Ruby master - Feature #9924
Revisiting GC.stat keys

06/09/2014 04:56 PM - ko1 (Koichi Sasada)

Status: Closed
Priority: Normal
Assignee: ko1 (Koichi Sasada)
Target version: 2.2.0

Description

Now, GC.stat returns internal information about GC.

However, key names are from internal implementations fields, so that it is difficult to understand what they mean.

To solve this problem, I reconsider about key names.

https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEiUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing

Any comments are welcome.

Related issues:
Copied from Ruby master - Feature #9923: Revisiting GC.stat keys

Associated revisions

Revision 8ee9071c - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages
- gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEiUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing Yellow color fields in this table are changed.
- test/ruby/test_gc.rb: catch up this change.

Revision 47471 - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages
- gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEiUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing Yellow color fields in this table are changed.
- test/ruby/test_gc.rb: catch up this change.

Revision 47471 - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages

03/17/2022
Revision 47471 - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages
- gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIukHJ5JzRR0NyZfrhEEnc8/edit?usp=sharing
  Yellow color fields in this table are changed.
- test/ruby/test_gc.rb: catch up this change.

Revision 47471 - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages
- gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIukHJ5JzRR0NyZfrhEEnc8/edit?usp=sharing
  Yellow color fields in this table are changed.
- test/ruby/test_gc.rb: catch up this change.

Revision 47471 - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages
- gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIukHJ5JzRR0NyZfrhEEnc8/edit?usp=sharing
  Yellow color fields in this table are changed.
- test/ruby/test_gc.rb: catch up this change.

Revision 47471 - 09/09/2014 09:33 AM - ko1 (Koichi Sasada)

- gc.c (rb_objspace_t::heap_pages): rename field names:
  - used -> allocated_pages
  - increment -> allocatable_pages
  - length -> sorted_length And remove unused 'limt' field.
- gc.c: rename macros:
  - heap_pages_used -> heap_allocated_pages
  - heap_pages_length -> heap_pages_sorted_length
  - heap_pages_increment -> heap_allocatable_pages
- gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIukHJ5JzRR0NyZfrhEEnc8/edit?usp=sharing
  Yellow color fields in this table are changed.
- test/ruby/test_gc.rb: catch up this change.

Revision 6c625427 - 09/10/2014 01:42 AM - ko1 (Koichi Sasada)

- gc.c (objspace_total_slot): rename objspace_available_slots.
- gc.c ( objspace_live_slot, objspace_free_slot): rename ..._slot() to ..._slots().
- gc.c (objspace_free_slot): should subtract heap_pages_final_slots.
Revision 47491 - 09/10/2014 02:35 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - remembered_shady_object -> remembered_wb_unprotected_objects
  - remembered_shady_object_limit -> remembered_wb_unprotected_objects_limit
  - old_object -> old_objects
  - old_object_limit -> old_objects_limit ref: [Feature #9924]

Revision 47491 - 09/10/2014 02:35 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - remembered_shady_object -> remembered_wb_unprotected_objects
  - remembered_shady_object_limit -> remembered_wb_unprotected_objects_limit
  - old_object -> old_objects
  - old_object_limit -> old_objects_limit ref: [Feature #9924]

Revision 47491 - 09/10/2014 02:35 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - remembered_shady_object -> remembered_wb_unprotected_objects
  - remembered_shady_object_limit -> remembered_wb_unprotected_objects_limit
  - old_object -> old_objects
  - old_object_limit -> old_objects_limit ref: [Feature #9924]

Revision 47491 - 09/10/2014 02:35 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - remembered_shady_object -> remembered_wb_unprotected_objects
  - remembered_shady_object_limit -> remembered_wb_unprotected_objects_limit
  - old_object -> old_objects
  - old_object_limit -> old_objects_limit ref: [Feature #9924]

Revision ed729d39 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 47493 - 09/10/2014 02:47 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename heap_used' to heap_allocated_pages'. ref: [Feature #9924]
- test/ruby/test_gc.rb: add constraints test for gc stat information.

Revision 52c41f1e - 09/10/2014 02:53 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - malloc_increase -> malloc_increase_bytes
  - malloc_limit -> malloc_increase_bytes_limit
  - oldmalloc_increase -> oldmalloc_increase_bytes
  - oldmalloc_limit -> oldmalloc_increase_bytes_limit ref: [Feature #9924]

Revision 47495 - 09/10/2014 02:53 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - malloc_increase -> malloc_increase_bytes
  - malloc_limit -> malloc_increase_bytes_limit
  - oldmalloc_increase -> oldmalloc_increase_bytes
  - oldmalloc_limit -> oldmalloc_increase_bytes_limit ref: [Feature #9924]

Revision 47495 - 09/10/2014 02:53 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - malloc_increase -> malloc_increase_bytes
  - malloc_limit -> malloc_increase_bytes_limit
  - oldmalloc_increase -> oldmalloc_increase_bytes
  - oldmalloc_limit -> oldmalloc_increase_bytes_limit ref: [Feature #9924]

