WFA test setup for 11n/ac/P2P certification for Infineon's Linux-based platforms

About this document

Scope and purpose

This document describes the setup and processes for Wi-Fi Alliance (WFA) certification testing for Linux-based platforms using Infineon AIROC™ CYW893xx Wi-Fi & Bluetooth® combo chips.

Intended audience

Field application engineers and customers.

Table of contents

Table of contents

1 Introduction

Wi-Fi Alliance (WFA) has various certification programs such as 11ac, 11n, WPS, and Wi-Fi Direct. It has station (STA) and access point (AP) modes. As shown in **Figure 1**, major components of the test setup are the test controller, testbed APs, testbed STAs, sniffer, and device under test (DUT).

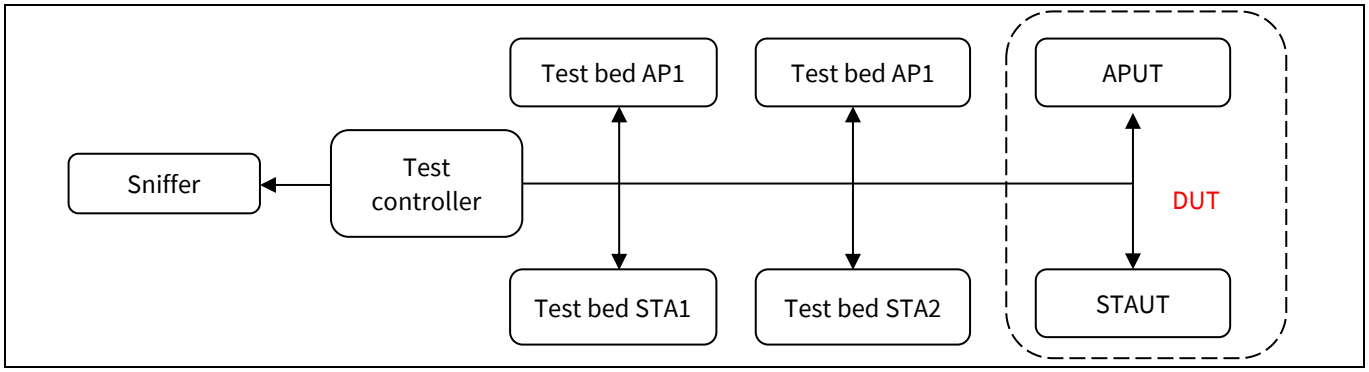

Figure 1 Typical WFA certification setup

1.1 Unified control API console (UCC)

WFA will provide the necessary software for the test controller in the form of a Wi-Fi test suite (WTS). The Test controller is called "unified CAPI console (UCC)" in the WTS. UCC uses the sigma control API (CAPI), a fundamental command language for device management, test configuration, and test execution within WTS.

The UCC core is architected and designed to encapsulate the complexity of test automation and control. It acts like a command interpreter taking the extended CAPI commands to drive test execution. Its job consists of CAPI interpretation, command delivery, and information handling that devices send back for use in subsequent commands.

