



≡ MENU

10 Linux brctl Command Examples for Ethernet Network Bridge

by RAMESH NATARAJAN on JUNE 19, 2017



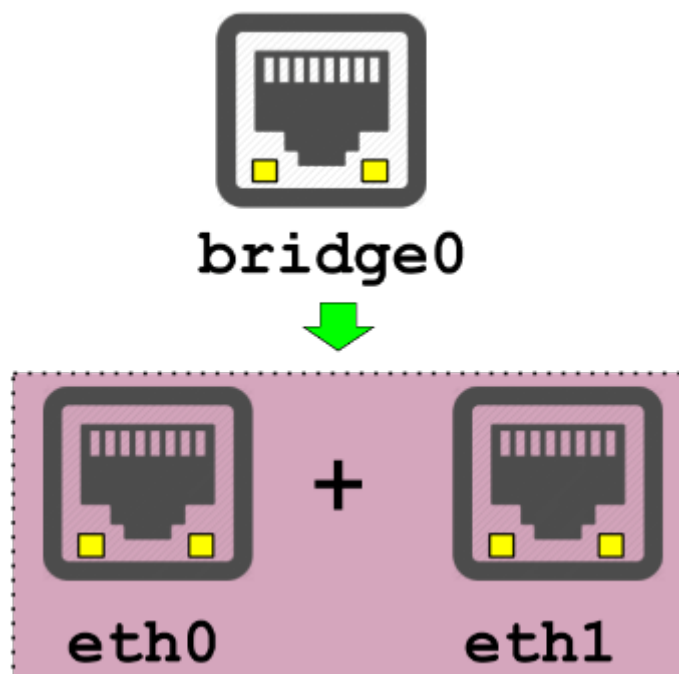
Tweet

brctl stands for Bridge Control.

In Linux, this command is used to create and manipulate ethernet bridge.

This is typically used when you have multiple ethernet networks on your servers, and you want to combine them and present it as one logical network.

For example, if you have eth0 and eth1, you can combine them and present it as just br0, which will inturn use both eth0 and eth1 for network traffic.



In this tutorial, we've covered the following:

1. Create New Ethernet Bridge using addbr
2. Display Available Ethernet Bridge using show
3. Delete Existing Ethernet Bridge using delbr
4. Add an Interface to Existing Bridge
5. Add Multiple Interfaces to Existing Bridge
6. Tracking MAC address of a Bridge
7. Set Ageing Time for Mac Address on a Bridge

8. Setup Spanning Tree on Ethernet Bridge
9. Display STP Parameter Values of a Bridge
10. Change Bridge Parameters Values

1. Create New Ethernet Bridge using addbr

Using `brctl addbr`, we can create a new ethernet bridge.

In the following example, we have created three ethernet bridges on this server: `dev`, `stage` and `prod`.

```
brctl addbr dev

brctl addbr stage

brctl addbr prod
```

Please note that at this stage, this is just an empty ethernet bridge which doesn't have any other ethernet network on it.

Please note that this `brctl` is bit different than the [Linux EtherChannel NIC bonding](#) that we discussed earlier.

2. Display Available Ethernet Bridge using show

Using `brctl show`, we can view all available ethernet bridges on your server.

The following example will display all the current instances of the three bridges that we just created.

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
dev               8000.00000000000000      no
prod             8000.00000000000000      no
stage            8000.00000000000000      no
```

If you notices the last column “interfaces” has nothing on it. This means that none of this bridge currently has any ethernet device associated with it yet.

On a related note: [9 Linux ethtool Examples to Manipulate Ethernet Card](https://www.thegeekstuff.com/2017/06/brctl-bridge/)

3. Delete Existing Ethernet Bridge using delbr

Using brctl delbr, we can delete an existing ethernet bridge.

The following example will delete the “stage” instance of the ethernet bridge.

```
brctl delbr stage
```

As you see from the show command, we don’t see the “stage” ethernet bridge anymore.

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
dev               8000.00000000000000      no
prod             8000.00000000000000      no
```

Please note that when a bridge is active (i.e up), you can remove it. You need to first bring down the bridge and then delete it.

