

Kerbal Space Program - Bug #11494

Non-public fields annotated with [KSPField] don't get loaded

09/15/2016 06:26 AM - IgorZ

Status:	New	Start date:	09/15/2016
Severity:	Normal	% Done:	0%
Assignee:			
Category:	Plugins/Add-Ons		
Target version:			
Version:	1.1.3	Language:	English (US)
Platform:	Windows	Mod Related:	No
Expansion:			

Description

If a configurable field is not public it doesn't get populated on part load.

```
public abstract class MyAbstractBaseClass : PartModule {
    [KSPField]
    protected float myNonPublicSetting = 0;
}
```

```
public class MyChildClass : MyAbstractBaseClass {
    void OnLoad(ConfigNode node) {
        // Here base.myNonPublicSetting is always 0 regardless to the config settings in the part.
    }
}
```

When field `myNonPublicSetting` is declared as public everything works as expected. While it looks a minor issue it's actually a bad one since it breaks OOP concept. Some settings may only be needed by the super classes. Not to mention that most of the config settings are usually read-only for the outer world. In the latter case such settings are usually declared protected/private, and a public/protected read-only field is created as a part of API:

```
public abstract class MyAbstractBaseClass : PartModule {
    [KSPField]
    private float myNonPublicSetting = 0;
```

```
    // An API accessor for the descendants, but not for any caller.
    protected cfgMyNonPublicSetting {
        get { return myNonPublicSetting; }
    }
}
```

```
public class MyChildClass : MyAbstractBaseClass {
    void OnLoad(ConfigNode node) {
        // Here cfgMyNonPublicSetting returns value from the config but cannot be changed by the descendants
        // unless base class explicitly allowed it by declaring setter for the property.
    }
}
```

I would admit that allowing annotated fields to be private may result in names conflict: super class and the subclass may define a field with the same name, and it will be a challenge to resolve such setups. Well, at least "protected" could be allowed without any side effects since reflection returns both public and protected fields when requesting in flattern mode.