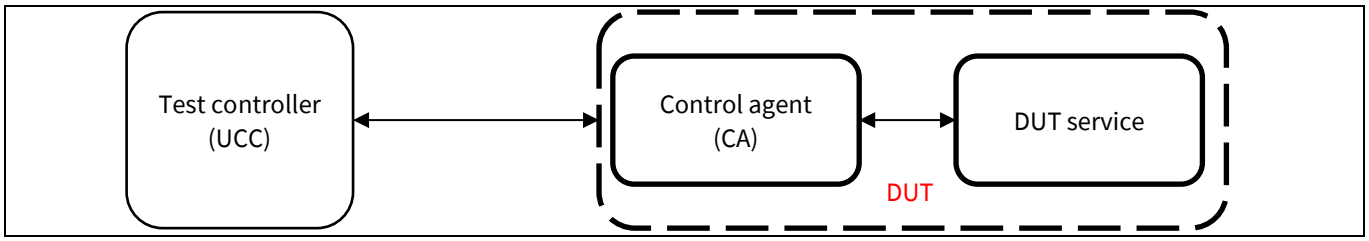

Figure 2 UCC-DUT setup

1.2 Control agent (CA)

A control agent (CA) is a proxy in which a CAPI control command is received for the device. As shown in **Figure 2**, you can run control agents inside the DUT, but some DUTs run these control agents outside of the DUT. The control agent typically opens a passive TCP socket on a selected port and waits for the UCC to connect to it. The UCC is configured to perform an active TCP socket to the control agent. The control agent collects the necessary information from the DUT service and returns it to the UCC.

1.3 DUT service

The DUT service processes the commands received from the control agent by executing commands on the DUT and sends the results back to the control agent.

The DUT is further classified into two categories. If the station is being tested, it is called "STAUT". If the AP is being tested, it is called "APUT".

