

## Foreman - Bug #10279

### erroneous collection of interface information in YAML output

04/27/2015 10:14 AM - Johannes Spanier

<b>Status:</b> Resolved	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b> Network	
<b>Target version:</b>	
<b>Difficulty:</b>	<b>Fixed in Releases:</b>
<b>Triaged:</b>	<b>Found in Releases:</b> 1.7.4
<b>Bugzilla link:</b>	<b>Red Hat JIRA:</b>
<b>Pull request:</b>	
<b>Description</b>	
in the presence multiple bridges AND vlans on a node foreman reports erroneous interface information in ENC output.  additional non-existent vlan sub-interfaces and duplicate entries are reported and additionally YAML generation takes a LONG time compared to nodes without vlans and bridges. facts are correct however!  an example erroneous YAML ENC output is attached as well as an "ip link show" and "brctl show" from the same node	
<b>Related issues:</b>	
Related to Foreman - Bug #10505: Missing values in node.rb output	<b>Resolved</b> 05/13/2015
Related to Foreman - Feature #7463: Support for bridge devices	<b>Closed</b> 09/16/2014

#### History

##### #1 - 04/27/2015 10:23 AM - Dominic Cleal

- Category set to Network

Could you also please attach the output of "facter" (or "facter --json") and the version of Facter installed on the host?

##### #2 - 04/27/2015 10:47 AM - Johannes Spanier

- File tionso-facter.json added

```
[root@tionso ~]# facter -v
2.4.1
```

##### #3 - 04/28/2015 02:48 AM - Marek Hulán

For some reason the interfaces are created repeatedly, probably with every fact upload. As a temporary workaround you can disable interfaces creation by setting `ignore_puppet_facts_for_provisioning` to false and remove all duplicated interfaces.

##### #4 - 04/28/2015 05:04 AM - Johannes Spanier

Thanks Marek for the proposed workaround, I will do that until the bug is fixed. Although I think you meant set `"ignore_puppet_facts_for_provisioning = TRUE"`. How to I go about removing the duplicated interfaces? There are a LOT of them and I do not want to do this in the WebUI for about 10 machines where this happened. Too much clicking ;-).

There remains the issue with the non-existent interfaces. In the provided example YAML output there are interfaces named `internal00.100`, `internal00.300` and `internal.700` that simply do not exist on the node at all, also the interface that is duplicated many times (`ovirtmgmt.300`) does not even exist on the node.

I looks like additional interfaces are erroneously reported for every possible mathematical combination of bridge names and vlan numbers on the node.

##### #5 - 04/28/2015 06:48 AM - Marek Hulán

Right, it should be TRUE, I'm glad you read my instructions but carefully :-). If you know ruby you can use rails console to delete interfaces from all hosts or you can script something using hammer or foreman API. In rails console a command might look like similar to this

```
Nic::Base.where(:identifier => ['internal00.100', ' internal00.300', ...]).map(&:destroy)
```

Make sure to backup your DB before any similar changes.

I have no idea where other interfaces come from. We use fact interfaces to find all interfaces on host, which in your case is this list

```
"_vdsmdummy_,bond0,eth0,eth1,eth1_100,eth1_300,eth1_700,eth2,internal00,lo,ovirtmgmt,public"
```

To reproduce it, an exact version of your Foreman might help.

#### #6 - 04/28/2015 07:09 AM - Johannes Spanier

FOREMAN: 1.7.4  
RUBY: ruby 1.9.3p484 (2013-11-22 revision 43786) [x86\_64-linux]  
PUPPET: 3.7.5  
OS: Ubuntu 14.04.02 LTS

Node:  
PUPPET: 3.7.5

#### #7 - 05/03/2015 11:06 AM - Johannes Spanier

OK. So I set "ignore\_puppet\_facts\_for\_provisioning = TRUE" and deleted all extraneous interface information. But now, after a few days, the interfaces are back in the ENC output! Probably accumulated with every facter run.

#### #8 - 05/03/2015 11:09 AM - Johannes Spanier

Also could some admin kindly set the "Found in release:" field to 1.7.4.

#### #9 - 05/05/2015 03:44 AM - Dominic Cleal

Ah, using 1.7.4 might explain it, as [#9104](#) made 1.8.0 disable updating of interfaces with that setting, but in 1.7, the setting only controls the updating of the primary.

