Kerbalism Documentation

Release v2.1.1

N70 PiezPiedPy Gordon-Dry

Sep 07, 2019

Contents

This wiki is not updated anymore. Please use the Kerbalism GitHub wiki.

Introduction

Kerbalism is a gameplay mod for **Kerbal Space Program** that tries to represent some of the problems a real space program must overcome. Anything will happen coherently to loaded and unloaded vessels alike, without exceptions. All mechanics can be enabled, disabled and utterly configured.

2.1 These are the mechanics that are simulated

2.2 Environment

- temperature, radiation, space weather

2.3 Habitat

- living space, comforts, pressure, CO2 levels, humidity

2.4 Reliability

- malfunctions, critical failures, manufacturing quality

2.5 Signal

- telemetry, relaying, signal strength, data rate attenuation

2.6 Science

- data storage, collection, analysis and transmission

2.7 Resources

- consumption and production in background

2.8 Kerbals

- biological and psychological needs, environmental hazards

So, you're about to go to Mun?

Kerbalism adds about six new ways to get your crew killed. If this is your first attempt at Mun with Kerbalism, read this tutorial or bring some body bags. You have been warned.

3.1 Before you go. . .

Let's assume you've just started a new career game and are about to do your first missions to Mun. You didn't expand your VAB or launch pad, so you're limited to vessels with a maximum of 30 parts and 18 tons. A crewed mission to Mun isn't trivial under these circumstances, and with Kerbalism it's going to be a new challenge, even if you've done many missions to Mun in earlier games.

In order to survive with Kerbalism, Kerbals need enough food, water and oxygen for the duration of the entire mission. Command pods contain enough of these to last for 5 days, which is just enough for a quick trip to Mun, like a flyby or a touch and go landing. Anything beyond that will require additional supplies, so stick to Mun - and Mun only. Do not stay for more than a few hours. And don't even think about Minmus, not yet. Any deviation from that will get your crew killed for a number of reasons, starvation just being one of them.

The most important resource on any crewed vessel, after oxygen, is electricity. Without electricity, your crew will suffocate, freeze or burn to death, whichever kills them first. Not having any food left won't be your number one problem when climatization fails and you're about to burn up in the sun. So, let's take care of that first.

3.2 Minimum requirements

You need a reliable method to generate electricity. If you haven't researched solar panels or fuel cell generators yet, go ahead and do that first. You won't survive a trip to Mun without a way to generate electricity.

If at all possible, don't rely on solar panels only, bring a fuel cell. As a general rule, fuel cells will keep your Kerbals alive, so get into the habit of using them early. The thing about solar panels is that they need to see the sun, and all of your missions will spend a lot of time in shadow.

3.3 Ship construction

If you're going with fuel cells (recommended), attach a H2+O2 (Hydrogen+Oxygen) fuel cell generator to your ship. Also attach a small pressurized tank that contains Hydrogen (by default, pressurized tanks contain oxygen, so make sure to configure it in the VAB), and if you can an extra tank for Oxygen. The fuel cell will consume hydrogen and oxygen while it runs, which might leave your kerbals with no O2 left to breahte. Solar panels will reduce the consumption of H2 and O2.

While a H2+O2 fuel cell is running, it also produces some water. Kerbalism won't run any process if the output of that process cannot be stored or dumped, so set your fuel cell to dump the water that cannot be stored. You can do that in flight, so don't panic if you forgot to do that in the VAB. Excess water won't be a problem later since your crew will be drinking it.

Other than that, you can build your vessel just like you did without Kerbalism. However, do not add any shielding to your command pod. You won't need it for Mun as the trip there is very short, and a fully shielded pod has its own issues. For instance, it won't float on water.

3.4 Launch window

If you don't have a fuel cell, don't launch unless you're certain that Mun will not pass through Kerbins shadow while you're up there. At the beginning of a new game, Mun is just about to pass behind Kerbin - so wait a couple of days before you launch. Otherwise your crew will freeze in the shadow.

3.5 Just before launch

If you have a fuel cell, you probably don't want it to be running all the time. Kerbalism comes with an on board device manager that can turn on your fuel cell when your batteries are empty, and turn it back off when your batteries are charged. Doing that will save you a lot of Hydrogen and Oxygen, especially when you also have solar panels that recharge your batteries.

3.6 What's next?

Launch! Fingers crossed, you'll make it back from Mun before you run out of oxygen, food, water, electricity or hydrogen.

Notice how humidity in your command pod rises throughout your trip. Remember all the water your crew is drinking? It's dripping from your instrument panels now. Not an ideal situation, but for a short trip not a reason of concern. However, for longer missions you will need extra ECLSS units with humidity controllers.

Your next steps will be towards Minmus, which takes considerably longer. Make sure to bring provisions for about 20 days, and bring a humidity controller this time. You will need it.

How to recycle O2 and Water

You can recycle O2 and water using Kerbalisms life support systems and chemical processors. Follow this guideline to extend your mission durations indefinitely and save literally tons of mass you don't have to bring on your journeys.

1. Add a small tank for waste water.

2. Have a water recycler in one of your life support units, set it to dump excessive CO2 and Ammonia.

3. Add a small tank for hydrogen and a small tank for CO2. They can be empty, but I'd fill up the hydrogen.

4. Have a chemical plant running a Sabatier process. Set that to dump any excessive liquid fuel.

5. Have a chemical plant running Water Electrolysis.

While this method is highly effective in reducing the mass of your vessel, it also is very risky. You now have 3 components that can break down, and if any one of them cannot be repaired, your crew will use up what little O2 is stored in the tanks and then die. So think about redundancy when using this method!

How to **Recycle O2 and Water**

```
           Waste        Food

                  Kerbals
                  Food: 0.27/day
                  Water: 0.14/day
                  Oxygen: 37.5/day

   WasteAtmosphere          WasteWater
   Excessive CO2 in Atmosphere
   (causes CO2 poisoning)

        Scrubber          WaterRecycler
                       *            *

           CO2              Ammonia

              Sabatier
                   *
```
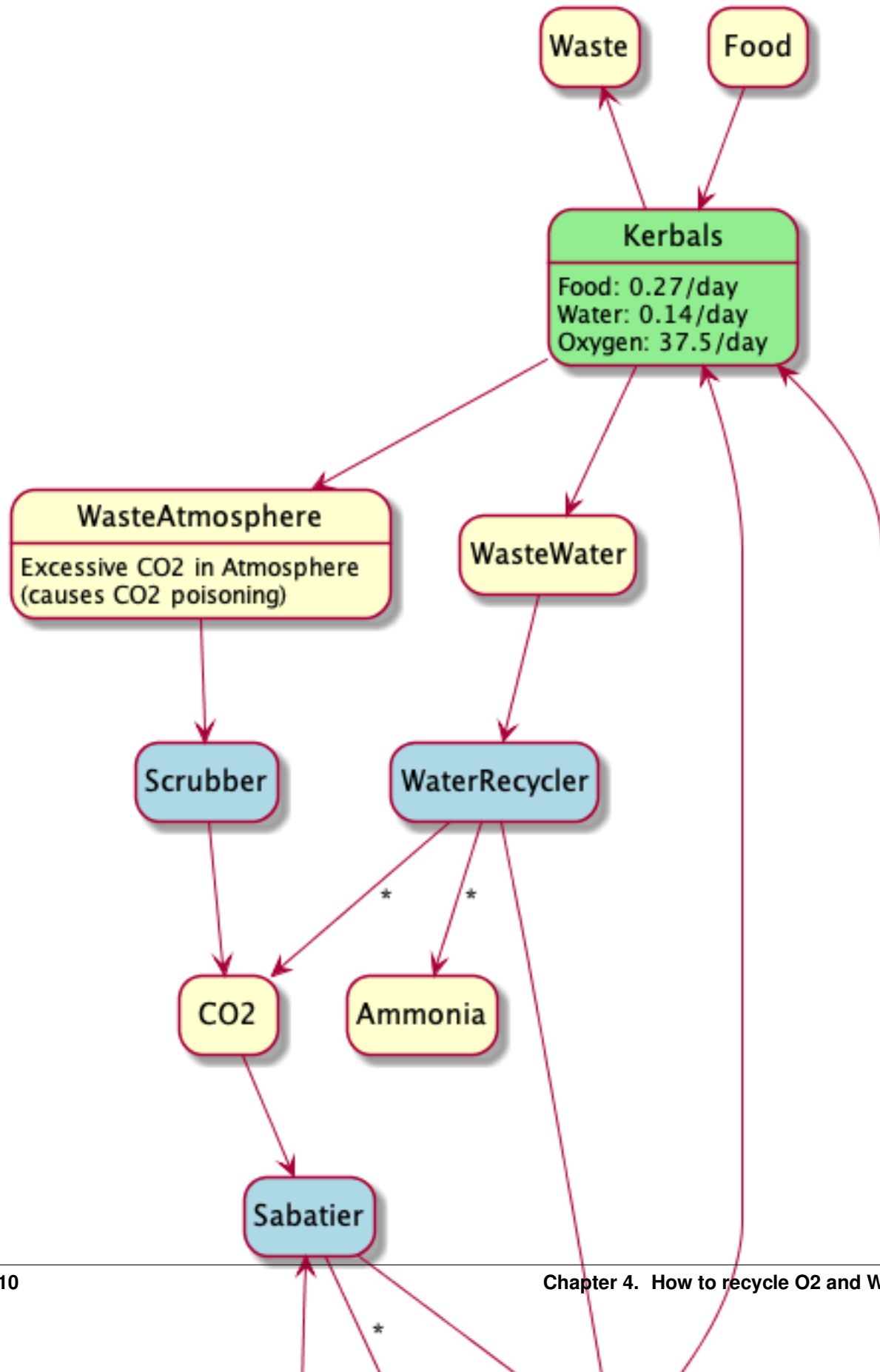
4.1 You can do more...

There are many processes available in Kerbalism, and you can do any number of things with them. This chart will help you navigate the various processes and resources.

