

Ruby - Feature #639

New String#encode_internal method

10/12/2008 08:13 AM - mike (Michael Selig)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description =begin Now that we have default_internal, I think there needs to be an easy way to check that a string is that encoding and if not transcode it. This is especially needed in libraries and other methods which are expected to return strings in default_internal encoding if it is set. Suggested implementation: class String def encode_internal intern = Encoding.default_internal # Do nothing if default_internal not set return self unless intern # Do nothing if already in default_internal encoding return self if encoding == intern # Just use "force_encoding" if compatible eg: ASCII/UTF-8 return force_encoding(intern) if Encoding.compatible(self, intern) == intern # Otherwise transcode encode(intern) end end =end		

History

#1 - 10/13/2008 04:02 AM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [\[ruby-core:19298\]](#) [Feature [#639](#)] New String#encode_internal method"
on Sun, 12 Oct 2008 08:11:17 +0900, Michael Selig redmine@ruby-lang.org writes:

[Now that we have default_internal, I think there needs to be an easy way to check that a string is that encoding and if not transcode it. This is especially needed in libraries and other methods which are expected to return strings in default_internal encoding if it is set.

I'd rather propose, default_internal to be the default target encoding
of the String#encode method, so that

str.encode

to be equivalent to

str.encode(default_internal)

with the optimization tricks you've suggested.

```
matz.
```

=end

#2 - 10/13/2008 07:57 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin
Applied in changeset r19764.
=end

#3 - 10/14/2008 05:57 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [\[ruby-core:19309\]](#) Re: [Feature [#639](#)] New String#encode_internal method"
on Mon, 13 Oct 2008 09:09:25 +0900, "Michael Selig" michael.selig@fs.com.au writes:

|I just tried String#encode with no parameter, but I was hoping to just be
|able to say:

|
| return str.encode

|to return a string from a method in the default_internal encoding if it is
|set. Unfortunately, unlike my suggested implementation, it raises an
|exception when default_internal is not set, so I have to say:

|
| return Encoding.default_internal ? str.encode : str

Indeed. I will fix.

matz.

=end