The following raises an error currently:

```ruby
require 'stringio'
require 'openssl'
io = StringIO.new(OpenSSL::ASN1::Integer.new(1).to_der)
asn = OpenSSL::ASN1.decode io
```

The reason is that `ossl_obj2bio[1]` looks for a T_FILE first, and then it tries to coerce the input to a String using `StringValue`. `StringValue` itself again expects the presence of `to_str`, which is currently missing for `StringIO`, but could be easily provided by aliasing `StringIO#string`.

I could imagine that the heuristic of `ossl_obj2bio` is quite common when working on binary data in a C extension. Would it therefore be OK to add `StringIO#to_str`? Patch attached.


**History**

#1 - 11/28/2011 10:51 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Feedback

Martin Bosslet wrote:

The following raises an error currently:

```ruby
require 'stringio'
require 'openssl'
io = StringIO.new(OpenSSL::ASN1::Integer.new(1).to_der)
asn = OpenSSL::ASN1.decode io
```

Why do you need to pass `io`, not `io.string` or the result of `to_der`?

#2 - 11/28/2011 11:16 AM - MartinBosslet (Martin Bosslet)

Nobuyoshi Nakada wrote:

Martin Bosslet wrote:

The following raises an error currently:

```ruby
require 'stringio'
require 'openssl'
io = StringIO.new(OpenSSL::ASN1::Integer.new(1).to_der)
asn = OpenSSL::ASN1.decode io
```

09/10/2021
Why do you need to pass `io`, not io.string or the result of to_der?

I don't really need to, it's just that I am working on making ASN1.decode streaming-aware. So I wanted it to work with any IO or IO-like object and additionally with Strings. That's why I thought if someone passes a StringIO, they would be surprised if an error was raised - they passed an IO-like object after all and there is no apparent reason why this shouldn't work.

I could, however, call StringIO#string as you suggested, but I would prefer to use StringValue, as a more generic way of coercing things into Strings. And it would potentially work for a wider variety of objects, not only limited to StringIO.

Do you see any negative aspects when adding StringIO#to_str?

---

#3 - 11/28/2011 03:23 PM - matz (Yukihiro Matsumoto)

Hi,

In message "Re: [ruby-core:41347] [ruby-trunk - Feature #5678] StringIO#to_str" on Mon, 28 Nov 2011 11:16:33 +0900, Martin Bosslet Martin.Bosslet@googlemail.com writes:

Do you see any negative aspects when adding StringIO#to_str?

Yes, to_str should be implemented for objects that have (almost) same method signature as strings, which is not true for StringIO.

matz.

#4 - 11/28/2011 06:50 PM - MartinBosslet (Martin Bosslet)

- Status changed from Feedback to Closed

Yukihiro Matsumoto wrote:

Hi,

In message "Re: [ruby-core:41347] [ruby-trunk - Feature #5678] StringIO#to_str" on Mon, 28 Nov 2011 11:16:33 +0900, Martin Bosslet Martin.Bosslet@googlemail.com writes:

Do you see any negative aspects when adding StringIO#to_str?

Yes, to_str should be implemented for objects that have (almost) same method signature as strings, which is not true for StringIO.

matz.

OK, thank you for clarifying this! I will use StringIO#string then.

#5 - 11/28/2011 06:59 PM - now (Nikolai Weibull)

On Mon, Nov 28, 2011 at 07:16, Yukihiro Matsumoto matz@ruby-lang.org wrote:

In message "Re: [ruby-core:41347] [ruby-trunk - Feature #5678] StringIO#to_str" on Mon, 28 Nov 2011 11:16:33 +0900, Martin Bosslet Martin.Bosslet@googlemail.com writes:

Do you see any negative aspects when adding StringIO#to_str?

Yes, to_str should be implemented for objects that have (almost) same method signature as strings, which is not true for StringIO.

How about implementing #to_s instead?

#6 - 11/28/2011 07:23 PM - matz (Yukihiro Matsumoto)

Hi,

In message "Re: [ruby-core:41354] [ruby-trunk - Feature #5678][Closed] StringIO#to_str" on Mon, 28 Nov 2011 18:50:12 +0900, Martin Bosslet Martin.Bosslet@googlemail.com writes:

OK, thank you for clarifying this! I will use StringIO#string then.
At the same time, having ASN1.decode to treat T_FILE but not StringIO is against duck typing. I think it is better if we can do something. But I don't have a good idea yet. #to_io does not work here. #to_str is not appropriate.

matz.

#7 - 11/29/2011 04:12 AM - MartinBosslet (Martin Bosslet)

Yukihiro Matsumoto wrote:

Hi,
In message "Re: [ruby-core:41354] [ruby-trunk - Feature #5678][Closed] StringIO#to_str" on Mon, 28 Nov 2011 18:50:12 +0900, Martin Bosslet Martin.Bosslet@googlemail.com writes:

At the same time, having ASN1.decode to treat T_FILE but not StringIO is against duck typing. I think it is better if we can do something. But I don't have a good idea yet. #to_io does not work here. #to_str is not appropriate.

matz.

We could handle StringIO exceptionally by trying to call #string on the object passed. But that's not too elegant and will probably only work in the very specific case of StringIO. Implementing #to_s like Nikolai suggested could be another solution? Then we could apply the following heuristic that should cover almost any (valid) case:

1. check for T_FILE
2. try #to_str
3. try #to_s ?

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>stringio.patch</td>
<td>582 Bytes</td>
<td>11/28/2011</td>
<td>MartinBosslet (Martin Bosslet)</td>
</tr>
</tbody>
</table>