

Ruby master - Misc #11570

Clarify autoload chaining behavior

10/06/2015 04:52 PM - mwpastore (Mike Pastore)

Status:	Open	
Priority:	Normal	
Assignee:		
Description		
<p>I've discovered a discrepancy between how MRI 2.1.7 and 2.2.3 handle autoload "chaining" (which I'll describe below) cf. RBX 2.5.8. I opened an issue with them but the lead contributor of Rubinius is pushing back on me to clarify the expected behavior with you guys. Any guidance you can provide would be appreciated.</p>		
Essentially:		
<ul style="list-style-type: none">• File A autoloads :Foo from file B, and attempts to invoke methods on class Foo.• File B autoloads :Foo from file C, and attempts to reopen class Foo in order to define additional methods and attributes.• File C defines the base Foo class.		
<p>In MRI 2.1.7 and 2.2.3, file A can see the methods defined in the base class defined in file C, as well as the extended methods and attributes added in file B. Both autoloads fire in the expected order and the composite class is computed and made available to the caller.</p>		
<p>In RBX 2.5.8, file A can only see the extended methods and attributes defined in file B. Only the first autoload fires and the base class definition is never loaded or used.</p>		
Which is the correct behavior?		