The current TAR header parsing code assumes the size is represented as an octal string. Because this is a 12-byte, null-terminated field, effectively this can represent up to 8GB (8^{11}). For bigger files the standard allows to define the field as a 12-byte INTEGER instead. When using this form, the first bit of the field should be turned on to signal that it is used.

Currently, TAR files containing files larger then 8GB in this format would fail parsing because size would be computed as 0.

(Wiki with some description of the logic, couldn't find a more "formal" document: http://en.wikipedia.org/wiki/Tar_(computing)#File_header)

The problem is with this code: http://yard.ruby-doc.org/stdlib-2.1.0/Gem/Package/TarHeader.html#from-class_method

The line that assigns the value to size should be conditioned on the value of the first bit, and should treat the two cases differently.

---

**History**

#1 - 02/18/2014 03:33 PM - luislavena (Luis Lavena)
- Status changed from Open to Assigned
- Assignee set to drbrain (Eric Hodel)

#2 - 07/01/2019 07:15 AM - ruby_learner (Jay Mav)
Hello,
Is there plan to fix this issue? Need this Windows because there is no native support for extracting tar contents. Is there any workaround for is there any other rubygem which can extract archived_file >= 8gb?
Thanks.

#3 - 07/30/2019 09:43 AM - hsbt (Hiroshi SHIBATA)
- Backport deleted (1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN)
- Assignee changed from drbrain (Eric Hodel) to hsbt (Hiroshi SHIBATA)
- Status changed from Assigned to Third Party's Issue

I'm not sure what usecase of this issue.

Can you file the details into the upstream repository? https://github.com/rubygems/rubygems

Thanks.