Kerbalism Processes

CHAPTER 5

Environment
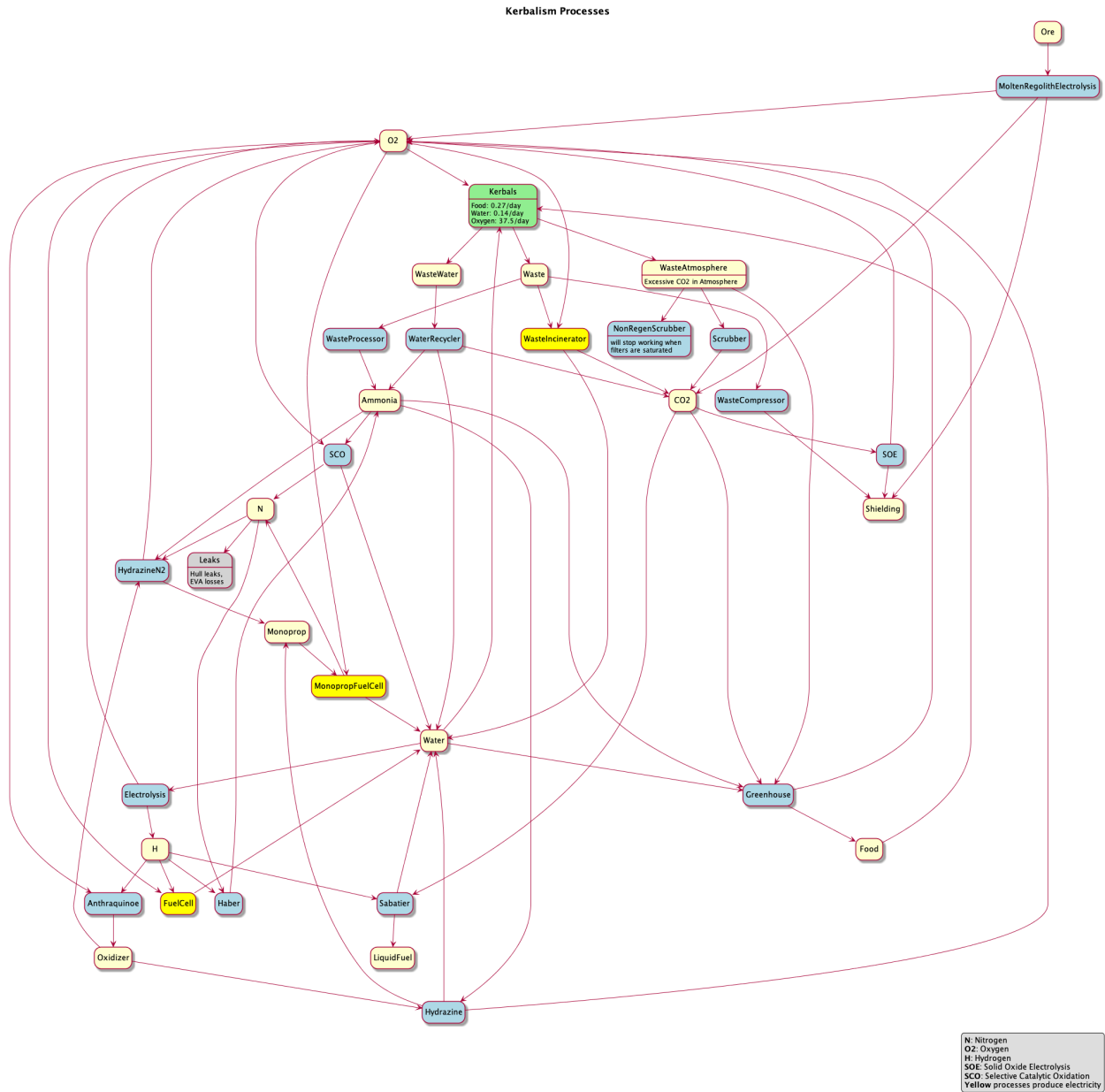
5.1 Temperature

Temperatures in space range from ridiculously low to extremely high. The temperature model in Kerbalism considers

- solar radiation *(the energy flux coming from a star, if not occluded)*
- albedo radiation *(the energy flux reflected from a celestial body towards a vessel)*
- body radiation *(the radiative cooling flux from a nearby celestial body)*
- cosmic background radiation

The temperature is then obtained according to the Stefan-Boltzmann law assuming the vessel is a perfect black body. Inside an atmosphere, the stock atmospheric temperature model is used instead.

5.2 Radiation

Celestial bodies interact in complex ways with radiation. Some have a magnetopause that shields radiation. Others have regions populated by extremely charged particles. The magnetopause is simply a sphere, possibly deformed along the body->star vector to define a magnetotail.

This is modeled with *radiation fields*, regions of space around a celestial body that have an associated radiation level. The overall radiation level for a vessel is determined by evaluating all the fields overlapping at the vessel position.

These fields are rendered in map view or the tracking station. They can be toggled by pressing *Keypad 0/1/2/3*, or by using the *Body Info* window.
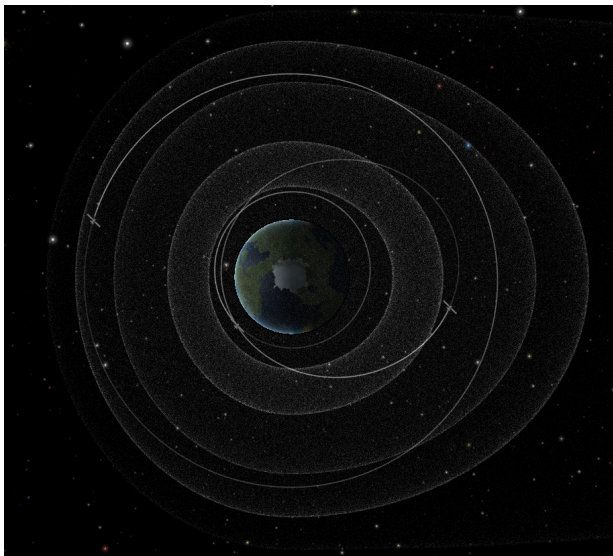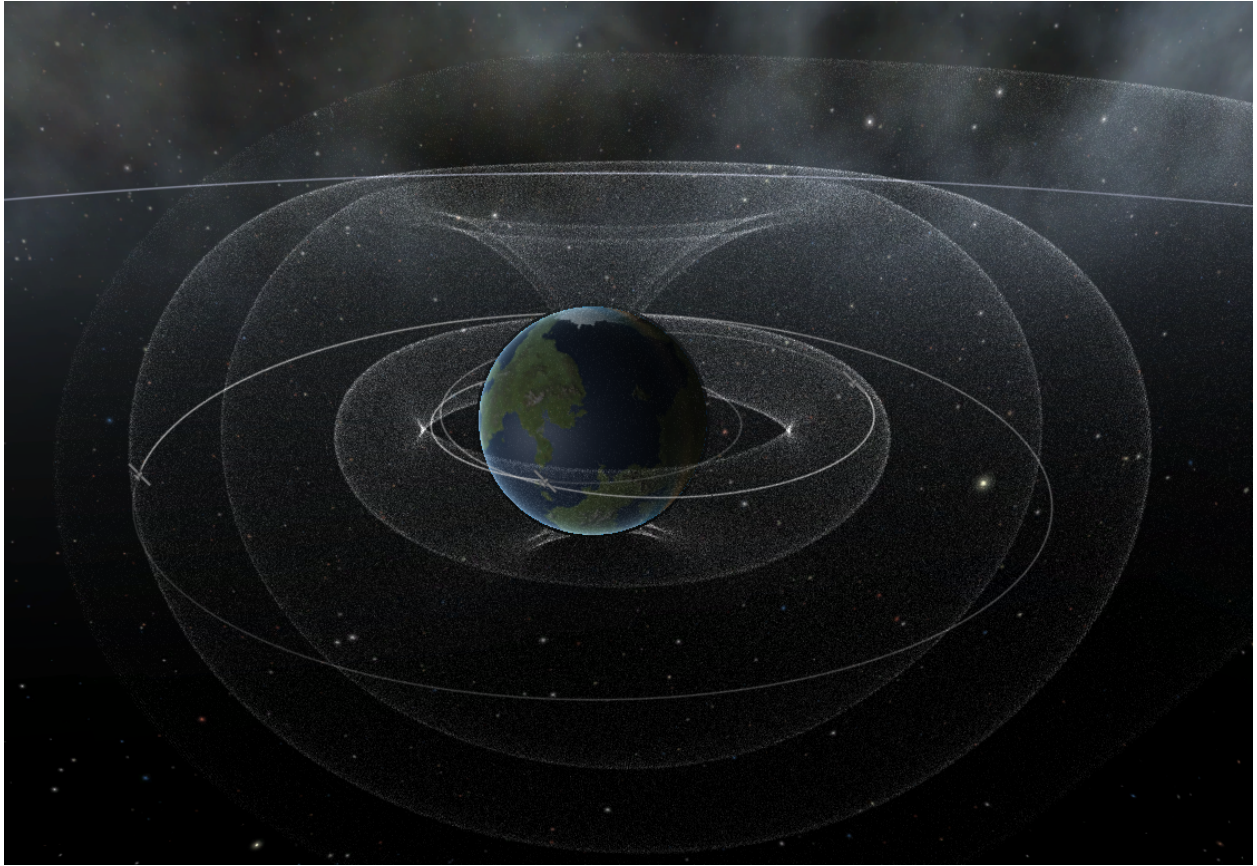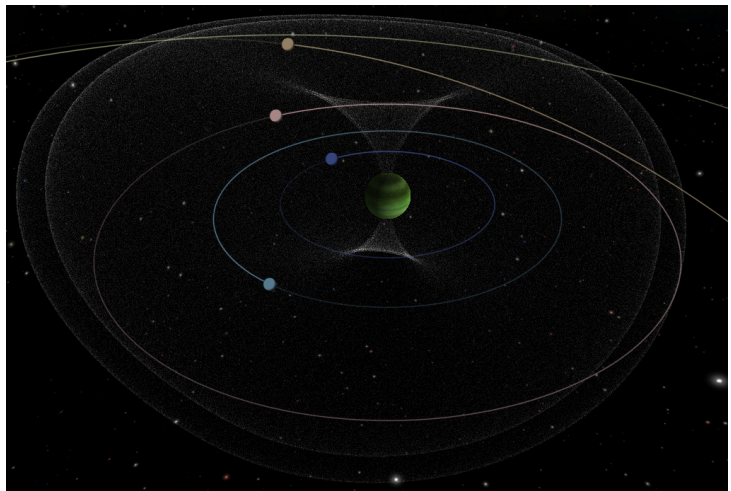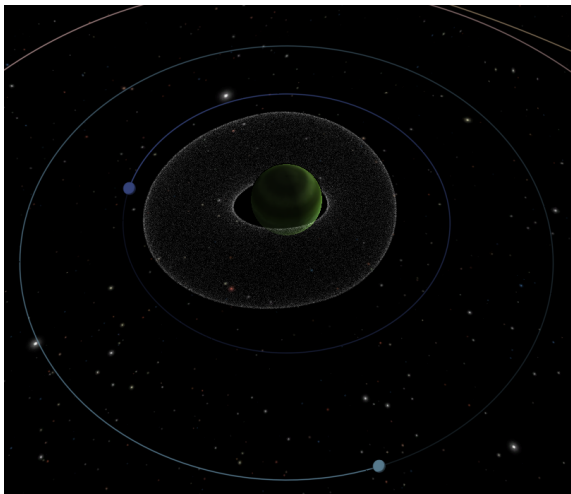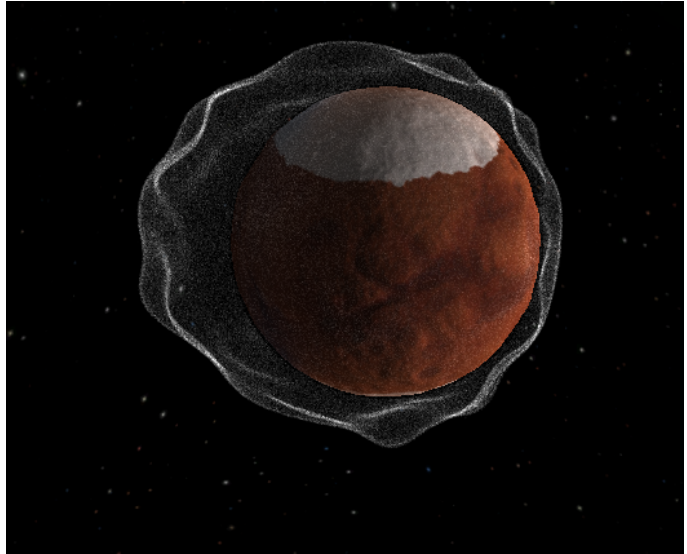
Radiation Models can be modified, see the Modding Kerbalism's Radiation Models section for more details.

5.3 Space weather

Coronal Mass Ejection events are generated in a stars corona, and move toward either a planetary system or a star-orbiting vessel. A warning will be issued as soon as the CME is ejected towards a body of interest. When the CME hits a planetary system or a star-orbiting vessel, all vessels outside of a magnetopause and in direct line of sight of a Star will receive extra radiation. Vessels inside of a magnetopause will suffer a communications *blackout*. The effects last for some time until the situation returns to normality.

Habitat

The internal habitat of a vessel is modeled as a set of individual parts flagged as habitats. Each part has an internal volume and an external surface, deduced automatically from their bounding box or specified by the part author.

From these basic properties, more complex ones are deduced and made available as modifiers to the rule framework.

