is modeled using three files : one file for each half-fairing, and one file for the "full" fairing.

Go to the <votre_Orbiter2024>\Meshes\Kourou_Rockets\Ariane6 folder.
And observe the different file names : I tried to maintain certain logic.

You'll notice that some files have "Fairing" in their name.

We will use a group of 3 files whose names begin with the same characters.

A6_GEN_Fairing_1.msh	rear half-fairing
A6_GEN_Fairing_2.msh	front half-fairing
A6_GEN_FairingS_1.msh	one-piece full fairing ("S" = plural because headdress 1+2 are "welded")



Warning : Don't make a mistake, don't use the files whose name contains an " L " because those are the **Long** fairings. (See further)

Make a copy of these 3 files. For example :

A6_foo_Fairing_1.msh
A6_foo_Fairing_2.msh
A6_foo_FairingS_1.msh

You don't have to keep the "A6" or the "Fairing". You can (for example) name your first file "my-beautiful-element_1.msh".

But it's essential to keep the "_1" and the "_2" at the end of the file name. Let me explain to you :



Let's assume that the fairing (the one declared in the **INI** file) is simply named "Fairing". Then the two meshes (because **N=2**, as you'll see later) that compose it are named "Fairing_1" and "Fairing_2." If instead of **N=2** we had **N=4**, we would have had four files for the headdress : Fairing_1, Fairing_2, Fairing_3, and Fairing_4. Conversely, if instead of **N=2** we had **N=1**, we would have only one file for the fairing, named Fairing_1.

2) Step 2: Specify the name of your DDS texture in the MSH file.

If we don't tell the msh files where to look for their dds file, then they won't find it.

Consolation: This does not result in a **CTD** with orbiter.

Edit each of these 3 files and look at what is written at the end (right after the "MATERIALS" section) :

```
TEXTURES 2
Kourou_Rockets\Ariane6\A6_GEN_ext.dds
Kourou_Rockets\Ariane6\A6_GEN_int.dds
```

We will modify line #2 with :

```
Kourou_Rockets\Ariane6\A6_foo_ext.dds
```

(If you have changed the folder, write the path and the names of your 3 **MSH** files).

Don't forget the ".dds" at the end, it's important.

Save these 3 modified files, and let's go on to the next step.

Let me conclude by summarizing my key points :

We have our custom fairing (3D models with **MSH** files and textures with **DDS** files).

Now Mr. *Meshe* knows where to find Ms. *Didihess*, everything's fine....

Now we need to tell the rocket which ~~headdress~~ fairing to wear. So we're going now to explore the **INI** file.....



C) INI FILE or how to change the fairing

There's only one file. To find it, we have to go to the ...\\Config\\Kourou_Rockets\\Ariane6 folder.

We have first to make a copy of the **A62.ini** file and rename it as **A6-foo.ini** (or another name of your choice).

Keep this file-copy in the same folder. (You can move it, but I won't go into detail here). Let's keep it simple....

1) Step 1 : Editing the INI file.

Use the *Notepad* program and look for the **[FAIRING]** section.

You will find two **[FAIRING]** sections :

- the first one is disabled (because there are semicolons (;) at the beginning of all lines)
- the second one is enabled.

For now, we'll use the **first** section (the one that's disabled). For the other one, wait a bit while you digest everything. You'll find all the explanations later...

1) remove the *semicolons* from the section : **[FAIRING] ; == Short Fairing (simplified mode) disabled ==**

2) write *semicolons* before all the lines in the section **[FAIRING] ; == Short Fairing (realistic mode) enabled ==**

Is it done ? Okay. So let's continue...

2) Step 2 : Modifying the INI file.

So we have this (without the semicolons (;) at the beginning of the lines, I remind you):

```
FAIRING] ; === Short Fairing (realistic mode) enabled =====  
N=2  
Meshname="Kourou_Rockets\Ariane6\A6_GEN_Fairing"  
Off=(0,0,50.4)  
Angle=0.000  
Height=20  
Diameter=5.4  
EmptyMass=900  
Speed=( 0,-10,0)  
Rot_speed=( -0.4,0,0)  
Module=Stage
```

?



The **red** line is the one you're going to modify. Replace the **path \ filename** with your own.
(The two **blue** lines are just to draw your attention : we'll come back to them later).

