

## Ruby trunk - Bug #9028

### Make SSLSocket Support Encodings

10/17/2013 09:31 AM - whitehat101 (Jeremy Ebler)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	drbrain (Eric Hodel)	
<b>Target version:</b>		
<b>ruby -v:</b>	1.9.3, 2.0.0-p0	<b>Backport:</b> 1.9.3: DONTNEED, 2.0.0: REQUIRED
<b>Description</b>		
<p>I was working on a bug in the xmpp4r project that caused REXML exceptions when receiving UTF-8 Strings. <a href="https://github.com/xmpp4r/xmpp4r/issues/13">https://github.com/xmpp4r/xmpp4r/issues/13</a></p> <p>The issue ended up being that SSLSocket#readline didn't always return strings with the same encoding. It gave plain ASCII strings an encoding of UTF-8, and UTF-8 strings an encoding of ASCII-8BIT. We were passing the SSLSocket directly to REXML::Parsers::SAX2Parser and REXML throws exceptions when the input is not UTF-8.</p> <p>Our solution, wrap the socket and always return consistently encoded strings:</p> <pre>class SSLSocketUtf8 &lt; OpenSSL::SSL::SSLSocket   def sysread *args     super.force_encoding ::Encoding::UTF_8   end end</pre> <p>Hello, I'm investigating some strange behavior with OpenSSL::SSL::SSLSocket and string encodings #readline returns UTF-8 encoded strings, until the string actually contains UTF-8, then it claims that the encoding is ASCII-8BIT I've been reading through the source, and I'm not sure where to try to patch it whitehat101: have an example script? whitehat101: can you reproduce it with #sysread? if you can, the problem lies in the C code if you cannot, the problem lies in the OpenSSL::Buffering module I don't have a concise example, I'm working with the xmpp4r project whitehat101: look at sample/openssl/echo_* you can probably make a simple example out of that I found that #sysread always returns 8BIT, but #readline usually gives UTF-8 Thank you, i'll look at those whitehat101: then I imagine the problem is that OpenSSL::Buffering#initialize creates a UTF-8 buffer (@rbuffer) I bet that # encoding: ASCII-8BIT at the very top of the file will fix it in buffering.rb? in ext/openssl/lib/openssl/buffering.rb My feeling is that these functions should be returning UTF-8 A patch that works for my project: class SSLSocketUtf8 &lt; OpenSSL::SSL::SSLSocket   def sysread *args     super.force_encoding ::Encoding::UTF_8   end end end hrm they should be returning the encoding of the SSLSocket It doesn't look like SSLSocket has any supportfor encodings I tried setting the encoding of the TCPSocket, but it had no effect since SSLSocket wraps the TCPSocket, I don't know if that has an effect on SSLSocket#sysread I'm guessing that SSLSocket has no idea what the encoding is, it just deals with bytes We're passing the SSLSocket directly to REXML::Parsers::SAX2Parser and REXML throws exceptions when the input is not UTF-8 possibly, since it isn't an IO subclass and doesn't seem to respond to #set_encoding setting the encoding on the TCPSocket probably has no effect because SSLSocket needs to read binary data off the TCPSocket the ultimate solution would be "make SSLSocket support encodings" That sounds right to me a short-term fix would be "make the SSLSocket methods return a consistent encoding, regardless of correctness"</p>		

whitehat101: if you file a bug, maybe I'll find the time to fix it for ruby 2.1  
you can file one here: <http://bugs.ruby-lang.org/projects/ruby-trunk/issues/new>  
That would be excellent, thanks  
Should I try to make an example, or just include this conversation?  
this conversation is enough

## Associated revisions

---

### Revision e45f56d5 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43964 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 43964 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

### Revision 43964 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

### Revision 43964 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

### Revision 43964 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

### Revision 43964 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

### Revision 43964 - 12/03/2013 01:44 AM - drbrain (Eric Hodel)

- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug #9028]
- test/openssl/test\_ssl.rb: Test for the above.

## History

---

### #1 - 12/03/2013 07:14 AM - drbrain (Eric Hodel)

- File *openssl.buffering.rb.encoding.patch* added
- Category set to *ext/openssl*
- Status changed from *Open* to *Assigned*
- Assignee changed from *drbrain (Eric Hodel)* to *MartinBosslet (Martin Bosslet)*
- Backport changed from *1.9.3: UNKNOWN, 2.0.0: UNKNOWN* to *1.9.3: DONTNEED, 2.0.0: REQUIRED*

The attached patch makes OpenSSL::Buffering methods return ASCII-8BIT strings for all results.

This doesn't bring encoding support to OpenSSL::SSL::Socket, but it avoids the bug of the buffering methods sometimes returning UTF-8 and sometimes returning ASCII-8BIT strings.

### #2 - 12/03/2013 10:39 AM - MartinBosslet (Martin Bosslet)

- Assignee changed from *MartinBosslet (Martin Bosslet)* to *drbrain (Eric Hodel)*

Sounds great. As long as there is no explicit encoding support for sockets, ASCII-8BIT also seems the most natural to me. Thank you and please go

ahead, Eric!

**#3 - 12/03/2013 10:44 AM - drbrain (Eric Hodel)**

- Status changed from *Assigned* to *Closed*

- % Done changed from 0 to 100

This issue was solved with changeset r43964.  
Jeremy, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

- 
- ext/openssl/lib/openssl/buffering.rb: Return ASCII-8BIT strings from SSLSocket methods. [ruby-trunk - Bug [#9028](#)]
  - test/openssl/test\_ssl.rb: Test for the above.

**Files**

---

openssl.buffering.rb.encoding.patch	1.02 KB	12/03/2013	drbrain (Eric Hodel)
-------------------------------------	---------	------------	----------------------