

Ruby master - Feature #13153

Inconsistent sprintf formatting for 0 value

01/24/2017 11:33 AM - RickHull (Rick Hull)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<pre>0.upto(5).map { i "%#0.2x" % i }</pre>	
<pre># expect: ["0x00", "0x01", "0x02", "0x03", "0x04", "0x05"]</pre>	
<pre># actual: ["00", "0x01", "0x02", "0x03", "0x04", "0x05"]</pre>	
<p>Notice that the 0x is omitted when <code>i == 0</code>. The 0x prefix is specified by '#' (and 'x') in the format string. If this is expected behavior, then it sure is surprising and undocumented AFAIK. <i>goes back to double check</i></p>	
<p>Behavior on 2.4.0 confirmed on 2.3.3 as well.</p>	

History

#1 - 01/24/2017 11:23 PM - zenspider (Ryan Davis)

FWIW, this is the exact same behavior in C since we're just calling through to `snprintf` for these format specifiers.

```
printf("%#0.2x %#04x %#0.2x %#04x\n", 0, 0, 1, 1)
```

#2 - 01/25/2017 01:24 AM - shyouhei (Shyouhei Urabe)

- Description updated

#3 - 01/26/2017 02:03 PM - znz (Kazuhiro NISHIYAMA)

Old ruby's results are 0x00.

It changed between 1.9.0-0 and 1.9.0-5.

```
% rbenv each ruby -ve 'puts "%#0.2x" % 0'
ruby 1.8.6 (2007-03-13 patchlevel 0) [x86_64-linux]
0x00
ruby 1.8.6 (2010-09-02 patchlevel 420) [x86_64-linux]
0x00
ruby 1.8.7 (2008-05-31 patchlevel 0) [x86_64-linux]
0x00
ruby 1.8.7 (2013-12-22 patchlevel 375) [x86_64-linux]
0x00
ruby 1.9.0 (2007-12-25 revision 14709) [x86_64-linux]
0x00
ruby 1.9.0 (2008-10-04 revision 19669) [x86_64-linux]
00
ruby 1.9.1p0 (2009-01-30 revision 21907) [x86_64-linux]
00
ruby 1.9.1p431 (2011-02-18 revision 30908) [x86_64-linux]
00
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
00
ruby 1.9.2p330 (2014-08-07 revision 47094) [x86_64-linux]
00
ruby 1.9.3p0 (2011-10-30 revision 33570) [x86_64-linux]
00
ruby 1.9.3p551 (2014-11-13 revision 48407) [x86_64-linux]
00
ruby 2.0.0p0 (2013-02-24 revision 39474) [x86_64-linux]
00
ruby 2.0.0p648 (2015-12-16 revision 53162) [x86_64-linux]
00
ruby 2.1.0p0 (2013-12-25 revision 44422) [x86_64-linux]
00
```

```
ruby 2.1.10p492 (2016-04-01 revision 54464) [x86_64-linux]
00
ruby 2.2.0p0 (2014-12-25 revision 49005) [x86_64-linux]
00
ruby 2.2.5p319 (2016-04-26 revision 54774) [x86_64-linux]
00
ruby 2.3.0p0 (2015-12-25 revision 53290) [x86_64-linux]
00
ruby 2.3.3p222 (2016-11-21 revision 56859) [x86_64-linux]
00
ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-linux]
00
```

#4 - 09/11/2020 07:44 PM - jeremyevans0 (Jeremy Evans)

- Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)
- ruby -v deleted (ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-linux])
- Tracker changed from Bug to Feature

sprintf has been documented that the prefix is only added on non-zero since [67151cd7af01f97524f0a14cf5abd9567f856155](https://github.com/ruby/ruby/commit/67151cd7af01f97524f0a14cf5abd9567f856155) in 2008, so this isn't a bug, but a feature request. Adding the prefix even on zero seems more reasonable to me, but I'm not sure the backwards compatibility loss is worth it. It's also questionable how this would interact with the space, plus, and minus flags. Here's a commit that implements it and passes the sprintf tests/specs: <https://github.com/jeremyevans/ruby/commit/2adaaa87f2cf40c21f2560b34afdce0ef45d321>