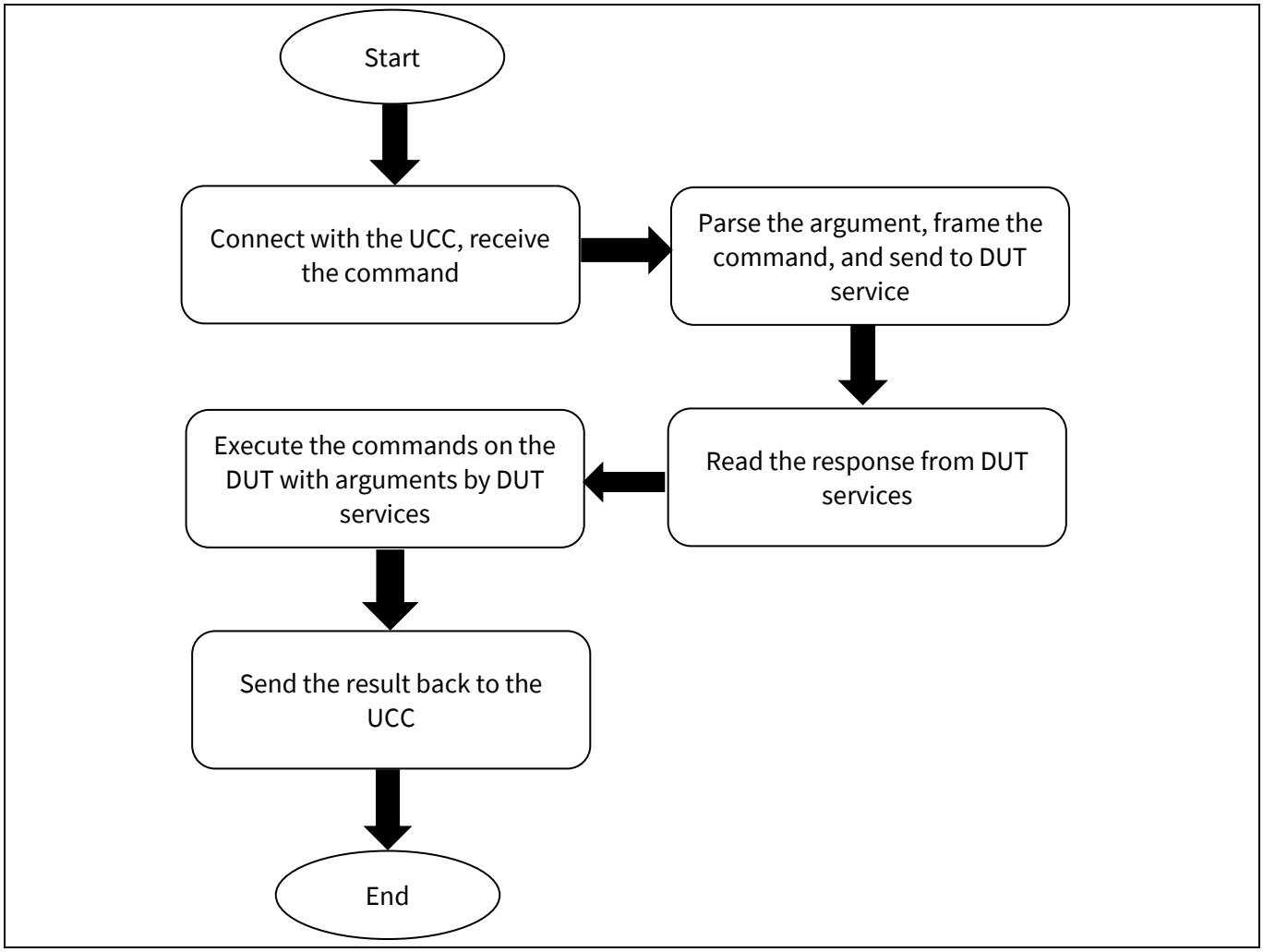

Figure 3 Flowchart for the WFA setup

Note: WFA uses different software like traffic generator and PC endpoint to run the data between two devices. See the WFA website **https://www.wi-fi.org/certification/wi-fi-test-suite** *for more information.*

2 Testing tools and setup

2.1 Wi-Fi test suite (WTS)

The WTS tool suite provides configuration, test control, traffic generation, and results analysis services. The test plan, in its entirety, can be executed in a fully automated manner through the WFA-distributed Wi-Fi test suite command scripts and the Wi-Fi test suite unified CAPI console.

2.2 Sniffer

Sniffers are required to capture and decode the 802.11 a/b/g/n/ac frames and packets. It could be wired or wireless. The most commonly used wireless sniffer is **Wireshark**.

2.3 Authentication server (RADIUS server)

These are PC-based servers, which perform the authentication server function on 802.1x ports. It is a client/server-based protocol that runs on the application layer and uses UDP. In this case, the client is an AP (also known as the authenticating agent) and the STA is a supplicant.

The **RADIUS server** checks the credentials of the supplicant on behalf of the client (authenticating agent) and it responds to the authenticating agent whether the supplicant is authorized to access the services.

2.4 Typical WTS configuration setup

Figure 4 shows the typical test configuration infrastructure for WTS.

