

## Ruby master - Bug #8994

add methods for Float to get if an NaN is quiet or not, also add class methods for Float to "generate" an quiet NaN and an loud NaN

10/08/2013 04:45 PM - Hanmac (Hans Mackowiak)

<b>Status:</b> Open	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> ruby 2.1.0dev (2013-10-01) [x86_64-darwin12.5.0]	<b>Backport:</b> 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
<b>Description</b>	
<p>Currently its not easy possible to know of an NaN in ruby is silent or not</p> <p>like Float::NAN.quiet?</p> <p>there should also methods to make an new NaN like Float::quiet_nan, maybe with the possibility to set user data</p> <p>also there is</p> <pre>[Float::NAN].pack("g") #=&gt; "\x7F\xC0\x00\x00" [0.0/0.0].pack("g")  #=&gt; "\xFF\xC0\x00\x00"</pre> <p>and</p> <pre>[-(0.0/0.0)].pack("g") #=&gt; "\x7F\xC0\x00\x00"</pre> <p>so it seems that - can turn an quiet NaN into an loud one? (i am not 100% clear about that) specially when using two different NaN in one operation like</p> <pre>[(0.0/0.0) + Float::NAN].pack("g") #=&gt; "\xFF\xC0\x00\x00" [Float::NAN + (0.0/0.0)].pack("g") #=&gt; "\x7F\xC0\x00\x00"</pre> <p>Wikipedia says: For example, a bit-wise IEEE floating-point standard single precision (32-bit) NaN would be: s111 1111 1axx xxxx xxxx xxxx xxxx xxxx where s is the sign (most often ignored in applications), a determines the type of NaN, and x is an extra payload (most often ignored in applications). If a = 1, it is a quiet NaN; if a is zero and the payload is nonzero, then it is a signaling NaN.</p> <p>so it seems that the negation does it a bit wrong? i mean -@ should not change the value like that?</p>	