

Please note:
The Command Matrix is
attached to this PDF file.



Wi-Fi Test Suite Control API Technical Specification Version 9.2.0

WI-FI ALLIANCE PROPRIETARY – SUBJECT TO CHANGE WITHOUT NOTICE

This document may be used with the permission of Wi-Fi Alliance under the terms set forth herein.

By your use of the document, you are agreeing to these terms. Unless this document is clearly designated as an approved specification, this document is a work in process and is not an approved Wi-Fi Alliance specification. This document is subject to revision or removal at any time without notice. Information contained in this document may be used at your sole risk. Wi-Fi Alliance assumes no responsibility for errors or omissions in this document. This copyright permission does not constitute an endorsement of the products or services. Wi-Fi Alliance trademarks and certification marks may not be used unless specifically allowed by Wi-Fi Alliance.

Wi-Fi Alliance has not conducted an independent intellectual property rights ("IPR") review of this document and the information contained herein, and makes no representations or warranties regarding IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions.

Wi-Fi Alliance owns the copyright in this document and reserves all rights therein. A user of this document may duplicate and distribute copies of the document in connection with the authorized uses described herein, provided any duplication in whole or in part includes the copyright notice and the disclaimer text set forth herein. Unless prior written permission has been received from Wi-Fi Alliance, any other use of this document and all other duplication and distribution of this document are prohibited. Unauthorized use, duplication, or distribution is an infringement of Wi-Fi Alliance's copyright.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY WI-FI ALLIANCE AND WI-FI ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS DOCUMENT AND ANY INFORMATION CONTAINED IN THIS DOCUMENT.

10900-B Stonelake Boulevard, Suite 126
Austin, TX 78759

Phone: 512.498.9434 • Fax: 512.498.9435 • Email: certifications@wi-fi.org
www.wi-fi.org

Table of Contents

1	Introduction	7
1.1	Audience	7
1.2	Command matrix	7
1.3	Reference documents	7
1.4	Acronyms	8
2	Basic communication paradigm	9
3	Command and response syntax	10
3.1	Command syntax	10
3.2	Response syntax	10
3.3	Response timeouts	10
4	Control Agent state machine	12
5	Data types	13
6	Unified CAPI Console (UCC)	14
6.1	UCC core	14
6.2	UCC command scripts	14
7	Station control API	15
7.1	CA_GET_VERSION	15
7.2	CONNECT_GO_START_WFD	16
7.3	DEVICE_GET_INFO	17
7.4	DEVICE_LIST_INTERFACES	18
7.5	REINVOKE_WFD_SESSION	19
7.6	STA_ADD_ARP_TABLE_ENTRY	20
7.7	STA_ADD_CREDENTIAL	21
7.8	STA_ASSOCIATE	23
7.9	STA_BLOCK_ICMP_RESPONSE	24
7.10	STA_BSSID_POOL	25
7.11	STA_DISCONNECT	26
7.12	STA_EXEC_ACTION	27
7.13	STA_GENERATE_EVENT	30
7.14	STA_GET_BSSID	32
7.15	STA_GET_EVENT_DETAILS	33
7.16	STA_GET_EVENTS	36
7.17	STA_GET_INFO	38
7.18	STA_GET_IP_CONFIG	39
7.19	STA_GET_MAC_ADDRESS	40
7.20	STA_GET_P2P_DEV_ADDRESS	40
7.21	STA_GET_P2P_IP_CONFIG	41
7.22	STA_GET_PARAMETER	42
7.23	STA_GET_PSK	44
7.24	STA_HS2_ASSOCIATE	45
7.25	STA_INVOKE_COMMAND	46
7.26	STA_IS_CONNECTED	49
7.27	STA_MANAGE_SERVICE	50
7.28	STA_OSU	52
7.29	STA_P2P_CONNECT	54
7.30	STA_P2P DISSOLVE	55
7.31	STA_P2P_RESET	55
7.32	STA_P2P_START_GROUP_FORMATION	56
7.33	STA_POLICY_UPDATE	57
7.34	STA_PRESET_TESTPARAMETERS	58
7.35	STA_REASSOC	64
7.36	STA_REASSOCIATE	65