6.1 Pseudo-resources

Some pseudo-resources are added to each habitat. Each one is used to simulate the individual properties of a vessels internal habitat volume and surface area. Their flow state is synchronized automatically from the habitat enabled/disabled state.

RESOURCE	CAPACITY	USE	DENSITY (per-unit)
Atmosphere	Set by volume	Pressure	1 m^3 of Nitrogen at STP
WasteAtmosphere	Set by volume	CO2 level	1 m^3 of CarbonDioxide at STP
MoistAtmosphere	Set by volume	Humidity level	1 m^3 of Saturated water vapor at STP
Shielding	Set by surface	Radiation shielding	1 m^2 of a 20mm Lead (Pb) layer

6.2 Atmospheric control

Atmospheric conditions inside a vessel are regulated by Life Support Systems (LSS) fitted into manned parts or by the External Life Support Unit (ECLSS). Each vessel has a number of configurable LSS slots that can be configured into an assortment of different LSS processes. The number of slots is upgradeable by purchasing the Slot Upgrade in the Electronics section of the Tech Tree. Also as you progress through the Tech Tree more options become available for the LSS slots.

The internal atmospheric pressure is regulated by the Pressure Controller, this unit is used to overcome the losses from leaks and for pressurizing inflatable habitats.

The CO2 level is regulated by the Scrubber, this unit is used to scrub from the atmosphere the CO2 that the Kerbal's exhale. The greenhouse can also be used to remove CO2 from the atmosphere.

The humidity level is regulated by the Humidity Controller, this unit removes the excess moisture in the atmosphere and recycles the moisture into clean water.

6.3 Radiation shielding

The user can choose the level of Shielding for each individual habitat part in the editor. The overall Shielding level on all enabled habitat parts is then used to reduce the environment radiation. It is possible to influence the level after launch by producing the Shielding resource.

6.4 Enable/disable habitats

The user can enable and disable habitat parts individually, both in flight and in the editor. This is used to configure and reconfigure the vessels internal volume, to influence its properties as the need arise.

6.5 Equalization and venting

When a habitat transitions from the enabled to the disabled state or vise versa, special care is used to avoid abrupt changes to the overall pressure of the whole vessels internal habitat. This is accomplished by two temporary states, in addition to *enabled* and *disabled*. These are *equalizing* that first matches the part pressure with the rest of the vessel and then switches to *enabled* and the other being *venting* that depressurizes the part completely by dumping the removed atmosphere either into the rest of the vessel, if there is room, or outside.

6.6 Inflatable habitats

If a habitat is inflatable, its inflate/deflate animation will be driven by the actual pressure of the part. Note that pressurizing a large habitat with a small pressure controller can take a long time. For example the mk1 pods pressure control will take approx 12 days (3 Earth days) to inflate the Gravity Ring. So remember to add enough pressure controllers for the job. And by the way, you also should bring enough nitrogen. You wouldn't want to blow up a bouncy castle with your mouth :/

To inflate an inflatable habitat on a celestial body surface with breathable atmosphere the "Air pump" should be used. Every crewable habitat got an air pump, including the non-inflatable ones. The more air pumps are enabled, the faster the inflation proceeds. Inflating the Gravity Ring with the air pump of the mk1 pod will take approx 3.5 hours. (Of course you will not bring the gravity ring to a planet's surface, or will you? This was just a comparision...)

6.7 Comforts

Comforts are provided by some vessel conditions, and parts implementing the Comfort module.

COMFORT	CONDITION	PART
firm-ground	vessel is landed	Gravity Ring
not-alone	more than 1 crew member in the vessel	
call-home	vessel can communicate with DSN via an antennas science rate	
exercise	Kerbal's can ride a bike or use a treadmill etc	Hitchhiker
panorama	Kerbal's can look out of a big window	Cupola

Reliability

7.1 MTBF

The Mean Time Between Failures is specified per-component and indicates how often it will experience a failure on average.

7.2 Failures

Failures comes in two variants: *malfunctions* and *critical failures*. The former can be repaired, the latter can't but are less frequent. Both types of failure disable the associated module: that is, the module will stop working.

Every time a component fails on an unmanned vessel, there is a chance that it will be fixed remotely by mission control engineers.

7.3 Quality

Manufacturing quality can be specified per-component in the VAB. A high quality will increase the MTBF, but also requires more money and mass. Thus there is a trade off between high reliability and cost/mass of components. Extra cost and mass are expressed in proportion to part cost and mass.

7.4 Inspection and Repair

All Kerbals can inspect components to reveal some vague information about the time left until the next failure.

Kerbals can also repair malfunctioned components, provided that they have the necessary specialization and experience level required.

7.5 Redundancy

The only way to plan around component failures is redundancy. To incentive this behavior, each component is assigned to a *redundancy group* and the planner will analyze redundancies on the vessel using this information. Optionally, when a component fails all others in the same *redundancy group* will be less likely to fail.

7.6 Supported modules

The system can trigger failures on arbitrary modules in a part, using the Reliability module. This module is added automatically for most stock components.

COMPONENT	MTBF std	MTBF high	REPAIR	REDUNDANCY	EXTRA COST	EXTRA MASS
Solar Panel (standalone)	4 years	16 years	Anyone	Power Generation	2.5	1.0
Solar Panel (embedded)	4 years	16 years	Anyone	Power Generation	0.25	0.1
Solar Panel (manned)	4 years	16 years	Anyone	Power Generation	0.125	0.05
Reaction Wheel (standalone)	4 years	16 years	Anyone	Attitude Control	2.0	1.0
Reaction Wheel (embedded)	4 years	16 years	Anyone	Attitude Control	0.25	0.15
Reaction Wheel (manned)	4 years	16 years	Anyone	Attitude Control	0.2	0.05
RCS (standalone)	8 years	32 years	Engineer	Attitude Control	2.0	1.0
RCS (embedded)	8 years	32 years	Engineer	Attitude Control	0.2	0.1
RCS (manned)	8 years	32 years	Engineer	Attitude Control	0.1	0.05
Light (standalone)	4 years	16 years	Anyone		5.0	1.0
Light (embedded)	4 years	16 years	Anyone		0.1	0.05
Light (manned)	4 years	16 years	Anyone		0.05	0.01
Parachute	8 years	32 years	Anyone	Landing	2.5	0.5
Engine	8 years	32 years	Engineer	Propulsion	1.0	0.1
Radiator* (standalone)	8 years	32 years	Engineer		1.0	0.25
Radiator* (embedded)	8 years	32 years	Engineer		0.2	0.1
Radiator* (manned)	8 years	32 years	Engineer		0.1	0.05
Resource Converter	8 years	32 years	Engineer		1.0	0.2
Resource Harvester	8 years	32 years	Engineer		1.0	0.2
Experiment (standalone)	8 years	32 years	Engineer		0.5	0.1
Experiment (embedded)	8 years	32 years	Engineer		0.05	0.01
Experiment (manned)	8 years	32 years	Engineer		0.025	0.005
Antenna (standalone)	8 years	32 years	Engineer	Comms	1.0	0.1
Antenna (embedded)	8 years	32 years	Engineer	Comms	0.5	0.01
Antenna (manned)	8 years	32 years	Engineer	Comms	0.05	0.001
Treadmill (in Hitchhiker)	4 years	16 years	Engineer		0.1	0.05

Table 1 – continued from previous page

COMPONENT	MTBF std	MTBF high	REPAIR	REDUNDANCY	EXTRA COST	EXTRA MASS
ECLSS (standalone)	8 years	32 years	Anyone	Life Support	2.5	0.1
LSS (manned)	8 years	32 years	Anyone	Life Support	0.625	0.025
Fuel Cell	8 years	32 years	Engineer	Power Generation	1.0	0.5
Chemical Plant	8 years	32 years	Engineer		1.0	0.2
Crustal Harvester	8 years	32 years	Engineer		1.0	0.2
Atmospheric Harvester	8 years	32 years	Engineer		1.0	0.5

This is valid for the "Radiator motor" and the "Radiator panel"

The above MTBF values are estimated average values and are mostly similar for standalone parts. For modules which are embedded into bigger parts, like for example the built in reaction wheels in manned pods, the MTBF values can vary much more. The MTBF depends on the mass of the part, or in the case of an embedded module a defined fraction of the part's mass. Also if a part has a crew capacity it is taken into account. To avoid weird numbers, the lowest possible MTBF is 4 years and the highest possible MTBF is 64 years. As a rule of thumb we can say that heavier parts have a shorter MTBF than lighter parts.

The EXTRA COST and EXTRA MASS values define a multiplier of the part's original values. So 0.1 means +10% and 2.5 means +250%.

Signal

8.1 Connections

To transmit data, vessels need a valid communications link with the Deep Space Network (DSN) on the surface of your home planet. For unmanned vessels, this communication link is also required for remote control. Celestial bodies occlude the signal, and other vessels can act as relays. Science data transmission speed is attenuated by signal strength and distance.

8.2 Antennas

Antennas comes in two types: *Internal* and *External*.

- Internal antennas as the name implies are fitted internally to probes and manned pods and can be used for short-range telemetry communications with the DSN and other vessels, they allow for operation of vessels via a control signal, these antennas also require constant power to operate.

- External antennas are the externally fitted antennas, these allow for longer distance communications and boost the telemetry and control signals, they are also used for transmitting science data or relaying data in the case of relay antennas. These antennas are also required for the Comfort bonus *call home*.

8.3 Range and Rate

When transmitting science, the data rate of that transmission depends on the type of antenna and on the distance. Data rates will be very low for very long distances, and reasonably high over short distances. Keep this in mind when planning missions into deep space.

Combining antennae will increase the communications range, but only to a very limited extend will it increase the data rate. When using multiple antennae at the same time, their combined data range will be the geometric mean of their individual rates.

What this means in practice is this: if you combine 2 or more antennae of the same type, you will increase the total communication range of the vessel and at the same time decrease the loss of data rate with distance. When you compare a vessel with one antenna with another vessel that has two antennae, they will both be able to transmit (almost) equally fast over short distances. But at long range, the vessel with 2 antennae will have an advantage.

This also means that you will want to combine antennae of same or similar capabilities only. If you combine one very fast antenna with one that is very weak, you will sacrifice almost all of the speed benefits you get from the fast antenna for the added range benefit of the weak antenna.

8.4 Transmission cost

Transmitting data consumes *ElectricCharge*. The cost is fixed and doesn't change with distance or signal strength.

Transmitters will use more EC during transmission, since they have to power their signal amplifier for sending. While passive (not sending), EC consumption will be a fraction of the transmission cost.

8.5 Extending antennas

Deployable Antennas need to be extended to work. This can be used by the player to configure what antennas are used for transmission, at any time. Extending and retracting antennas is possible even when the vessel is not controllable.

8.6 Control loss

Kerbalism will use the stock CommNet system if it is enabled allowing for 3 different models for control loss in vessels without a connection, the **none** model, causes complete loss of control on unlinked vessels. The **limited** model instead permits partial control of the vessel, the **full** model causes no loss of control, so that signal loss only affects the science data transmission. If CommNet is not enabled then connections will always be available unless you run out of Electric Charge or antennas break.

Science

The science system in Kerbalism is very different from how you know it in KSP. Science isn't generated at the click of one button any more, most experiments will take time to complete. Some will just take minutes, others will need years. The good news is that experiments will keep running in the background on unloaded vessels while you're busy with other missions, and they will generate a constant stream of scientific value while they're running.

While this doesn't sound like a big thing, it will change the way you build vessels, it will change the way you plan your missions and it will force you to make tough decisions. You will have to choose which experiments to run, because you won't be able to run them all at the same time. Sometimes you will have to delete valuable data to be able to collect new science. And you will have to come up with engineering solutions for problems you never had before. Or when was the last time you had to sustain a base with crew for months, submerged at least 100m deep on the ocean floor?

