

Kerbal Space Program - Feature #2414

Improvements to the way modules are loaded.

04/23/2014 07:46 AM - swamp_ig

Status:	New	% Done:	0%
Severity:	Normal		
Assignee:			
Category:	Parts		
Target version:			
Platform:	Any	Mod Related:	No
Expansion:			
Description			
It would be really great if the way modules were loaded in the game were a bit more robust.			
Currently modules are loaded from the save files and saved ships in strict order of definition as in the part config. This presents a problem if you want to add in extra modules when you're doing any modding, especially when those modules have tweakable controls in them because the order of the modules determines the order the controls appear. It will also cause stock KSP issues if you ever want to add new modules to an existing part, especially if they are tweakable.			
What I might suggest is to change Part.LoadPart(ConfigNode node, int moduleIndex) to ignore the index, pull the name field out of the node, and just use PartModuleList[string] with the name instead of using PartModuleList[int]. This would be way more robust.			
The other modification I might request will require a small amount of extra work, but would enable having more than one copy of the same PartModule in a part, plus the ability to switch out a different PartModule class if you needed / wanted to for modding purposes.			
The way to do it would be to have an optional 'class' attribute for MODULE definitions that can be used to define what class you want to use. If this is defined then the module name is just used as a name and nothing else.			
The easiest place to do this is in Part.AddModule(ConfigNode), plus you'd also need to modify PartModule.Load to store the new name in a field, and also alter PartModuleList[string] to use the name attribute.			
Incidentally, there's a potential and hard-to-track down bug in PartModuleList[string] which I'll report separately.			

History

#1 - 04/23/2014 08:00 AM - swamp_ig

change Part.LoadPart(ConfigNode node, int moduleIndex) should read Part.LoadModule(ConfigNode node, int moduleIndex).

Here's the fixed code:

```
public PartModule LoadModule(ConfigNode node, int moduleIndex)
{
    if (!node.HasValue("name"))
    {
        Debug.LogWarning("In part " + base.name + " cannot load unnamed module from save file", gameObject);
        return null;
    }

    string moduleName = node.GetValue("name");

    if (!Modules.Contains(moduleName))
    {
        Debug.LogWarning("In part " + base.name + " module " + moduleName + " does not exist so cannot be load
ed from the save file", gameObject);
        return null;
    }

    PartModule partModule = this.Modules[name];
    partModule.Load(node);
    return partModule;
}
```

#2 - 04/26/2014 11:06 PM - hermes47

- Tracker changed from Bug to Feature

#3 - 05/07/2014 12:34 AM - swamp_ig

I'm working on a solid fix for this which will be part of the ModuleManager plugin. I would of course be happy to contribute the code back in the longer term.