



MediaTek LinkIt™ Development Platform for RTOS Wi-Fi Profile API Migration Guide

Version: 1.0

Release date: 4 November 2016

© 2015 - 2016 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc. ("MediaTek") and/or its licensor(s). MediaTek cannot grant you permission for any material that is owned by third parties. You may only use or reproduce this document if you have agreed to and been bound by the applicable license agreement with MediaTek ("License Agreement") and been granted explicit permission within the License Agreement ("Permitted User"). If you are not a Permitted User, please cease any access or use of this document immediately. Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited. THIS DOCUMENT IS PROVIDED ON AN "AS-IS" BASIS ONLY. MEDIATEK EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF ANY KIND AND SHALL IN NO EVENT BE LIABLE FOR ANY CLAIMS RELATING TO OR ARISING OUT OF THIS DOCUMENT OR ANY USE OR INABILITY TO USE THEREOF. Specifications contained herein are subject to change without notice.

Document Revision History

Revision	Date	Description
1.0	4 November 2016	Initial release

Table of contents

1. Overview	4
2. Wi-Fi profile API Migration	5
2.1. Using <code>wifi_init()</code> API	5
2.1.1. Migration steps from deprecated Wi-Fi profile APIs	6
2.2. Other deprecated Wi-Fi profile APIs	8
2.2.1. <code>wifi_profile_set_mac_address()</code>	8
2.2.2. <code>wifi_profile_set_pmk()</code>	9
2.2.3. <code>wifi_profile_set_country_region()</code>	9
2.2.4. <code>wifi_profile_commit_setting()</code> and <code>wifi_profile_get_profile()</code>	9
2.2.5. <code>wifi_profile_get_xxxx()</code> APIs	9

Lists of tables and figures

Table 1. Deprecated APIs 5
Table 2. wifi_init() to replace the deprecated APIs 5
Table 3. Deprecated profile API and wifi_init() mapping 6
Figure 1. Initialization flow before and after migration 7

1. Overview

The Wi-Fi profile APIs are deprecated starting from MediaTek LinkIt™ SDK v4.1.0. This guide offers a smooth transition from old deprecated APIs to the new APIs.

This document guides you through an example to replace the deprecated APIs.

2. Wi-Fi profile API Migration

All Wi-Fi profile APIs are deprecated starting from LinkIt SDK v4.1.0. The deprecated profile APIs are listed in Table 1. **Error! Reference source not found.**

Table 1. Deprecated APIs

Deprecated Profile API	Deprecated Profile API
wifi_profile_set_opmode()	wifi_profile_get_opmode()
wifi_profile_set_channel()	wifi_profile_get_channel()
wifi_profile_set_bandwidth()	wifi_profile_get_bandwidth()
wifi_profile_set_mac_address()	wifi_profile_get_mac_address()
wifi_profile_set_ssid()	wifi_profile_get_ssid()
wifi_profile_set_wireless_mode()	wifi_profile_get_wireless_mode()
wifi_profile_set_security_mode()	wifi_profile_get_security_mode()
wifi_profile_set_wpa_psk_key()	wifi_profile_get_wpa_psk_key()
wifi_profile_set_pmk()	wifi_profile_get_pmk()
wifi_profile_set_wep_key()	wifi_profile_get_wep_key()
wifi_profile_set_country_region()	wifi_profile_get_country_region()
wifi_profile_set_dtim_interval()	wifi_profile_get_dtim_interval()
wifi_profile_set_listen_interval()	wifi_profile_get_listen_interval()
wifi_profile_set_power_save_mode()	wifi_profile_get_power_save_mode()
wifi_profile_commit_setting()	wifi_profile_get_profile()

2.1. Using wifi_init() API

The list of deprecated Wi-Fi profile APIs that can be replaced with wifi_init() API is shown in Table 2:

Table 2. wifi_init() to replace the deprecated APIs

Deprecated Profile API	Deprecated Profile API
wifi_profile_set_opmode();	wifi_profile_set_channel();
wifi_profile_set_bandwidth();	wifi_profile_set_ssid();
wifi_profile_set_wireless_mode();	wifi_profile_set_security_mode();
wifi_profile_set_wpa_psk_key();	wifi_profile_set_wep_key();
wifi_profile_set_dtim_interval();	wifi_profile_set_listen_interval();
wifi_profile_set_power_save_mode()	

The wifi_init() API initializes the Wi-Fi module at boot up. Before calling this API, configure the profile settings in wifi_config_t and wifi_config_ext_t structures. Initializing the wifi_config_t settings is mandatory, while initializing wifi_config_ext_t is optional. More details on wifi_init() can be found in Wi-Fi API reference.

