Contents

Introduction

This page explains proper command line configuration of VLANs on the router's internal switch (internal and external ports), usually after using the GUI (DD-WRT *Setup->Switch Config* (formerly *VLANs*) tab.

These instructions are necessary:

- Only for Broadcom-based routers
- For a variety of "basic" setups that are not properly handled in NVRAM by the GUI
- For advanced setups that the GUI cannot handle at all.
- **NOTE:** not all Broadcom routers are capable of VLAN configuration. E.g. the WRT320N does not use VLANs for the switch, as they have two independent Ethernet phys, so the switch is non-configurable.

Important: Before following the instructions below, '*first* use the DD-WRT Setup->VLANs page to make your initial configuration changes, if possible.

- If the only changes involve the checkboxes at the bottom of the page (e.g. Auto-Negatiate...Enabled), nothing else is needed.
- Otherwise, by setting it up first on the VLANs page, most of your work is already done.

- Some changes made on the VLANs page do require additional manual configuration

Note: The WLAN is a separate interface from the switch and it does not support VLANs, although it can create virtual interfaces which are similar in some aspects. The WLAN interface is bridged to the LAN VLAN by default. It is possible to bridge VLAN's to different physical or virtual wireless interfaces using Multiple WLANs.

Visual mapping chart

Default internal device network (vlan0, vlan1, eth0, eth1, eth2 etc)

Preserve Settings Before you start

Telnet or SSH to the router (do **not** use the *Commands* GUI!) and run these commands to check the nvram variables for your default vlan configuration before you start so that you know the default configuration if you need to revert your changes without hard resetting.

```
nvram show | grep vlan.*ports | sort
nvram show | grep port.*vlans | sort
nvram show | grep vlan.*hwname | sort
```

These commands will output a bunch of lines that have a name, an equal sign (=), and then some more text which is the *value* of the variable like so:

```
vlan1ports=1 2 5*
vlan2ports=0 5u
...
```

Save that list!

If you run into trouble, to revert your changes you will have to modify each line in the list and paste it back as follows: put **nvram set** in front of each line, and encapsulate each *value* with quotes:

```
nvram set vlan1ports="1 2 5*"
nvram set vlan2ports="0 5u"
...
```

Then paste that into the Telnet or SSH command line (to reset those values), and save them permanently:

```
nvram set vlan1ports="1 2 5*"
nvram set vlan2ports="0 5u"
...
nvram commit
```

Explaining the connection between Hardware, VLANs, and NVRAM Variables

You've saved the existing NVRAM information already, as described above. Right? You DO need that info!

Visible Hardware, Internal Ports, and VLANs

First, let's describe how your router hardware is mapped to ports, bridges, ethernet devices, etc. This is also described in Default_Configuration_Overview#Illustration_of_port_and_vLAN_mappings but lots of people get confused... so we'll make another attempt here.

You have the advantage of the actual NVRAM data for your router, so that helps.

Switch

Inside the router is a (gigabit or 100mbit) switch with six ports. Five of the ports map to five physical ethernet ports on the router box. One port is internally connected to the CPU.

CPU Port
: First, one of the ports in one of the "vlan*ports" values has an asterisk. That port is either 5 or 8. That's the port number for the invisible backbone trunk between the switch and the *eth0 device* that the CPU talks to. We often think of that port as the "CPU" port. We'll use 5(8) sometimes in the documentation because many popular devices use port 5, but others use 8.

Other Ports
: This is tricky because it varies by manufacturer and router switch speed. Here's a general rule of thumb that mostly applies but don't be surprised if your situation is different, particularly if you have an unusual router brand. *At the very least, take note whether or not you have vlan0! This is crucial.*
  - ◊ *If you have vlan0:* probably your WAN port is port 0, and the other ports are 1-4. But it could be the other way around (WAN is internal port 4, other ports are 0-3.) vlan0 is your *default internal LAN;* vlan1 is your *default WAN.*
  - ◊ *If you have no vLAN 0:* probably your WAN port is port 0, and the others are 1-4. vlan1 is your *default internal LAN;* vlan2 is your *default WAN.*