Revision 47495 - 09/10/2014 02:53 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - malloc_increase -> malloc_increase_bytes
  - malloc_limit -> malloc_increase_bytes_limit
  - oldmalloc_increase -> oldmalloc_increase_bytes
  - oldmalloc_limit -> oldmalloc_increase_bytes_limit ref: [Feature #9924]

Revision 47495 - 09/10/2014 02:53 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - malloc_increase -> malloc_increase_bytes
  - malloc_limit -> malloc_increase_bytes_limit
  - oldmalloc_increase -> oldmalloc_increase_bytes
  - oldmalloc_limit -> oldmalloc_increase_bytes_limit ref: [Feature #9924]

Revision 47495 - 09/10/2014 02:53 AM - ko1 (Koichi Sasada)

- gc.c (gc_stat_internal): rename:
  - malloc_increase -> malloc_increase_bytes
  - malloc_limit -> malloc_increase_bytes_limit
  - oldmalloc_increase -> oldmalloc_increase_bytes
  - oldmalloc_limit -> oldmalloc_increase_bytes_limit ref: [Feature #9924]

Revision 5d5c95f2 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)
• gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object (new key istotal_allocated_objects') Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :
  • hash.c (rb_hash_set_default_proc): export (in internal).
  • internal.h: ditto.

Revision 48423 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)

• gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object (new key istotal_allocated_objects'). Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :
  • hash.c (rb_hash_set_default_proc): export (in internal).
  • internal.h: ditto.

Revision 48423 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)

• gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object (new key istotal_allocated_objects'). Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :
  • hash.c (rb_hash_set_default_proc): export (in internal).
  • internal.h: ditto.

Revision 48423 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)

• gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object (new key istotal_allocated_objects'). Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :
  • hash.c (rb_hash_set_default_proc): export (in internal).
  • internal.h: ditto.

Revision 48423 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)

• gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object (new key istotal_allocated_objects'). Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :
  • hash.c (rb_hash_set_default_proc): export (in internal).
  • internal.h: ditto.

Revision 48423 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)

• gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object (new key istotal_allocated_objects'). Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :
  • hash.c (rb_hash_set_default_proc): export (in internal).
  • internal.h: ditto.
• internal.h: ditto.

History

#1 - 06/09/2014 04:56 PM - ko1 (Koichi Sasada)
- Copied from Feature #9923: Revisiting GC.stat keys added

#2 - 07/16/2014 07:27 AM - ko1 (Koichi Sasada)
No objection?

#3 - 07/26/2014 08:12 AM - ko1 (Koichi Sasada)
I got Matz's approval.

#4 - 09/09/2014 09:34 AM - ko1 (Koichi Sasada)
- Status changed from Open to Closed
- % Done changed from 0 to 100

Applied in changeset r47471.

• gc.c (rb_objspace_t::heap_pages): rename field names:
  ◦ used -> allocated_pages
  ◦ increment -> allocatable_pages
  ◦ length -> sorted_length And remove unused 'limit' field.
• gc.c: rename macros:
  ◦ heap_pages_used -> heap_allocated_pages
  ◦ heap_pages_length -> heap_pages_sorted_length
  ◦ heap_pages_increment -> heap_allocatable_pages
• gc.c (gc_stat_internal): fix symbol names ref: [Feature #9924]
  Yellow color fields in this table are changed.
• test/ruby/test_gc.rb: catch up this change.

#5 - 09/09/2014 12:51 PM - ko1 (Koichi Sasada)
- Status changed from Closed to Open

r47472
r47473
r47474

are also related changes (and not finished).

These fixes break programs depends on specific GC.stat() keys.

Any other objections?

#6 - 09/10/2014 01:42 AM - ko1 (Koichi Sasada)
- Status changed from Open to Closed

Applied in changeset r47488.

• gc.c (objspace_total_slot): rename objspace_available_slots.
• gc.c (objspace_live_slot, objspace_free_slot): rename __slot() to __slots().
• gc.c (objspace_free_slot): should subtract heap_pages_final_slots.
• gc.c (gc_stat_internal):
  ◦ add 'heap_available_slots' field
  ◦ rename heap_live_slot to heap_live_slots
  ◦ rename heap_free_slot to heap_free_slots ref: [Feature #9924]

#7 - 09/10/2014 02:54 AM - ko1 (Koichi Sasada)
- Status changed from Closed to Open

fixed.

https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nQRzZLEIUIkJJRzRR0NyZfrhEEnc8/edit?usp=sharing

03/17/2022
Feedbacks are welcome.

#8 - 11/14/2014 07:29 AM - ko1 (Koichi Sasada)
- Status changed from Open to Closed

Applied in changeset r48423.

- gc.c (gc_stat_internal): add compatible layer. From Ruby 2.2, keys of GC.stat are changed [Feature #9924]. To provide compatible layer, GC.stat add a default_proc (if default_proc of given Hash object is not set). At first use of this compatible layer of interpreter process, show a warning message like that: program: GC.stat[:total_allocated_object] warning message: "warning: GC.stat keys were changed from Ruby 2.1. In this case, you refer to obsolete total_allocated_object" (new key istotal_allocated_objects'). Please check https://bugs.ruby-lang.org/issues/9924 for more information." Please correct my English message :)
- hash.c (rb_hash_set_default_proc): export (in internal).
- internal.h: ditto.