

Ruby master - Bug #10902

require("enumerator") scans LOAD_PATH 2x on every invocation

02/24/2015 10:34 PM - tmm1 (Aman Gupta)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.1.5	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

On every invocation of require "enumerator" (for example during boot when many gems require it), the VM will scan the load path twice: once for enumerator.rb and again for enumerator.so. Of course, no file is found because enumerator is now shipped within the VM by default.

```
$ ruby -e 'p $LOADED_FEATURES[0] '
"enumerator.so"
```

```
$ ruby -e 'p $LOAD_PATH.size '
8
```

```
$ strace -e trace=open ruby -e ' 1.times{ require "enumerator" } ' 2>&1 | grep enumerator.rb | wc -
1
8
```

```
$ strace -e trace=open ruby -e ' 1.times{ require "enumerator" } ' 2>&1 | grep enumerator.so | wc -
1
8
```

```
$ strace -e trace=open ruby -e ' 10.times{ require "enumerator" } ' 2>&1 | grep enumerator.so | wc -
-1
80
```

```
$ strace -e trace=open ruby -e ' 100.times{ require "enumerator" } ' 2>&1 | grep enumerator.so | wc -
-1
800
```

In enumerator.c, we call rb_provide("enumerator.so") which adds it to \$LOADED_FEATURES. This means require "enumerator.so" can be optimized, but most libraries do not include the .so extension when requiring enumerator.

History

#1 - 02/24/2015 10:39 PM - tmm1 (Aman Gupta)

This gets much worse as \$LOAD_PATH grows. For example in our app \$LOAD_PATH.size is 351, causing 702 failed open() syscalls per require.

#2 - 02/28/2015 09:49 PM - tmm1 (Aman Gupta)

Having a hard time coming up with a clean patch here. The following works but it's pretty hacky.

```
diff --git a/load.c b/load.c
index fa225fa..68d15e7 100644
--- a/load.c
+++ b/load.c
@@ -952,6 +952,9 @@ rb_require_safe(VALUE fname, int safe)
     } volatile saved;
     char *volatile ftptr = 0;

+   if (strcmp(RSTRING_PTR(fname), "enumerator") == 0)
+   return Qfalse;
+
     if (RUBY_DTRACE_REQUIRE_ENTRY_ENABLED()) {
         RUBY_DTRACE_REQUIRE_ENTRY(StringValuePtr(fname),
             rb_sourcefile(),
```

#3 - 07/04/2015 09:28 PM - normalperson (Eric Wong)

I tried to fix it in `load.c`, but failed `rubyspec`:

<http://80x24.org/spew/m/1436044472-18990-1-git-send-email-e@80x24.org.txt>

However, this one-liner seems to work:

```
--- a/enumerator.c
+++ b/enumerator.c
@@ -2060,6 +2060,7 @@ InitVM_Enumerator(void)
     rb_define_method(rb_cYielder, "yield", yielder_yield, -2);
     rb_define_method(rb_cYielder, "<<", yielder_yield_push, -2);

+   rb_provide("enumerator.rb");
+   rb_provide("enumerator.so"); /* for backward compatibility */
}
```

I'm not sure if we need something similar for complex and rational,
not sure if they were really used via `require` in the past...

<http://80x24.org/spew/m/1436044803-31588-1-git-send-email-e@80x24.org.txt>

#4 - 07/15/2015 11:58 PM - normalperson (Eric Wong)

Ping? <https://bugs.ruby-lang.org/issues/10902>

I'd like to commit my second patch soon since this scan bothers me.

#5 - 07/16/2015 07:44 AM - nobu (Nobuyoshi Nakada)

It will disallow a library named "enumerator.rb".