Some examples (no we don't want to list every device here):

```
WRT54G v2/v3, WRT54GS v1/v2 : CPU port is 5, WAN port is 0 (vlan0 and vlan1?)
WRT54G v4, WRT54GS v3, WRT54GL v1.1 : CPU port is 5, WAN port is 4 (vlan0 and vlan1?)
Netgear R7000 gigabit : CPU is 5, WAN is 0 (vlan1 and vlan2)
Most non-Netgear gigabit : CPU 8, WAN 0
```

Network devices at the CPU end

eth0 device
: eth0 is mentioned above. This is what the actual CPU sees when it comes to the switch CPU port (5 or 8).

eth1 device
: eth1 connects to the wifi devices. Note that this is not part of the switch, nor do they have port numbers. Below we explain the default bridge (br0) that connects eth1 to the default internal LAN vlan.

VLANs

VLANs are virtual constructs. They do have meaning, but basically they are a flexible way to accomplish a few simple things:

- One or more physical switch ports can be linked to a given vlan. They are isolated from other vlans on the same switch unless bridges or other methods are used to allow a connection.
- vlan traffic on one router can be tagged and travel over a single trunk to another router -- and the same vlans are available there. A simple way for traffic to be appropriately connected.

DD-WRT has two default vlans that are mapped to the hardware:

Default internal LAN vLAN
: By default your first vLAN (1 or 0) is the internal LAN. By default a bridge (br0) is automagically created to link the WiFi devices (eth1) to this vlan. To break that you may have to do significant custom configuration that may well vary by router brand/etc. This is the default path for untagged packets.

Default WAN vLAN
: By default, the second vlan (2 or 1) is the WAN connection. By default packets on this vLAN are also untagged (marked with a "u" next to the CPU port in nvram... see below.)

A few important notes

- The CPU port must be included in any vlan if that vlan is to be seen by the cpu. Which is basically all the time.
- We urge that the default WAN vlan (2 or 1) be reserved for that purpose. If you don't need a WAN function, use other vlan numbers because DD-WRT probably assumes this purpose in various places.
- The same is a safe assumption for the default LAN vlan (1 or 0) as well.
- We've spent a lot of effort on these low-numbered vlan ID's. The DD-WRT GUI allows you to easily define much higher vlan numbers. And the newest releases support vlan id's as high as 4096, but you have to use those via direct NVRAM configuration as shown below, rather than in the GUI.

NOTE: As of revs 35079-35165 (depending on the kernel), 4096 VLAN ID support is now built in, for Broadcom-based devices. [This link] is a good starting place.]]

Summary

At this point, we have not explained anything about those interesting NVRAM variables. But you should be able to think about your hardware ports and internal port numbering (and vlan numbering) schemes. And you can now plan how you actually want to configure your vlans for the router.

What you get to play with:

The CPU

- Connects to eth0 which trunks to the internal switch "CPU" port (5 or 8)
- Connects to eth1, which trunks to the WiFi devices

One default bridge

- br0 links eth1 (wifi devices) and the default LAN vlan (vlan1 or vlan0), all untagged

Two default vlans

- Default LAN vlan (vlan1 or vlan0), by default connecting to all LAN hardware ports (0-3 or 1-4)
- Default WAN vlan (vlan2 or vlan1), by default connecting to the WAN hardware port (4 or 0)

Five switch ports

- 100mbit or gigabit
- 0-4 or 4-0... it varies by brand and model

Available vlan numbers
: could be 0-15, 1-16 or even 1-4096 depending...

Whew! OK, now we're ready to tackle the NVRAM variables.

NVRAM variables--defining the switch and VLANS

As noted above, the DD-WRT GUI is able to handle properly setting some of these values, for basic configurations. However, if you create new VLANs or reconfigure in any of a number of ways, or make non-default use of these (eg set up a new bridge, etc), you will likely find that things don't fully work correctly unless you ensure all of these NVRAM values are correctly set.

We don't mean to scare you. There are only three sets of VLAN/Port-related NVRAM variables, and they truly are not that hard to set correctly.

Here they are, in quick summary:

vlan#ports
: Each vlan is assigned one or more ports plus the CPU port. Default vlan and tagging also specified.

port#vlans
: Each port is assigned to one vlan (CPU to all vlans). Plus other per-port settings for speed, duplex, etc.

vlan#hwname
: Activates a vlan, ensuring it is properly initialized.

