Ruby master - Feature #13934

[Feature request] Being able to set a default encoding other than Unicode on a "per-project" basis

09/24/2017 02:07 PM - shevegen (Robert A. Heiler)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Hello ruby-core team and everyone else,

I propose a "project-wide encoding" or a "toplevel namespace wide encoding".

I'll soon explain what I mean there; but first, allow me to start with a problem description or more, a slight inconvenience.

In Ruby we can use an Encoding String in a .rb file such as "# Encoding: ISO-8859-1" or any of the other encodings, after the shebang line. This is also described here:

https://docs.ruby-lang.org/en/2.4.0/Encoding.html

This is fine and works but if you have lots of .rb files and also files under the bin/ subdirectory of a gem, then this is not really a great way to handle encoding IMO. It can become cumbersome, in particular if you consider switching to another encoding again one day. (I may have to switch to Unicode, at the least for some projects).

For Unicode, this is probably not a big deal because it is the default anyway, but for other encodings, this is a bit annoying. The more files you have, the more often you may have to use the magic encoding comment.

Thus, I propose some easier way to set the Encoding on a per-project, that is, per "namespace" way.

I think most ruby hackers will use a toplevel constant, usually a module, sometimes a class, for their projects. So this could be used to designate another default encoding to use other than Unicode - but only on a per-project basis (per-toplevel constant), rather than hardcoded in every individual .rb file.

So, for example, let's illustrate this, as this may make things easier to understand.

Given the "module Foobar", and two classes called "A" and "B" inside of this module, we can set the default encoding to be used in this way then:

```
module Foobar
  class A; end
  class B; end
end
Foobar.default_encoding = 'ISO-8859-1'
```

The name for the method is just a suggestion; any other method name could be used. The more important part is whether the functionality would be good/useful. (In theory one could also use other ways, such as via Kernel or Object, and specifying the encoding to be used on another namespace, but I thought that setting the default encoding on the particular constant which the ruby hacker wants to modify, would make the most sense.)

The above would be essentially equivalent to using a line such as "# Encoding: ISO-8859-1" in all the individual .rb files (at the least in the .rb files that require this information).

I am not sure if I have missed anything here, perhaps it is not possible or unwanted, I have no idea. But I wanted to suggest it anyway since it may give people the ability to quickly change the default encoding for all .rb files, rather than having to specify it via post-shebang lines. Note that invoking this method would also overrule any individual shebang definitions, as the ruby hacker who invokes this method, will be assumed to know what he/she is doing. Feel free to close this issue at any moment in time.

Thanks for reading!

May ruby save many ducklings.

History

#1 - 09/24/2017 02:29 PM - phluid61 (Matthew Kerwin)

If it's a method, that means you have to parse the file and execute it, before you can know how to parse it ..?

#2 - 09/25/2017 05:18 AM - naruse (Yui NARUSE)

In addition to what Matthew and Yui said, I have some more questions:

What about files with more than one module, or with code outside a module and code inside the module?

Are you planning to have actual content e.g. in Latin-1 in one of these files? What happens to that content if the encoding is changed? If you yourself don't plan this, what should happen in this case, anyway?

If the content of the file is in fact only US-ASCII, then Ruby is quite permissive in combining this with data that is ASCII-compatible. Where do you have actual specific problems?

"Your project" isn't an island. How do you plan to make sure that this project works together with libraries that don't assume the same encoding?

What about projects that use more than one module? Or no module?

What about having an option on the ruby command for this functionality?