9.1 Transmission

Science results can be flagged for transmission home and they will be sent to DSN at the first opportunity. The transmission happens over time, even when the vessels are unloaded. Transmission times can range from mere seconds to years, depending on the size of the file transmitted and the transmission rate of the connection.

9.2 Samples

Some experiment results are not transmissible, these are considered samples and need to be recovered or analyzed in a lab. Samples are stored in slots that can be flagged for analysis, and then will be analyzed over time in a laboratory. As the analysis proceeds the sample will be slowly converted into transmissible data.

Samples have mass. Some Experiments like the Goo Container contain just a few grams of sampling material (mystery goo) that is used up while the goo observation takes place. Once the sampling material is used up, the experiment

cannot be rerun. Other experiments will collect samples from the atmosphere or the surface. A surface sample for example will collect 25kg of mass that should be accounted for when it needs to be hauled off the surface.

9.3 Supported experiments

The Science system supports stock experiments and other experiments that use the Science Dialog. Stock experiments get some tweaks, the data is transmissible completely or not at all, situation and biome combinations have been altered, the need to repeat the same experiment in a situation multiple times has been removed and the science credits returned have been rebalanced.

9.4 Tips

- **Slow Down** As science takes time to collect it can be advisable to set your parachutes to open at the maximum altitude to allow enough time to complete science collection as you float slowly down.

- **Early Power** Experiments need electric charge. Your early pods don't have a lot of power so you need to invest early in other sources. Fuel cells are a great way of resolving your power issues until solar panels. Make them a priority.

- **Explore** It's not all about getting to space! At least not in your first few launches. Shores, Water and Grasslands are all within easy reach!

- **All The Science** With power in short supply you may have to pick and choose your experiments carefully. Don't expect to be able to put all science experiments on a single vessel.

- **Repeat, Repeat** Science takes time to complete, but it doesn't need to finish to get some science. If the experiment doesn't finish you can run it again next mission to pick up the science left behind.

- **Gonna Need a Bigger HDD** Different experiments produce different amounts of data and hard drives can fill up quickly. Make sure you're transmitting that data home! Antennas can be automated to respond to a full hard drive.

- **Transmitting. . .** Experiments produce data at different rates so ensure you have enough bandwidth to transmit it as quickly as you gather it or your hard drive will fill up fast. Choose the number and type of antennas wisely.

- **Unlimited Power!** You don't have it. Make sure you switch on your experiments (including those in the capsule) in the VAB and check the Kerbalism planner to see how much electrical power your vessel uses.

- **How Much?** Generally the longer it takes and the more electrical charge it uses the more science reward you'll get. Those difficult experiments are worth doing.

- **Automate** Just like everything else in Kerbalism those science experiments can be set to turn on and off depending on situation. Check the auto tab of the Kerbalism panel in flight.

- **Polar Orbits** You can run biome dependent experiments on a probe in polar orbit. That way your experiment will fly over all the biomes and collect data for all of them.

Resources

10.1 Containers

The containers are configurable in the VAB. The inline supply containers can store.

- Food and Water *(Supplies)*
- Food
- Water
- Waste and WasteWater *(Sewage)*
- Waste
- WasteWater

The radial pressurized containers can store.

- Oxygen gas
- Nitrogen gas
- Hydrogen gas
- Ammonia gas
- CarbonDioxide gas
- Xenon gas

10.2 ISRU

Configurable ISRU's can execute a set of available chemical processes that can be configured in the VAB. Processes will stop running if they don't have the available resources or if there is no capacity for the output resources to go into.

Excess resources can be dumped overboard by using the Dump function in order to enable a process to keep running if output capacity is not available.

LiquidFuel, Oxidizer and MonoPropellant chemical components regarding the processes are.

- LiquidFuel = Methane *CH4*

- Oxidizer = HydrogenPeroxide *H2O2*

- MonoPropellant = Hydrazine *N2H4*

CHEMICAL PROCESS	INPUT RESOURCES	OUTPUT RE-SOURCES	TECH REQUIRED
Water electrolysis	EC, Water	Hydrogen, Oxygen	
Sabatier process	EC, CO2, Hydrogen	Water, LiquidFuel	
Haber process	EC, Nitrogen, Hydrogen	Ammonia	
Waste incinerator	Waste, Oxygen	CO2, Water, EC	Precision Engineering
Waste compressor	EC, Waste	Shielding	Precision Engineering
Anthraquinone process	Hydrogen, Oxygen	Oxidizer	Advanced Science
Hydrazine production	EC, Ammonia, Oxidizer	Water, O2, Monoprop	Advanced Science
Hydrazine production N2 inj	EC, Ammonia, Oxidizer, Nitrogen	O2, Monoprop	Experimental Science
Solid oxide electrolysis	EC, CO2	Oxygen, Shielding	Experimental Science
Molten regolith electrolysis	EC, Ore [Regolith]	Oxygen, CO2, Shielding	Experimental Science
Selective catalytic oxidation	EC, Ammonia, Oxygen	Nitrogen, Water	Experimental Science

10.3 Harvesters

Crustal, Oceanic and Atmospheric harvesters can be configured to extract one among a set of resources.

HARVESTER	RESOURCE
Crustal	Water
Crustal	Ore
Crustal	Nitrogen
Oceanic	Water
Oceanic	Nitrogen
Oceanic	Ammonia
Atmospheric	CarbonDioxide
Atmospheric	Oxygen
Atmospheric	Nitrogen
Atmospheric	Ammonia

10.4 Fuel cells

Fuel cells can be configured to use a number of resources to produce *ElectricCharge*. They operate under the same conditions as the chemical processes on the ISRU's

CELL TYPE	INPUT RESOURCE	EXTRA OUTPUT
H2+O2	Hydrogen and Oxygen	Water
Monoprop+O2	MonoPropellant and Oxygen	Water and Nitrogen

Kerbals

11.1 Biological needs

Kerbals need a constant supply of basic resources, Food, Water and Oxygen otherwise they will eventually perish.

RULE	CONSUME	PRODUCE	MASS PER-DAY (Kg)	UNITS PER-DAY
Eating	Food	Waste	0.07375	0.27
Drinking	Water	WasteWater	0.14	0.14
Breathing	Oxygen	WasteAtmosphere	0.05287	37.5

Individual consumption may vary.

11.2 Psychological needs

Kerbals will suffer mental breakdown after some time, it can be increased by providing.

- More habitat volume per-capita.
- A pressurized habitat.
- Basic comforts.

11.3 Environmental hazards

Kerbals will die if environmental conditions get out of hand, such as.

- *CO2 poisoning* from being exposed to high *CO2* levels (above 2%) in the internal atmosphere for too long. *CO2* levels are maintained by using Scrubbers and/or Greenhouses.

- Exposed to extreme levels of *humidity* (above 95%) in the internal atmosphere for too long. *Humidity* is maintained at a constant 60% by using the Humidity controllers.

- Exposed to temperatures outside of the *survivable range*. The internal temperature in a vessel is maintained constantly within the *survivable range* if there is enough *ElectricCharge* present. The climatization system uses *ElectricCharge* in proportion to the volume of the habitat to climatize and the difference between the *external* temperature and the *survivable range*.

- Exposed to extreme levels of *radiation*. Radiation belts have extremely high levels, and *solar storms* will dramatically increase the radiation for all vessels in a region temporarily. *Shielding* can be specified per-part in the VAB to reduce the environmental radiation reaching the internal habitat.

11.4 LSS

Each pod or the External LSS Unit (ECLSS) can be configured with Life Support System setups from among the following.

ECLSS SETUP	DESCRIPTION	TECH REQUIRED
Scrubber	Sequester CO2 from the internal atmosphere	
Pressure control	Consume Nitrogen to keep the internal pressure at an acceptable level	Engineering 101
Humidity controller	Extract potable water from the internal atmosphere and maintain humidity at 60%	Survivability
Water recycler	Extract potable water, ammonia and CO2 out of waste water	Space Exploration
Waste processor	Extract ammonia out of organic waste	Advanced Exploration
Monoprop fuel cell	Burn monoprop and O2, producing EC with by-products of water and nitrogen	Advanced Electrics

11.5 Greenhouse

The Greenhouse is based on design targets from the Prototype Lunar Greenhouse which is designed primarily to support the oxygen needs of one person rather than food needs. Thus one Greenhouse supports one kerbal with 100% of his/her O2 needs and half of its food needs per crop, a full crop can be harvested every 200 days, or you can use the emergency harvest function to harvest whatever amount of food has grown earlier.

The greenhouse growth is configured to.

- consume *Water* and *Ammonia*.

- consume *CO2* from waste atmosphere and/or pressurized tanks.

- consume *ElectricCharge* for the artificial lighting lamps, when their use is required.

- require an internal pressure of at least 10kPA.

- require radiation levels not in excess of 0.03 rad/h.

- produce *Oxygen* and *Food*.

CHAPTER 12

Graphics User Interface (GUI)

12.1 Planner

A planner is provided in the editors to help design around the mechanics introduced in Kerbalism.

The calculations are done relative to a target body and situation, whether a star is visible or occluded and considering the number of crew currently assigned to the vessel (keep ALT pressed to consider the whole vessel crew capacity instead).

Target body, situation and whether in shadow or not can be changed by clicking on the relative icon in the planner title bar.

Hover the mouse over an entry to show related tooltips with explanations and additional information.

12.2 Monitor

The *vessel monitor* is available in the space center, the tracking station and in flight. It shows the state of your vessels and the Kerbals inside them.

The monitor will show a list of vessels, and a summary of its state represented with indicator icons.

From left to right:

- vessel name
- body name
- problem icons
- battery level
- supplies level
- reliability status
- signal status

Clicking on a vessel in the list will select it. The monitor will then show a panel about the vessel. Right-clicking anywhere in the monitor will de-select the vessel and return back to the vessel list view.

In the vessel details view, select what panel to display by left-clicking on the bottom menu entries. Middle-click on the menu entries instead to popout the panel as a window.

Groups and Filtering

The last bottom menu button in *vessel details* allows you to see and change the *group* a vessel is assigned to. When at least one of the vessels is assigned to a group, the *filter bar* will appear at the bottom of the monitor. Click on it and type a group to hide all vessels except the members of that group.

Problem icons

	Vessel is not in direct sunlight
	Solar storm inbound
	Solar storm in progress
	Exposed to intense radiation
	Exposed to extreme radiation
	Kerbal doesn't feel well
	Kerbal is about to die
	Kerbal is stressed
	Kerbal is breaking down
	CO2 level is above warning level
	CO2 level is above danger level
	Greenhouse is not growing

Battery icon

	ElectricCharge above warning threshold
	ElectricCharge below warning threshold
	ElectricCharge is depleted

Supply icon

	Supply resources above warning threshold
	Supply resources below warning threshold
	Supply resources are depleted

Reliability icon

	No failures
	One or more *malfunctions*
	One or more *critical failures*

Signal icon

	Transmission rate above 5Kbps
	Transmission rate below 5Kbps
	No signal / Blackout

12.3 Telemetry

This panel shows readings from the vessel, including crew vitals, resource supply levels, habitat and environmental information.