Required Manual NVRAM Changes

VERY IMPORTANT For a fully working VLAN Config, you must make the following manual changes (if needed) because the current GUI doesn't do it! Each of these NVRAM variables is explained below.

- *Add* vlan**N**hwname for every VLAN other than the defaults (which already should have these values in your saved entries.) Each one has a value of "et0". ie (if you're adding vlan 103)

```
nvram set vlan103hwname=et0
```

- *Add* a correct vlan**N**ports value for each new VLAN, and correct any existing ones to remove/change ports no longer in those VLANs. Each entry contains the ports in that VLAN, plus the CPU port (5 or 8). Also add tagging (if needed) for the new VLANs. For example, if port 4 is now in VLAN 103 instead of VLAN 1, you might need:

```
nvram set vlan1ports="1 2 3 5*"
nvram set vlan103ports="4 5"
```

- *Add* a correct port**N**vlans value, if you're using vlans outside of GUI capabilities (eg more than 16). In our sample case with port 4 now in VLAN 103, we might use:

```
nvram set port4vlans="103 18 19 21"
nvram set port5vlans="1 2 103 16"
```

- *Verify* your changes and save
    - You have made the changes but they are not yet active nor saved!
    - Use the above nvram show commands to view your updated settings. Are you sure they're correct? If so, move ahead. Just know that once you commit, your router will follow the new rules, for better or worse
    - Commit the changes.

```
nvram commit
```

NVRAM vlan#ports

The vlan#ports variables are per-VLAN. They define which ports are assigned to each VLAN, plus other VLAN-specific attributes (default VLAN, 802.1q tagging, and forcing the WAN VLAN to be untagged.)

As noted above,

- You must know your CPU internal port number (5 or 8).
- Be aware of your default LAN and WAN vlans (1 and 2 or 0 and 1)

With 100 mbps ports, the vlan#ports variables typically appear like this:

```
nvram show | grep vlan.*ports | sort
vlan0ports=1 2 3 4 5*
vlan1ports=0 5
```

With gigabit ports, the vlan#ports variables may instead appear like this:

```
nvram show | grep vlan.*ports | sort
vlan1ports=1 2 3 4 5*
vlan2ports=0 5u
```

or

```
nvram show | grep vlan.*ports | sort
vlan1ports=1 2 3 4 8*
vlan2ports=0 8
```

As noted above, there are default VLANs for the LAN and WAN. They are either

- vlan0ports and vlan1ports

or

- vlan1ports and vlan2ports (there won't be a vlan0)

Other VLAN attributes

These are suffixes for the final CPU port number.

"t"
: tag with 802.1q vlan header *(e.g. for trunking). Can be added to one or more ports in a vlan.*

"u"
: used on WAN vlan, to ensure WAN traffic is *untagged*.

"*"
: used on default LAN VLAN, untagged incoming traffic goes here.

See below and References for more information.

NVRAM port#vlans

The port#vlans variables define the vlan(s) for each port, and other per-port attributes.

Note that each port is normally only in one vlan, except that the CPU port is connected to all vlans. (There are unusual cases where a port can be placed in multiple vlans.) ALSO note that *for this NVRAM variable, the CPU port is always port 5, never port 8.*

By default, then WAN port is connected to the default WAN VLAN (1 or 2), and all other ports are connected to the default LAN VLAN (0 or 1).

Normally, every port also has one or more additional hardware attributes assigned.

Port Attributes

```
0 or 1 to 15, or even 4096 This is the VLAN for the port. NOTE that due to the other attributes b
16 Tagged
17 NOT Auto-Negotiate. (If #17 is off, you'll get 18, 19, and 21 on a gigabit switch)
18 = NOT forced-100 MBit (either Auto-Negotiate is used, or some other speed is forced)
19 = NOT Full-Duplex (either Auto-Negotiate is used, or Half-duplex is forced)
20 = NOT Enabled (ie. disabled)
21 = NOT forced-Gigabit (either Auto-Negotiate is used, or some other speed is forced)
```