7.37	STA_RESET_DEFAULT	66
7.38	STA_RESET_PARM.....	68
7.39	STA_SCAN	68
7.40	STA_SEND_ADDDBA.....	69
7.41	STA_SEND_COEXIST_MGMT	70
7.42	STA_SEND_FRAME.....	71
7.43	STA_SEND_P2P_INVITATION_REQ	72
7.44	STA_SEND_P2P_PRESENCE_REQ.....	73
7.45	STA_SEND_P2P_PROVISION_DIS_REQ	73
7.46	STA_SEND_SERVICE_DISCOVERY_REQ	74
7.47	STA_SET_11N.....	75
7.48	STA_SET_EAPAKA.....	77
7.49	STA_SET_EAPFAST	78
7.50	STA_SET_EAPSIM.....	79
7.51	STA_SET_EAPTLS	80
7.52	STA_SET_EAPTTLS	81
7.53	STA_SET_ENCRYPTION.....	83
7.54	STA_SET_IP_CONFIG.....	84
7.55	STA_SET_MACADDR.....	85
7.56	STA_SET_OPPORTUNISTIC_PS.....	85
7.57	STA_SET_P2P	86
7.58	STA_SET_PEAP.....	88
7.59	STA_SET_POWER_SAVE	89
7.60	STA_SET_PSK.....	90
7.61	STA_SET_PWRSAVE	92
7.62	STA_SET_RADIO.....	92
7.63	STA_SET_RFEATURE.....	93
7.64	STA_SET_SECURITY	96
7.65	STA_SET_SLEEP.....	97
7.66	STA_SET_SYSTIME	98
7.67	STA_SET_UAPSD.....	99
7.68	STA_SET_WIRELESS.....	100
7.69	STA_SET_WMM.....	104
7.70	STA_SET_WPS_PBC.....	106
7.71	STA_START_AUTONOMOUS_GO	107
7.72	STA_WPS_ENTER_PIN.....	108
7.73	STA_WPS_READ_LABEL.....	109
7.74	STA_WPS_READ_PIN.....	110
7.75	START_WFD_CONNECTION.....	111
7.76	START_WPS_REGISTRATION.....	112
7.77	TRAFFIC_AGENT_CONFIG	113
7.78	TRAFFIC_AGENT_RECEIVE_START.....	117
7.79	TRAFFIC_AGENT_RECEIVE_STOP.....	118
7.80	TRAFFIC_AGENT_RESET.....	119
7.81	TRAFFIC_AGENT_SEND.....	120
7.82	TRAFFIC_AGENT_VERSION	121
7.83	TRAFFIC_SEND_PING	122
7.84	TRAFFIC_STOP_PING	123
8	Access Point Configuration API.....	124
8.1	Access Point Commands.....	124
8.2	AP_CA_VERSION	124
8.3	AP_CONFIG_COMMIT.....	124
8.4	AP_DEAUTH_STA.....	125
8.5	AP_GET_MAC_ADDRESS	125
8.6	AP_RESET_DEFAULT	126
8.7	AP_SEND_ADDDBA_REQ	127

