

[More](#)[KSPTV](#)[KSP Wiki](#)[Get Mods!](#)[Store](#)[Activity](#)[Forum](#)[Home](#) > [Add-ons](#) > [Add-on Releases](#) >[Unread Threads since my last visit](#)[Mark site read](#)[\[0.90\] TweakScale - Rescale Everything! \(v1.50 - 2014-12-24 10:40 UTC\)](#)

# [0.90] TweakScale - Rescale Everything! (v1.50 - 2014-12-24 10:40 UTC)

[Follow](#)

1

By Biotronic, May 16, 2014 in Add-on Releases

Rate this topic

[Start new topic](#)

1

2

3

4

5

6

[NEXT](#)

&gt;&gt;

Page 1 of 67

## Biotronic

Sr. Spacecraft Engineer



Members

+ 263

359 posts

Posted May 16, 2014 (edited)

[Report post](#)

### [TweakScale is under new administration](#)

Due to pressures outside the game, and the fact I rarely play KSP these days, I do not have the time to maintain TweakScale.

In my absence, pellinor has accepted the burden and is currently the maintainer of TweakScale.

Please visit his [new release thread](#) for updates.

TweakScale lets you change the size of a part. Not just that, but it will figure out how much fuel is in the resized part. And if it's an engine, it will become more powerful by scaling it bigger, or weaker by scaling it smaller.



This plugin was initially made by Gaius Goodspeed, and I heartily recommend you also take a look at [his parts](#).

TweakScale is already used by Gaius' parts above, and [ZodiusInfuser's Infernal Robotics model rework](#), which I also recommend to anyone who likes having fun.

TweakScale uses [Swamp-Ig's KSPAPIExtensions](#).

TweakScale is made available under the terms of the [Do What the ... You Want to Public License](#).

## DOWNLOAD

[Alternative Download from Dropbox](#)

[KerbalStuff](#)

[Curse Page](#)

[Source on GitHub](#)

Current Version: v1.50 (2014-12-24 10:40 UTC)

HALP! It no worky! / FAQ

Alright, calm down. We will get to the bottom of this. What symptoms are you experiencing?

**The scale slider is missing, and/or there's a gaping hole in the right-click menu!**

This is usually caused by an outdated

[KSPAPIExtensions.dll](#). If you have any of [these mods](#) installed, please update them. Also make sure you have the latest TweakScale installed.

Sometimes when updating, you need to completely remove the old mod first.

There may be other mods that use KSPAPIExtensions.dll than those listed there, so if that doesn't solve your problem, try a search for that file name.

Lastly, this may be caused by [Goodspeed Aerospace Parts](#). If you have it installed, find and remove its Scale.dll file.

### **Part X of mod Y isn't rescalable!**

Great! That means I either have not updated the mod, or it's not on the list of supported mods, as seen below. I'm very happy to get updated cfgs, or just a note that some parts are not working. I don't use all the mods listed, so it's hard to keep them all updated.

### **Part Z becomes too heavy when scaled up!**

Yeah, I'm a bit lazy, so all parts' mass scale with the cube of the size. That's the equivalent of saying every part is solid, or filled with heavy cream. Some parts are probably not filled, and may consist of but a thin skin over an empty inner. Maybe some day I'll get around to updating that kind of stuff. Otherwise, if someone else were to do it, I'd be more than happy to include their work!

### **TweakScale and some other mod are having problems cooperating!**

First make sure both mods are up to date. Occasionally, a mod may change the names of files, so just overwriting with an updated version might not be enough. If necessary, take a backup. Then delete the folder in which the mod is installed, and install the updated

version.

If this does not solve your problem, or you're scared of breaking your game, please post here. As below, please post your output\_log.txt.

### **I'm having a problem and it's not listed here!**

Great! That means I can make TweakScale even better! Please post here and describe your problem. Make sure you have the newest version of TweakScale, and please post your output\_log.txt. It's going to be too big to place in a single post, so upload it to [Dropbox](#), [Pastebin](#) or some other service that lets you upload large text files.

Please add TweakScale to your mod!

If you are a mod author and you want to bundle TweakScale with your mod, please do! A few notes:

- Please place your TweakScale .cfgs in your mod's folder, not in the TweakScale folder. This way users can delete TweakScale and install a new version without breaking your mod.
- If your mod is already on the list of supported mods, please post here or [PM me](#), and I will remove support, giving you full control over the .cfgs.

Anyways, features:

### Scaling Control

You as the author of a part or addon get complete control over which parts you want to offer in which sizes. Should that fuel tank only be available in size 2.5m and 3.75m? Make it so! That RCS thruster on the other hand, could be scalable freely between half regular scale and double regular scale.

### Mass Control

For heavy, solid parts, mass increases with the cube of the scale - you scale it in three dimensions, after all. For parts that are a thin layer of aluminium plates over a rigid skeleton - like fairings, crew compartments, empty fuel

tanks - mass probably scales closer to the square of the scale.

### Control Over Every Feature of a Part

This is not all! Engines can be made more powerful as they increase in size, generators more efficient, and wings more lift. All by writing a simple config file, and then automatically for all rescalable parts of that type.

### How to Use

First add a part that's the wrong size:



Right click:





See how it says 3.75m? Well, the command capsule is 2.5m, so let's change it. You do this by pressing the << >> buttons or dragging the slider.



See how well it fits?



## Example Configurations

For a part that should be available in 62.5cm, 1.25m, 2.5m, 3.75m and 5m configurations, and by default is 1.25m, use the following definition:

```
{  
  name = TweakScale  
  defaultScale = 1.25  
  type = stack  
}
```

MODULE

If the part should instead be freely rescalable, use this:

```
{  
  name = TweakScale  
  type = free  
}
```

## MODULE

And for a part that should be available in 25%, 50%, 100% and 200% scales, use:

```
{  
name = TweakScale  
type = surface  
}
```

## MODULE

But I said you had full control of scales, didn't I? If you want your parts to be available only in 2.5m and 3.75m versions, use this definition:

```
{  
name = TweakScale  
type = stack  
scaleFactors = 2.5, 3.75  
scaleNames = 2.5m, 3.75m  
}
```

## MODULE

If your mod has a collection of parts that will all be available in the same set of sizes, you might want to make your own scale type:

```
{  
name = MyMod  
freeScale = false  
scaleFactors = 0.5, 1.0, 2.0  
scaleNames = Small, Medium, Large  
defaultScale = 1.0  
}
```

## SCALETYPE



After defining this once, you can then start using it for your parts:

```
{  
name = TweakScale  
type = MyMod  
}
```

MODULE

As you can see the scale type uses the same names as the module definition, and they can even inherit from other scale types (if you want to change just a small detail):

```
{  
name = spaceplane  
type = stack  
scaleFactors = 1.25, 2.5, 3.75, 5.0  
scaleNames = 1.25m, 2.5m, 3.75m, 5m  
}
```

SCALETYPE

This will use all the defined properties of the 'stack' scaletype, but will not be available in 62.5cm size (because some poor kerbal is going to sit inside there).