So, in our example we would have : **Meshname="Kourou_Rockets\Ariane6\A6_foo_Fairing"**
*Did you see that ? No **_1** or **_n** at the end of the name of the file...*
Easy, right ? (Remember to use quotation (" ") marks).

Now we move on to the **CFG** file and luckily, it's the simplest one.

D) CFG FILE or how everything becomes so easy

In the same folder as your **INI** file, you'll find a file called **A62.cfg**. We'll proceed as usual :
Make a copy of this file, rename it **A6-foo.cfg**, and then **edit** it. Here's the change to do :

ClassName = A62	➔	ClassName = A6-foo	(the name of your INI file)
Module = Multistage2015	➔	Module = Multistage2015	(no change)
(You can ignore the other lines)			

We're almost finished : all that's left is to create the **SCN** file (the scenario).



E) SCN FILE or the launch scenario

Again, the easiest way is to use a copy of an existing script.
For example, let's take the **Ariane 6.2 (Generic).scn** file and make a copy that we'll rename as **Ariane 6.2 (foo).scn**.
The name of this file is flexible ; you can name it whatever you want.

Now edit this file, and look for the rocket section :

```
Ariane6:Kourou_Rockets\Ariane6\A62  
STATUS Landed Earth  
POS -52.7604642 5.2711303  
HEADING 341  
AFCMODE 7  
FUEL 1  
CONFIG_FILE Config\Kourou_Rockets\Ariane6\A62.ini  
GUIDANCE_FILE Config\Kourou_Rockets\Ariane6\A62_guidance.txt  
CONFIGURATION 0  
END
```

Here too, the lines to modify are highlighted in **red**.

- Line 1: the name and path of your **CFG** file
- Line 7: the name and path of your **INI** file
- Line 8: the name and path of your **guidance** TXT file. (optional, as a reminder)



If we use the names of the examples we've seen (the path remains unchanged for simplicity), we should have this :

```
Ariane6:Kourou_Rockets\Ariane6\A6-foo
STATUS Landed Earth
POS -52.7604642 5.2711303
HEADING 341
AFCMODE 7
FUEL 1
CONFIG_FILE Config\Kourou_Rockets\Ariane6\A6-foo.ini
GUIDANCE_FILE Config\Kourou_Rockets\Ariane6\A6-foo.ini _guidance.txt
CONFIGURATION 0
END
```

If your Orbiter crashes when you launch your new scenario, you've (probably) made an error in declaring the path or name of your payload/satellite/spacecraft. In that case, check the **Orbiter.log** file in your Orbiter's root directory.

Of course, if your files aren't in this directory, you'll need to correct their paths in this **SCN** file.

You'll also notice that I've modified the "guidance" file... which doesn't exist yet.

So, let's take care of that file.

E) TXT FILE or the guide file that tells the rocket what to do

And here too, the easiest way is to use a copy of an existing file. You can use any *_**guidance***.txt file located in the same folder as the rocket's **INI** and **CFG** files as a template.

Make a copy, rename it whatever you want (the name and path are entirely up to you, but make sure you copy it correctly from the **SCN** file), and have fun modifying this "guidance" file as you want.

If you have trouble understanding the command lines, refer to the **Multistage documentation** on pages 23 and 24. This documentation by **Fred18** is in the...\Doc\Fred18's Modules folder (Multistage 2015.pdf file).

F) LUA FILE or the file that speaks on the screen

But what the heck is this ?! Isn't it over yet ? Nope...

Just a quick tip : if you made a copy of an existing **SCN** file — as I advised — you might have noticed that this scenario is linked to a "script" file (**LUA** extension) used to display comment lines on the screen during flight. Yes, indeed....

For example, edit one of the **SCN** and look in this section :

```
BEGIN_ENVIRONMENT
System Sol
Date MJD 59064.5381944444
Script Ariane6/Ariane 6.2
END_ENVIRONMENT
```

Do you see the red line ? Then either delete it, or modify the **LUA** file (by renaming it) to better match your flight. I won't say any more!

Oh, and one more thing... Remember to change the launch date (the number in green) if you wish....

THAT'S IT FOR TODAY !!!

Your rocket is ready; all you have to do is to launch it.

But before we part ways, we still have two topics to discuss :

- 1) The story of the headdress in "realistic" mode (*next chapter*)
(If you're tired of over thinking it, you can skip this chapter)
- 2) How to install your own satellite, the one you bought on internet.