Also, please note that earlier brctl command was called as brcfg.

4. Add an Interface to Existing Bridge

Using brctl addif, we can add an interface to an existing ethernet bridge.

The following example will add the ethernet network eth0 to the bridge “dev” that we just created.

Warning: Don’t try this on any critical machines, as you might loose network connection to that machine when something goes wrong. Try this only on a test instance where you have console access.

```
brctl addif dev eth0
```

In the above:

- This will really make the eth0 as a port of the bridge “dev”.
- So, all the frames that arrives on eth0 will be processed as if it is really arriving at the bridge.
- Also, when the frames are sent-out on “dev” bridge, it will use eth0. When multiple interfaces are part of “dev” bridge, then eth0 will be a potential candidate to sent-out the outgoing frames from the bridge.

After adding the eth0 to “dev” bridge, the brctl show command will display the following.

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
dev              8000.0050568954bc       no             eth0
prod            8000.00000000000000      no
```

As you see from the above output, the last column “interfaces” now has eth0 for “dev” bridge.

Note: If something goes wrong while adding eth0 to the “dev” bridge, execute the following command from the console to delete the dev bridge.

```
# brctl delbr dev
```

Also, note that if you try to add the loopback interface to the bridge, you’ll get the following invalid argument command. # brctl addif dev lo
can’t add lo to bridge dev: Invalid argument

As you can imagine, you can’t add an interface that doesn’t exist on the system to the bridge.

```
# brctl addif dev eth2
interface eth2 does not exist!
```

5. Add Multiple Interfaces to Existing Bridge

In the above example, we added only one ethernet device (eth0) to the bridge.

But, the whole idea of using a bridge is to add more interfaces to the bridge.

In the following example, we are adding both eth0 and eth1 to the bridge “dev”.

```
# brctl addif dev eth0 eth1
```

As you see from the following output, for the dev bridge we see two lines. In the last column “interfaces”, we see both eth0 and eth1 for “dev” ethernet bridge.

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
dev              8000.0050568954bc       no             eth0
                                                         eth1
prod            8000.00000000000000      no
```

Note: If a particular ethernet interface is already part of a bridge, then you can’t add it to another bridge.

Basically, one network can be part of only one bridge. If not, we’ll get the following error message when trying to add it to another bridge.

```
# brctl addif prod eth1
device eth1 is already a member of a bridge; can't enslave it to bridge p
```

6. Tracking MAC address of a Bridge

Using brctl showmacs, we can see all the learned MAC addresses of a bridge.

In the following example, as of now, the following are the two mac addresses that connected to the ethernet bridge “prod”. This data will keep changing depending on the current status of what is connected to the bridge.

```
# brctl showmacs prod
```

port	no	mac addr	is local?	ageing timer
1		00:50:56:89:54:bc	yes	0.00
2		00:60:50:46:bc:40	no	0.00

7. Set Ageing Time for Mac Address on a Bridge

In the following example, we are setting the mac address ageing time to 120 seconds on “dev” ethernet bridge.

```
brctl setaging dev 120
```

In the above:

- dev is the name of the bridge where we are setting this value
- 120 is seconds
- So, if a frame is not seen for “dev” bridge in 120 seconds, then “dev” bridge will delete the mac address from the forwarding database.

8. Setup Spanning Tree on Ethernet Bridge

Using brctl stp, we can setup spanning tree on ethernet bridge.

But default, spanning tree is not enabled when we create a bridge.

Spanning tree is helpful when you have multiple bridges on your network, and they can all collaborate to find the shortest path between two ethernet.

In the following example, we are turning on the spanning tree on “dev” ethernet bridge.

We can use either on or yes to enable spanning tree. So, both of the following command will do exactly the same.

```
brctl stp dev on

brctl stp dev yes
```

As you see from the following show output, the values in the “STP enabled” column for “dev” bridge is now “yes”.

```
# brctl show
bridge name      bridge id          STP enabled      interfaces
dev              8000.000000000000  yes
prod            8000.000000000000  no
```

To turn off spanning tree on your ethernet bridge, do the following:

```
# brctl stp dev off

# brctl show
bridge name      bridge id          STP enabled      interfaces
dev              8000.000000000000  no
prod            8000.000000000000  no
```

9. Display STP Parameter Values of a Bridge

STP stands for Spanning Tree Protocol.