### Adding module support

You can now add support for your own modules! For a simple module that's happy with having its values changed when OnLoad is called, this is how:

```
{  
name = MyPartModule  
  
flooferRate = 2  
}
```

## TWEAKSCALEEXPONENTS

When a user rescales a part with a `MyPartModule` module, `TweakScale` will automatically update the `flooberRate` of the part with the square of the scale (so if it's a 2.5m part and it's scaled to 3.75m, the `flooberRate` will be 2.25 times its usual value [ $3.75/2.5 = 1.5$ ;  $1.5^2 = 2.25$ ]).

New in 1.10 is the ability to change fields of fields - that is, `myPartModule.someStruct.value` or `myPartModule.someList[x]`:

```
{
  name = ModuleGenerator
  outputList
  {
    rate = 3
  }
}
```

## TWEAKSCALEEXPONENTS

The above config will scale all members of the list `outputList` on `ModuleGenerator`, to the cube of the current scale.

Note that this system works for any depth:

```
{
  name = ModuleMyModule
  foo
  {
    bar
    {
      quxRate = 3
    }
  }
}
```

## TWEAKSCALEEXPONENTS

The above would scale  
ModuleMyModule.foo[x].bar.quxRate.

New in 1.19 is tech requirements:

```
{  
name = TweakScale  
type = stack  
techRequired = basicRocketry, start,  
generalRocketry, advRocketry, heavyRocketry  
}
```

## MODULE

Here, each option in the stack SCALETYPE will be unlocked by a corresponding tech. If there are fewer techRequired than scaleNames/scaleFactors, the unmatched scales will be unlocked by default.

It might be that your module would benefit more from using a list of values than an exponent. In that case, you may specify the list in the tweakscale module statement in the .cfg:

```
{  
name = TweakScale  
type = stack  
defaultScale = 3.75
```

## MODULE

```
{  
name = ModuleEngines  
maxThrust = 1, 2, 3, 4, 5  
}  
}
```

## MODULE

Here, a 62.5cm version would have a maxThrust of 1, 1.25m would be 2, and so on until the 5m version has a maxThrust of 5.

New in 1.23: Forced relative scaling!

This is invaluable for persistent values. If they were to use absolute scaling, the default value would be loaded from the prefab part, scaled up, and used without a second thought. With relative scaling, the current value is loaded, the multiplied by the scale factor difference since last we looked. And it's super easy to use:

```
{  
name = MyModule  
!MyValue = 2  
}
```

TWEAKSCALEEXPONENTS

New in 1.30: Cost scaling!

As 0.24.1 introduced the beautiful IPartCostModifier, TweakScale now supports changing the cost of a part:

```
{  
name = TweakScale  
dryCost = 2  
}
```

TWEAKSCALEEXPONENTS

The above configuration will make the cost of a part scale with its surface area. Useful for fuel tanks, solar sails, balloons and rugs.

Thanks to NathanKell, 1.44 added support for not doing scaling when a specific module is added to a part. This is useful when TweakScale can do some updates on its own,



but should keep its hands off when another module can do better.

```
{  
name = MyModule  
socksPerMinute = 2  
ignore = SockManager  
}
```

#### TWEAKSCALEEXPONENTS

This code will scale socksPerMinute by the square of the scale, or if there's a SockManager partmodule on the part, leave it all alone and let the SockManager do its thang.

1.48 adds Auto Scaling - a part will automatically choose an appropriate scale, based on the scale of the parent. This feature can be toggled with Ctrl+L, and temporarily disabled by holding Left Shift.

1.48 also adds Chain Scaling - if two 1.25m parts are placed atop one another, and you rescale the parent to 2.5m, the child will be scaled along with it. Chain Scaling can be toggled with Ctrl+K, and temporarily disabled by holding Left Shift.

Lastly, 1.48 Auto Scaling and Chain Scaling both support asymmetric scaling. This means if you choose a 1.25m to 2.5m adapter as your first (root) part, and add a fuel tank in both ends of it, one will be scaled to 1.25m, the other to 2.5m.

In order to wield this amazing power, TweakScale must be made aware of the sizes of the attach nodes of a part. This can be done on a per-part or per-scaletype basis. For a regular stack-mounted fuel tank, the nodes may be something like this:

```
node_stack_bottom = 0.0, -0.3125, 0.0, 0.0,  
1.0, 0.0, 1
```

```
node_attach = 0.5, 0.0, 0.0, 1.0, 0.0, 0.0
```

```
node_stack_top = 0.0, 0.3125,  
0.0, 0.0, 1.0, 0.0, 1
```

node\_attach is the surface node. You should probably not auto-scale based on that. For the other two, we get this config:

```
{  
MODULE  
{  
name = TweakScale  
defaultScale = 1.25  
type = stack  
ATTACHNODES  
{  
base = 1  
top = stack:1  
bottom = stack:1  
}  
}  
}
```

```
@PART[MyFuelTank]
```

For a 1.25m to 2.5m adapter, we instead get this:

```
{  
MODULE  
{  
name = TweakScale  
defaultScale = 2.5  
type = adapter_2_1  
ATTACHNODES  
{  
base = 2
```

```

top = stack:1
bottom = stack:2
}
}
}

```

```
@PART[MyAdapter]
```

This basically means that node\_stack\_top for the adapter has a scale of  $(\text{top}/\text{base}) = (1/2)$  of the scale of the part, so 1.25m, while node\_stack\_bottom has a scale of  $(\text{bottom}/\text{base}) = (2/2) = 1$  times the scale of the part, so 2.5m. The reason it says stack:1 is to have a system of families. This is the stack family, mostly intended for ...you guessed it - stack-mounted parts. This is just a string, so if you need a 'floopnurger' family, go right ahead. Only nodes with matching families will auto-scale, and one node can belong to more than one family.