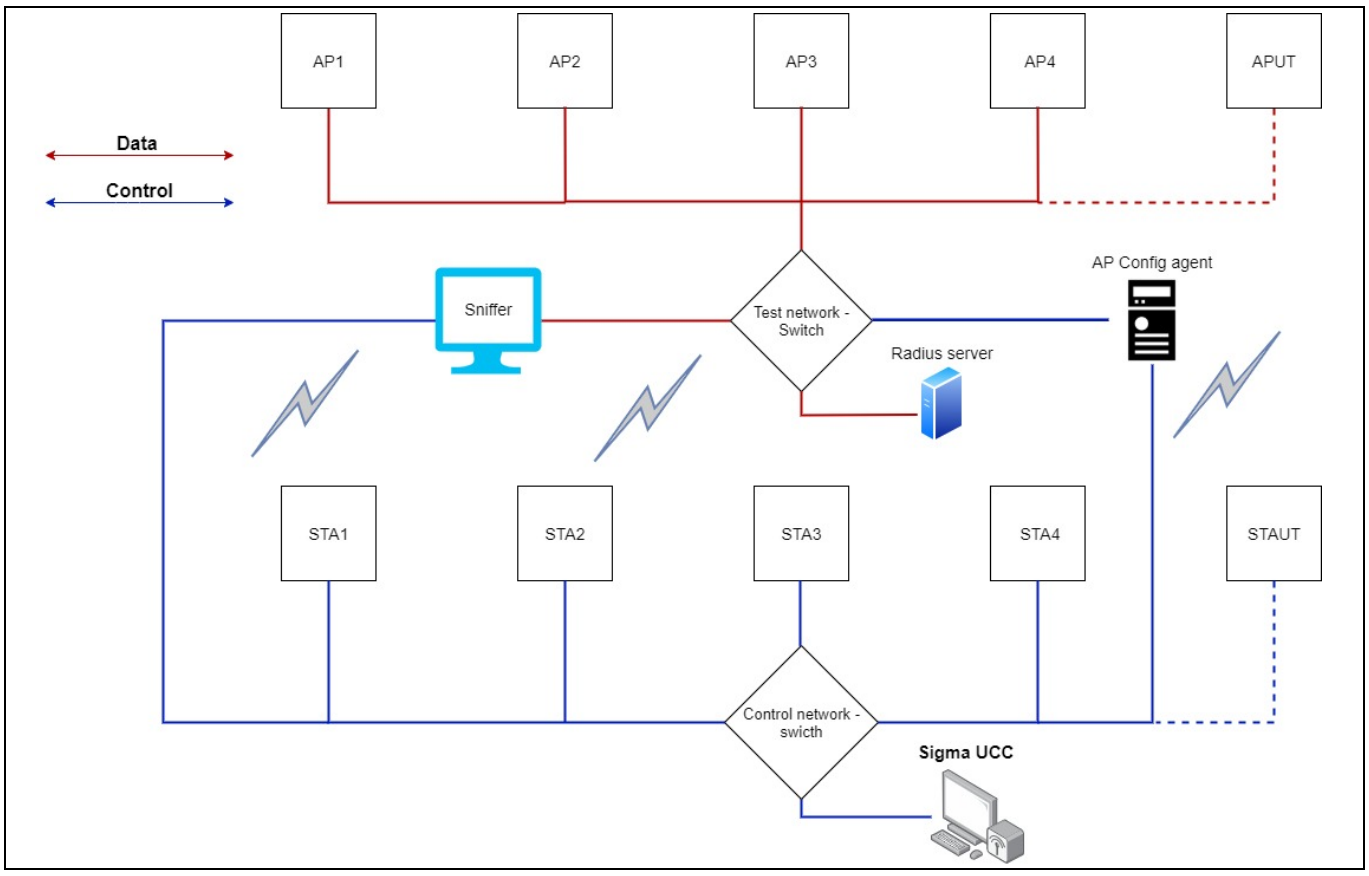

Figure 4 Wi-Fi test suite configuration

3 Test setup configurations

3.1 Edit the Windows UCC-side configuration for SoftAP certification - *init_802.11n.txt*

1. Set `$DUT` to `$APUT`.
2. Set the DUT control agent IP address and port number.
3. Set the DUT wireless IP address.
4. Set the PC endpoint IP address to the DUT's IP address.
5. Set the traffic generator IP address to the DUT's wireless IP address.
6. Set the test bed control agent (STA1) IP address to the wired IP address of the WTS-compliant testbed1.
7. Set testbed (STA1) wireless IP address to the wireless IP address of the WTS-compliant testbed1.
8. Set the test bed control agent (STA2 – Broadcom-VHT as well as broadcom11n) IP address to the wired IP address of the WTS-compliant testbed2.
9. Set the test bed (STA2 – Broadcom-VHT as well as broadcom11n) wireless IP address to the WTS-compliant testbed2's wireless IP address.
10. Set the testbed control agent (STA3) IP address to the wired IP address of the WTS-compliant testbed3.
11. Set the test bed (STA3) wireless IP address to the WTS-compliant testbed3's wireless IP address.
12. Set the test bed control agent (STA4) IP address to the wired IP address of the WTS-compliant testbed4.
13. Set the test bed (STA4) wireless IP address to the WTS-compliant testbed4's wireless IP address.

