

## Ruby master - Bug #11268

### Zlib::DataError: incorrect header check

06/16/2015 02:15 PM - stevenspiel (Steven Spiel)

<b>Status:</b>	Rejected		
<b>Priority:</b>	Normal		
<b>Assignee:</b>	akr (Akira Tanaka)		
<b>Target version:</b>			
<b>ruby -v:</b>	2.2.2p95	<b>Backport:</b>	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

#### Description

I'm having an issue opening a webpage with open-uri

```
$ ruby -v
ruby 2.2.2p95 (2015-04-13 revision 50295) [x86_64-darwin14]
$ irb
```

```
require 'open-air'
=> true
open('https://www.hoveround.com')
Zlib::DataError: incorrect header check
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:377:in inflate'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:377:in block in inflate_adapter'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/protocol.rb:411:in call_block'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/protocol.rb:402:in <<'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/protocol.rb:102:in read'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:399:in read'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:317:in read_chunked'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:281:in block in read_body_0'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:260:in inflater'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:279:in read_body_0'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:201:in read_body'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:333:in block (2 levels) in open_http'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http.rb:1421:in block (2 levels) in transport_request'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http/response.rb:162:in reading_body'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http.rb:1420:in block in transport_request'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http.rb:1411:incatch'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http.rb:1411:in transport_request'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http.rb:1384:in request'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:324:in block in open_http'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/net/http.rb:853:in start'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:318:in open_http'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:736:in buffer_open'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:211:in block in open_loop'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:209:incatch'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:209:in open_loop'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:150:in open_uri'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:716:in open'
from /Users/stevenspiel/.rbenv/versions/2.2.2/lib/ruby/2.2.0/open-uri.rb:34:in open'
from (irb):2
from /Users/stevenspiel/.rbenv/versions/2.2.2/bin/irb:11:in ``
```

#### Associated revisions

##### Revision 5105240b - 12/16/2019 02:20 PM - mame (Yusuke Endoh)

lib/net/http/response.rb: support raw deflate correctly

Net::HTTP had used Zlib::Inflate.new(32 + Zlib::MAX\_WBITS) for all content encoding (deflate, zlib, and gzip).  
But the argument 32 + Zlib::MAX\_WBITS means zlib and gzip decoding

with automatic header detection, so (raw) deflate compression had not been supported.

This change makes it support raw deflate correctly by passing an argument `-Zlib::MAX_WBITS` (which means raw deflate) to `Zlib::Inflate.new`. All deflate-mode tests are fixed too.

[Bug #11268]

### Revision 251f5d82 - 12/16/2019 02:39 PM - mame (Yusuke Endoh)

Revert "lib/net/http/response.rb: support raw deflate correctly"

This reverts commit 5105240b1e851410020b3b3f1a2bead7ffdd4291.

In RFC 2616:

```
deflate
    The "zlib" format defined in RFC 1950 [31] in combination with
    the "deflate" compression mechanism described in RFC 1951 [29].
```

So "Content-Encoding: deflate" means zlib format, not raw deflate.

[Bug #11268]

## History

---

### #1 - 06/16/2015 03:03 PM - Hanmac (Hans Mackowiak)

i can reproduce it with "ruby 2.3.0dev (2015-06-15 trunk 50908) [x86\_64-linux]"

but i did use require 'open-uri' instead of require 'open-air'

### #2 - 08/20/2015 04:22 PM - nkmrya (Yasuhiro Nakamura)

I check the response from <https://www.hoveround.com>.

```
$ openssl s_client -connect www.hoveround.com:443
```

Request header

```
GET / HTTP/1.1
Host: www.hoveround.com
Accept-Encoding: gzip,deflate
```

Response

```
HTTP/1.1 200 OK
Cache-Control: private, no-store, must-revalidate
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
Content-Encoding: deflate
Server: Microsoft-IIS/8.5
X-Frame-Options: SAMEORIGIN
Set-Cookie: CMSPreferredCulture=en-US; expires=Sat, 20-Aug-2016 14:26:58 GMT; path=/; HttpOnly
Set-Cookie: ASP.NET_SessionId=ksufrr40usfo3ecjvxnudpus; path=/; HttpOnly
Set-Cookie: CurrentContact=788a7c31-f770-4eba-bd0f-f322b42b3f71; expires=Thu, 20-Aug-2015 14:26:58 GMT; path=/; HttpOnly
X-UA-Compatible: IE=Edge
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Strict-Transport-Security: max-age=10886400
Date: Thu, 20 Aug 2015 14:26:59 GMT
```

```
5361
... encoded content
```

encoded content

```
edbde972db48b230fa...
```

That server encodes the content by Deflate. Deflate has not header. (see RFC 1951) Encoded content is not satisfy Gzip header and Zlib header, so `Zlib::Inflate` can not inflate it.

I think, the solution is to remove deflate from request header or to make new method for inflate raw 'Deflate'. Which solution is better?