III - WHY MAKE IT SO EASY WHEN IT CAN BE SO DIFFICULT ?



Because it's prettier and funnier... Okay, but it's more complicated. So if you had trouble understanding the previous chapter, don't insist and go your way....

Are you still determined ? Okay, then let's go.

I don't know if you've noticed, but in the videos of the launch of flight VA265 (the real one) you can clearly see the fairings playing accordion-like when they are jettisoned... So I had fun simulating this phenomenon.

Let's go back to the INI file from earlier : you saw that these are the lines that are activated *by default* :

```
FAIRING] ; === Short Fairing (realistic mode) enabled =====
N=1
Meshname="Kourou_Rockets\Ariane6\A6_GEN_FairingsS" ; instead of A6_GEN_Fairing without "S"
Off=(0,0,50.4)
Angle=0.000
Height=20
Diameter=5.4
EmptyMass=1800
Speed=(0.000,0.000,0.800) " ; instead of "Speed=(0,-10,0)"
Rot_speed=(0.000,0.000,0.000) ; instead of "Rot_speed=(-0.4,0,0)"
Module=..\Kourou_Rockets\Ariane6\A6_GEN_FairingsS ; instead of "stage"
Name=Fairing ; line to add "
```



I've highlighted the relevant line in *orange*. What does it mean ?

Quite simply, when the fairing is jettisoned, instead of using the "stage.dll" module from *Multistage*, it looks for the A6_GEN_FairingsS.cfg file located in the ... \Config\Kourou_Rockets\Ariane6 folder.

And this makes a **VesselBuilder vessel** that will appear in the list **F3** with the name **Fairing**.

But... very quickly, this "**Fairing**" vessel will create to two new **VesselBuilder vessels**, which will be the two half-fairings (**Fairing1** and **Fairing2**), and which will play accordion. And the **Fairing vessel** will disappear at the same time as the birth of its two Childs, Fairing1 and Fairing2.

Have you understood ?

So, if you decide to use this fairing's version, you'll need 3 new files...

But once again, you just need to make a copy of these 3 files and rename them as in the following example :

```
A6_GEN_FairingsS.cfg ⇒ copy ⇒ A6_GEN_FairingS - copy.cfg ⇒ renamed ⇒ A6_foo_FairingsS
A6_GEN_Fairing1.cfg ⇒ copy ⇒ A6_GEN_Fairing1 - copy.cfg ⇒ renamed ⇒ A6_foo_Fairing1
A6_GEN_Fairing2.cfg ⇒ copy ⇒ A6_GEN_Fairing2 - copy.cfg ⇒ renamed ⇒ A6_foo_Fairing2
```

(I kept "foo" to keep things simple, but the name can be different for you)

Once the copies are made, you need to :

- 1) In the INI file (orange line) correct A6_GEN_FairingsS to A6_foo_FairingsS
- 2) Edit the A6_foo_FairingsS file then look for the section :

```
;<-----MESHES DEFINITIONS----->
MESH_0_NAME = Kourou_Rockets\Ariane6\A6_GEN_fairingsS_1
```

Rename A6_GEN_FairingsS_1 to A6_foo_FairingsS_1

And then look for the section; <-----EVENTS DEFINITIONS----->

Then the lines :

```
EVENT_0_SPAWNEDCLASS = Kourou_Rockets\Ariane6\A6_GEN_Fairing1
EVENT_1_SPAWNEDCLASS = Kourou_Rockets\Ariane6\A6_GEN_Fairing2
```

Rename A6_GEN_Fairing1 to A6_foo_Fairing1

Rename A6_GEN_Fairing2 to A6_foo_Fairing2

- 3) Edit the A6_foo_Fairing1 file then look for the section :

```
;<-----MESHES DEFINITIONS----->
MESH_0_NAME = Kourou_Rockets\Ariane6\A6_GEN_fairing_1
```

Rename A6_GEN_FairingsS_1 to A6_foo_FairingsS_1

- 4) Edit the A6_foo_Fairing2 file then look for the section :

```
;<-----MESHES DEFINITIONS----->
MESH_0_NAME = Kourou_Rockets\Ariane6\A6_GEN_fairingsS_2
```

Rename A6_GEN_FairingsS_2 to A6_foo_FairingsS_2



COMPLETE !!! If you haven't made any spelling mistakes, it will work. Otherwise, you'll have to figure out why...