12.4 File manager

This panel allows you to visualize files stored in the vessels hard drive, flag them for transmission or analysis, or delete them. Hovering over a file will display a tooltip with additional information.
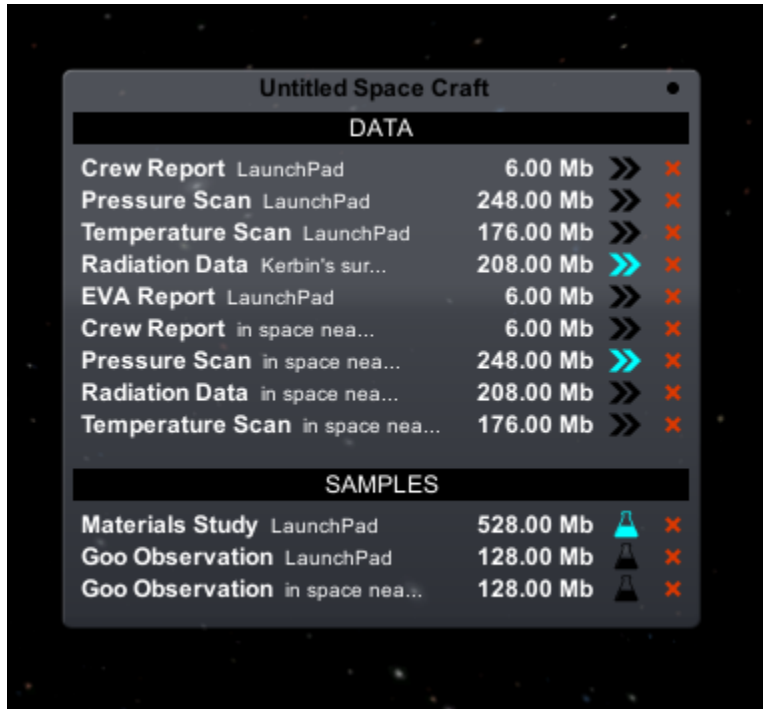
12.5 Device manager

This panel will show the control status of all components in a vessel, it is also used as the editor for the automation scripts.
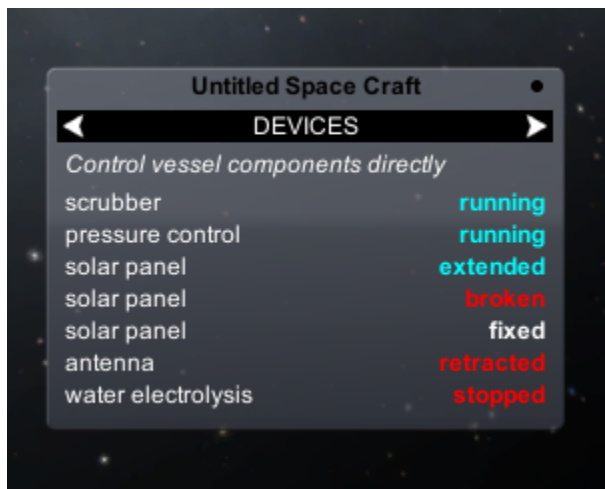
12.6 Configuration manager

This panel allows you to set what messages will be shown for the vessel.
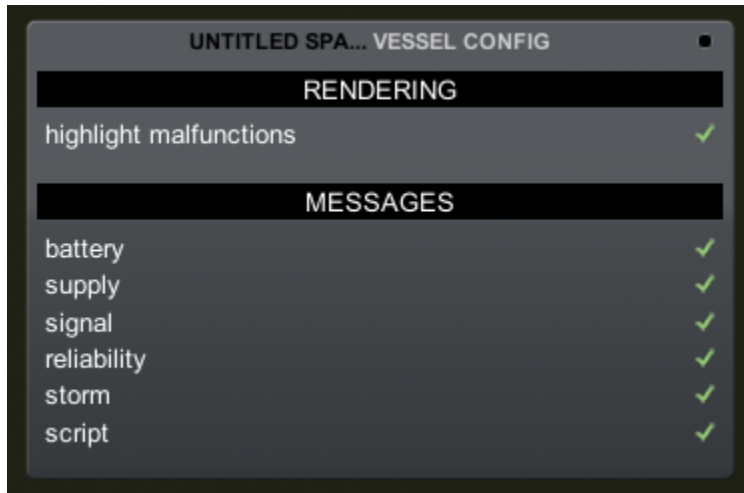
12.7 Body info

When in the tracking station or map view, press **B** to open the *body info window*. Here some information is shown about the body atmosphere and radiation environment, also the rendering of the radiation fields can be controlled here.
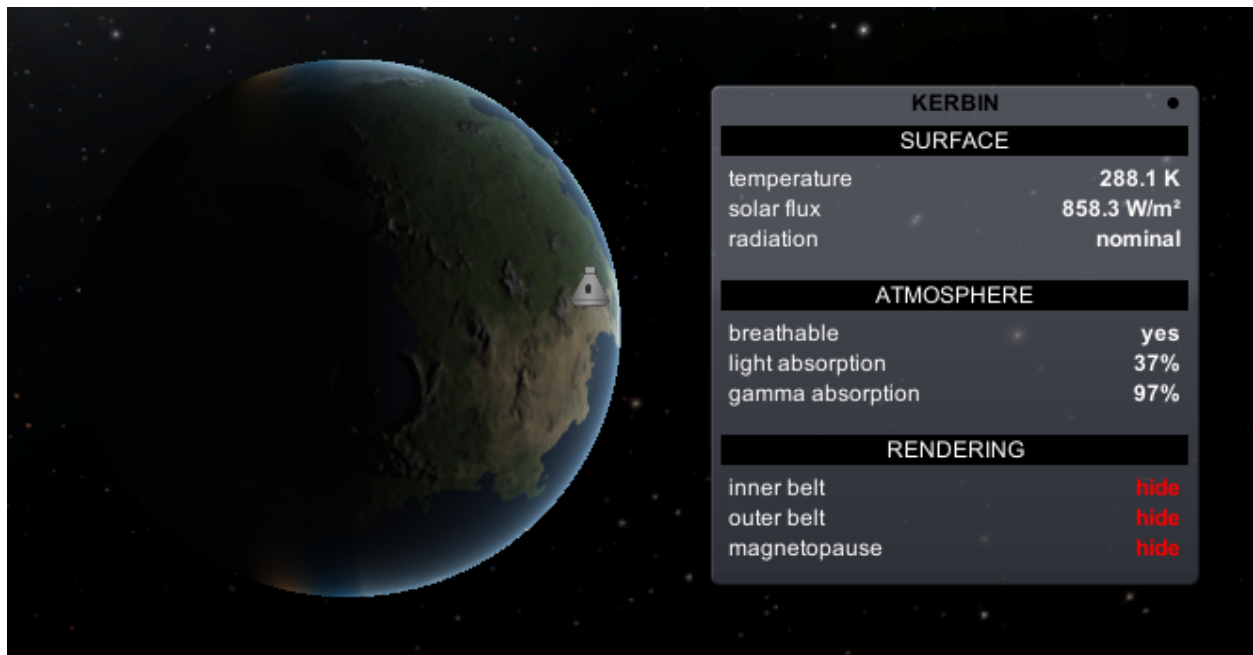
Kerbalism's Parts

13.1 Containers
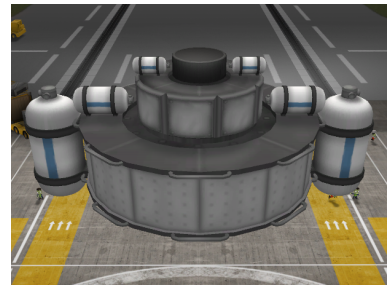
A set of supply resource containers.

13.2 Greenhouse
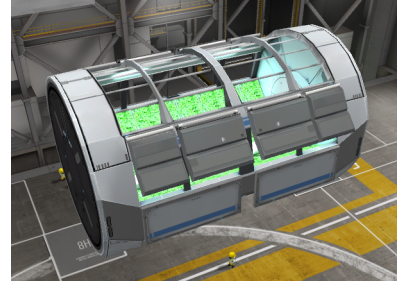
A greenhouse.

13.3 Gravity Ring
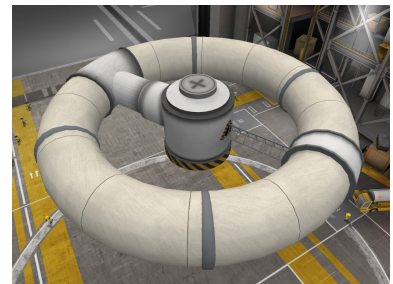
A rotating ring inflatable habitat.

13.4 Active Shield
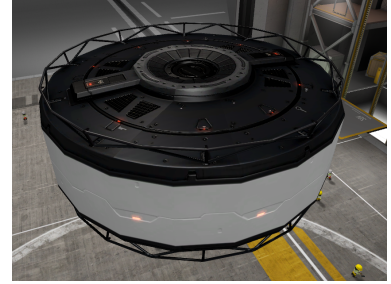
A late-game active radiation shield.

13.5 Geiger Counter

An environment radiation sensor.

13.6 ECLSS Unit

An External Life Support System (ECLSS).

13.7 Chemical Plant

A very small ISRU plant.

13.8 Short Antenna
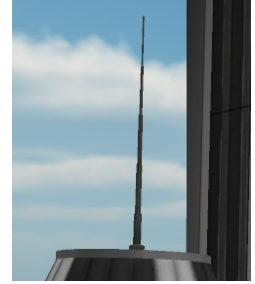
A small antenna.

Automation

Components in a vessel can be turned on and off automatically by environmental conditions. The set of component changes is stored in scripts, and a simple editor UI is provided. When a specified change in conditions is detected, the relative script is executed on a vessel. This works transparently for loaded and unloaded vessels.

14.1 Scripts

A script represents a list of state changes for all vessel components. Each component can be set in one of three states: *don't care*, *on* or *off*.

14.2 Editor

There is a simple graphical editor for the scripts conditions. It can be opened by clicking on the **auto** icon in the Monitor UI. Click on the arrows in the panel title to select one of the scripts. Then click on the components to change their state. Components states can be manually controlled by using the *direct control* page.

14.3 Direct control

The Script editor UI can also serve as a simple way to change the state of single components without clicking on the part first. This works even for unloaded vessels. The state of each component is also reported. This is not that informative usually, but can act as a sort of summary of the overall vessel status.

14.4 Conditions

Scripts are triggered by the following conditions.

CONDITION	TRIGGER
landed	vessel state switched to landed
atmo	entering the atmosphere
space	reaching space
sunlight	star returns to visible
shadow	star gets occluded
power_high	EC level goes above 80%
power_low	EC level goes below 20%
rad_low	radiation goes below 0.02 rad/h
rad_high	radiation goes above 0.05 rad/h
linked	signal is regained
unlinked	signal is lost
eva_out	going out on Eva
eva_in	coming back from Eva
drive_full	drives are at 90% capacity
drive_empty	drives are below 10% capacity
action[0-5]	press [0-5], on the active vessel

14.5 Supported modules

Only these modules are supported by the automation system.

MODULE	ACTION
Antenna	Extend/Retract
Experiment	Enable/Disable
Emitter	Enable/Disable
Gravity Ring	Enable/Disable
Greenhouse	Enable/Disable
Harvester	Start/Stop
Laboratory	Start/Stop
Process Controller	Start/Stop
ModuleDeployableSolarPanel	Extend/Retract
ModuleGenerator	Start/Stop
ModuleLight (and some derivatives)	Turn on/off
ModuleResourceConverter (and some derivatives)	Start/Stop
ModuleResourceHarvester	Start/Stop
SCANsat	Start/Stop scanning

Background Simulation

15.1 Resources

Modules that consume and produce resources are simulated for unloaded vessels.

MODULE
Greenhouse
GravityRing
Emitter
Experiment
Harvester
Laboratory
ModuleCommand
ModuleDeployableSolarPanel
ModuleGenerator
ModuleResourceConverter (and some variants)
ModuleResourceHarvester
ModuleAsteroidDrill
ModuleScienceConverter
ModuleLight (and some variants)
SCANsat
ModuleSCANresourceScanner
ModuleCurvedSolarPanel
FissionGenerator
ModuleRadioisotopeGenerator

15.2 Solar panels

Solar panel output is simulated. Fixed panel orientation is taken into account, and tracking panels are simulated around the pivot. A portion of the flux is blocked by the atmosphere depending on density and path length.

