

## Ruby trunk - Feature #9924

### Revisiting GC.stat keys

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

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	ko1 (Koichi Sasada)	
<b>Target version:</b>	2.2.0	
<b>Description</b>		
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.		
<a href="https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing">https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing</a>		
Any comments are welcome.		
<b>Related issues:</b>		
Copied from Ruby trunk - Feature #9923: Revisiting GC.stat keys		Closed 06/09/2014

#### 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 '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]  
<https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing> Yellow color fields in this table are changed.
- test/ruby/test\_gc.rb: catch up this change.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47471 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### 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 '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]  
<https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/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 '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] <https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/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 '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] <https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/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 '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] <https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/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 '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] <https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/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 '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] <https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/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.

- 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]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47488 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 47488 - 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.
- 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]

**Revision 47488 - 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.
- 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]

**Revision 47488 - 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.
- 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]

**Revision 47488 - 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.
- 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]

**Revision 47488 - 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.
- 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]

**Revision 47488 - 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.
- 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]

**Revision 2d581d6c - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47490 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 47490 - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

**Revision 47490 - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

**Revision 47490 - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

**Revision 47490 - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

**Revision 47490 - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

**Revision 47490 - 09/10/2014 02:13 AM - ko1 (Koichi Sasada)**

- gc.c (gc\_stat\_internal): support:
  - total\_allocated\_pages
  - total\_freed\_pages ref: [Feature #9924]

**Revision 6b398398 - 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]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47491 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**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 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.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47493 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### 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 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]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47495 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### 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 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." Pleaes correct my English message :)
- hash.c (rb\_hash\_set\_default\_proc): export (in internal).
- internal.h: ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48423 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### 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." Pleaes 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." Pleaes 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." Pleaes 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." Pleaes 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." Pleaes 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." Pleaes correct my English message :)
- hash.c (rb\_hash\_set\_default\_proc): export (in internal).

- 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]  
<https://docs.google.com/spreadsheets/d/11Ua4uBr6o0k-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing> 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-nORrZLEIIUkHJ9JRzRR0NyZfrhEEnc8/edit?usp=sharing>



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." Pleaes correct my English message :)
- hash.c (rb\_hash\_set\_default\_proc): export (in internal).
- internal.h: ditto.