AND WHAT ABOUT LONG FAIRING ???

Ah yes, of course... So if you insist... Indeed, up until now we've only dealt with the short fairing. For the long fairing, it's exactly the same... but instead of the files mentioned previously (*I'm not going to repeat myself, no way...*) you need to use the following files :

For the rocket :

A64.cfg and A64.ini

For the fairings

- model 3D :

A6_GEN_FairingSL_1.msh

A6_GEN_FairingL_1.msh

A6_GEN_FairingL_2.msh

- texture :

A6_GEN_Lext.dds

And if you want the accordion fairings :

A6_GEN_FairingSL.cfg

A6_GEN_FairingL1.cfg

A6_GEN_Fairing2L.cfg



Did you see the " L " ?... Well, it means "Long".

IV - HOW TO MODIFY and/or CHANGE the PAYLOAD(S)

This is done in the "ini" file which now holds no more secrets for you...

A) Start by editing your INI file (the one from your new rocket).

Next, look for the "Payload" section and here's what you'll find :

```
[PAYLOAD_1]
Meshname=" Kourou_Rockets\Satellites\A6-Payload"
Off=(0.000,0.000, 55.430)
Rotation=(0.000,0.000,0.000)
Height=4.000
Diameter=2.000
Mass=5000
Module=Kourou\Satellites\A6_Payload
Name= Name=Payload
Speed=(0.000,0.000,1.000)
Rot_speed=(0.000,0.000,0.300)
Render=0
Live=0
```

B) Modify the lines in your INI file (those that are in red).

Suppose you have decided to create a folder called "**Personal**" and your files are named "**My_addon**".

- 1) Declare your satellite **mesh**, assuming it is named "**My_addon.msh**" and is located in the <Your-Orbiter> \Meshes\ **Personal** folder
Like this : **Meshname="Personal\My_addon"** ; (*Don't forget the quotation marks [" "]*)
- 2) Adjust the position of your mesh (yes, your satellite) so that it is properly positioned.
For example : Off=(0.000,0.000,**72.000**)
- 3) Declare your satellite's configuration file, assuming it's named "**My_addon.cfg**" and located in the <Your-Orbiter> \Config\Vessels\ **Personal** folder
Like this : **Module=Personal\My_addon**
- 4) Give a name for your satellite. This name can be different from the one in the configuration file.
For example : **Name=Foo**

You will get this :

```
[PAYLOAD_1]
Meshname="Personal\My_addon"
Off=(0.000,0.000,72.000)
Rotation=(0.000,0.000,0.000)
Height=4.000
Diameter=2.000
Mass=5000
Module= Personal\My_addon
Name= Foo
Speed=(0.000,0.000,1.000)
Rot_speed=(0.000,0.000,0.300)
Render=0
Live=0
```



And there you have it. Your satellite (or spacecraft or payload) is declared and attached on your rocket..

If you want to install a second satellite, repeat the same process with a new other section named [PAYLOAD_2]

V - SYLDA and DISPENSER

A) SYLDA

The **SYLDA** allows for the integration of two payloads within the same fairing :

- For a single launch, the satellite is placed on the **ULPM**, optionally with a payload adapter.
- For a dual launch, one satellite is placed under the **SYLDA** into the fairing.

Then, the second satellite is placed on the support structure, again optionally with an adapter..

The **SYLDA** consists of a support structure with three separation systems: one for each satellite and one for the structure itself.

The use of this advanced technology fully guarantees the required performance: high rigidity, minimal mass, large payload capacity, and flexibility to adapt the configuration to specific customer requirements.

At the end of the flight, the top satellite is released first. The **SYLDA** is split in two by pyrotechnic charges, allowing the second satellite to be released by spring.