Note: *STA1, STA2, STA3, and STA4 are WTS-compliant STA test beds from Atheros, Broadcom, Intel, and Marvell respectively.*

3.2 Automated SoftAP setup configuration

Edit the Linux WTS Tcl-side configuration and make the following changes:

1. Open *Sigma11N-android-SoftAP_rev1/TCL/linux/linux.tcl*.
2. Set `BIN_PATH` to the path where all WLAN binaries and conf files are kept.
3. Set `WLAN_IFACE` to `wlan0`.
4. Set `WLAN_IPADDR` to desired wlan0 IP address.

No other settings are required.

3.3 Run WFA tests

1. Start a test on the UCC (a Windows laptop, for example):

```
cd Wi-FiTestSuite_UCC-Windows_v9.1.0\bin
```

2. Run the 4.2.2 soft-AP certification test on the SoftAP DUT side:

```
cd Wi-FiTestSuite_UCC-Windows_v9.1.0\bin
wfa_ucc N N-4.2.2
```

Note: *This SoftAP DUT must be qualified with WTS-compliant test beds on another end, as mentioned earlier.*

3.4 Edit the configuration for 11ac STAUT – *init_VHT.txt*

Note: In VHT-Testbed-APs.txt, comment out `TestbedAPConfigServer`.

1. Set `sniffer_enable!0!`.
2. Set the DUT control agent IP address and port number.
3. Set the DUT wireless IP address.
4. Set `DUT_Name` to `STAUT-VHT`.
5. Set the PC endpoint IP address to the DUT's Ethernet IP address.
6. Set the traffic generator IP address to the DUT's wireless IP address.
7. Comment out all test bed STA entries.
8. In the test bed AP configuration, set all AP's IP address to the wired IP address of the test bed AP.
9. For the packet sniffer, comment out the following line and set `sniffer_enable` to `0`.

   ```
   #wfa_sniffer!ipaddr=192.168.250.5,port=9999!
   ```

10. Comment out *PoweronOffAPs.txt* and *RadioOff.txt* as follows:

    ```
    #wfa_test_commands!PowerOnOffAPs.txt!
    #wfa_test_commands!RadioOff.txt!
    ```

4 Preparation to run the WFA certification test

4.1 Prerequisites

Download and install the following software in the DUT device (STAUT or APUT) to run the test:

- **Tcl-Tk:** Download from **https://www.tcl.tk/**
- **Iptables:** Download from **https://www.netfilter.org/projects/iptables/downloads.html**

4.2 Software package required for certification

The WTS package can be downloaded from the WFA website. Infineon provides necessary modifications for CA and DUT services and releases the WTS package, but this contains only the modifications. Therefore, you need to use both the WTS packages downloaded from the WFA website and the package from Infineon.

The Infineon WTS package contains the following:

Package content	Description
AP_SIGMA_REL_X_X	Tcl-based sigma agent package for supplicant-based SoftAP certification evaluation
P2P_SIGMA_REL_X_X	Tcl-based sigma agent package for supplicant-based P2P certification evaluation
STA_SIGMA_REL_X_X	C-based sigma agent package for supplicant-based STA certification evaluation

4.3 Compile the services

You need to compile CA (wfa_ca) and DUT (wfa_dut) services for the Infineon WLAN platform. Because each device mode has different requirements of wfa_ca and wfa_dut, you need to have different wfa_ca and wfa_dut for different modes.

4.3.1 11n/ac STAUT

1. Enter the following commands to compile wfa_ca and wfa_dut:

```
#cd STA_SIGMA_REL
#make
```

2. Update the `PATH` variable in *run.sh*:

- For non-FMAC:
  - Specify `BIN_PATH` to the path where all WLAN binaries (bcmdhd.ko, rtecdc.bin, nvram.txt, clm_blob, wl, dhd, wpa_supplicant, and wpa_cli) and conf files are kept.
- For FMAC:
  - Specify `BIN_PATH` to the path where all WLAN binaries *(wl, wpa_supplicant and wpa_cli)* and *conf* files are kept.
  - Copy the *ko, firmware, NVRAM,* and *clm_blob* files to `FMAC_PATH` (*/lib/firmware/brcm/* for AIROC™ CYW89373 automotive Wi-Fi & Bluetooth® combo chip).