15.3 Algorithm details

Dependency information is preserved in *recipes*, all the recipes are executed using an iterative algorithm that is order-less, works at arbitrary time steps and is not limited by storage capacity.

Settings

The settings for Kerbalism are stored in the GameData/Kerbalism folder and are in the Settings.cfg file.

These settings can be used to select a profile, enable/disable features, fine-tune the environment etc, and customize the UI.

SETTING	DESCRIPTION	DEFAULT
Profile	name of profile to use, if any	default
Reliability	enable/disable component malfunctions and critical failures	true
Deploy	enable/disable the addition of EC cost to ExtendRetract modules	true
Science	enable/disable science data storage, transmission and analysis	true
SpaceWeather	enable/disable coronal mass ejections	true
Automation	enable/disable controlling vessel components using scripts	true
SurvivalTemperature	ideal living temperature in K	295
SurvivalRange	sweet spot around survival temperature	5.0
IdealLivingSpace	ideal living space per-capita in m^3	40.0
ComfortFirmGround	firm-ground comfort factor	0.4
ComfortExercise	exercise comfort factor	0.2
ComfortNotAlone	not-alone comfort factor	0.1
ComfortCallHome	call-home comfort factor	0.1
ComfortPanorama	panorama comfort factor	0.1
PressureFactor	*pressurized* modifier value for vessels below the threshold	10.0
PressureThreshold	level of atmosphere resource that determine *pressurized* modifier status	0.9
PoisoningFactor	*poisoning* modifier value for vessels below threshold	0.0
PoisoningThreshold	level of waste atmosphere resource that determine *poisoning* modifier status	0.02
HumidityFactor	*humidity* modifier value for vessels below the threshold	1.0
HumidityThreshold	level of atmosphere resource that determine *humidity* modifier status	0.95
ShieldingEfficiency	proportion of radiation blocked by shielding (at max amount)	0.9
StormRadiation	radiation during a solar storm, in rad/h	5.0
ExternRadiation	radiation outside the heliopause, in rad/h	0.04
StormMinTime	minimum interval between storms over a system, in seconds	2160000

Table 1 – continued from previous page

SETTING	DESCRIPTION	DEFAULT
StormMaxTime	maximum interval between storms over a system, in seconds	8640000
StormDuration	how long a storm last once it hit, in seconds	21600
StormEjectionSpeed	CME speed in m/s	1000000
ScienceDialog	keep the stock science results dialog around	true
QualityScale	scale applied to MTBF for high-quality components	4.0
CriticalChance	proportion of malfunctions that lead to critical failures	0.25
SafeModeChance	proportion of malfunctions fixed remotely for unmanned vessels	0.5
IncentiveRedundancy	if true, each malfunction will increase the MTBF of components in the same redundancy group	false
EnforceCoherency	detect and avoid issues at high timewarp in external modules	true
TrackingPivot	simulate tracking solar panel around the pivot	true
HeadLampsCost	EC/s cost if Eva headlamps are on	0.002
DeathReputation	reputation to remove in case of death	100
BreakdownReputation	reputation to remove in case of breakdown	10
StockMessages	use the stock messages instead of our own message box	false
MessageLength	duration of messages on screen in seconds	4
LowQualityRendering	use less particles to render the magnetic fields	false
UIScale	scale UI elements by this factor, relative to KSP scaling settings	1.0
UIPanelWidthScale	scale UI Panel Width by this factor, relative to KSP scaling settings	1.0

Modding Kerbalism

This section is for those who wish to modify Kerbalism, it contains information regarding changing its default behavior.

17.1 Modding Kerbalism's Profiles

17.1.1 Profiles

A profile is a named set of rules, supplies and processes. Multiple profiles can coexist in the same install and can be defined anywhere inside the GameData folder. At any time only the one specified in the Profile parameter in Settings is used.

17.1.2 Supply

Supply details about a resource are described by a Supply definition. These resources are shown in the Planner and Monitor UI. They are used to determine.

- amount added to all manned pods, in proportion of crew capacity

- capacity added to EVA Kerbals, that will then take the resources when going out on EVA

- amount gifted to rescue contract victims

- warning and danger levels for the resource, and messages to show

PROPERTY	DESCRIPTION	DEFAULT
resource	name of resource	
on_pod	how much resource to add to manned parts, per-kerbal	0.0
on_eva	how much resource to take on Eva, if any	0.0
on_rescue	how much resource to gift to rescue missions	0.0
empty	set initial amount to zero	false
low_threshold	at what level the resource is considered low	0.15
low_message	messages shown when level goes below low threshold, you can use *message macros* here	
empty_message	messages shown when level reach zero, you can use *message macros* here	
re-fill_message	messages shown when level goes back above low threshold, you can use *message macros* here	

17.1.3 Rule

A rule describes a mechanic that increments an accumulator per-kerbal based on the environment and the availability of resources. When the accumulator reaches the fatal threshold the rule can be configured to *kill* the kerbal, or to trigger an *unplanned event* instead.

PROPERTY	DESCRIPTION	DEFAULT
name	unique name for the rule	
input	resource consumed, if any	
output	resource produced, if any	
interval	if 0 the rule is executed continuously, else it is executed every 'interval' seconds	0.0
rate	amount of input resource to consume at each execution	0.0
ratio	ratio of output resource in relation to input consumed, deduced automatically from input and output density ratio if not specified	0.0
degeneration	amount to add to the property at each execution, when we must degenerate	0.0
variance	variance for degeneration, unique per-kerbal and in range [1.0 +/- variance]	0.0
individuality	variance for rate, unique per-kerbal and in range [1.0 +/- variance]	0.0
modifiers	comma-separated list of modifiers influencing the rule	
breakdown	trigger a unplanned event in the vessel, instead of killing the kerbal	false
lifetime	value will not be reset when recovering the kerbal on kerbin. used for things that cannot be cured, like radiation.	false
warning_threshold	threshold of degeneration used to show warning messages and yellow status color	0.33
danger_threshold	threshold of degeneration used to show danger messages and red status color	0.66
fatal_threshold	threshold of degeneration used to show fatal messages and kill/breakdown the kerbal	1.0
warning_message	messages shown when degeneration goes above warning threshold, you can use *message macros* here	
danger_message	messages shown when degeneration goes above danger threshold, you can use *message macros* here	
fatal_message	messages shown when degeneration goes above fatal threshold, you can use *message macros* here	
relax_message	messages shown when degeneration goes back below warning threshold, you can use *message macros* here	

17.1.4 Process

Processes are 'vessel-wide' resource producers/consumers, with rates that can be influenced by the environment, habitat conditions, and level of resources.

PROP-ERTY	DESCRIPTION	DE-FAULT
name	unique name for the process	
modifiers	comma-separated list of modifiers	
input	zero or more input resource names and rates, in the format **input = resource@rate**	
output	zero or more output resource names and rates, in the format **output = resource@rate**	
dump	comma-separated list of resources to dump excess output of overboard	false

17.1.5 Modifiers

Rule and Process rates can be influenced by the environment, habitat conditions and resource levels. This is accomplished by multiplying the rates with a set of modifiers, that can be specified as a comma-separated list.

MODIFIER	RATES ARE MULTIPLIED BY
breathable	zero inside an atmosphere containing oxygen and above 25kPA of atmospheric pressure, 1.0 otherwise
temperature	absolute difference between the external temperature, and the survival range that is specifiable in settings
radiation	incoming radiation at the vessel position, in rad/s
shielding	*shielding factor*, computed from the level of Shielding resource
volume	volume in m^3 of all enabled habitats in the vessel
surface	surface in m^2 of all enabled habitats in the vessel
living_space	the *volume* per-capita, normalized against an ideal living space
comfort	the *comfort factor*, computed from the Comfort providers in the vessel
pressure	the *pressure factor*, or 1.0 if *Atmosphere* level is above threshold (both specified in settings)
poisoning	the *poisoning_factor*, or 1.0 if *WasteAtmosphere* level is above threshold (both specified in settings)
humidity	the *humidity_factor*, or 1.0 if *MoistAtmosphere* level is above threshold (both specified in settings)
per_capita	the inverse of number of crew members, the effect on rates is a division by number of crew members
zerog	1 if the ship is above the atmosphere of a body, 0 if in atmospheric flight or landed
landed	1 if the ship is on the ground or in water, 0 otherwise
resource name	the level of resource specified

17.1.6 Message macros

The messages specified in a *Rule* or a *Supply* can contain macros.

MACRO	REPLACED BY
$NEWLINE	The new line character
$VESSEL	The vessel name
$KERBAL	The Kerbal name. Empty for the resource level messages
$ON_VESSEL	Replaced by 'On $VESSEL, ' if the vessel is not the active one, or an empty string otherwise
$HIS_HER	Replaced by 'his' or 'her' based on Kerbal gender

17.1.7 Unplanned events

If breakdown is set to true in a Rule then one of these events will trigger at random when it reaches its fatal threshold.

TYPE	DESCRIPTION	EFFECT
Mumbling	A Kerbal has been in space for too long	*none*
Fat Finger	The wrong button was pressed on the control panel	Loss of science data
Wrong Valve	The wrong valve was opened for lack of concentration	Loss of supply resources
Rage	A component was the victim of your Kerbal's rage	A component fail

17.2 Modding Kerbalism's Radiation Models
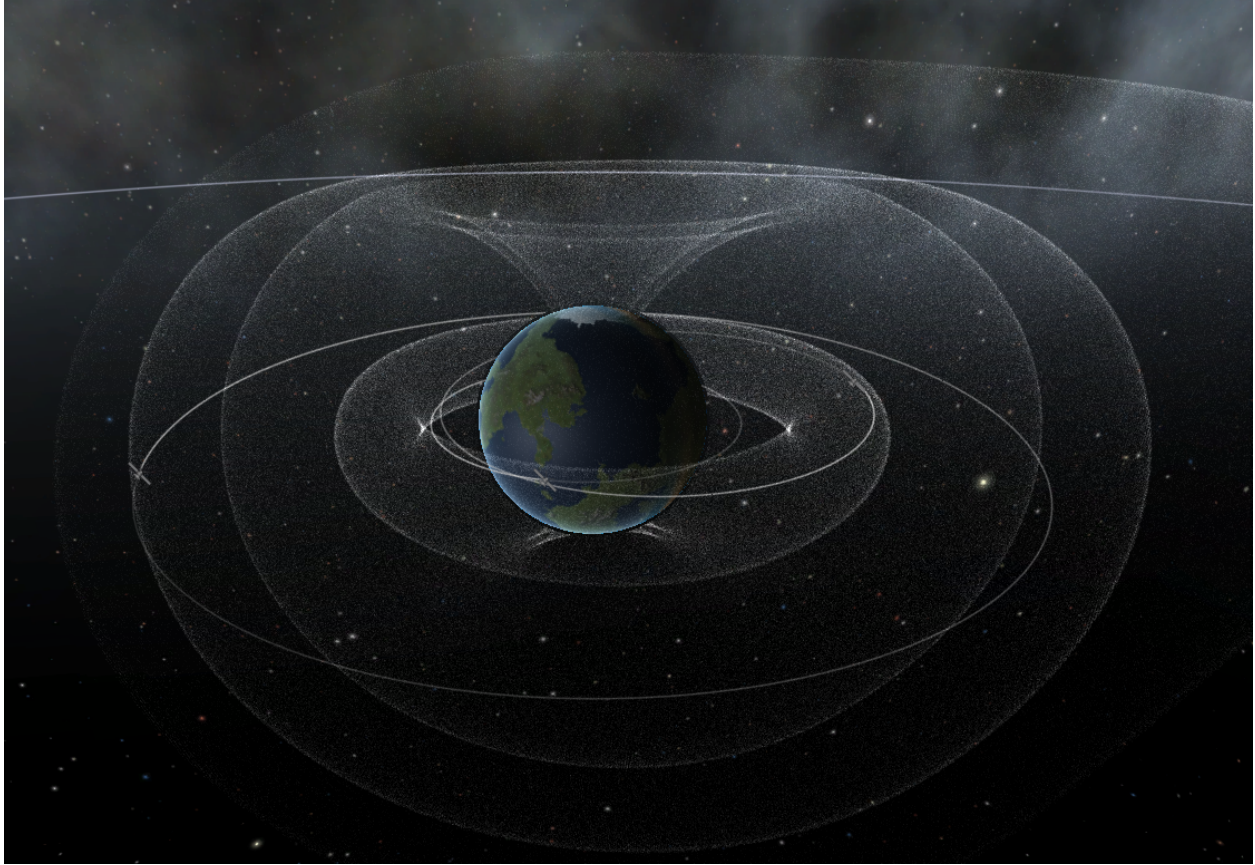
17.2.1 Radiation Model

A *RadiationModel* defines the signed distance function parameters that determine the shapes of the inner belt, outer belt and magnetopause. The model can be assigned to one or more celestial bodies using *RadiationBody*.