Some sample attribute settings:

```
18 19 21 = Auto-negotiate Gigabit
18 19 = Auto-negotiate 100 MBit
17 19 = Force 100 MBit, half-duplex
17     = Force 10 MBit, full duplex (on a 100 MBit router)
17 18 = Force Gigabit full duplex (on a gigabit router)
16     = For some reason, the CPU port is generally given this "tagged" attribute
```

As noted in the quick example above, if we're assigning port 4 to VLAN 103, we would need to set two port**N**vlans variables. On a Gigabit router (with WAN port 0) it would look like this:

```
nvram set port5vlans="1 2 103 16"
nvram set port4vlans="103 18 19 21"
```

NVRAM vlan#hwname

This ensures each vlan is initialized properly. There must be one for each vlan. **The gui never adds these variables as of today.** *You must do this* to ensure DD-WRT features all work correctly.

So far, the correct value is always "et0"

For instance, if you were enabling VLAN 103 you would set:

```
nvram set vlan103hwname=et0
```

Finishing up

After you've set all your nvram variables you need to make sure to commit them so that they will be saved! Then you need to reboot to have your new configuration take effect.

```
nvram commit
reboot
```

Usage examples

Reminder:' Available VLAN's are 0-15 on 100mbit models and 1-15 on gigabit models, except that as noted above newer Broadcom gigabit firmware supports VLANs up to 4096.

Second WAN port

You can move ports to the WAN VLAN to make them act as WAN ports residing outside the router's NAT or to just replace a damaged WAN port. Devices attached to these ports will request their IP addresses from the ISP instead of getting a private IP address from your router. One might want to have such behavior when having unsorted firewall issues with a SIP operator. This example assumes a 100mbit router withvlan0 for LAN and vlan1 for WAN, and adds internal port 4 to vlan1:

```
nvram set vlan0ports="1 2 3 5*"
nvram set vlan1ports="0 4 5"
nvram set port4vlans="4 17 18"
nvram commit
reboot
```

Disable LAN ports

Another use is to disable most of the wired LAN ports, leaving only the WAN port that connects you to the rest of the world. You can use your router as a WiFi access point and no one can plug in a network cable to get around your Radius authentication (requested in the German forum). NOTE that in case of trouble you probably want to keep one LAN port available, as in this example:

Example to only enable LAN external port 1 (internal port 3) (tested with a WRT54Gv5):

```
nvram set vlan0ports="3 5*"
nvram set vlan1ports="4 5"
nvram unset port0vlans
nvram unset port1vlans
nvram unset port2vlans
nvram commit
reboot
```

Separate LAN ports (into another subnet)

You can create new VLAN's to separate LAN ports into multiple subnets. After you have them separated you can use iptables to restrict access between the VLAN's/subnets. Here on a 100 mbit router we'll move port 4 to its own vlan3:

```
nvram set vlan0ports="1 2 3 5*"
nvram set vlan1ports="0 5"
nvram set vlan3ports="4 5"
nvram set vlan3hwname=et0
nvram set port4vlans="3 18 19"
nvram set port5vlans="0 1 3 16"
nvram commit
reboot
```

802.1q VLAN trunk

VLAN trunking allows you to connect multiple VLAN's between two routers without having to use a cable for each VLAN. For instance, if you had VLAN 1 and VLAN 3 on two different devices then you can create a trunk port on both devices to tag the Ethernet frames with an 802.1q header. Then devices in VLAN 1 on router 1 can communicate with devices in VLAN 1 on router 2 and devices in VLAN 3 on router 1 can communicate with devices in VLAN 3 on router 2.

Note: Broadcom's 100mbit switches use VLAN 0 as the default LAN VLAN but 802.1q specifies that a tag of "0" means that the frame doesn't belong to any VLAN. This is the reason gigabit models use VLAN 1 for their LAN and you should avoid using VLAN 0 in a trunk because of this.

For every VLAN that you want trunked you must put the trunk port into the VLAN and add a "t" after the port number to indicate that it will be tagged.