8.8	AP_SEND_BCNRPRT_REQ.....	128
8.9	AP_SEND_BSSTRANS_MGMT_REQ.....	129
8.10	AP_SEND_LINK_MEA_REQ.....	130
8.11	AP_SEND_TSMRPT_REQ.....	131
8.12	AP_SET_11D.....	132
8.13	AP_SET_11H.....	133
8.14	AP_SET_11N_WIRELESS.....	134
8.15	AP_SET_APQOS.....	135
8.16	AP_SET_HS2.....	136
8.17	AP_SET_PMF.....	139
8.18	AP_SET_RADIUS.....	140
8.19	AP_SET_RFEATURE.....	141
8.20	AP_SET_RRM.....	143
8.21	AP_SET_SECURITY.....	144
8.22	AP_SET_STAQOS.....	146
8.23	AP_SET_WIRELESS.....	147
9	Wi-Fi Test Suite Sniffer Control API.....	153
9.1	Sniffer Control Commands.....	153
9.2	SNIFFER_CONTROL_START.....	153
9.3	SNIFFER_CONTROL_STOP.....	155
9.4	SNIFFER_CONTROL_UPLOAD.....	155
9.5	SNIFFER_CONTROL_FIELD_CHECK.....	156
9.6	SNIFFER_CHECK_PMK_ID.....	178
9.7	SNIFFER_CHECK_RETRY.....	179
9.8	SNIFFER_FRAME_CHECK.....	180
9.9	SNIFFER_CHECK_CHNI_SWITCH.....	181
9.10	SNIFFER_GET_FIELD_VALUE.....	182
9.11	SNIFFER_CHECK_TIME_DIFFERENCE.....	187
9.12	SNIFFER_CHECK_FRAME_FIELD.....	188
9.13	SNIFFER_CHECK_PVB_PSPOLL_DATA.....	188
9.14	SNIFFER_CHECK_PVB_PSNONPOLL_DATA.....	189
9.15	SNIFFER_CHECK_P2P_NOA_DESCRIPTOR.....	189
9.16	SNIFFER_CHECK_P2P_OPPTS_CLIENT.....	190
9.17	SNIFFER_CHECK_P2P_OPPTS_GO.....	190
9.18	SNIFFER_CHECK_P2P_NOA_START_TIME.....	191
9.19	SNIFFER_CHECK_P2P_NOA_WMMPS_RETRIGGER.....	192
9.20	SNIFFER_CHECK_P2P_CLIENT_PS_RETRIGGER.....	193
9.21	SNIFFER_CHECK_P2P_NOA_DURATION.....	193
9.22	SNIFFER_CONTROL_FIELD_CHECK_ALL.....	194
9.23	SNIFFER_CONTROL_FILTER_CAPTURE / WFA_SNIFFER_CONTROL_FILTER_CAPTURE.....	195
9.24	SNIFFER_INJECT_FRAME.....	199
9.25	SNIFFER_CLEAR_COUNTERS.....	200
9.26	SNIFFER_DECRYPT_TRACE.....	201
9.27	SNIFFER_FETCH_FILE.....	202
9.28	WFA_SNIFFER_CONTROL_CAPTURE_DECRYPT.....	203
9.29	WFA_AV_CAPTURE.....	204
9.30	SNIFFER_MEDIA_CHECK.....	205
9.31	SNIFFER_PTS_CALC.....	206
9.32	WFA_MERGE_TRACE.....	206
9.33	SNIFFER_CHECK_DSCV_WINDOW.....	207
9.34	SNIFFER_GENERATE_HASH.....	209
9.35	SNIFFER_GENERATE_RAND_MACS.....	210
9.36	SNIFFER_CONTROL_SUBTASK.....	210
10	Common commands for AP/STA/PCP/Sniffer-Injector.....	211
10.1	DEV_SEND_FRAME.....	211
11	STA command line interface utilities.....	216