To add this **SYLDA** to the Ariane 6 rocket, edit your rocket's **INI** file and add these lines :

```
[PAYLOAD_1] ; # Satellite or payload #2 positioned above the Sylda #
(...)
```

```
[PAYLOAD_2] ; # SYLDA #
Meshname="Kourou_Rockets\Ariane6\A6_Syllda01"
Off=(0.000,0.000,54.820)
Rotation=(0.000,0.000,0.000)
Height=4.000
Diameter=2.000
Mass=500
MODULE="Kourou_Rockets\Ariane6\A6_Syllda01"
NAME="Syllda"
Speed=(0.000,0.000,1.000)
Rot_speed=(0.000,0.000,0.000)
Render=0
Live=0
```

```
[PAYLOAD_3] ; # Satellite or payload #1 positioned
(...) ; # below (or inside) the Syllda #
```



B) DISPENSER

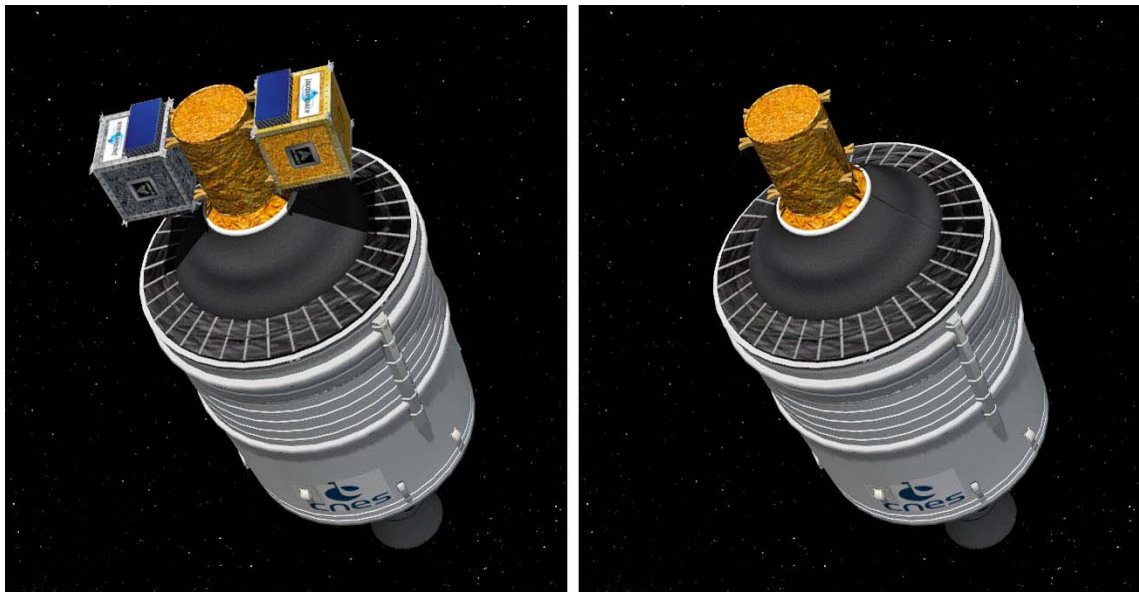
The Ariane 6 **DISPENSER** is a mechanical structure installed at the top of the launcher, designed to carry, protect, and separate multiple satellites during a single flight. It provides structural support for the payloads during launch and ensures their precise, sequential separation once in orbit.

During flight VA 266, two Galileo satellites, future additions to the European navigation system, were installed on this **DISPENSER**, which allowed them to be securely docked in the fairing.

To add this dispenser to the Ariane 6 rocket, edit your rocket's **INI** file and add these lines :

```
[ADAPTER] ; # The Dispenser #  
Meshname="Kourou_Rockets\Ariane6\A6_Dispenser"  
Off=(0.000,0.000,52.350)  
Height=1.200  
Diameter=6.100  
Emptymass=1000.000
```

Note : only one [ADAPTER] is possible under the rocket fairing.



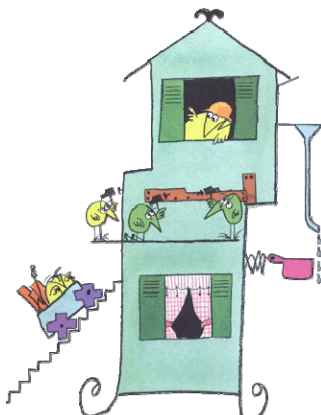
VI - HOW TO MODIFY THE ROCKET'S CHARACTERISTICS

Everything is located in your rocket's **INI** configuration file :
empty mass, dimensions, position, thrust value, fuel, captain's age...



Regarding modifying these **INI** files, I recommend you to read the [Fred18's Multistage2015](#) documentation, which is in the...\Doc\ Fred18's Modules folder (Multistage 2015.pdf file).

That's it !



JacquesMomo 2025