### Advanced features

If more advanced logic is required, TweakScale offers an IRescalable interface. It is defined in Scale\_Redist.dll, and its definition is as follows:

```

{
void OnRescale(ScalingFactor factor);
}

```

```

public interface
IRescalable<T>

```

ScalingFactor has the properties 'absolute' and 'relative'. Most likely, you want to use absolute. relative is the change in scale since last time OnRescale was called, while absolute is the change in scale in relation to defaultScale. absolute and relative have properties linear, quadratic and cubic, which

are shorthands for different scaling factors. If you want something different, you'll have to figure out the math yourself.

IRescalable can be used in conjunction with .cfg exponents. In that case, TweakScale will first call IRescalable's OnRescale, followed by updates from .cfgs. (this may be changed in the future, as I'm not sure it's the best solution)

An example implementation of an IRescalable may be:

```
class MyModuleUpdater :  
TweakScale.IRescalable<MyPartModule>  
{  
    MyPartModule _module;  
  
    public MyModuleUpdater(MyPartModule  
        module)  
    {  
        _module = module;  
    }  
  
    public void  
    OnRescale(TweakScale.ScalingFactor factor)  
    {  
        _module.flooberRate = _module.flooberRate *  
            factor.relative.quadratic;  
        _module.ReactToFlooberRate(13);  
    }  
}
```

For an example implementation, check out [how Modular Fuel Tanks/Real Fuels support is implemented](#).

Version History:

[v1.50 \(2014-12-24 10:40 UTC\)](#)

- Fixed erroneous placement of attach



nodes when duplicating parts.

- Updated Scale\_Editor.dll. Easier than removing it.

#### [v1.49 \(2014-12-18 11:03 UTC\)](#)

- Now saving hotkey states correctly.
- 'Free' scaletype actually free.
- Fixed bug in OnStart.
- First attempt at scaling offsets.

#### [1.48 \(2014-12-16 22:54 UTC\)](#)

- Updated for .90!
- Cleaned up autoscaling and chain scaling.

#### [1.47 \(2014-11-17 20:03 UTC\)](#)

- Removed [RealChute](#) support.
- Fixed a bug where TweakScale would try to set erroneous values for some fields and properties, which notably affected [Infernal Robotics](#).
- Fixed a bug where cloned fuel tanks would have erroneous volumes.

#### [1.46 - 2014-11-16 20:03 UTC\)](#)

- Fixed an issue where features were incorrectly scaled upon loading a ship,
- Scaling a part now scales its children if they have the same size.
- Parts now automatically guess which size they should be.

#### [1.44 \(2014-10-10 19:20 UTC\)](#)

- Now updating UI sliders for float values.
- Better support for [KSP Interstellar](#).
- Added support for [TweakableEverything](#).
- Added support for [FireSpitter's](#) FSFuelSwitch.

#### [1.44 \(2014-10-10 19:20 UTC\)](#)

- Updated for KSP 0.25
- Added ability to not update certain properties when a specific partmodule is on the part.

Many thanks to NathanKell, who has all the credit for this release!

#### [1.43 \(2014-08-24 23:52 UTC\)](#)

- Added licence file (sorry, mods!)
- No longer chokes on null particle emitters.

#### [1.41 \(2014-08-15 11:43 UTC\)](#)

- Fixed scaling of Part values in unnamed TWEAKSCALEEXPONENTS blocks.

#### [1.40 \(2014-08-13 12:55 UTC\)](#)

- Removed [Karbonite](#) cfg, since that project does its own TweakScale config.

#### [1.39 \(2014-08-12 23:03 UTC\)](#)

- Fixed cost calculation for non-full tanks.

#### [1.38 \(2014-08-12 20:45 UTC\)](#)

- Added scaling of FX.
- Added support for [Banana for Scale](#).
- Updated [Karbonite](#) support.
- Fixed a bug where no scalingfactors available due to tech requirements would lead to unintended consequences.

#### [1.37 \(2014-08-05 13:35 UTC\)](#)

- Updated cost calculation.

#### [1.36 \(2014-08-04 21:58 UTC\)](#)

- Updated [Real Fuels](#) and [Modular Fuel Tanks](#) support.

- Added [KSPX](#) support.

#### [1.35 \(2014-08-03 16:24 UTC\)](#)

- Corrected cost calculation.
- Updated to [KSPAPIExtensions](#) 1.7.0

#### [1.34 \(2014-08-02 00:49 UTC\)](#)

- Fixed a bug where repeated scaling led to inaccurate placing of child parts.
- Added [Karbonite](#) support.

#### [1.33 \(2014-07-28 20:00 UTC\)](#)

- Updated for RealFuels 7.1

#### [1.32 \(2014-07-25 22:29 UTC\)](#)

- Updated with KSPAPIExtensions 1.6.3 for 0.24.2

#### [1.31 \(2014-07-25 15:58 UTC\)](#)

- Fixed a bug where parts with defaultScale inaccessible due to tech requirements were incorrectly scaled.

#### [1.30 \(2014-07-25 11:23 UTC\)](#)

- Updated KSPAPIExtensions with 24.1 support.
- Re-enabled Real Fuels support.
- Added support for IPartCostModifier

#### [1.29 \(2014-07-24 11:50 UTC\)](#)

- Fixed [Modular Fuel Tanks](#) support.

#### [1.28 \(2014-07-24 10:32 UTC\)\)](#)

- Fixed the rest of the bug that was partly fixed in 1.27.
- Added support for Tantaress Space Technologies.

#### [1.27 \(2014-07-23 21:44 UTC\)](#)

- Fixed a bug on non-Windows platforms.

#### [1.26 \(2014-07-23 16:60 UTC\)](#)

- Fixed typo in DefaultScales.cfg. type = free, type = surface and so on should now work.
- Updated support for [KW Rocketry](#) and [NFT](#).

#### [1.25 \(2014-07-23 12:31 UTC\)](#)

- [Modular Fuel Tanks](#) yet again supported! (Still waiting for Real Fuels)
- Refactored IRescalable system to be easier for mod authors.
- Fixed a bug where one field could have multiple exponents, and thus be rescaled multiple times.

#### [1.24 \(2014-07-21 20:51 UTC\)](#)

- Nodes are now correctly scaled when parts are duplicated.
- Per-scaletype exponents are now used correctly.

#### [1.23 \(2014-07-21 12:44 UTC\)](#)

- Multiple TweakScale .dlls no longer interfere with each other.

#### [1.22 \(2014-07-20 22:21 UTC\)](#)

- Fixed tanks that refill when you change between ships.
- Fixed tech lookup time, which was way too slow.

#### [1.21 \(2014-07-18 18:47 UTC\)](#)

- Updated for 0.24!
- More flexibility of configuration. Mod authors can now have feature scale definitions per part, per scaletype or global.