In addition to these steps, ensure that the certification paths are set as follows:

- Keep all certificates in `BIN_PATH`.
- Store the private key in `BIN_PATH` with the name *wifiuser.pfx*.
- Store ca_cert in `BIN_PATH` by name *cas.pem*.
- Store Client cert in `BIN_PATH` with the name *wifiuser.pem*.

- Set the identity in `CERTIFICATES_IDENTITY`.
- Set the private key password in `CERTIFICATES_PRIVATE_KEY_PASSWD`.
- Kill any running instance of wfa_dut before starting *run.sh.*

4.3.2 11n/ac APUT

Get the *Wi-FiTestSuite_DUT_Code-Linux_vX.Y.Z* package from the original WFA WTS package because Infineon does not provide changes in wfa_dut and wfa_ca for AP mode.

Enter the following commands in the APUT:

```
#cd Wi-FiTestSuite_DUT_Code-Linux_vX.Y.Z
#cd lib
#make
#cd ../dut
#make
#cp wfa_dut AP_SIGMA_REL/tcl/linux/bin/
#cp wfa_dut AP_SIGMA_REL/C/dut/gcc
#cd ../ca
#make
#cp wfa_ca AP_SIGMA_REL/tcl/linux/bin/
#cp wfa_ca AP_SIGMA_REL/C/ca/gcc
```

4.3.3 P2P DUT

Enter the following commands in the DUT:

```
#cd Wi-FiTestSuite_DUT_Code-Linux_vX.Y.Z
#cd lib
#make
#cd ../dut
#make
#cp wfa_dut P2P_SIGMA_REL
#cd ../ca
#make
#cp wfa_ca P2P_SIGMA_REL
```

5 Starting the CA and DUT services

5.1 11n STA

Run the scripts provided by Infineon to start wfa_dut and wfa_ca in the STAUT as follows:

```
#cd STA_SIGMA_REL
#killall wfa_dut
#sh -x run.sh <CHIPID>
```

Example: `#sh -x run.sh 89373sdio`

Table 1 Chip ID with bus

Chip	Chip ID	Bus	Usage
AIROC™ CYW89359 automotive Wi-Fi & Bluetooth® combo chip	89359	PCIe	`sh -x run.sh 89359pcie` `tclsh xyz.tcl linux 89359pcie`
AIROC™ CYW89359 automotive Wi-Fi & Bluetooth® combo chip	89359	SDIO	`sh -x run.sh 89359sdio` `tclsh xyz.tcl linux 89359sdio`
AIROC™ CYW89373 automotive Wi-Fi & Bluetooth® combo chip	89373	PCIe	`sh -x run.sh 89373pcie` `tclsh xyz.tcl linux 89373pcie`
AIROC™ CYW89373 automotive Wi-Fi & Bluetooth® combo chip	89373	SDIO	`sh -x run.sh 89373sdio` `tclsh xyz.tcl linux 89373sdio`
AIROC™ CYW89335 automotive Wi-Fi & Bluetooth® combo chip	89335	SDIO	`sh -x run.sh 89335` `tclsh xyz.tcl linux 89335`

5.2 11n AP

Commands in the following sections must be run in superuser mode. Before running the DUT script, you must clear the firewall.

1. Run the following commands to clear the firewall. You may need to add a delay of 1 second in between these commands.

   ```
   #iptables -F
   #iptables -P INPUT ACCEPT
   #iptables -P FORWARD ACCEPT
   #iptables -P OUTPUT ACCEPT
   ```

2. Run the following commands to execute wfa_ca and wfa_dut.

   ```
   #cd AP_SIGMA_AGENT/tcl/linux
   #tclsh linux.tcl linux <CHIPID>
   ```

   See **Table 1** for the value of `<CHIPID>`.

