Ruby master - Feature #10305

Method for resolving all autoload statements / Add warning on autoload when used with chroot

09/29/2014 02:05 PM - Quintus (Marvin Gülker)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Hi everyone,

I'm currently trying to make my way to create a chrooted process using Dir.chroot. However, once these lines are executed:

```
Dir.chroot("/some/path/here")
Dir.chdir("/")
```

all libraries that use the #autoload method (which are quite a lot) fail with LoadErrors everywhere. This is natural, because after being locked into the chroot the referenced paths are simply gone. I could live with that if there was a method to load all the files referenced by #autoload immediately, which I would then simply call before I lock the process into the chroot. However, it appears there is not even a way to get a list of all constants that are to be autoloaded; the #autoload? method only allows checks for specific constants I simply do not have at hand without digging through all the library code, which is infeasable.

Therefore, I suggest do add one of the following methods to Ruby:

1. Kernel#load_autoloads that loads all modules referenced with #autoload right now.
2. Kernel#autoload_list that returns a list of symbols for the constants that are to be autoloaded.

The methods could probably also be on Module rather than on the Kernel module, but I guess this is topic for discussion.

I also suggest that when calling Dir::chroot() and there are constants that are marked as autoloadable, a warning should be printed that the referenced files cannot be loaded if the constants are ever used.

Steps to reproduce the LoadError problem after chroot:

Create a file "foo.rb" with this content:

```
autoload :Bar, File.join(Dir.pwd, "bar.rb")

Dir.chroot "/var/empty"
Dir.chdir "/"

Bar.new
```

Create a file "bar.rb" with this content:

```
class Bar
end
```

Ensure you have the directory "/var/empty", and it is empty.

Execute "foo.rb" with root rights.

```
$ sudo ruby foo.rb
foo.rb:6:in `<main>': cannot load such file -- /home/quintus/foo/bar.rb (LoadError)
```

Valete,
Marvin