Ruby master - Feature #5643
require/load options and binding option
11/17/2011 07:41 AM - trans (Thomas Sawyer)

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

Description
Current Kernel#load is defined as:

```
load(filename, wrap=false)
```

I purpose that it be modified to work as option argument, e.g.

```
load(filename, :wrap=>true)
```

Right off the bat this has better name connascence.

Then support an additional option :binding, such that, given:

```
$ cat lib/example.rb
def a
  1
end

then

class X
  load('example.rb', :binding=>binding)
end

X.new.a #=> 1
```

The binding option should also work with #require (which would also support option parameter) differing from #load in the it would only allow the feature to be loaded once per-binding's self regardless of being required again.

This ability would greatly benefit systems that need "plugin" capability. Presently, a great deal of coding has to go into simulating this functionality to create plugin systems, which are often imperfect nor robust.

Related issues:
Related to Ruby master - Feature #10320: require into module

History
#1 - 03/28/2012 12:27 AM - mame (Yusuke Endoh)
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)

#2 - 10/27/2012 06:27 AM - ko1 (Koichi Sasada)
- Target version changed from 2.0.0 to 2.6

I changed target to next minor because no discussion on it.

#3 - 12/22/2012 01:27 AM - trans (Thomas Sawyer)
I think someone else recently suggested that the wrap argument be able to be a module which the code is then evaluated under, instead of toplevel. Considering that, maybe :wrap option should be used for all cases and it can just handle any of boolean, module or binding accordingly.

#4 - 10/03/2014 04:48 PM - nobu (Nobuyoshi Nakada)
- Related to Feature #10320: require into module added

03/25/2022
- Target version deleted (2.6)