   Example –> `#tclsh linux.tcl linux 89373sdio`

5.3 11ac STA

Run the following command for Linux – semi-automated STA setup:

```
#cd STA_SIGMA_REL
#killall wfa_dut
#sh -x run.sh <CHIPID>
```

See **Table 1** for the value of <CHIPID>.

5.4 11ac AP

In AP_SIGMA_REL, wfa_dut and wfa_ca are referred from *AP_SIGMA_REL/tcl/linux/bin/.*

Enter the following commands to start CA and DUT services for each section (11n, 11ac, P2P, etc.):

```
#cd AP_SIGMA_REL/VHT
#tclsh vht_softap_dut.tcl linux <CHIPID>
```

See **Table 1** for the value of <CHIPID>.

Example: `#tclsh vht_softap_dut.tcl linux 89373sdio`

5.5 P2P

1. Before running the DUT script, run the following commands to clear the firewall:

   ```
   #iptables -F
   #sleep 1
   #iptables -P INPUT ACCEPT
   #sleep 1
   #iptables -P FORWARD ACCEPT
   #sleep 1
   #iptables -P OUTPUT ACCEPT
   ```

2. Run the following commands to load the driver and supplicant manually:

   ```
   #cd P2P_SIGMA_REL
   #tclsh main.tcl 9000
   ```

Note: *wfa_ca and wfa_dut are not required for P2P tests because there are no throughput-related validations.*

5.6 Sample test logs

5.6.1 Test case information

The following logs show the information of the test case that is being run:

```
2021-07-13 14:18:27.779 - INFO - WiFiTestSuite Version [10.10.1]
2021-07-13 14:18:27.796 - INFO - Logging started in file - ./log/N-5.2.31_Jul-13-2021__14-18-27/log_N-5.2.31.log
2021-07-13 14:18:27.796 - INFO - ----------------------------------------
2021-07-13 14:18:27.796 - INFO - Test Info
```

```
2021-07-13 14:18:27.796 - INFO - ------------------------------------------
2021-07-13 14:18:27.812 - INFO - Test Prog Name    : N
2021-07-13 14:18:27.812 - INFO - Test ID           : N-5.2.31
2021-07-13 14:18:27.812 - INFO - Test Usermode     : precert
2021-07-13 14:18:27.812 - INFO - Test CmdPath      : ..\cmds\WTS-11n
2021-07-13 14:18:27.812 - INFO - Test InitFile     : AllInitConfig_N.txt
2021-07-13 14:18:27.826 - INFO - Test TB File      : \802.11n-Testbed-APs.txt
2021-07-13 14:18:27.826 - INFO - Test Start Time   : 2021-07-13 14:18:27
2021-07-13 14:18:27.826 - INFO - ------------------------------------------
```

5.6.2 Logs showing sample CAPI commands

The following logs show some sample CAPI commands that are passed and the information received:

```
2021-07-13 14:18:56.628 - INFO - ~~~~~ [ Getting TB STA Version Info ] ~~~~~
2021-07-13 14:18:56.628 - INFO - realtek11n (IP:Port) ---> ca_get_version
2021-07-13 14:18:57.220 - INFO - realtek11n (IP:Port) <--
status,COMPLETE,version,WIN7_WIN8_DUT-v9.1.0

2021-07-13 14:18:57.236 - INFO - realtek11n (IP:Port) ---> device_get_info
2021-07-13 14:19:05.723 - INFO - realtek11n (IP:Port) <--
status,COMPLETE,vendor,Realtek,model,RTL8812BU,version,10.9.1222.2016

2021-07-13 14:19:07.236 - INFO - broadcomvht (IP:Port) ---> ca_get_version
2021-07-13 14:19:07.453 - INFO - broadcomvht (IP:Port) <--
status,COMPLETE,version,1.139.1

2021-07-13 14:19:08.904 - INFO - intel11n (IP:Port) ---> ca_get_version
2021-07-13 14:19:08.921 - INFO - intel11n (IP:Port) <--
status,COMPLETE,version,WIN7_WIN8_DUT-WTSv9.2.0
```