#### #10 - 05/11/2015 08:19 AM - Marek Hulán

I just tested to import your facts into my Foreman (running develop branch) and I realized what's going on. Some of your interfaces seems to have same MAC but they don't follow the naming conventions on which we rely.

internal00 and public seems to have the same MAC as eth1 but since they don't have dot in their name we don't detect them as VLANs (they are VLANs right? or Alias...). This has impact on next import because then we consider them as physical so we update their attributes (identifiers). This is present in Foreman 1.8 as well and TBH I'm not sure how to properly solve this since we don't have enough information in facter output to detect proper type of interfaces with custom identifier.

One thing that might help is precreating interfaces manually with correct types (making public and internal00 VLANs before first puppet facts upload). Second solution is to upgrade to 1.8 and disable interface auto-updating.

#### #11 - 05/13/2015 08:26 PM - Ben Tullis

- Related to Bug #10505: Missing values in node.rb output added

#### #12 - 05/18/2015 02:53 AM - Johannes Spanier

Some of your interfaces seems to have same MAC but they don't follow the naming conventions on which we rely.

What happens is that I have ONE real interface. I create TWO VLAN interfaces from these. Namely eth1.100 and eth1.700. These have the SAME MAC address. This is the normal situation and the way it is with vlan interfaces... Can't just make up more MACs. eth1.100 and eth1.700 DO follow the usual naming convention for VLAN (sub-)interfaces.

internal00 and public seems to have the same MAC as eth1 but since they don't have dot in their name we don't detect them as VLANs (they are VLANs right?)

NO. They are bridge interfaces. They use eth1.100 and eth1.700 respectively as one of their bridge ports (which both have the same MAC, inherited from eth1). The other bridge Ports are tun/tap Interfaces for the virtual machines. They do not show up here. Same again here regarding the MAC. The MAC of the bridge interface is inherited from the (first?) bridge member.

So in the end eth1, eth1.100, eth1.700, internal00 and public all have the same MAC, which is just the normal operating mode of linux networking.

```
eth1      <- host interface |
+- eth1.100 <- vlan interface | | +- internal00 <- bridge interface |
+- eth1.700 <- vlan interface |
```

+ public <- bridge interface

### #13 - 05/18/2015 02:57 AM - Johannes Spanier

Dang, nice ASCII art with interface associations broken. Should have known... This time with formatting...

```
eth1          <- host interface
|
+ eth1.100    <- vlan interface
| |
| + internal00 <- bridge interface
|
+ eth1.700    <- vlan interface
|
+ public      <- bridge interface
```

### #14 - 05/18/2015 04:51 AM - Marek Hulán

Thanks, the overview helps. VLAN interfaces are OK. The issue in this case I think are bridges because we don't detected that they are bridges. We consider interface being a bridge only if it's name is br# or virbr# ([https://github.com/foreman/foreman/blob/develop/app/services/fact\\_parser.rb#L4](https://github.com/foreman/foreman/blob/develop/app/services/fact_parser.rb#L4)). Since your bridges doesn't have underscore in name they are not even considered as virtuals (bridge would be) so they are considered physical and hence they can be renamed (same MAC as eth1).

### #15 - 10/27/2015 01:00 PM - Marek Hulán

- Related to Feature #7463: Support for bridge devices added

### #16 - 04/13/2016 09:59 AM - Marek Hulán

- Status changed from New to Feedback

There's no way we could identify bridge interface type based only on facter output. In your case I would suggest naming bridges like br1 br2 which should help. If this is not possible you can set rule to ignore these interfaces completely so they wouldn't be created in Foreman. In order to do this, customize Provisioning Setting with name "Ignore interfaces with matching identifier" that was added in Foreman 1.10. In your case, the value would be "[ lo, usb\*, vnet\*, macvtap\*, internal00, public ]" or if internal suffix differs per host, you could specify it as internal\*.

If none of suggested solution works for you, please let us know, I'd reopen the issue.

### #17 - 05/17/2017 11:23 AM - Anonymous

- Status changed from Feedback to Resolved

no reaction, closing

## Files

tionso-net-info.txt	1.99 KB	04/27/2015	Johannes Spanier
tionso.lab.yaml	808 KB	04/27/2015	Johannes Spanier
tionso-facter.json	5.89 KB	04/27/2015	Johannes Spanier