The structures `wifi_config_t` and `wifi_config_ext_t` have a set of parameters to map the deprecated Wi-Fi profile APIs, as shown in Table 3.

Table 3. Deprecated profile API and `wifi_init()` mapping

Deprecated Profile APIs	<code>wifi_config_t</code>	<code>wifi_config_ext_t</code>
<code>wifi_profile_set_opmode()</code>	<code>opmode</code>	–
<code>wifi_profile_set_channel()</code>	<code>ap_config.channel</code>	–
<code>wifi_profile_set_bandwidth()</code>	<code>ap_config.bandwidth</code>	–
<code>wifi_profile_set_mac_address()</code>	–	–
<code>wifi_profile_set_ssid()</code>	<code>sta_config.ssid</code> <code>sta_config.ssid_length</code> <code>ap_config.ssid</code> <code>ap_config.ssid_length</code>	–
<code>wifi_profile_set_wireless_mode()</code>	–	<code>sta_wireless_mode</code> <code>ap_wireless_mode</code>
<code>wifi_profile_set_security_mode()</code>	<code>ap_config.auth_mode</code> <code>ap_config.encrypt_type</code>	–
<code>wifi_profile_set_wpa_psk_key()</code>	<code>sta_config.password</code> <code>sta_config.password_length</code> <code>ap_config.password</code> <code>ap_config.password_length</code>	–
<code>wifi_profile_set_pmk()</code>	–	–
<code>wifi_profile_set_wep_key()</code>	<code>sta_config.password</code> <code>sta_config.password_length</code> <code>ap_config.password</code> <code>ap_config.password_length</code>	–
<code>wifi_profile_set_country_region()</code>	–	–
<code>wifi_profile_set_dtim_interval()</code>	–	<code>ap_dtim_interval</code>
<code>wifi_profile_set_listen_interval()</code>	–	<code>sta_listen_interval</code>
<code>wifi_profile_set_power_save_mode()</code>	–	<code>sta_power_save_mode</code>
<code>wifi_profile_commit_setting()</code>	–	–
<code>wifi_profile_get_profile()</code>	–	–

2.1.1. Migration steps from deprecated Wi-Fi profile APIs

Before migrating the Wi-Fi profile APIs:

- 1) Profile APIs are used to get or set the Wi-Fi profile in NVDM. The profile settings can be read by Wi-Fi profile get APIs.
- 2) At system boot up, the Wi-Fi driver configurations were initialized automatically based on the profile settings.

After migration:

- 1) Store the Wi-Fi settings in a storage, such as NVDM, flash, macro or more supported by the HDK, instead of using `wifi_profile_set_xxxx()` APIs.
- 2) At system boot up, the Wi-Fi driver configurations cannot be initialized automatically. Call the function `wifi_init()` to initialize the Wi-Fi driver. User can load the Wi-Fi settings from the storage to initialize the structure of `wifi_init()`.

The Wi-Fi initialization flow before and after migration is shown in Figure 1.

