

Ruby master - Feature #8027

add the possibility to raise an exception in #included, #extended, #prepending, #inherited and break the calling feature

03/06/2013 05:26 PM - Hanmac (Hans Mackowiak)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
Description	
imo an exception in this hook methods should be possible to prevent the adding in the ancestor chain	
<pre>module A def self.included(mod) raise "dont include me" end end</pre>	
<pre>module E include A end ##>> RuntimeError: dont include me</pre>	
<pre>E.ancestors #=>[E,A] what i want: E.ancestors #=>[E]</pre>	
maybe other hook methods like method_defined should get an "prevent" option too	
maybe for not breaking code we could add "before_included" where an exception could be raised	

History

#1 - 03/08/2013 08:50 AM - semaperepelitsa (Semyon Perepelitsa)

I think you are looking for Module.append_features:

```
module A
def self.append_features(mod)
raise "dont include me"
super
end
end
```

```
module E
begin
include A
rescue
end
end
```

```
p E.ancestors #=>[E]
```

If you don't call super it also won't be included.

#2 - 03/08/2013 04:45 PM - Hanmac (Hans Mackowiak)

ah good to know, but not the best way. What if i want that a Module is extendable but not includable? Then append_features may not what i want

#3 - 06/02/2013 03:22 PM - zzak (Zachary Scott)

- Status changed from Open to Feedback