### #3 - 09/08/2015 02:38 AM - nagachika (Tomoyuki Chikanaga)

- Status changed from Open to Assigned
- Assignee set to akr (Akira Tanaka)

Yasuhiro, thank you for your investigation!  
So is it server side issue?

Anyway, you can specify request header via open-uri.

```
require "open-uri"  
open("https://www.hoveround.com", "Accept-Encoding" => "plain") do |f| puts f.read end
```

This works fine on my environments.

### #4 - 09/08/2015 01:40 PM - nkmrya (Yasuhiro Nakamura)

I think it is issue of ruby.

Test case of test/net/http/test\_httpresponse.rb is wrong.  
That test case say "\x9C\xCBH\xCD\xC9\xC9\xa\x00\x06,\x02\x15" is raw Deflate format (at line 85) but actually that hex is Zlib format.  
A case of deflated 'hello' is "\xCBH\xCD\xC9\xC9\xa\x00".  
I think ruby can not inflate a data which is compressed by raw Deflate format.  
So ruby can not inflate the response of <https://www.hoveround.com> .  
I examined these problems by using the php.

I am writing a path for problem of "open-uri".

### #5 - 09/08/2015 03:34 PM - nkmrya (Yasuhiro Nakamura)

- File net\_http.patch added

This patch has passed tests.

### #6 - 12/16/2019 02:29 PM - mame (Yusuke Endoh)

- Status changed from Assigned to Closed

Applied in changeset [git|5105240b1e851410020b3b3f1a2bead7fdd4291](https://github.com/ruby/ruby/commit/5105240b1e851410020b3b3f1a2bead7fdd4291).

---

lib/net/http/response.rb: support raw deflate correctly

Net::HTTP had used Zlib::Inflate.new(32 + Zlib::MAX\_WBITS) for all content encoding (deflate, zlib, and gzip).  
But the argument 32 + Zlib::MAX\_WBITS means zlib and gzip decoding with automatic header detection, so (raw) deflate compression had not been supported.

This change makes it support raw deflate correctly by passing an argument -Zlib::MAX\_WBITS (which means raw deflate) to Zlib::Inflate.new. All deflate-mode tests are fixed too.

[Bug [#11268](#)]

### #7 - 12/16/2019 02:31 PM - mame (Yusuke Endoh)

I've changed Net::HTTP to support (raw) deflate correctly. Now it works correctly.

```
$ ./local/bin/ruby -ropen-uri -e 'URI.open("https://www.hoveround.com")'
```

[nkmrya \(Yasuhiro Nakamura\)](#) Thank you for the great investigation.

### #8 - 12/16/2019 02:50 PM - mame (Yusuke Endoh)

- Status changed from Closed to Rejected

No! RFC2616 says:

```
deflate  
    The "zlib" format defined in RFC 1950 [31] in combination with  
    the "deflate" compression mechanism described in RFC 1951 [29].
```

<https://tools.ietf.org/html/rfc2616#section-3.5>

So, "Content-Encoding: deflate" means zlib format, not raw deflate. In this case, <https://www.hoveround.com> violates the spec.

The previous commit breaks communication with a valid HTTP server that returns zlib format with a header "Content-Encoding: deflate". So reverted.

**#9 - 12/16/2019 02:58 PM - mame (Yusuke Endoh)**

[https://zlib.net/zlib\\_faq.html#faq39](https://zlib.net/zlib_faq.html#faq39)

39 . What's the difference between the "gzip" and "deflate" HTTP 1.1 encodings?

"gzip" is the gzip format, and "deflate" is the zlib format. They should probably have called the second one "zlib" instead to avoid confusion with the raw deflate compressed data format. While the HTTP 1.1 RFC 2616 correctly points to the zlib specification in RFC 1950 for the "deflate" transfer encoding, there have been reports of servers and browsers that incorrectly produce or expect raw deflate data per the deflate specification in RFC 1951, most notably Microsoft. So even though the "deflate" transfer encoding using the zlib format would be the more efficient approach (and in fact exactly what the zlib format was designed for), using the "gzip" transfer encoding is probably more reliable due to an unfortunate choice of name on the part of the HTTP 1.1 authors.

Bottom line: use the gzip format for HTTP 1.1 encoding.

**#10 - 12/16/2019 03:30 PM - naruse (Yui NARUSE)**

RFC 7230, which obsoletes RFC 2616, says "Note: Some non-conformant implementations send the "deflate" compressed data without the zlib wrapper". But it doesn't say the actual fallback algorithm.

<https://httpwg.org/specs/rfc7230.html#deflate.coding>

If WHATWG fetch Standard or RFC defines an algorithm, Ruby will introduce it.

**Files**

---

net_http.patch	6.01 KB	09/08/2015	nkmyra (Yasuhiro Nakamura)
----------------	---------	------------	----------------------------