#### [1.20 \(2014-06-12 22:15 UTC\)](#)

- New algorithm for rescaling attach nodes. Tell me what you think!
- Added [Deadly Reentry Continued](#) and



#### [1.19 \(2014-06-06 10:56 UTC\)](#)

- Added support for tech requirements for non-freeScale parts.

#### [1.18 \(2014-06-04 00:58 UTC\)](#)

- Factored out Real Fuels and Modular Fuel Tanks support to separate dlls.

#### [1.17 \(2014-06-03 22:21 UTC\)](#)

- Fixed bug where attachment nodes were incorrectly scaled after reloading. This time with more fix!
- Added support for [Near Future Technologies](#).

#### [1.16 \(2014-06-03 21:31 UTC\)](#)

- Fixed bug where attachment nodes were incorrectly scaled after reloading.

#### [1.15 \(2014-06-03 20:50 UTC\)](#)

- Finally squished the bug where crafts wouldn't load correctly. This bug is present in 1.13 and 1.14, and affects certain parts from Spaceplane+, MechJeb, and KAX.

#### [1.14 \(2014-06-03 19:50 UTC\)](#)

- Fixed a bug where nodes with the same name were moved to the same location regardless of correct location. (Only observed with KW fairing bases, but there could be others)

#### [1.13 \(2014-06-02 18:25 UTC\)](#)

- Added support for [MechJeb](#), [Kerbal Aircraft eXpanion](#), [Spaceplane+](#), [Stack eXTensions](#), [Kerbal Attachment System](#), [Lack Luster Labs](#), [Firespitter](#), [Taverio's Pizza and Aerospace](#), [Better RoveMates](#), and [Sum Dum Heavy](#)

## [Industries Service Module System.](#)

- Fixed a bug where Modular Fuel Tanks were not correctly updated.

### [1.12 \(2014-06-02 11:10 UTC\)](#)

- Added support for [ÄÏ;ÄÏ¾ÄÏ"ÄÏ¾ÄÏ](#).
- No longer scaling heatDissipation, which I was informed was a mistake.

### [1.11 \(2014-06-01 20:24 UTC\)](#)

- Removed silly requirement of 'name = \*' for updating all elements of a list.
- Added .cfg controlled scaling of Part fields.

### [1.10 \(2014-06-01 14:53 UTC\)](#)

- Added support for nested fields.

### [1.9 \(2014-05-31 00:57 UTC\)](#)

- Fixed a bug where rescaleFactor caused erroneous scaling.
- Added (some) support for [Kethane](#) parts.

### [1.8.1 \(2014-05-30 22:41 UTC\)](#)

Don't use 1.8.1, either. 🤖

### ~~1.8~~

Don't use 1.8

### [1.7 \(2014-05-22 21:09 UTC\)](#)

- Removed spurious debug log printing.
- You can now use a list of value instead of an exponent!
- Exponents and value lists can be specified per part, not just per module type. (And per-part values take precedence)
- Fixed erroneous scale for Rockomax 48-7S, and added missing scale for KW Rocketry 2.5m Nose Cone.

#### [1.6 \(2014-05-22 14:04 UTC\)](#)

- Fixed a problem where parts were scaled back to their default scale after loading, duplicating and changing scenes.
- Fixed defaultScale for Rockomax 48-7S (was 625, should be .625. Who'd guess someone'd notice that, eh?)

#### [1.5.0.1 \(2014-05-21 00:36 UTC\)](#)

- Fixed a bug in 1.5 where scale was not correctly preserved.
- Updated KSPAPIExtensions.dll - please delete the version in GameData/.
- Added support for [KSO](#).

#### [1.5 \(2014-05-20 22:23 UTC\)](#)

- Changed from hardcoded updaters to a system using .cfgs.
- Made registration of custom IRescalables possible.

#### [1.4 \(2014-05-20 17:53 UTC\)](#)

- Fixes compatibility with GoodspeedTweakScale (but not old versions of TweakScale, but that should be a much, much smaller problem anyway).
- Added scaling support for [KW Rocketry](#), [NovaPunch](#) and [KSP Interstellar](#).

#### [1.3 \(2014-05-19 22:49 UTC\)](#)

- Fixed a bug where parts would get rescaled to stupid sizes after loading.
- Breaks compatibility with old version of the plugin (pre-1.0) and GoodspeedTweakScale. 🤖

#### [1.2 \(2014-05-18 22:00 UTC\)](#)

- Fixed default scale for freeScale parts.
- Fixed node sizes, which could get

absolutely redonkulous. Probably not perfect now either.

- [B9 Aerospace](#), [Talisar's Cargo Transportation Solutions](#), and NASA Module Manager configs.
- Now does scaling at onload, removing the problem where the rockets gets embedded in the ground and forcibly eject at launch.
- Fixed a silly bug in surface scale type.

#### 1.1 (2014-05-17 23:30 UTC):

- Added scaling support for [B9 Aerospace](#) and [Talisar's Cargo Transportation Solutions](#).
- Will now correctly load (some) save games using an older version of the plugin.

#### 1.0 (2014-05-16 18:00 UTC):

- Initial Release!

#### Supported Mods

- [B9 Aerospace](#)
- [Cargo Transportation Solutions](#)
- [KW Rocketry](#)
- [NovaPunch 2](#)
- [KSP Interstellar](#)
- [Kerbin Shuttle Orbiter System](#)
- [Kethane](#)
- [KOSMOS](#)
- [MechJeb](#)
- [Kerbal Aircraft eXpansion](#)
- [Spaceplane+](#)
- [Stack eXTensions](#)
- [Kerbal Attachment System](#)
- [Lack Luster Labs](#)
- [Firespitter](#)
- [Taverio's Pizza and Aerospace](#)
- [Better RoveMates](#)
- [Sum Dum Heavy Industries Service Module System](#)
- [Near Future Technologies](#)



- [Deadly Reentry Continued](#)
- [Large Structural/Station Components](#)
- [Infernal Robotics](#)
- [USI Kolonization Systems \(MKS/OKS\)](#)
- [Klockheed Martian Special Parts](#)
- [Tantares Space Technologies](#)
- [Karbonite](#)
- [Modular Fuel Tanks](#)
- [Real Fuels](#)
- [KSPX](#)
- [RLA Stockalike](#)
- [Smart Parts](#)

Known Issues/TODO:

- More mods should be supported. I'm open to suggestions.
- Enlarged engines burn too hot.
- Exhaust trails are the wrong size. I think.
- Support for more modules (addons)?
- Suboptimal interactions with [Procedural Parts](#) and [Procedural Fairings](#) - parts may intersect after loading.
- I have no idea how [FAR](#) feels about the updating of drag and lift.
- Weird TWR in MechJeb. ([ref post 22](#))
- Camera can end up in weird positions. ([ref post 22](#))
- Add ModuleEngineConfig support.
- Do what [Taverius says](#) about engines and possibly other stuff.

