

## Ruby master - Feature #6612

### Add streaming inflate and deflate to Zlib

06/20/2012 03:35 PM - drbrain (Eric Hodel)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	2.0.0
<b>Description</b>	
<p>=begin Currently there is no way to control the inflate output size of a deflate stream from ruby. For example, 50MB of "0"s compress to just under 50KB:</p> <pre>ruby -rzlib -e 'p Zlib.deflate("0" * 50_000_000, Zlib::BEST_COMPRESSION).length' #=&gt; 48611</pre> <p>When inflating this, 50MB are allocated which is undesirable.</p> <p>The attached patch allows Zlib::Inflate#inflate, Zlib::Deflate#deflate, Zlib::ZStream#finish and other methods that end up calling zstream_expand_buffer or zstream_detach_buffer to be called with a block which gives the user more control over the amount of memory allocated in their process. (A fixed maximum chunk size of 16384 bytes is used in this patch.)</p> <p>The new API looks like:</p> <pre>z = Zlib::Inflate.new z.inflate deflate_string do  chunk  # write chunk to output stream # nil is returned from inflate end  footer = z.finish # flush buffer to output stream</pre> <p>Here's a comparison of resource usage:</p> <pre>\$ dd if=/dev/zero of=/dev/stdout bs=1m count=1024   gzip -c &gt; 1G.gz \$ cat test.rb require 'zlib'  gzipped = File.read '1G.gz'  z = Zlib::Inflate.new Zlib::MAX_WBITS + 32  z.inflate gzipped do  chunk  # do nothing with chunk # current ruby will ignore this block end  z.finish</pre> <p>With existing ruby (trunk 35758) 1GB of memory is allocated:</p> <pre>\$ /usr/bin/time -l ruby20 test.rb 3.86 real    3.24 user    0.61 sys 1080475648 maximum resident set size 0 average shared memory size 0 average unshared data size 0 average unshared stack size 263860 page reclaims 0 page faults 0 swaps 0 block input operations 0 block output operations 0 messages sent</pre>	

```
0 messages received
0 signals received
1 voluntary context switches
34 involuntary context switches
```

With the patch only 37MB are allocated:

```
$/usr/bin/time -l ./ruby20 -l .ext/x86_64-darwin11.4.0 test.rb
3.47 real    3.43 user    0.03 sys
36724736 maximum resident set size
0 average shared memory size
0 average unshared data size
0 average unshared stack size
8981 page reclaims
0 page faults
0 swaps
0 block input operations
9 block output operations
0 messages sent
0 messages received
0 signals received
1 voluntary context switches
90 involuntary context switches
```

Some notes about this patch:

`zstream_expand_buffer()` yields each block of inflate (or deflate) output instead of expanding the buffer (as in non-block output). `zstream_expand_buffer_into()` is used to reduce duplication as well.

`zstream_detach_buffer()` yields the buffer and returns nil if a block was given. `OBJ_INFECT` was moved into `zstream_detach_buffer()` as well.

A new stream flag `ZSTREAM_FLAG_GZFILE` is added to prevent a block given to a gzip method that invokes `zstream_expand_buffer()` or `zstream_detach_buffer()` from yielding.

To ensure proper handling of `((|z->buf|))` such as resetting the buffer after yielding a chunk, `rb_protect` is used in `zstream_expand_buffer` and `zstream_run`. In `zstream_run`, the for loop has been extracted to `zstream_run_loop`.

In `zstream_run_loop`, `Zlib::BufError` is no longer raised when flushing (finishing) the stream. This allows the stream to be flushed into one string when the inflate block is interrupted.

=end

## Associated revisions

### Revision 3e419e6c - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- `ext/zlib/zlib.c`: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- `ext/zlib/zlib.c` (`zstream_expand_buffer`): Uses `rb_yield` when a block is given for streaming support. Refactored to use `zstream_expand_buffer_into` to remove duplicate code.
- `ext/zlib/zlib.c` (`zstream_expand_buffer_protect`): Added wrapper function to pass jump state back through GVL-free section to allow `zstream` clean-up before terminating the ruby call.
- `ext/zlib/zlib.c` (`zstream_expand_buffer_without_gvl`): Acquire GVL to yield processed chunk of output stream.
- `ext/zlib/zlib.c` (`zstream_detach_buffer`): When a block is given, returns `Qnil` mid-stream and yields the output buffer at the end of the stream.
- `ext/zlib/extconf.rb`: Update `INCFLAGS` to find `internal.h` for `rb_thread_call_with_gvl`
- `test/zlib/test_zlib.rb`: Updated tests

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36349 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 36349 - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- `ext/zlib/zlib.c`: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- `ext/zlib/zlib.c` (`zstream_expand_buffer`): Uses `rb_yield` when a block is given for streaming support. Refactored to use `zstream_expand_buffer_into` to remove duplicate code.

- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36349 - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36349 - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36349 - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36349 - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36349 - 07/10/2012 12:04 AM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.

- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 3cf7d1b5 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36356 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 36356 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36356 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36356 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36356 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.

- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36356 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

#### Revision 36356 - 07/10/2012 06:28 PM - drbrain (Eric Hodel)

- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature #6612]
- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- test/zlib/test\_zlib.rb: Updated tests

## History

---

### #1 - 06/20/2012 03:43 PM - drbrain (Eric Hodel)

- File deleted (zlib.inflate\_deflate\_chunked.patch)

### #2 - 06/20/2012 03:43 PM - drbrain (Eric Hodel)

- File zlib.inflate\_deflate\_chunked.patch added

Oops, I uploaded the wrong patch.

### #3 - 07/04/2012 05:38 AM - drbrain (Eric Hodel)

- File zlib.inflate\_deflate\_chunked.2.patch added

Updated patch that will apply after [#6615](#)

I committed the OBJ\_INFECT refactoring separately.

### #4 - 07/04/2012 11:35 AM - nobu (Nobuyoshi Nakada)

Don't modify a String object with realloc().  
It causes a crash when CALC\_EXACT\_MALLOC\_SIZE is set.

### #5 - 07/06/2012 03:47 AM - drbrain (Eric Hodel)

It looks like using ruby\_xrealloc() will solve this, correct?

### #6 - 07/10/2012 09:04 AM - drbrain (Eric Hodel)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r36349.  
Eric, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

- 
- ext/zlib/zlib.c: Added streaming support to inflate processing. This allows zlib streams to be processed without huge memory growth. [Feature [#6612](#)]

- NEWS: ditto
- ext/zlib/zlib.c (zstream\_expand\_buffer): Uses rb\_yield when a block is given for streaming support. Refactored to use zstream\_expand\_buffer\_into to remove duplicate code.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_protect): Added wrapper function to pass jump state back through GVL-free section to allow zstream clean-up before terminating the ruby call.
- ext/zlib/zlib.c (zstream\_expand\_buffer\_without\_gvl): Acquire GVL to yield processed chunk of output stream.
- ext/zlib/zlib.c (zstream\_detach\_buffer): When a block is given, returns Qnil mid-stream and yields the output buffer at the end of the stream.
- ext/zlib/extconf.rb: Update INCFLAGS to find internal.h for rb\_thread\_call\_with\_gvl
- test/zlib/test\_zlib.rb: Updated tests

## Files

---

zlib.inflate_deflate_chunked.patch	18.1 KB	06/20/2012	drbrain (Eric Hodel)
zlib.inflate_deflate_chunked.2.patch	16 KB	07/04/2012	drbrain (Eric Hodel)