Once you’ve enabled stp on your bridge, you can use showstp as shown below to view all the stp parameter values of your bridge.

The following will display the stp parameter and its current value for “dev” bridge.

```
# brctl showstp dev
dev
bridge id          000a.000000000000
designated root     000a.000000000000
root port          0          path cost          0
```

max age	19.99	bridge max age	19.99
hello time	1.99	bridge hello time	1.99
forward delay	14.99	bridge forward delay	14.99
ageing time	299.95		
hello timer	0.00	tcn timer	0.00
topology change timer	0.00	gc timer	0.00
hash elasticity	4	hash max	512
mc last member count	2	mc init query count	2
mc router	1	mc snooping	1
mc last member timer	0.99	mc membership timer	259.96
mc querier timer	254.96	mc query interval	124.98
mc response interval	9.99	mc init query interval	31.24
flags			

10. Change Bridge Parameters Values

The spanning tree parameter default values for a specific bridge can be changed. You can also change other parameter values of the bridge that you created.

All of the following set commands mentioned in the table below starts with brctl. For example:

```
brctl setageing dev 100
```

The following table displays available set commands for brctl. The time value is in seconds.

brctl command	Description
setageing bridge time	Set ageing time
setbridgeprio bridge prio	Set bridge priority (between 0 and 65535)
setfd bridge time	Set bridge forward delay
sethello bridge time	Set hello time
setmaxage bridge time	Set max message age
setgcint bridge time	Set garbage collection interval in seconds
sethashel bridge int	Set hash elasticity

sethashmax bridge int	Set hash max
setmclmc bridge int	Set multicast last member count
setmcrouter bridge int	Set multicast router
setmcsnoop bridge int	Set multicast snooping
setmcsqc bridge int	Set multicast startup query count
setmclmi bridge time	Set multicast last member interval
setmcmi bridge time	Set multicast membership interval
setmcqpi bridge time	Set multicast querier interval
setmcqi bridge time	Set multicast query interval
setmcqri bridge time	Set multicast query response interval
setmcqri bridge time	Set multicast startup query interval
setpathcost bridge port cost	Set path cost
setportprio bridge port prio	Set port priority (between 0 and 255)
setportmcrouter bridge port int	Set port multicast router
sethashel bridge int	Set hash elasticity value

Tweet



Add your comment

If you enjoyed this article, you might also like..

1. [50 Linux Sysadmin Tutorials](#)
 2. [50 Most Frequently Used Linux Commands \(With Examples\)](#)
 3. [Top 25 Best Linux Performance Monitoring and Debugging Tools](#)
 4. [Mommy, I found it! – 15 Practical Linux Find Command Examples](#)
 5. [Linux 101 Hacks 2nd Edition eBook](#)
Free
- [Awk Introduction – 7 Awk Print Examples](#)
 - [Advanced Sed Substitution Examples](#)
 - [8 Essential Vim Editor Navigation Fundamentals](#)
 - [25 Most Frequently Used Linux IPTables Rules Examples](#)
 - [Turbocharge PuTTY with 12 Powerful Add-Ons](#)





Tagged as: [CentOS brctl Command](#), [CentOS brctl Enable STP](#), [CentOS Ethernet Bridge](#), [Debian brctl Command](#), [Linux brctl Disable STP](#), [Linux brctl NIC Bridge](#), [Linux brctl STP Example](#), [Linux Network Bridge Examples](#), [RedHat Ethernet Bridge](#), [Ubuntu brctl Command](#)

Comments on this entry are closed.

Pete June 20, 2017, 4:18 am

I've never used a bridge to bond interfaces, but have been using them for years to provide networking to virtual machines.

LINK

Magnetix June 23, 2017, 3:38 am

Bridge vs bonding with link aggeration? In this sbridge setup there's no lacp mode setup needed on the switch right? What about performance?