Edited March 8, 2015 by goldenpeach

Limbo!

Lack, frizzank, DaniDE and 14 others like this



Developer of [TweakScale - Rescale Everything!](#) and [StripSymmetry](#).



**DasBananenbrot**

Senior Rocket Scientist



Posted May 16, 2014

Report post

This looks interesting definitely will check this out .



Members

+ 75

1,217 posts

Location: Kermany



## Gaalidas

Capsule Communicator



Members

+ 266

1,707 posts

Posted May 16, 2014

Report post 

Wow, this is amazing. I was just thinking I needed to try to expand my own custom tweaks to include more scaling using the older plugin.

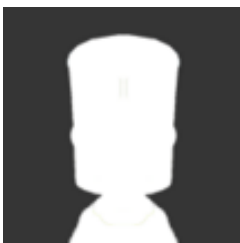


*...It all made a LOT more sense when it was still in my head.*

Official mascot of Kerbal Foundries.

## sirkut

Capsule Communicator



Members

+ 490

2,560 posts

Posted May 16, 2014

Report post 

One thing. What's the license for this project? Are modders allowed to include this dll with our mods?

This is pure awesome. I'm glad it works with the robotic parts that ZodiussInfuser has done. Can't wait to put this into place. Great work!



Want robotic parts?

Download the Magic Smoke Industries Infernal Robotics plugin and parts!

[Magic Smoke Industries Infernal Robotics](#)

## Biotronic

Sr. Spacecraft Engineer



Members

+ 263

359 posts

Posted May 16, 2014 (edited)

Report post

▼ sirkut said:

One thing. What's the license for this project?

Sorry, forgot about that. It's [CC BY-NC-SA](#). I want to make it [WTFPL](#) (my favorite licence), but I've asked Gaius for permission to do that and haven't yet received an answer. (I'm not holding that against him, just clarifying)

▼ sirkut said:

Are modders allowed to include this dll with our mods?

Absolutely. That's basically what it's made for.

Edited May 16, 2014 by Biotronic



Developer of [TweakScale - Rescale Everything!](#) and [StripSymmetry](#).



## camlost

Junior Rocket Scientist



Members

+ 78

514 posts

Posted May 16, 2014

Report post

This is revolutionary.



[Advanced Jet Engine](#)



[Advanced VTOL Control System](#)

## ZodiusInfuser

Doctor of Robotics



Posted May 16, 2014

Report post

This is a great way for mod creators to reduce part spam!!  
Thanks Biotronic for taking this over and expanding on it.



Members

+ 456

1,240 posts

[Infernal Robotics - Next](#) - For all your actuation needs!



Check out the [Model Rework Thread!](#)

## NathanKell

Keepin' it Real



Members

+ 5,792

13,276 posts

Posted May 16, 2014

Report post



This is awesome!

A suggestion: my plan for this (and boy am I glad I don't have to write it 🤖) was to do the stats changing via confignodes and reflection. For instance, in the MODULE:

```
ChangeStats
{
    // key = member to change,
    value = power (applied to scaleFactor)
    to use
    mass = 2.0
    ModuleDeployableSolarPanel
    {
        chargeRate = 2.0
        panelMass = 2.0
        //note that flowRate is
        set automatically each tick by the
        module and doesn't need to be changed.
    }
}
```

Note that this would allow the plugin to be applied to arbitrary parts/modules via only cfg changes, not source changes, and would allow users to specify how stats should be changed.

Two questions:

1. How do you handle parts with multiple MODELS? I assume you're changing the part's root model transform, not the individual submodels' transforms?

2. Have you considered allowing non-linear scaling? That

would be very useful for, say, scaling fuselages.



[Real Solar System](#) -- [Real Fuels](#)



[Realism Overhaul](#) -- [Realistic Progression Zero](#)

[Reaching for the Stars: Earth's Rocket Age](#),

[\[RftS\] Jane's All the World's Hunters of the French Civil War](#)

## Biotronic

Sr. Spacecraft Engineer



Members

+ 263

359 posts

Posted May 16, 2014

Report post

▼ **NathanKell said:**

This is awesome!

A suggestion: my plan for this (and boy am I glad I don't have to write it 🤖) was to do the stats changing via confignodes and reflection. For instance, in the MODULE:

```
ChangeStats
{
    // key = member to change,
    value = power (applied to
    scaleFactor) to use
    mass = 2.0
    ModuleDeployableSolarPanel
    {
        chargeRate = 2.0
        panelMass = 2.0
        //note that
        flowRate is set automatically each
        tick by the module and doesn't need
        to be changed.
    }
}
```

Note that this would allow the plugin to be applied to arbitrary parts/modules via only cfg changes, not source changes, and would allow users to specify



how stats should be changed.

That is kinda neat, and maybe I'll try it out. Given I already use scale types to make templates, one could define these once and be done with it. No more need for a new dll version to support a new addon's modules. I'm pretty sure there will be exceptions, though - where one should call an Update() function or somesuch after changing things.

▼ **NathanKell said:**

Two questions:

1. How do you handle parts with multiple MODELS? I assume you're changing the part's root model transform, not the individual submodels' transforms?

To be frank, that's not code I've written, so I had to check the source. This line does the scaling:

```
savedScale =  
part.transform.GetChild(0).localScale =  
Vector3.Scale(basePart.transform.GetChild(0).localScale, rescaleVector);
```

So it changes only the transform of the part, no special handling is done for multiple models.

▼ **NathanKell said:**

2. Have you considered allowing non-linear scaling? That would be very useful for, say, scaling fuselages.

Actually, I have. Logarithmic scaling for instance, would be nice for at least some use cases.



Developer of [TweakScale - Rescale Everything!](#) and [StripSymmetry](#).



Resident Kurmudgeon



Members

+ 987

4,002 posts

You know, if this plugin could change textures along with (some or all) sizes (should be up to the modder) it would be an almost complete solution for part bloat. Imagine not having stretched textures for a 3.75m part scaled up from a 0.625 base model. Modders could make scaled down versions of textures and define them along with each size.