Here is an example of how to put three LAN ports (3, 2, and 1) into separate VLAN's (1, 3 and 4) and use the fourth LAN port (4) as a trunk on a gigabit model. VLAN 2 is still the WAN default VLAN; port 0 is the WAN port here. The commands that are commented out should already be set by default.

```
nvram set vlan1ports="3 4t 8*"
#nvram set vlan2ports="0 8"
nvram set vlan3ports="2 4t 8"
nvram set vlan4ports="1 4t 8"
#nvram set port0vlans="2 18 19"
nvram set port1vlans="4 18 19"
nvram set port2vlans="3 18 19"
#nvram set port3vlans="1 18 19"
nvram set port4vlans="1 3 4 16 18 19"
nvram set port5vlans="1 2 3 4 16"
#nvram set vlan1hwname=et0
#nvram set vlan2hwname=et0
nvram set vlan3hwname=et0
nvram set vlan4hwname=et0
nvram commit
reboot
```

It is also possible to have only the trunk port in a VLAN. This is useful for connecting VAP's on different devices together by bridging an 'empty' VLAN with the VAP.

```
nvram set vlan1ports="1 2 3 4t 8*"
nvram set vlan3ports="4t 8"
nvram set port4vlans="1 3 16 18 19"
nvram set port5vlans="1 2 3 16"
nvram set vlan3hwname=et0
nvram commit
reboot
```

On some models it is also possible to create a default VLAN (untagged VLAN on a trunk port) which is not possible at all in the GUI even if the GUI works for your model. This example makes VLAN 1 the default VLAN for port 4 while also tagging VLAN 3 on port 4.

```
nvram set vlan1ports="1 2 3 4 8*"
nvram set vlan3ports="4t 8"
nvram set port4vlans="1 3 16 18 19"
nvram set port5vlans="1 2 3 16"
nvram set vlan3hwname=et0
nvram commit
reboot
```

Broadcom VLAN limit

Pre-k3.10 builds *OR any* before 35244, a separate switch-robo.ko kernel module is required for Broadcom.

- Reference: patches, and modules: VIDs beyond 15 possible?, VLAN config, Updated Guide (search the forum for more).

For k3.10+ starting in build 35244, the 15 VLAN Broadcom limit has been extended to 4096. Note that /sys/switch/eth0/vlan will contains 4096 (0-4095) directories instead of just 16 (0-15), which are used for VLAN configuration.

Example startup script to configure *VLAN for VID 1000 and 2000* (**customize VLAN configuration & VID(s) as needed!**):

```
# Clear VLAN 0 & 2
echo "" > /proc/switch/eth0/vlan/0/ports
echo "" > /proc/switch/eth0/vlan/2/ports

# Configure VLAN 1 with LAN port 1, 2, 3 and CPU port
echo "0 1 2 5t*" > /proc/switch/eth0/vlan/1/ports

# Configure VLAN 1000 with WAN port and CPU port (both tagged)
echo "4t 5t" > /proc/switch/eth0/vlan/1000/ports

# Configure VLAN 2000 with LAN port 4, WAN port (tagged) and CPU port (tagged)
echo "3 4t 5t" > /proc/switch/eth0/vlan/2000/ports

# Setting up VLAN interfaces ...

# We don't need the vlan2 interface now ...
/sbin/ifconfig vlan2 down
/sbin/vconfig rem vlan2

# Setup vlan1000 interface
/sbin/vconfig add eth0 1000
/sbin/ifconfig vlan1000 up
/sbin/ifconfig vlan1000 txqueuelen 0

# Setup vlan2000 interface
/sbin/vconfig add eth0 2000
/sbin/ifconfig vlan2000 up
/sbin/ifconfig vlan2000 txqueuelen 0

# Tell DD-WRT PPPoE startup code to use the new WAN interface
/usr/sbin/nvram set pppoe_wan_ifname=vlan1000
```

- One could instead continue to use the NVRAM method to interface with DD-WRT configuration subsystems.
- Avoid storing too much in NVRAM (i.e. scripts). **Recommended:** use USB or JFFS partitions if possible.
- Reference: quarkysg post in VLAN config, Updated Guide

References

- VLAN Support - List of devices tested for VLAN support (not up to date).
- Enabling VLAN Support for BCM4704 - VLAN support with separate internal WAN & LAN interfaces
- Reconfigure VLANs for 802.1q Compatibility
- VLAN Default Configuration
- VLAN Configuration
- VLAN Detached Networks (Separate Networks With Internet)
- VLAN Detached Networks each with Wireless and Internet