

# How to enable driver to support 802.11d

2013/01/21

## Introduction

In 802.11d spec. there's a IE(element id is 7, country IE) in AP's beacon to specify what channel plan the AP applied now, and STA can follow the country IE to decide its channel plan.

In Realtek wireless LAN (WLAN) driver, it can support 802.11d function when setting is configured correctly. If 802.11d support is enabled, driver would retrieve channel plan from first scanned AP which its beacon contains country IE, then driver will keep the channel plan and follow it to do site survey.

Besides, there's one more function will be enabled when 802.11d support is enabled. That is if WLAN received some AP's beacon in certain channel which is passive scan channel of WLAN's channel plan, then driver would change the scan type of channel from passive to active. And next time when WLAN need to do site survey, it will send probe request in that channel. This can help WLAN to scan more available APs in that legal channel and connect to hidden-SSID APs in passive scan channels.

Summary the functions Realtek WLAN would be active when enable **802.11d support**:

1. Change channel scan type to active if some AP is sending beacon in that channel
2. Following the channel plan of AP's IE.

## Configure steps

If need to enable Realtek WLAN driver 802.11d support, there're two steps to do.

1. Add define "CONFIG\_80211D" to file "include/autoconf.h", for example:

```
#define CONFIG_80211D
```

ps. If you can't find "include/autoconf.h" or the driver you get is compound driver package, you should add this definition to file "autoconf\_XXX\_YYY\_linux.h", where "XXX" is IC type and "YYY" is interface type, not "include/autoconf.h". For example, the autoconf.h of RTL8723A SDIO interface IC should be

“autoconf\_rtl8723a\_sdio.h”.

Once you finish this step, the function 1, automatic change scan type of channel to be active, is enabled. But function 2, follow the channel plan by AP, doesn't be active yet until step 2 is done. If you don't need function 2, you can also just stop here.

## 2. Set driver parameter “rtw\_80211d” to 1, default is 0.

There're two ways to set the value of “rtw\_80211d”, and you can decide which one is your favor.

### 2.1 Modify code of “os\_dep/linux/os\_intf.c”

Variable “rtw\_80211d” can be found in this file, and it would be like

```
#ifdef CONFIG_80211D
int rtw_80211d = 0;
#endif
```

Change the value of “rtw\_80211d” from 0 to 1.

```
#ifdef CONFIG_80211D
int rtw_80211d = 1;
#endif
```

### 2.2 Bring value of “rtw\_80211d” from outside when load driver

In Linux OS, the value of driver parameters could be set when load driver. So you can also set the value of “rtw\_80211d” when use “insmod” to load wlan driver.

For example:

```
# insmod wlan.ko rtw_80211d=1
```

On the other hand, you can also disable function 2 by set “rtw\_80211d” to 0, for example:

```
# insmod wlan.ko rtw_80211d=0
```

ps. You need to be super user or root when use “insmod” to load driver.

After these two steps, the driver now supports both function 1&2 of 802.11d.