Maybe something like this:

```
MODULE
{
    modulename = scaleTextures
    textures
    {
        name =
modfolder/parts/somepart/partColourMap1
        name =
modfolder/parts/somepart/partColourMap1
        name =
modfolder/parts/somepart/partColourMap2
    }
}
```

Here for example you have the same texture for the first two sizes, but a larger texture for the third size.

Snjo might be able to help you with the texture switching code.

In fact this could well be achieved with firespitter by letting the player change textures only if they want to, but an automatic solution could also be useful.

Edited May 16, 2014 by Cpt. Kipard



**Public Service  
Announcements**

[Mod Dev Links](#)

[Compilation](#)

**Requests**

[...](#)

## ialdabaath

Junior Rocket Scientist



Members

+ 89

549 posts

Posted May 16, 2014

Report post 

This may sound crazy, but can you include options for adding a 7.5m and 10m scale?



## ZodiusInfuser

Doctor of Robotics



Members

+ 456

1,240 posts

Posted May 16, 2014

Report post 

### Cpt. Kipard said:

You know, if this plugin could change textures along with (some or all) sizes (should be up to the modder) it would be a complete solution for part bloat. Imagine not having stretched textures for a 3.75m part scaled up from a 0.625 base model. Modders could make scaled down versions of textures and define them along with each size.

Maybe something like this:

```
MODULE
{
    modulename = scaleTextures
    textures
    {
        name =
modfolder/parts/somepart/partColour
Map1
        name =
modfolder/parts/somepart/partColour
Map1
        name =
modfolder/parts/somepart/partColour
Map2
    }
}
```

Here for example you have the same texture for the first two sizes, but a larger texture for the third size.

Snjo might be able to help you with the texture switching code.

In fact this could well be achieved with firespitter by letting the player change textures only if they want to, but an automatic solution could also be useful.

Extending on this, having a mod like tweakscale that can switch between a set of models of various lengths would be extremely useful for reducing structural part bloat in IR, whilst retaining the visual style. For instance I could model various lengths of truss and using a tweakable, be able to switch between the models (with the collision and attachment getting scaled along a specific axis). I could then have 1m, 2m, 3m, 4m, 5m lengths all appear as a single part in the editor 🤖

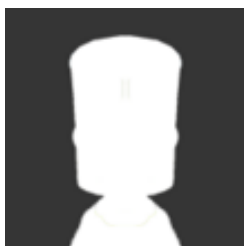


[Infernal Robotics - Next](#) - For all your actuation needs!  
Check out the [Model Rework Thread!](#)



## quantumpion

Rocketry Enthusiast



Members

+ 41

90 posts

Posted May 16, 2014 (edited)

Report post



Does this mod work with stock parts, or does it require modded parts with the added code?

Edited May 16, 2014 by quantumpion



## Taki117

Flight Director

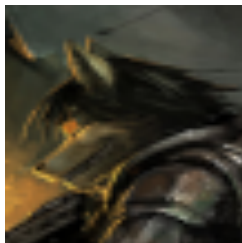


Posted May 16, 2014

Report post



This is amazing! I certainly hope more mods get on the band wagon.

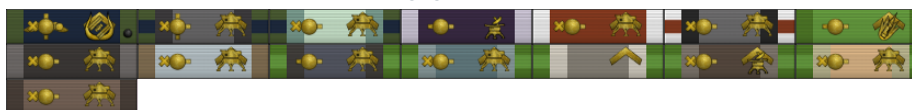


Members

+ 456

3,704 posts

Click for my youtube Channel



## CaptainKipard

Resident Kurmudgeon



Members

+ 987

4,002 posts

Posted May 16, 2014

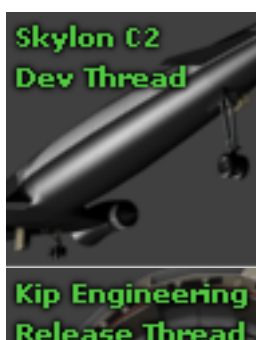
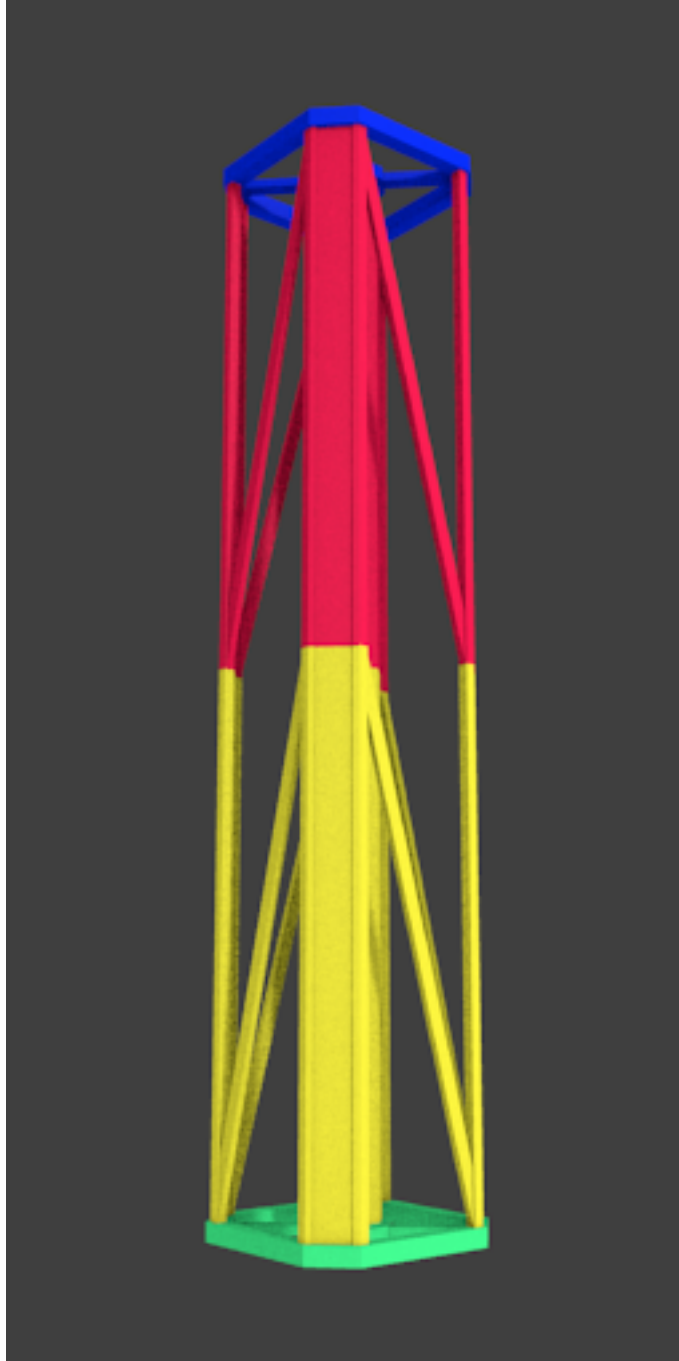
Report post



### ▼ ZodiusInfuser said:

Extending on this, having a mod like tweakscale that can switch between a set of models of various lengths would be extremely useful for reducing structural part bloat in IR, whilst retaining the visual style. For instance I could model various lengths of truss and using a tweakable, be able to switch between the models (with the collision and attachment getting scaled along a specific axis). I could then have 1m, 2m, 3m, 4m, 5m lengths all appear as a single part in the editor 🤖

One idea I tried to push for stretchy tanks several months ago was a modular way of scaling, by repeating middle section models and capping with end cap models. This could be implemented in a couple different ways.