LINK

Next post: [15 Git Branch Command Examples to Create and Manage Branches](#)

Previous post: [How to Fix DELL PowerEdge Memory/battery problems were detected Error](#)

[RSS](#) | [Email](#) | [Twitter](#) | [Facebook](#) | [Google+](#)

EBOOKS

Free

[Linux 101 Hacks 2nd Edition eBook](#) - Practical Examples to Build a Strong Foundation in Linux

[Bash 101 Hacks eBook](#) - Take Control of Your Bash Command Line and Shell Scripting

[Sed and Awk 101 Hacks eBook](#) - Enhance Your UNIX / Linux Life with Sed and Awk

[Vim 101 Hacks eBook](#) - Practical Examples for Becoming Fast and Productive in Vim Editor

[Nagios Core 3 eBook](#) - Monitor Everything, Be Proactive, and Sleep Well

POPULAR POSTS

[15 Essential Accessories for Your Nikon or Canon DSLR Camera](#)

[12 Amazing and Essential Linux Books To Enrich Your Brain and Library](#)

[50 UNIX / Linux Sysadmin Tutorials](#)

[50 Most Frequently Used UNIX / Linux Commands \(With Examples\)](#)

[How To Be Productive and Get Things Done Using GTD](#)

[30 Things To Do When you are Bored and have a Computer](#)

[Linux Directory Structure \(File System Structure\) Explained with Examples](#)

[Linux Crontab: 15 Awesome Cron Job Examples](#)

[Get a Grip on the Grep! – 15 Practical Grep Command Examples](#)

[Unix LS Command: 15 Practical Examples](#)

[15 Examples To Master Linux Command Line History](#)

[Top 10 Open Source Bug Tracking System](#)

[Vi and Vim Macro Tutorial: How To Record and Play](#)

[Mommy, I found it! -- 15 Practical Linux Find Command Examples](#)

[15 Awesome Gmail Tips and Tricks](#)

[15 Awesome Google Search Tips and Tricks](#)

[RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams](#)

[Can You Top This? 15 Practical Linux Top Command Examples](#)

[Top 5 Best System Monitoring Tools](#)

[Top 5 Best Linux OS Distributions](#)

[How To Monitor Remote Linux Host using Nagios 3.0](#)

[Awk Introduction Tutorial – 7 Awk Print Examples](#)

[How to Backup Linux? 15 rsync Command Examples](#)

[The Ultimate Wget Download Guide With 15 Awesome Examples](#)

[Top 5 Best Linux Text Editors](#)

[Packet Analyzer: 15 TCPDUMP Command Examples](#)

[The Ultimate Bash Array Tutorial with 15 Examples](#)

[3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id](#)

[Unix Sed Tutorial: Advanced Sed Substitution Examples](#)

[UNIX / Linux: 10 Netstat Command Examples](#)

[The Ultimate Guide for Creating Strong Passwords](#)

[6 Steps to Secure Your Home Wireless Network](#)

[Turbocharge PuTTY with 12 Powerful Add-Ons](#)

CATEGORIES

[Linux Tutorials](#)

[Vim Editor](#)

[Sed Scripting](#)

[Awk Scripting](#)

[Bash Shell Scripting](#)

[Nagios Monitoring](#)

[OpenSSH](#)

[IPTables Firewall](#)

[Apache Web Server](#)

[MySQL Database](#)

[Perl Programming](#)

[Google Tutorials](#)

[Ubuntu Tutorials](#)[PostgreSQL DB](#)[Hello World Examples](#)[C Programming](#)[C++ Programming](#)[DELL Server Tutorials](#)[Oracle Database](#)[VMware Tutorials](#)

ABOUT THE GEEK STUFF



My name
is
Ramesh

Natarajan. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about [Ramesh Natarajan](#) and the blog.

CONTACT US

Email Me : Use this [Contact Form](#) to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

[Follow us on Google+](#)

[Follow us on Twitter](#)

[Become a fan on Facebook](#)

SUPPORT US

Support this blog by purchasing one of my ebooks.

[Bash 101 Hacks eBook](#)

[Sed and Awk 101 Hacks eBook](#)

[Vim 101 Hacks eBook](#)

[Nagios Core 3 eBook](#)