11.1	Error codes and return values.....	216
11.2	RESET_DEFAULT	216
11.3	SET_11N_CHANNEL_WIDTH	217
11.4	SET_40_INTOLERANT	217
11.5	SET_ADDBA_REJECT	218
11.6	SET_AMPDU	218
11.7	SET_AMSDU	219
11.8	SET_GREENFIELD	219
11.9	SET_MCS	220
11.10	SET_RIFS_TEST	220
11.11	SET_SGI20	221
11.12	SET_STBC_RX	221
11.13	SET_SMPS	222
11.14	SEND_ADDBA	222
11.15	SEND_COEXIST_MGMT	223
11.16	SET_NOACK	223
11.17	SET_TXSP_STREAM	224
11.18	SET_RXSP_STREAM	224
12	Remote power switch control API	225
12.1	POWER_SWITCH_CTRL	225
12.2	POWER_SWITCH_RESET	225
13	Wi-Fi Alliance Extended MAC Tester CAPI commands	226
13.1	WFAEMT_CONFIG_NAV	226
13.2	WFAEMT_START_NAV_TEST	226
13.3	WFAEMT_STOP_NAV_TEST	227
13.4	WFAEMT_CONFIG_STAUT_MIC	227
13.5	WFAEMT_START_STAUT_MIC_TEST	228
13.6	WFAEMT_STOP_STAUT_MIC_TEST	228
13.7	WFAEMT_CONFIG_APUT_MIC	229
13.8	WFAEMT_START_APUT_MIC_TEST	230
13.9	WFAEMT_STOP_APUT_MIC_TEST	230
14	Server control APIs	231
14.1	SERVER_SET_PARAMETER	231
14.2	SERVER_REQUEST_STATUS	233
14.3	SERVER_RESET_DEFAULT	236
14.4	SERVER_CA_GET_VERSION	237
14.5	SERVER_GET_INFO	238
15	WPS-NFC Commands	239
15.1	STA_ER_CONFIG	239
15.2	STA_NFC_ACTION	240
15.3	AP_NFC_ACTION	243
15.4	AP_WPS_READ_PIN	244
16	Command sequence for traffic generation and testing	245
Appendix A	Document revision history	246



List of Figures

Figure 1. Control Agent state machine..... 12

Figure 2. UCC standard configuration..... 14

Figure 3. UCC command sequence flow diagram 245

List of Tables

Table 1. List of acronyms 8

Table 2. Defined status values..... 10

Table 3. Wi-Fi Test Suite data types 13

Table 4. Document revision history..... 246

1 Introduction

This document describes the API that is used to control both Wi-Fi Alliance CERTIFIED™ Test-bed Devices and Device under Test (DUTs), and the communication protocol used to invoke the API between the Unified CAPI Command Console (UCC) and the Device Agent. Refer to the Sigma Test Environment System Architecture Document [1] for a complete description of the test framework and test bed.

The UCC runs a test program which orchestrates the test. At points in the test where the DUT is required to perform some action or the UCC needs information from the device, the UCC issues an appropriate command using one of the standardized API calls described in this document. The API command passes through the local network stack and arrives at the PC or system that runs the device control agent. Here, the command is processed by the device Control Agent program to produce a command or series of commands to control the device. The device then executes these commands and hence performs the action required for the test.

The UCC uses the same API and protocol to communicate with a Control Agent for a PC-based Traffic Agent.

CAPI commands may be either mandatory, optional, or not required depending on the platform component and certification program.

1.1 Audience

The intended audience for this document is engineering, marketing, sales and any other group or person with an interest in the development of DUT, Access Point (AP), or Station (STA) Control Agents.

1.2 Command matrix

Readers should become familiar with the Command Matrix spreadsheet before reading this document. The Command Matrix is a companion guide to this document. The Command Matrix will help readers identify the CAPI commands and parameters that are required for their particular Wi-Fi Alliance program. The command matrix may be found at <https://www.wi-fi.org/members/certification-testing/wi-fi-test-suite>.

The Command Matrix spreadsheet contains two sheets: one sheet lists the station (STA) commands and the other sheet lists the access point (AP) commands. The commands are listed vertically in alphabetical order. The program names are listed horizontally across the top of the sheet.

Commands that are mandatory for a particular program test plan are marked with "M" while commands that are optional are marked with "O". Parameters that are required for a particular command are marked "R", while parameters that may be used in the command are marked "NR" for not required. The Command Matrix spreadsheet may be sorted to display only the commands and parameters for a particular Wi-Fi Alliance program.