**Public Service  
Announcements**

[Mod Dev Links](#)

[Compilation](#)

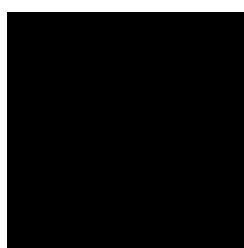
[Engineer's](#)

**Requests**

[Ask Squad to](#)

**phoenix\_ca**

Hrmph.



Posted May 16, 2014 (edited)

Report post

Congrats on release. 🤖

One question: Is this backward compatible with the original TweakScale you forked it from? That is, is it okay to get rid of it now?

▼ **ialdabaath said:**

Members

+ 146

1,429 posts

This may sound crazy, but can you include options for adding a 7.5m and 10m scale?

Technically that's already implemented. All it should take is a change to the part config.

▼ **quantumpion said:**

Does this mod work with stock parts, or does it require modded parts with the added code?

That's explained in the OP (which you should've read).  
Short answer: Yes.

Edited May 16, 2014 by phoenix\_ca



[Geostationary Orbit Constellations for Dummies \[MechJeb\]](#) x ▼

## Biotronic

Sr. Spacecraft Engineer



Members

+ 263

359 posts

Posted May 16, 2014

Report post

▼ **ialdabaath said:**

This may sound crazy, but can you include options for adding a 7.5m and 10m scale?

I won't, but if you want them, just take a look at defaultscales.cfg in the tweakscale folder - you can easily add it yourself.

▼ **quantumpion said:**

Does this mod work with stock parts, or does it require modded parts with the added code?

The download includes a module manager config for stock parts, so many stock parts are resizable after installing this mod. (Note to self: Add module manager to download)



Developer of [TweakScale - Rescale Everything!](#) and [StripSymmetry](#).





## Biotronic

Sr. Spacecraft Engineer



Members

+ 263

359 posts

Posted May 16, 2014

Report post

▼ phoenix\_ca said:

Congrats on release. 🤖

Thanks.

▼ phoenix\_ca said:

One question: Is this backward compatible with the original TweakScale you forked it from? That is, is it okay to get rid it now?

Should be. I've added code that's supposed to handle that, but I will not guarantee it's bug free.



Developer of [TweakScale - Rescale Everything!](#) and [StripSymmetry](#).



## NathanKell

Keepin' it Real



Members

+ 5,792

13,276 posts

Posted May 17, 2014

Report post

▼ Biotronic said:

That is kinda neat, and maybe I'll try it out. Given I already use scale types to make templates, one could define these once and be done with it. No more need for a new dll version to support a new addon's modules. I'm pretty sure there will be exceptions, though - where one should call an Update() function or somesuch after changing things.

You should be safe doing it OnLoad.

Speaking of--the "jumping" on launch is symptomatic of scale being set OnStart rather than OnLoad. Stretchy moved to setting scale OnLoad and it doesn't have that issue anymore.

▼ Biotronic said:

Actually, I have. Logarithmic scaling for instance, would be nice for at least some use cases.

Sorry, I wasn't clear, I meant more-than-one-axis, i.e. non-uniform scaling.



[Real Solar System](#) -- [Real Fuels](#)



[Realism Overhaul](#) -- [Realistic Progression Zero](#)

[Reaching for the Stars: Earth's Rocket Age](#),

[\[RftS\] Jane's All the World's Hunters of the French Civil War](#)

