

Exporting 3D Model for Multi Stage and Assembling the Stack

Preparing to Export

Before exporting your file for conversion, make things easier for yourself and set up a separate directory containing the Anim8tor file you are working on and Meshmaker (if that's the mesh converter you're using). Also copy two random meshes into this directory and rename them 'input' and 'shadow', as shown in Fig. 1 below. I'll explain why later.

It's also a good idea when developing an add-on to use a fresh Orbiter installation for your testing and development; so into this folder place shortcuts to Orbiter.exe and any other files you will be working with during development. The two folder shortcuts, 'multistage' and 'spacecraft' in Fig.1 are pointing to folders within the Orbiter installations Meshes folder. It's into these folders that you will drop any newly created and appropriately named meshes written my Meshmaker.

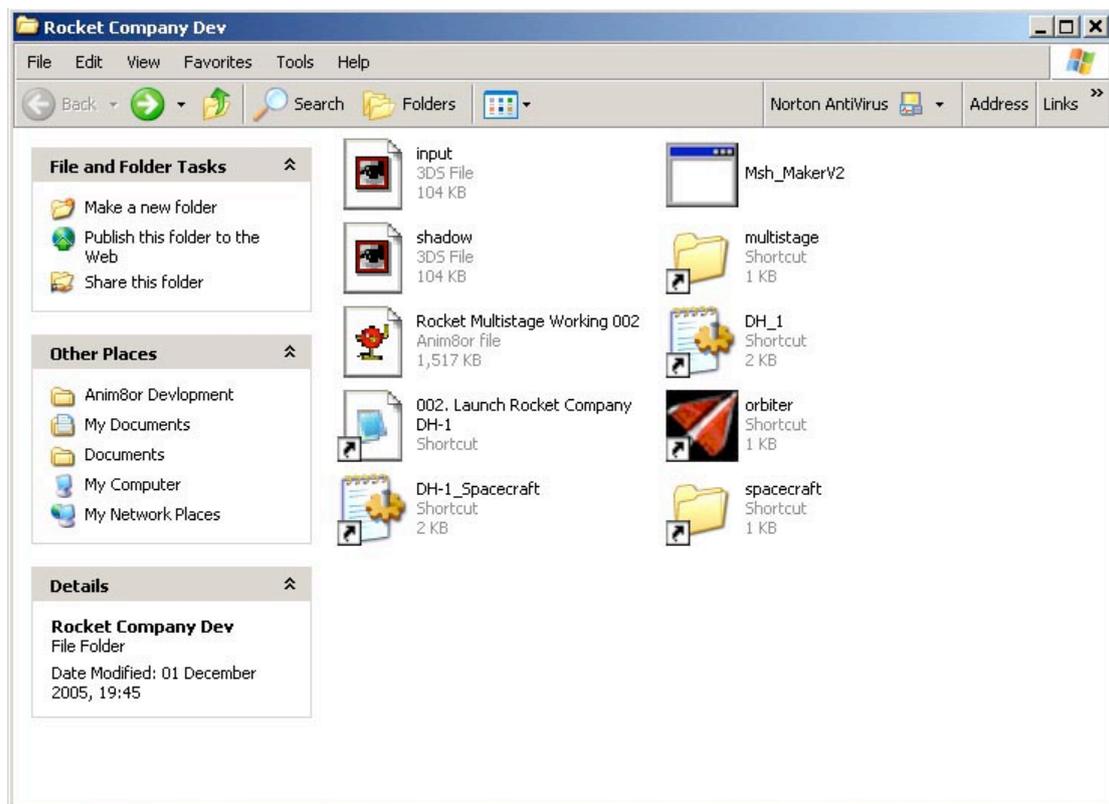


Fig 1.

1. Exporting your meshes

Two 3DS files need to be exported from the Anim8tor model to enable meshmaker to create a mesh file suitable for use in Orbiter. The easiest way to do this is to export from Anim8tor twice. For the first export, name your file 'input', and for the second name the file 'shadow'. By creating those two files

earlier as described in the first paragraph, simply selecting the file already in the directory and over writing it saves having to type the name each time.

Note:

- All parts of the model need to be centred on origin and in in 3ds format.

2. Converting the Exported files to Orbiter's mesh format.

Assuming you have set up your directory as described, continue on,

- Double-click mesh maker.
- A file called Debug is produced
- Double-click mesh maker again
- A file called output is produced
- Re-name this output file to the mesh name you have chosen.
- Copy this file to the meshes folder in your Orbiter installation *
- **Delete the file named Debug**
- Repeat parts 1 and 2 above for the remaining parts of your launcher

* You may have decided to place all the mesh files directly into the Meshes folder, or into the separate folders called multistage & spacecraft as shown in Fig. 1. You may also create a different folder name of your choice. As long as the path is correctly cited in the config file it will work.

Now assuming that you have all the meshes exported and converted its time to assemble the launcher on the pad. There appear to be two common methods of doing this.

3. Assembling the Stack

- Create a new Object in Anim8tor and copy & paste the various parts into the Object window
- Carefully assemble the whole stack. *You've probably done this already as you constructed your model.*
- Group all the parts together and stand the complete assembly on the origin.

Method 1

- Open your config.ini file for the launcher
- Ungroup the various stages in Anim8tor
- Double-click the first stage and get the Y co-ordinate from the group editor. This number is the height of the centre point of the first stage. Just note it for now.
- In the Config.ini file set offset [OFF] to zero for all 3 co-ordinates for the first stage. Also set Cog to zero at the start of the file. **We do this because in this example the launcher will stand on the ground. If**

we were launching from a higher platform or launch pad, then the COG would need to be increased.

- Double-click on the second stage and get the Y co-ordinate. Subtract the previously noted first stage offset from this figure. **We do this because we initially set the offset for the first stage to zero, so the following stages & parts of the launcher have to have this number subtracted from them each time to position them correctly.** Use this calculated number as the vertical offset for the Z co-ordinate for the second stage in the config file.

The reason for using the Y co-ordinate for the Z coordinate in the config file is that multistage.dll re-orientates the launcher on the pad so Z is 'up' not Y as in Anim8tor.

- Continue in the same manner for the remaining stages, payload and fairing *, continuing to subtract the first number previously noted for the first stage.

Method 2

Alternatively you can dispense with 'zeroing' the Z coordinate for the first stage and all the subsequent calculations for the positioning of the remaining stages, payloads and fairings. It's just that I noticed Vinka's demo config.ini file that comes with multistage.dll does it that way, which though seeming logical does create more work.

Positioning of the fairings is a little different to what's gone before.

The number of parts for the fairing is declared at the start of the [FAIRING] statement. In my example we will have a two-part fairing, e.g. N=2.

To obtain the numbers for the offset coordinates, open the object editor for either of the fairing parts. The X coordinate is the first number (discard any '- sign), the Y co-ordinate should be set to zero to centre the mesh about the Y-axis. The Z coordinate is calculated in a similar manner to the stage offset numbers – **Remember it's the Y coordinate you want to use for Orbiter's Z-axis in multistage.**