

Ruby - Bug #21339

Namespace: `RubyVM::InstructionSequence.load\_iseq` isn't called for the root namespace

05/14/2025 02:20 PM - byroot (Jean Boussier)

Status:	Assigned	
Priority:	Normal	
Assignee:	tagomoris (Satoshi Tagomori)	
Target version:		
ruby -v:	ruby 3.5.0dev (2025-05-14T12:41:46Z master b5575a80bc) +PRISM [arm64-darwin24]	Backport: 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

```
File.write("/tmp/compile-cache.rb", <<~'RUBY')
  class << RubyVM::InstructionSequence
    def load_iseq(path)
      p [:load_iseq, path]
      RubyVM::InstructionSequence.compile_file(path)
    end
  end
end
RUBY
File.write("/tmp/test-file.rb", "")

puts "main:"
require "/tmp/compile-cache.rb"
require "/tmp/test-file.rb"

if ENV["RUBY_NAMESPACE"]
  puts "namespace:"
  ns = Namespace.new
  ns.require("/tmp/compile-cache.rb")
  ns.require("/tmp/test-file.rb")
end
```

Expected behavior:

I would expect load_iseq to be invoked both in the root namespace and the user namespace when RUBY_NAMESPACE=1.

Actual behavior:

Somehow it's only invoked in the user namespace:

```
main:
[:compile_cache_setup]
[:load_iseq, "/tmp/test-file.rb"]

main:
[:compile_cache_setup]
namespace:
[:compile_cache_setup]
[:load_iseq, "/tmp/test-file.rb"]
```

But I don't understand why.

Related issues:	
Related to Ruby - Bug #21324: Namespace loads RubyGems in root Namespace but ...	Open

History

#1 - 05/15/2025 01:58 AM - mame (Yusuke Endoh)

It works as expected with --disable=gems.

```
$ RUBY_NAMESPACE=1 ./local/bin/ruby --disable=gems test.rb
./local/bin/ruby: warning: Namespace is experimental, and the behavior may change in the future!
```

```
See doc/namespace.md for known issues, etc.
main:
[:load_iseq, "/tmp/test-file.rb"]
namespace:
[:load_iseq, "/tmp/test-file.rb"]
```

But it does not by default.

```
$ RUBY_NAMESPACE=1 ./local/bin/ruby test.rb
./local/bin/ruby: warning: Namespace is experimental, and the behavior may change in the future!
See doc/namespace.md for known issues, etc.
main:
namespace:
[:load_iseq, "/tmp/test-file.rb"]
```

The reason seems to be that rubygems's require monkey patch does not work in sub namespace.

```
# with RUBY_NAMESPACE=1
puts "main:"
require "csv" #=> OK

if ENV["RUBY_NAMESPACE"]
  puts "namespace:"
  ns = Namespace.new
  # csv is a bundled gem, so rubygems is needed to require it
  ns.require("csv") #=> cannot load such file -- csv (LoadError)
end
```

Thus, there are two problems

- load_iseq does not fire via RubyGems require monkey patch.
- RubyGems require monkey patch does not work in sub namespaces.

#2 - 05/15/2025 10:48 AM - Eregon (Benoit Daloze)

- Related to Bug #21324: Namespace loads RubyGems in root Namespace but it should not added

#3 - 05/15/2025 10:49 AM - Eregon (Benoit Daloze)

Basically dependent or even caused by [#21324](#) then

#4 - 06/03/2025 05:46 AM - hsbt (Hiroshi SHIBATA)

- Tags set to namespace