The inner belt is a torus. The *a* radius defines the distance from the section center to the origin. The *b* radius defines the radius of the section.

The outer belt is the boolean subtraction of a torus with another torus. The second torus is equal to the first, except for the fact that the *b* radius is reduced by a border factor. This in turn is not constant everywhere but fades from the *outer_border_start* at the origin to the *outer_border_end* at the domain boundary.

The magnetopause is simply a sphere, possibly deformed along the *body->star* vector to define a magnetotail.

All values are in body radii.

PROPERTY	DESCRIPTION	DEFAULT
name	Unique name for the radiation model	
has_inner	True if the model has an inner radiation belt	false
inner_dist	Inner belt torus *a* radius	
inner_radius	Inner belt torus *b* radius	
inner_compression	Deform space along the *body->star* vector, in direction of the star	1.0
inner_extension	Deform space along the *body->star* vector, in opposite direction of the star	1.0
inner_quality	Quality of border for rendering purposes, only influence pre-computation time	30.0
inner_deform	Deform the surface using a sum of sine waves	0.0
has_outer	True if the model has an outer radiation belt	false
outer_dist	Outer belt torus *a* radius	
outer_radius	Outer belt torus *b* radius	
outer_compression	Deform space along the *body->star* vector, in direction of the star	1.0
outer_extension	Deform space along the *body->star* vector, in opposite direction of the star	1.0
outer_border_start	Outer belt border extension at the origin	0.1
outer_border_end	Outer belt border extension at the domain boundary	1.0
outer_deform	Deform the surface using a sum of sine waves	0.0
outer_quality	Quality of border for rendering purposes, only influence pre-computation time	40.0
has_pause	True if the model has a magnetopause	false
pause_radius	Magnetopause radius	
pause_compression	Deform space along the *body->star* vector, in direction of the star	1.0
pause_extension	Deform space along the *body->star* vector, in opposite direction of the star	1.0
pause_height_scale	Deform space along the magnetic axis vector	1.0
pause_deform	Deform the surface using a sum of sine waves	0.0
pause_quality	Quality of border for rendering purposes, only influence pre-computation time	20.0

17.2.2 Radiation Body

The *RadiationBody* associates a *RadiationModel* to a celestial body and defines the radiation contribution inside the zones delimited by the signed distance function. Radiation values in a zone can be negative, that is usually the case for a magnetopause's contribution.

PROPERTY	DESCRIPTION	DEFAULT
name	Name of the celestial body	
model	Name of the *RadiationModel* associated	
radiation_inner	Radiation contribution inside the inner belt, in rad/h	
radiation_outer	Radiation contribution inside the outer belt, in rad/h	
radiation_pause	Radiation contribution inside the magnetopause, in rad/h	
reference	Index of the body used to determine radiation fields orientation	0

Radiation is *computed* at a point by walking the *body chain* and summing all contributions for that point from all the fields overlapping with that point. When the top of the chain is reached the radiation value parameter *ExternRadiation* from the Settings file is added.

17.3 Kerbalism's Part Modules

17.3.1 Comfort

The part provides comforts for the crew.

PROPERTY	DESCRIPTION
bonus	the comfort bonus provided
desc	short description shown in part tooltip

see Comforts for a list of allowed bonus's

17.3.2 Configure

The part allows for different *setups* of modules and resources that can be selected by the user in-game.

PROPERTY	DESCRIPTION	DEFAULT
title	short string to show on part UI	
slots	number of setups that can be selected concurrently	1
reconfigure	string in the format *trait@level*, specifying that the part can be reconfigured in flight by the crew	
symmetric	if true enforces same configuration on all parts in the symmetry group (useful for tanks)	false
SETUP	one or more sub-nodes that describe a setup	

A **SETUP** sub-node has the following properties.

PROPERTY	DESCRIPTION
name	short string describing the setup
desc	longer description of the setup
tech	id of technology required to unlock the setup
cost	extra cost, in space-bucks
mass	extra mass, in tons. (1 = 1000Kg)
MODULE	zero or more sub-nodes associating a module with the setup
RESOURCE	zero or more sub-nodes defining a resource included in the setup

A **MODULE** sub-node, inside a *SETUP* node, associates a specific module (that is already defined in the part) to that particular setup. The module will then be disabled (effectively acting like it wasn't there) unless the user selects the setup. A module can be associated to only a setup. Not all modules in the part need to be associated to a setup, and those that aren't will behave as usual.

PROPERTY	DESCRIPTION
type	module name
id_field/id_value	the name of a field in the module definition and its value respectively, used to identify a module in particular if multiple ones of the same type exist in the part
id_index	the zero-based index, selecting a specific module of *type* among all the ones present in the part

A **RESOURCE** sub-node, inside a *SETUP* node, adds a specific resource amount and/or capacity to the setup. The resource definition is the same as the stock one you are familiar with. The resource doesn't need to be defined in the part directly but only in the setup. When the setup is selected, the resource will be added to the part. If the part already contain the same resource, the amount and/or capacity will simply increase when the setup is selected.

17.3.3 Emitter

The part emits radiation. Use a negative radiation value for absorption.

PROPERTY	DESCRIPTION	DEFAULT
radiation	radiation in rad/s, can be negative	
ec_rate	EC consumption rate per-second (optional)	
toggle	true if the effect can be toggled on/off	false
active	name of animation to play when enabling/disabling	

17.3.4 Experiment

Hooks experiments into the Kerbalism science system.

PROP- ERTY	DESCRIPTION	DE- FAULT
experi- ment_id	The ID of the experiment (which must be defined elsewhere)	
experi- ment_desc	A nice description of the experiment.	
data_rate	sampling rate in Mb/s	0.01
ec_rate	EC consumption rate per second while recording	0.01
al- low_shrouded	Allow the experiment to run while it's part is shrouded	true
sam- ple_mass	If not 0, this is a sample and cannot be transmitted	0.0
sam- ple_reservoir	Amount of sampling material stored on the part	= sam- ple_mass
sam- ple_collecting	If set to true, the experiment will produce mass.	false
sci- ence_cap	A factor for the max. attainable science value	1.0
requires	Additional requirements that must be met for recording. See below.	
re- sources	Resources consumed while the experiment is running. Will stop if one of the resources is depleted. Rate is per sec. Malformed definitions or unknown resources will be ignored. Example: resources = Water@0.01,Food@0.02	
crew_operate	Requirements for crew on vessel for recording. If this is not set, the experiment can run on unmanned probes.	
crew_reset	Requirements for crew to reset the experiment. If this is set, the experiment will only record data from within the situation where recording was started, until it is reset (either by a kerbal that has to match the requirement, or by a lab.	
crew_prepare	If set, a kerbal has to prepare the experiment before it can record data. Once prepared, the experiment will only record data while it remains in the situation it was prepared for. The kerbal doing the preparation has to match the requiremens	
hide_when_unavailable	Don't show the UI when the experiment is unavailable.	
anim_deploy	Name of the part animation to trigger when recording starts.	

Crew specifications (used in crew_operate, crew_reset or crew_prepare as well as in some other Kerbalism mods) have to be given according to *true|trait|[trait]@level*

Examples:

- "true": any kerbal will do.

- "Scientist": you need a Scientist, doesn't matter how experienced. Other traits are "Pilot" and "Engineer". We're not assuming that you'll want to use "Tourist". . .

- If the value is "@3" any Kerbal with 3 or more stars will do

- If the value is "Scientist@2" you need a Scientist with 2 or more stars.

- Empty values usually turn the feature off.

Requirements of the experiments work as additional filters, and work ON TOP OF what the underlying experiment uses. If you create a Kerbalism Experiment for 'seismicScan'it won't work in orbit. The underlying experiment restrictions are checked first, then the additional requirements are checked.

The restrictions are case sensitive and comma-separated, and must ALL be met for recording. *restriction = Shadow,Space,Body:Kerbin* will only record data while in space near Kerbin AND in shadow. *restriction = Alti-*

---

tudeMin:250000,Surface will never record anything for plainly obvious reasons.

Here is a list of currently supported requirements:

- OrbitMinInclination, OrbitMaxInclination: min./max. inclination of the orbit (f.i. *OrbitMinInclination:30*)

- OrbitMinEccentricity, OrbitMaxEccentricity: min./max. eccentricity of the orbit (f.i. *OrbitMaxEccentricity:0.1*)

- OrbitMinArgOfPeriapsis, OrbitMaxArgOfPeriapsis: min./max. argument of periapsis

- TemperatureMin, TemperatureMax: min./max. Temperature in Kelvin

- AltitudeMin, AltitudeMax: min./max. Altitude in Meters

- RadiationMin, RadiationMax: min./max. radiation in rad/h

- Microgravity: not on a surface, not in atmosphere.

- Body: body on which the experiment can run. More than one name can be given (separate with semicolon), to exclude a body prefix it with ! (f.i. *Body:Eve;Duna;!Kerbin*)

- Shadow: vessel must not be exposed to sunlight

- Sunlight: vessel must be in the presence of a supreme being that radiates warmth and light upon it

- Surface: vessel must be on a surface

- Atmosphere: vessel must be within an atmosphere

- AtmosphereBody: vessel must be within the SOI of a body with atmosphere

- AtmosphereAltMin / AtmosphereAltMax: Altitude of vessel as a multiplier of atmosphere thickness. On Kerbin, AtmosphereAltMin:1 equals 70km.

- Vacuum: the opposite of Atmosphere

- BodyWithAtmosphere, BodyWithoutAtmosphere: does what it says on the tin.

- Ocean: vessel must be submerged

- PlanetarySpace: in planetary space, i.e. not around the sun

- AbsoluteZero: temperature < 30 K

- InnerBelt: vessel must be in a inner Van Allen Belt

- OuterBelt: vessel must be in a outer Van Allen Belt

- MagneticBelt: vessel must be in any Van Allen Belt

- Magnetosphere: vessel must be inside a magnetosphere

- Thermosphere: vessel must be inside a thermosphere

- Exosphere: vessel must be inside an exosphere

- InterPlanetary: vessel must be in interplanetary space, i.e. in the SOI of the Sun

- InterStellar: vessel must be outside the sun magnetopause

- Greenhouse: there must be one greenhouse on the vessel.

- CrewMin, CrewMax: min./max. amount of crew on vessel

- CrewCapacityMin, CrewCapacityMax: min./max. crew capacity

- VolumePerCrewMin, VolumePerCrewMax: min./max. habitat volume per crew member

- Facility building levels: MissionControlLevelMin, MissionControlLevelMax, AdministrationLevelMin, AdministrationLevelMax, TrackingStationLevelMin, TrackingStationLevelMax, AstronautComplexLevelMin, AstronautComplexLevelMax

- MaxAsteroidDistance: max. distance to the nearest asteroid. For unloaded vessels this only works if the asteroid is set as the target.

