

## Ruby - Feature #1275

### IPAddr unnecessarily destroys information on creation

03/12/2009 10:11 PM - chrylis (Christopher Smith)

<b>Status:</b>	Rejected	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	knu (Akinori MUSHHA)	
<b>Target version:</b>	2.0.0	
<b>Description</b>		
<p>=begin</p> <p>The IPAddr class internally stores both a complete IP address and a network mask. For many applications, client code needs to be able to track an interface's IP address (say, 10.253.0.20) and its netmask (/24), and in IPv6, both the 128-bit identifier and the netmask are generally considered part of the address.</p> <p>The current implementation of IPAddr immediately clears all masked bits, making it impossible to store information like "10.253.0.20/24"; instead, the class stores only the <i>network</i> address of 10.253.0.0/24.</p> <p>Since it is always possible to mask off bits later, IPAddr should internally store the entire address alongside the netmask and provide the masked address on request (as already provided in #mask). I suggest adding an option "mask_host_part=true" to initialize and changing the check condition at 483 to "if prefixlen and mask_host_part" (to keep from breaking current code).</p> <p>=end</p>		
<b>Related issues:</b>		
Related to Ruby - Feature #1905: How about IPAddr#<=> to take care of mask_addr?		<b>Rejected</b> <b>08/07/2009</b>

#### History

##### #1 - 07/16/2009 06:02 PM - yugui (Yuki Sonoda)

=begin  
Patch is welcome.  
=end

##### #2 - 11/29/2009 09:53 AM - ujihisa (Tatsuhiko Ujihisa)

- Status changed from Open to Assigned
- Assignee set to knu (Akinori MUSHHA)

=begin  
=end

##### #3 - 04/02/2010 08:15 AM - znz (Kazuhiro NISHIYAMA)

- Category changed from core to lib
- Target version set to 2.0.0

=begin  
=end

##### #4 - 02/13/2012 11:44 PM - mame (Yusuke Endoh)

- Status changed from Assigned to Rejected

I'm rejecting this feature ticket because no progress has been made for a long time. See [\[ruby-core:42391\]](#).

I'm not sure if this use case is valid for IPAddr library.  
But if you still want the feature seriously, it might be good to start with creating a patch, as Yugui said.

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)