## ZodiusInfuser

Doctor of Robotics



Members

+ 456

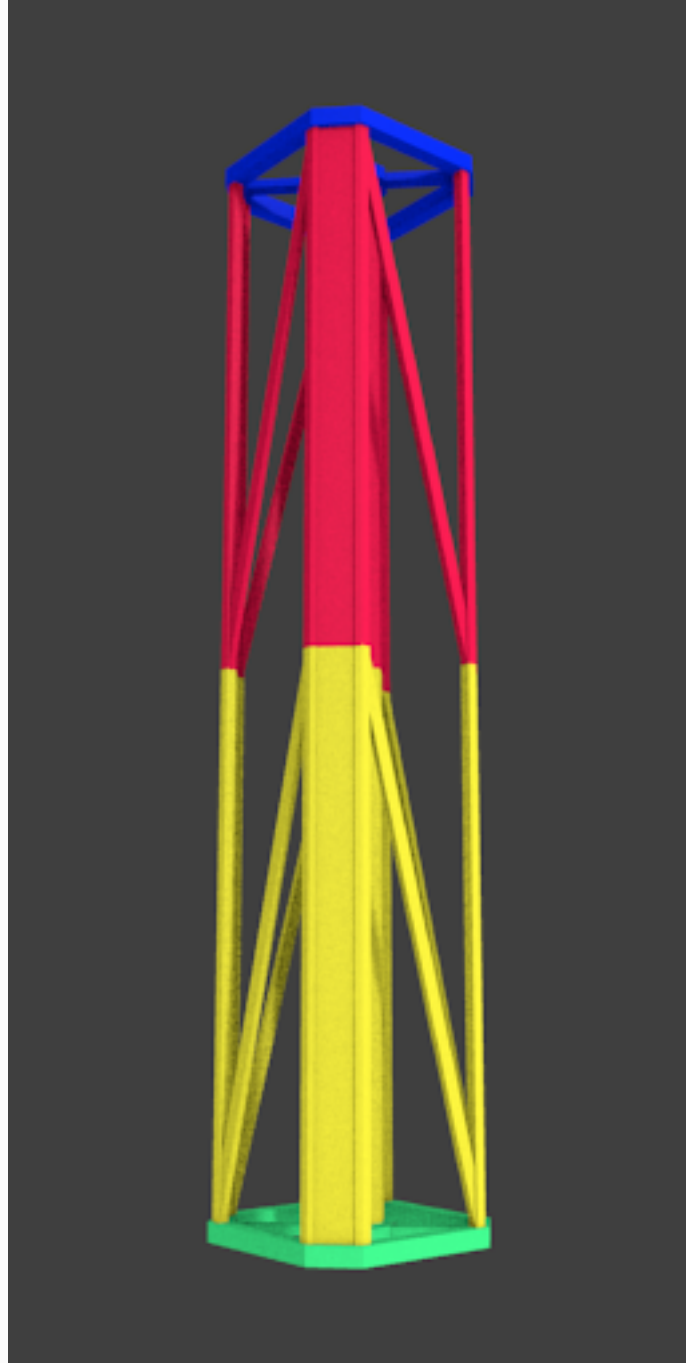
1,240 posts

Posted May 17, 2014

Report post

### Cpt. Kipard said:

One idea I tried to push for stretchy tanks several months ago was a modular way of scaling, by repeating middle section models and capping with end cap models. This could be implemented in a couple different ways.



🤖 If that was a real feature, I'd definitely put the effort into redesigning my parts for it. I would say what's lacking from that concept are mid-caps to place between repeated sections, and the ability to apply a rotation from one section and cap to the next (since I use a 90 degree offset). Obviously the collision/attachment nodes could just be scaled along the length.

I presume no one took you up on your idea at the time?



[Infernal Robotics - Next](#) - For all your actuation needs!  
Check out the [Model Rework Thread!](#)





Members

+ 515

860 posts

This is fantastic. Done a quick experiment with this and LLL. Looks promising, very promising.

Seems to handle multiple MODEL{}s just fine.



**LACK LUSTER LABS**  
SYNERGISING ROCKETS AND STUFF

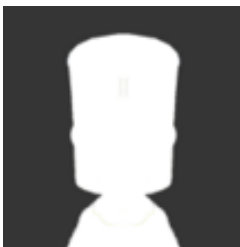


Mods: [LLL](#), [SXT-Stock extension](#), [KSC++](#)

[Addon development thread](#)

## Xobotun

Bottle Rocketeer



Members

0

16 posts

Posted May 17, 2014

Report post

That's a really great mod! However there are some issues:

1. Visual bug: Engine fairings still remain of original size (e.g. I shrunk mainSail engine into 0.625m teritrarySail, but it still draws an unbelievably big fairing around my satellite xD)

2. An incompatibility: mechJeb shows weird TWR and times. You should definitely add a characteristics preview beneath sizeChanger bar.

3. Annoying bug: Having too many shrunk parts on your vessel (especially from NASA to .625) makes your flight camera behave oddly - it thinks that rocket is much longer and moves some meters downwards. As a result, I see only engine trails:)

Besides that, it is a really big mod!



## CaptainKipard

Resident Kurmudgeon



Posted May 17, 2014 (edited)

Report post



**ZodiusInfuser said:**

I presume no one took you up on your idea at the



Members

+ 987

4,002 posts

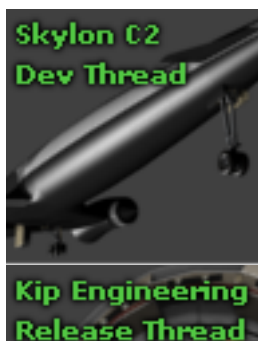
time?

To their detriment 🙄 no. It's been ages and it didn't even make its way into a "todo" list for the people who continued the development of that plugin.

Actually I lied a little. The idea I proposed back then was more similar to the way the stability enhancers work, i.e. stretching the model, keeping the UVs proportional, and capping with end caps.

What I'm proposing here should be simpler to make, and frankly better IMO. After all IRL things are made of sections.

Edited May 17, 2014 by Cpt. Kipard



**Public Service  
Announcements**

[Mod Dev Links](#)

[Compilation](#)

[Engineer's](#)

**Requests**

[Ask Squad to](#)

## ZodiusInfuser

Doctor of Robotics



Members

+ 456

1,240 posts

Posted May 17, 2014 (edited)

Report post

▼ **Cpt. Kipard said:**

To their detriment 🙄 no. It's been ages and it didn't even make its way into a "todo" list for the people who continued the development of that plugin.

Actually I lied a little. The idea I proposed back then was more similar to the way the stability enhancers work, i.e. stretching the model, keeping the UVs proportional, and capping with end caps.

The question is whether KSP lets you instance part geometry. If not then it would be a case that the model would need to contain all the geometry needed to cover all planned size variations. I PM'ed Biotronic with a suggestion for this but have no idea if it's viable, as it relies

on the "showing" and "hiding" of specific model nodes from code. This approach may also have a performance impact.

Quote

What I'm proposing here should be simpler to make, and frankly better IMO. After all IRL things are made of sections.

It could also give modders more control over the final visual appearance of the scaled parts, which I like 🤖

Edited May 17, 2014 by [ZodiusInfuser](#)



[Infernal Robotics - Next](#) - For all your actuation needs!  
Check out the [Model Rework Thread!](#)

## CaptainKipard

Resident Kurmudgeon



Members

+ 987

4,002 posts

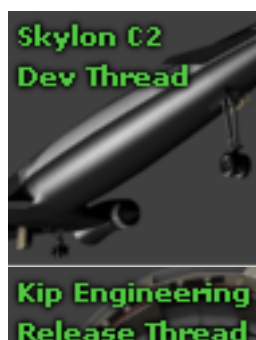
Posted May 17, 2014

Report post

ZodiusInfuser said:

The question is whether KSP lets you instance part geometry.

It must. Welding the same mu files together works just fine, so why not?



Public Service  
Announcements

[Mod Dev Links](#)

[Compilation](#)

[Requests](#)

1

2

3

4

5

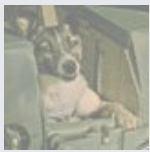
6

NEXT



Page 1 of 67

⚠ This topic is now closed to further replies.



**GO TO TOPIC LISTING**


Add-on Releases

**NEXT UNREAD TOPIC**

[KSP V1.4.5] TAC Fuel Balancer v2.20



[Home](#) > [Add-ons](#) > [Add-on Releases](#) >

 Unread Threads since my last visit

☒ Mark site read

[0.90] TweakScale - Rescale Everything! (v1.50 - 2014-12-24 10:40 UTC)

Language ▼

[Privacy Policy](#)

[Contact Us](#)

©2018 Take-Two Interactive Software, Inc.

Powered by Invision Community