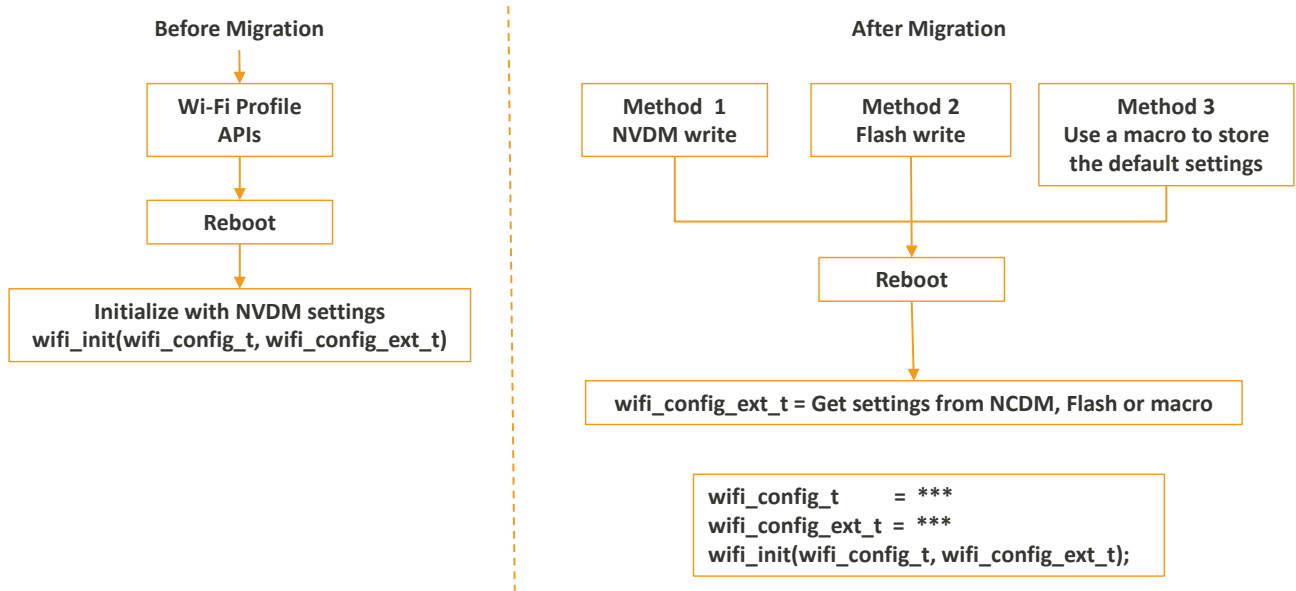


Figure 1. Initialization flow before and after migration

An example implementation of using NVDM (Method 1) to initialize the opmode, SSID, password and wireless mode to initialize the Wi-Fi settings, is shown below. Note that, you can also use Method 2 and Method 3 (see Figure 1).

- 1) Write settings in NVDM.

```

char opmode = '1';
char ssid[] = "AP1";
char ssid_len[] = "3";
char password[] = "12345678";
char password_len = '8';
char wireless_mode = '9';
if (NVDM_STATUS_OK != nvdm_write_data_item("common", "OpMode",
    NVDM_DATA_ITEM_TYPE_STRING,
    (uint8_t *)opmode, os_strlen(opmode))) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "Ssid",
    NVDM_DATA_ITEM_TYPE_STRING,
    (uint8_t *)ssid, os_strlen(ssid))) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "SsidLen",
    NVDM_DATA_ITEM_TYPE_STRING,
    (uint8_t *)ssid_len, os_strlen(ssid_len))) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "PassWord",
  
```



```

        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)password, os_strlen(password)) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "PassWordLen",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)password_len, os_strlen(password_len))) {
    return -1;
}

if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "WirelessMode",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)wireless_mode, os_strlen(wireless_mode))) {
    return -1;
}

```

2) Load Wi-Fi settings from storage medium when boot up.

```

wifi_config_t config = {0};
wifi_config_ext_t config_ext = {0};

uint8_t buff[128];
uint32_t len = sizeof(buff);
nvdm_read_data_item("common", "OpMode", buff, &len);
config.opmode = (uint8_t)atoi((char *)buff);

len = sizeof(buff);
nvdm_read_data_item("STA", "SsidLen", buff, &len);
config.sta_config.ssid_length = (uint8_t)atoi((char *)buff);

len = sizeof(buff);
nvdm_read_data_item("STA", "Ssid", buff, &len);
memcpy(config.sta_config.ssid, buff, config.sta_config.ssid_length);

len = sizeof(buff);
nvdm_read_data_item("STA", "PassWordLen", buff, &len);
config.sta_config.password_length = (uint8_t)atoi((char *)buff);

len = sizeof(buff);
nvdm_read_data_item("STA", "PassWord", buff, &len);
memcpy(config.sta_config.password, buff,
config.sta_config.password_length);

len = sizeof(buff);
nvdm_read_data_item("STA", "WirelessMode", buff, &len);
config_ext->sta_wireless_mode_present = 1;
config_ext->sta_wireless_mode = (uint8_t)atoi((char *)buff);

```

3) Call the function `wifi_init()` to initialize the Wi-Fi profile.

```
wifi_init(&config, &config_ext);
```

2.2. Other deprecated Wi-Fi profile APIs

2.2.1. `wifi_profile_set_mac_address()`

This API should no longer be used and can be removed from the source code, as eFuse is used to load and set the MAC address.

2.2.2. `wifi_profile_set_pmk()`

This API should no longer be used and can be removed from the source code.

2.2.3. `wifi_profile_set_country_region()`

This API should be removed from the source code, as the country code is set through `wifi_init()`.

2.2.4. `wifi_profile_commit_setting()` and `wifi_profile_get_profile()`

These two APIs should no longer be used and can be removed from the source code.

2.2.5. `wifi_profile_get_xxxx()` APIs

The deprecated `wifi_profile_get_xxxx()` APIs were used to read settings from the NVDM. However, with current APIs the settings can be read from NVDM, flash, or macro see section 2.1.1, "Migration steps from deprecated Wi-Fi profile APIs".