5.6.3 Sample sniffer logs

The following logs show some sample sniffer logs:

```
SNIFFER CHECKS LOG - Testcase: N-5.2.31
Jul: 13:2021-14:19:45 | SNIFFER ---> sniffer_get_info,
Jul: 13:2021-14:19:51 | SNIFFER <--
status,COMPLETE,WfaSnifferVersion,SnifferSTA,Other,SwInfo,UnKnown_2.6.39.4,Wi
resharkVersion,1.6.2,MapConfVersion,
Jul: 13:2021-14:22:43 | SNIFFER --->
sniffer_control_start,filename,SnifferTrace_N-5.2.31,channel,6
Jul: 13:2021-14:22:43 | SNIFFER <-- status,COMPLETE

Jul: 13:2021-14:25:38 | SNIFFER ---> sniffer_control_stop,
Jul: 13:2021-14:25:38 | SNIFFER <-- status,COMPLETE
```

5.6.4 Final test results

The following logs show the final test result:

```
2021-07-13 14:27:34.608 - INFO -
----------------RESULT--------------------------


2021-07-13 14:27:34.608 - INFO - Expected  <= 115 %
2021-07-13 14:27:34.608 - INFO - Actual -   100.0 %
2021-07-13 14:27:34.608 - INFO - TEST RESULT ---> PASS
2021-07-13 14:27:39.632 - INFO ------------------------------------------------
2021-07-13 14:27:39.648 - INFO - SNIFFER (IP:Port) --->
sniffer_frame_check,name,wireShark_wlan,filename,N-
5.2.31_4,srcmac,xx:xx:xx:xx:xx:xx,bssid,yy:yy:yy:yy:yy:yy,framename,anydata,t
ype,AC_Video,Present,No
2021-07-13 14:27:56.214 - INFO - SNIFFER (IP:Port) <--
status,COMPLETE,CheckResult,SUCCESS
2021-07-13 14:27:56.230 - INFO - FINAL TEST RESULT  --->            PASS
2021-07-13 14:27:56.246 - INFO - END: TEST CASE [N-5.2.31]
2021-07-13 14:27:57.183 - INFO - PASS
2021-07-13 14:27:57.183 - INFO - Execution Time [569.81] seconds
```

6 Appendix: Recommendations from Infineon

See the WFA sigma documentation provided by Infineon as a reference—it is consistent with the WFA requirements. You can use follow the recommendations from Infineon use concurrent operation of 2.4-GHz and 5-GHz bands for automotive infotainment and telematics applications.

6.1 Getting support for your certification process

Do the following to get support from Infineon to debug test failures on your platforms:

- Rule out issues related to test bed configuration and setup:
  - Run the failed test cases on the WFA test bed platform with the DUT replaced by a third-party WFA-certified device.
  - Ensure that the changes that you make to the scripts, driver, and supplicant conform to your product design. The scripts and tools provided by Infineon are based on the Infineon reference platform.
  - Ensure that device configurations provided by Infineon are used: *Sample .conf* files are provided in release packages.
  - Verify that configuration files are not being modified by the device.
- Tune the host platform software for optimal performance for throughput-related test cases. Infineon sigma release documentation provides host-tuning parameters on the reference platform for Linux as a guide.

Do the following if WFA test cases fail after following these steps:

- Raise a support ticket with Infineon and provide all UCC log files and DUT device log files.
- In the support ticket, confirm that you have complied with the steps mentioned in this section. Infineon will provide log files showing the 'Pass' status for these tests on an Infineon platform, which you can use for comparison with the results obtained from your tests.

Revision history

Document version	Date of release	Description of changes
**	2021-12-07	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2021-12-07
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to www.cypress.com/support

Document reference
002-34191 Rev. **

IMPORTANT NOTICE
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.