

Module GC

The GC module provides an interface to Ruby's mark and sweep garbage collection mechanism. Some of the underlying methods are also available via the ObjectSpace module, described beginning on page 635.

Module methods

count GC.count → *int*

1.9 / Returns a count of the number of times GC has run in the current process.

```
GC.count # => 1
res = ""
10_000.times { res += "wibble" }
GC.count # => 38
```

disable GC.disable → true or false

Disables garbage collection, returning true if garbage collection was already disabled.

```
GC.disable # => false
GC.disable # => true
```

enable GC.enable → true or false

Enables garbage collection, returning true if garbage collection was disabled.

```
GC.disable # => false
GC.enable # => true
GC.enable # => false
```

start GC.start → nil

Initiates garbage collection, unless manually disabled.

```
GC.start # => nil
```

stress GC.stress → true or false

1.9 / Returns the current value of the stress flag (see GC.stress=).

stress= GC.stress = *bool* → *bool*

1.9 / Ruby will normally run garbage collection periodically. Setting the stress flag to true forces garbage collection to occur every time Ruby allocates a new object. This is typically used only for testing extensions (and Ruby itself).

```
GC.stress = true
```

Instance methods

garbage_collect garbage_collect → nil

Equivalent to GC.start.

```
include GC
garbage_collect # => nil
```