Ruby trunk - Feature #8217

OpenSSL::BN.new with integers

04/04/2013 03:59 PM - naruse (Yui NARUSE)

Status:	Closed
Priority:	Normal
Assignee:	MartinBosslet (Martin Bosslet)
Target version:	2.6

Description

Current OpenSSL::BN.new gets only strings, so users must do integer.to_s, it costs extra resource.
Therefore I propose OpenSSL::BN.new to allow Fixnu/Bignum.

```
diff --git a/ext/openssl/ossl_bn.c b/ext/openssl/ossl_bn.c
index 1038135..1f1ebba 100644
--- a/ext/openssl/ossl_bn.c
+++ b/ext/openssl/ossl_bn.c
@@ -120,6 +120,44 @@ ossl_bn_initialize(int argc, VALUE *argv, VALUE self)
base = NUM2INT(bs);
}

    if (RB_TYPE_P(str, T_FIXNUM)) {
    long i;
    unsigned char *bin = (unsigned char*)ALLOC_N(long, 1);
    long n = FIX2LONG(str);
    unsigned long un = abs(n); +
    for (i = sizeof(VALUE) - 1; 0 <= i; i--) {
    bin[i] = un&0xff;
    un >>= 8;
    } +
    GetBN(self, bn);
    if (!BN_bin2bn(bin, sizeof(long), bn)) {
    ossl_raise(eBNError, NULL);
    }
    if (n < 0) BN_set_negative(bn, 1);
    return self;
    }
    else if (RB_TYPE_P(str, T_BIGNUM)) {
    long i, j;
    BDIGIT *ds = RBIGNUM_DIGITS(str);
    unsigned char *bin = (unsigned char*)ALLOC_N(BDIGIT, RBIGNUM_LEN(str)); +
    for (i = 0; RBIGNUM_LEN(str) > i; i++) {
    BDIGIT v = ds[i];
    for (j = sizeof(BDIGIT) - 1; 0 <= j; j--) {
    bin[(RBIGNUM_LEN(str)-1-i)*sizeof(BDIGIT)+j] = v&0xff;
    v >>= 8;
    }
    } +
    GetBN(self, bn);
    if (!BN_bin2bn(bin, sizeof(BDIGIT)*RBIGNUM_LEN(str), bn)) {
    ossl_raise(eBNError, NULL);
    }
    if (!RBIGNUM_SIGN(str)) BN_set_negative(bn, 1);
    return self;
    }  if (RTEST(rb_obj_is_kind_of(str, cBN))) { BIGNUM *other;

diff --git a/test/openssl/test_bn.rb b/test/openssl/test_bn.rb
index af1c72c..758cc54 100644
--- a/test/openssl/test_bn.rb
+++ b/test/openssl/test_bn.rb
@@ -8,6 +8,13 @@ class OpenSSL::TestBN < Test::Unit::TestCase
end

def test_integer_to_bn
```

- assert_equal(999.to_bn, OpenSSL::BN.new(999))
- assert_equal((2 ** 107 - 1).to_bn, OpenSSL::BN.new(2 ** 107 - 1))
- assert_equal(-999.to_bn, OpenSSL::BN.new(-999))
- assert_equal((-(2 ** 107 - 1)).to_bn, OpenSSL::BN.new(-(2 ** 107 - 1)))
- end +
- def test_integer_str_to_bn assert_equal(999.to_bn, OpenSSL::BN.new(999.to_s(16), 16)) assert_equal((2 ** 107 - 1).to_bn, OpenSSL::BN.new((2 ** 107 - 1).to_s(16), 16)) end

Associated revisions

Revision 8b29525d - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@40461 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 40461 - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

Revision 40461 - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

Revision 40461 - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

Revision 40461 - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

Revision 40461 - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

Revision 40461 - 04/25/2013 07:02 AM - naruse (Yui NARUSE)

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]

History

#1 - 04/25/2013 04:02 PM - naruse (Yui NARUSE)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r40461.
Yui, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- ext/openssl/ossl_bn.c (ossl_bn_initialize): allow Fixnum and Bignum. [ruby-core:53986] [Feature #8217]