Ruby trunk - Feature #6376

Feature lookup and checking if feature is loaded

04/29/2012 07:03 AM - trans (Thomas Sawyer)

Status:	Assigned
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	

Description

$LOADED_FEATURES is useful to know what "files" have been loaded. But it doesn't really tell us what "features" have been loaded. If there where were a way to look-up a load path, without actually loading it then it would be possible to compare that to $LOADED_FEATURES and thus know. e.g.

```
require 'ostruct'
$LOADED_FEATURES #=> [..., "/home/trans/.local/lib/ry/rubies/1.9.3-p125/lib/ruby/1.9.1/ostruct.rb"
]

path = require_path('ostruct') #=> "/home/trans/.local/lib/ry/rubies/1.9.3-p125/lib/ruby/1.9.1/ost
ruct.rb"
$LOADED_FEATURES.include?(path)
```

Of course, it would be nice to also have:

```
required?('ostruct')  #=> true
```

These methods could be class methods of special module, if it's important to keep the Kernel more tidy, e.g. Ruby.required?('ostruct').

I am currently working on a project where I need this (and have a couple of other projects that could use it too) and I've had to implement the whole thing from scratch, which isn't simple, nor fast, nor am I 100% confident that it specs exactly to Ruby's own lookup procedure. So it would be much better if Ruby would expose its lookup functionality.

History

#1 - 04/29/2012 07:06 AM - trans (Thomas Sawyer)

Oh, I forget to mention that there seems to be no way to see what the "current loading feature" is either, as it appears that it is not added to $LOADED_FEATURES until after loading is completed, which kind of surprised me. Is that right?

#2 - 04/29/2012 05:29 PM - now (Nikolai Weibull)

On Sun, Apr 29, 2012 at 00:03, trans (Thomas Sawyer)
transfire@gmail.com wrote:

> These methods could be class methods of special module, if it's important to keep the Kernel more tidy, e.g. Ruby.required?('ostruct').

Why not add it to $LOADED_FEATURES?

#3 - 05/03/2012 01:02 PM - mame (Yusuke Endoh)

- Status changed from Open to Feedback

Of course you know what is defined by the feature you loaded, don't you?
(If not, you must not load such a file; it is very dangerous)

So you can use: defined?(OpenStruct)

--
Yusuke Endoh mame@tsg.ne.jp

#4 - 05/03/2012 01:23 PM - now (Nikolai Weibull)

On Thu, May 3, 2012 at 6:02 AM, mame (Yusuke Endoh) mame@tsg.ne.jp wrote:

> Issue #6376 has been updated by mame (Yusuke Endoh).

> Status changed from Open to Feedback
>
> Of course you know what is defined by the feature you loaded, don't you?
> (If not, you must not load such a file; it is very dangerous)
>
> So you can use: defined?(OpenStruct)

Except that defined? won't work for paths:

```
class A
end
class B
end
p defined?(A::B)
```

so if you have a file that provides A::B and B has already been
provided by another library, then defined? won't work.

#5 - 05/03/2012 01:53 PM - mame (Yusuke Endoh)

2012/5/3 Nikolai Weibull now@bitwi.se:

> so if you have a file that provides A::B and B has already been
> provided by another library, then defined? won't work.

Then, defined?(A::B.some_class_method) or else.

Anyway, I don't think it is a good idea to depend whether a feature is
"loaded" by require or not.
What we really need to know is, whether a feature that you need is
"defined", doesn't it?

--
Yusuke Endoh mame@tsg.ne.jp

#6 - 05/03/2012 02:04 PM - trans (Thomas Sawyer)

I think it depends. For on thing, a library's api can change over time. So that might not be the best fit, if what your asking is if library xyz is being used?. That's a more general question. Two different libraries might share some of the same module namespaces.

#7 - 05/03/2012 02:53 PM - now (Nikolai Weibull)

On Thu, May 3, 2012 at 6:53 AM, Yusuke Endoh mame@tsg.ne.jp wrote:

> 2012/5/3 Nikolai Weibull now@bitwi.se:
>
> > so if you have a file that provides A::B and B has already been
> > provided by another library, then defined? won't work.
>
> Then, defined?(A::B.some_class_method) or else.

That's far from good enough.

> Anyway, I don't think it is a good idea to depend whether a feature is
> "loaded" by require or not.
> What we really need to know is, whether a feature that you need is
> "defined", doesn't it?

Yes, certainly. Add loaded?/feature? that works like defined? except
that it doesn't leak its lookup into parent namespaces.

#8 - 10/27/2012 06:57 AM - ko1 (Koichi Sasada)

- Assignee set to mame (Yusuke Endoh)

mame-san, could you judge this ticket?

#9 - 11/20/2012 02:44 AM - mame (Yusuke Endoh)

- Status changed from Feedback to Assigned

- Target version changed from 2.0.0 to 2.6

If I had a right to judge a feature request, I would reject this.
But actually I have no right. Assigning to matz. (virtual rejection?)

I don't see what OP really want to do. Doesn't Rescue'ing a LoadError from require help?

```
begin
require "foo"
FooExist = true
rescue LoadError
FooExist = false
end
```

A dirty operation should be performed by a dirty code.

--
Yusuke Endoh mame@tsg.ne.jp

#10 - 11/20/2012 03:36 AM - trans (Thomas Sawyer)

=begin
mame (Yusuke Endoh) Your example would load the library. The request is to know where a library comes from ((({require_path('ostruct')}))). As an additional benefit it would allow us to know if a library has been loaded or not. It's nothing to do with catching an load error.
=end

#11 - 11/20/2012 09:09 PM - mame (Yusuke Endoh)

- Assignee changed from mame (Yusuke Endoh) to matz (Yukihiro Matsumoto)

Yes I know what you want. But I don't know why you want it. In general, I don't think that it is a good idea to depend on whether a feature is loaded or not. Rather, you should make sure to require what feature you need.

So, please elaborate your use case. I guessed one use case: you want to use one of many alternative libraries, for example, either eventmachine or Celluloid::IO. In such a case, you may want to try to require one, and if a LoadError is raised, then require the other.

--
Yusuke Endoh mame@tsg.ne.jp

#12 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)