- Part: name (or any of multiple names, separated by comma) of a part that has to be anywhere on the vessel

- Module: name of a module that is required anywhere on the vessel

- SunAngleMin, SunAngleMax: min./max. angle of sunlight on the surface of the body

The following might or might not work for unloaded vessels, please udpate this list when you find out:

- SurfaceSpeedMin,SurfaceSpeedMax: Speed above surface

- VerticalSpeedMin,VerticalSpeedMax: Vertical speed

- SpeedMin,SpeedMax: speed

- DynamicPressureMin,DynamicPressureMax: current dynamic pressure

- StaticPressureMin,StaticPressureMax: current static pressure

- AtmDensityMin,AtmDensityMax: current atmospheric density

- AltAboveGroundMin,AltAboveGroundMax: Altitude above ground. Note that this value can change rapidly as KSP loads/unloads the terrain of a body

17.3.5 GravityRing

Used by the *Gravity Ring* part.

PROPERTY	DESCRIPTION
ec_rate	EC consumed per-second when deployed
deploy	a deploy animation can be specified
rotate	a rotate loop animation can be specified

17.3.6 Greenhouse

The part simulates a greenhouse. The crop grows over time, then it is harvested as a resource. Growth has lighting requirements that can be satisfied from the environment and/or the integrated lamps. Additional requirements can be specified, such as input resources, minimal pressure and maximal radiation. By-product resources can be produced.

PROPERTY	DESCRIPTION
crop_resource	name of resource produced by harvests
crop_size	amount of resource produced by harvests
crop_rate	growth per-second when all conditions apply
ec_rate	EC/s consumed by the lamp at max capacity, set to 0 to disable the lamp
light_tolerance	minimum lighting flux required for growth, in W/m^2
pressure_tolerance	minimum pressure required for growth, in sea level atmospheres (optional)
radiation_tolerance	maximum radiation allowed for growth in rad/s, considered after shielding is applied (optional)
lamps	object with emissive texture used to represent intensity graphically
shutters	animation to manipulate shutters
plants	animation to represent plant growth graphically

Resource requirements and by-products (other than EC for the lamps) are specified using the stock *resHandler* specification

```
INPUT_RESOURCE
{
  name = Water
  rate = 0.00023148
}

OUTPUT_RESOURCE
{
  name = Oxygen
  rate = 0.00463
}
```

17.3.7 Habitat

The part has an internal habitat.

PROPERTY	DESCRIPTION	DEFAULT
volume	habitable volume in m^3, deduced from bounding box if not specified	
surface	external surface in m^2, deduced from bounding box if not specified	
inflate	inflate animation, if any	
toggle	show the enable/disable toggle	true

17.3.8 HardDrive

The part has an interface to access the vessel hard drive, where the science data files are stored.

PROP-ERTY	DESCRIPTION	DE-FAULT
dataCa-pacity	Storage capacity for transmissible data, in Mb (=Mib)	102400
sample-Capacity	Capacity for experiment samples, in slots (=Mib). Note that Kerbalism will not display sample sizes in Mb, but uses a virtual size unit instead (slots, bags) (TBD)	100
title	Name displayed in file manager	
experi-ment_id	If set, restricts write access to the experiment with that id ON THE SAME PART with the given experiment_id.	

17.3.9 Harvester

The part harvests resources, similar to the stock resource harvester. The differences are that the output doesn't scale with concentration, instead it has the specified rate when above a threshold and zero below it.

PROPERTY	DESCRIPTION	DEFAULT
title	name to show on UI	
type	type of resource, same values accepted by stock harvester	0
resource	resource to extract	
min_abundance	minimal abundance required, in the range [0.0, 1.0]	
min_pressure	minimal pressure required, in kPA	
rate	amount of resource to extract per-second, when abundance is above threshold	
ec_rate	amount of EC consumed per-second, irregardless of abundance	
drill	the drill transform	

17.3.10 Laboratory

The part transforms non-transmissible science samples into transmissible science data over time.

PROPERTY	DESCRIPTION
ec_rate	EC consumed per-second
analysis_rate	analysis speed in Mb/s
researcher	required crew for analysis, in the format *trait@level*

17.3.11 PlannerController

The Part has a toggle to enable/disable simulation in the *Planner*. The *Planner* simulates resource consumption and production for many types of modules, and most of the time it is useful to be able to toggle these on and off in the VAB/SPH to simulate different scenarios for the vessel.

Some modules do not offer any way to toggle them on and off in the VAB/SPH and that's where the *PlannerController* comes in, once added to a part it will add an editor-only toggle button. The *Planner* will then consider or ignore all modules in that part depending on the toggle button state.

PROPERTY	DESCRIPTION	DEFAULT
toggle	show the toggle button in the editor	true
considered	default button state	false

17.3.12 ProcessController

The part has resource processing capabilities. This module allows the implementation of a scheme to provide converter-like modules on a vessel, while keeping the computation independent of the number of individual converters.

The trick is by using a Process which uses a hidden pseudo-resource created ad-hoc e.g. _WaterRecycler_.

This module then adds that resource to its part automatically, and provides a way to *start/stop* the process by a part UI button. Under the hood, starting and stopping the process is implemented by merely setting the resource flow to true and false respectively.

PROPERTY	DESCRIPTION	DEFAULT
resource	pseudo-resource to control	
title	name to show on UI	
desc	description to show on tooltip	
capacity	amount of pseudo-resource to add	1.0
toggle	show the enable/disable toggle	true
running	start the process by default	false

17.3.13 Reliability

The part has the capability of module failure. This module disables other modules when a *failure* happens.

PROPERTY	DESCRIPTION	DEFAULT
string	component module name	
mtbf	mean time between failures, in seconds	21600000.0
repair	trait and experience required for repair, in the form *trait@experience*	
title	short description of component	
redundancy	redundancy group	
extra_cost	extra cost for high-quality, in proportion of part cost	0.0
extra_mass	extra mass for high-quality, in proportion of part mass	0.0

17.3.14 Sensor

The part has sensor capabilities that adds environmental readings to a parts UI and to the *telemetry* panel on the *Monitor* UI.

PROPERTY	DESCRIPTION
type	type of sensor
pin	pin animation driven by telemetry value

The types of sensors available are.

TYPE	READINGS
temperature	external vessel temperature in K
radiation	environment radiation at vessel position, in rad/s (before shielding is applied)
pressure	environment pressure in kPA
gravioli	number of negative gravioli particles detected

17.3.15 Patch injection

Enabled features are specified by the user in the Settings file and are detected automatically from the modifiers used in the current profile. They are then used to inject MM patches on-the-fly at loading time, so that it is possible to do conditional MM patching depending on the features enabled by using **:NEEDS[FeatureXXX]**. Likewise it is possible to use **:NEEDS[ProfileXXX]** to do conditional MM patching depending on the current profile.

FEATURE	HOW IT IS DEFINED	WHAT DOES IT ENABLE
Reliability	user-specified in Settings file	component malfunctions and critical failures
Deploy	user-specified in Settings file	the deployment system
Science	user-specified in Settings file	the science system
SpaceWeather	user-specified in Settings file	coronal mass ejections
Automation	user-specified in Settings file	script UI and automatic execution
Radiation	detected from modifiers used	simulation and rendering of radiation
Shielding	detected from modifiers used	shielding resource added to habitats
LivingSpace	detected from modifiers used	volume is calculated for habitats
Comfort	detected from modifiers used	comfort parts are added
Poisoning	detected from modifiers used	atmospheric CO2 is simulated in habitats
Pressure	detected from modifiers used	atmospheric pressure is simulated in habitats
Humidity	detected from modifiers used	atmospheric humidity is simulated in habitats
Habitat	one or more features require it	the habitat module is added to parts

Downloads and Links

18.1 Latest release

- The Kerbalism release zip from v2.0.0 onwards will work for all the stated versions of KSP, basically one zip for all versions, no longer are there separate zips needed for a specific version of KSP.

- You can find the latest release of Kerbalism on SpaceDock or GitHub Releases

- If you like taking risks you can try the the latest Travis CI development build at https://builds.spaceball.cf

- Source code can be found on GitHub

- **Warning** There may be unfinished features and more bugs than you originally had using a Development build, use it at your own risk and check the CHANGELOG.md file for the latest changes.

- Also A source zip from the GitHub repository is not the same as a release zip from SpaceDock, GitHub Releases or the Travis CI Development build but it is not much different, just make sure not to copy the entire contents into your GameData folder. Basically download the source zip from GitHub and only copy the content that is in the GameData folder. **Note** using the source GameData folder is no different than using the Travis CI Development build.

18.2 Requirements

- Any KSP version from 1.3.1 to 1.6.x

- Community Resource Pack (CRP)

- ModuleManager 4.0.2+

18.3 Change Log

- A Change Log is available in your **GameData/Kerbalism** folder that contains the latest changes of the version you have installed.

- You can view the latest changes that have been developed but not released, on GitHub here CHANGELOG.md

18.4 Online Community

- There is a Kerbalism Thread on the Kerbal Space Program's forum and also a Discord Server for Kerbalism

Supported Mods

Most mods work together with Kerbalism, others don't. Such is life. For a complete list of supported mods have a look inside the Support folder. Some of the interactions deserve a special mention though:

19.1 SCANsat

- sensors consume EC in the background and their EC cost is evaluated by the planner
- sensors are shut down and restarted in background depending on EC availability

19.2 RemoteTech

- antenna EC cost is evaluated by the planner
- failures will disable the antenna

19.3 DeepFreeze

- all rules are suspended for hibernated Kerbals
- the vessel info window shows frozen Kerbals with a different color

19.4 NearFuture

- curved solar panels, reactors, fission generators and RTGs produce EC in background and are evaluated by the planner

19.5 PlanetaryBaseSystem

- the converters will work in the background and are evaluated by the planner

19.6 OrbitalScience

- experiments data size has been tweaked for background data transmission

19.7 OPM/RSS/NewHorizons

- custom radiation definitions for these planet packs are provided

About Kerbalism

Kerbalism was originally created by ShotgunNinja. It is now under active development by N70 and a number of contributors.

20.1 REQUIREMENTS

- Any KSP version from 1.3.1 to 1.6.x
- Community Resource Pack (CRP)
- ModuleManager 4.0.2+

The Kerbalism release zip from v2.0.0 onwards will work for all the stated versions of KSP, basically one zip for all versions, no longer are there separate zips needed for a specific version of KSP.

This mod includes version checking using MiniAVC. If you opt-in, it will use the Internet to check whether there is a new version available. Data is only read from the Internet and no personal information is sent. For a more comprehensive version checking experience, please download the KSP-AVC Plugin.

20.2 FAQs

There is a help file on GitHub for those wishing to report bugs or contribute to Kerbalism, see CONTRIBUTING.md.

I think I have found a bug, and I have just a few mods installed

- Try to reproduce it consistently, then provide us with reproduction steps that demonstrates the issue. You may be asked to supply log files, screen shots and maybe a save game. Post the report on the Kerbalism KSP forums thread, or raise an issue on GitHub Kerbalism Issues.

I want to add support for Kerbalism to my parts

- Add the appropriate modules to your parts. Check the Kerbalism modules documentation for the module specifications.

I want to interact with Kerbalism in my code

- Have a look at the System/API.cs source code on GitHub. Raise an issue to request more functions.

20.3 CONTRIBUTORS

This project wouldn't have been possible without the contributions of an awesome community of people, too many to mention individually. Thanks guys.

Also a special thanks goes out to the artists that provided all the parts:

- mehka: Gravity ring
- Nazari1382: Geiger counter, small supply container
- tygoo7: Medium and big supply containers, radial pressurized container
- zzz: Greenhouse, active shield

20.4 LICENSE

This mod is released under the Unlicense. For more information, please refer to unlicense.org