If a parameter is marked required by all programs, then future programs must maintain that designation to ensure backward compatibility.

1.3 Reference documents

All Wi-Fi Test Suite Documents may be found at <https://www.wi-fi.org/members/certification-testing/wi-fi-test-suite>.

[1] Wi-Fi Test Suite Getting Started Guide

[2] Wi-Fi Test Suite Release Notes

[3] Command Matrix Spreadsheet

[4] Sigma Test Environment System Architecture Document

[5] Wi-Fi CERTIFIED Passpoint™ Release 2 APUT Wi-Fi Test Suite Execution Guide

[6] Wi-Fi CERTIFIED Passpoint™ Release 2 STAUT Wi-Fi Test Suite Execution Guide

1.4 Acronyms

Table 1. List of acronyms

Acronym/Term	Definition
ADDBA	Add Block Acknowledgement
AMPDU	Aggregated Medium Access Control Protocol Data Unit
AMSDU	Aggregated Medium Access Control Service Data Unit
ASD	Application Specific Device – A non PC based Wi-Fi device
CA	Control Agent
DUT	Device under test
EMT	Extended MAC Tester
MIC	Message Integrity Check
NAN (Wi-Fi Aware™)	Neighbor Awareness Networking
P2P (Wi-Fi Direct®)	Peer to Peer
PMF	Protected Management Frame
PSW	Power
PLCP	Physical Layer Convergence Procedure
SMPS	Spatial Multiplexing Power Save
STA	Station
STAUT	Station under Test
STBC	Space-Time Block Code
TA	Traffic Agent
TCP	Transmission Control Protocol
TID	Traffic Identifier
UCC	Unified CAPI Command Console
UIBC	User Input Back Channel
VHT (Wi-Fi Certified™ ac)	Very High Throughput in 5 GHz
WFD (Miracast®)	W-Fi Display
WMM®	Wi-Fi Multimedia™
WMM-AC	Wi-Fi Multimedia™ Admission Control
WPA™	Wi-Fi Protected Access®
WPS	Wi-Fi Protected Setup™

2 Basic communication paradigm

The Control Agent performs a passive TCP open on a selected port and waits for the UCC to connect to it. The UCC is configured to perform an active TCP open to the Control Agent.

Once the TCP connection is established, any API calls made by the UCC are sent to the Control Agent as ASCII formatted commands over the TCP connection. The UCC must handle one, and only one, command at a time. Each command is terminated by a new-line (CR+LF). The Control Agent uses the same format for sending responses.

If the TCP connection terminates for any reason, the Control Agent should wait passively for the next connection attempt from the UCC.

A typical CAPI command sequence is shown below.

```
UCC: device_list_interfaces,interfaceType,802.11
CA: status,RUNNING
CA: status,COMPLETE,interfaceType,802.11,interfaceID,interfaceId_1 interfaceId_2
UCC: sta_get_info,interface,interfaceId_1
CA: status,RUNNING
CA: status,COMPLETE,vendorInfo_1,value_1,vendorInfo_2,value_2
```

Commands are executed sequentially. In this example, the UCC will not respond with “STA_GET_INFO” until the device CA responds with the completed device list.

Note that users need to append the characters line feed \r\n to every response so that a new line will separate the commands on the UCC console output or the logs.

3 Command and response syntax

3.1 Command syntax

To run a command on the Device Control Agent, the function name and any parameters are entered, separated by commas and terminated by a CR+LF ('Enter'). Some parameters are mandatory and are marked accordingly.

To aid in debugging, and to allow optional parameters with default values, each parameter consists of the parameter's name followed by the value, also comma separated. The general command format is:

```
<Command Name>,<Parameter 1 Name>,<Parameter 1 Value>[,...]
```

There is no escape character defined in this version of the protocol so tokens must not contain commas or new-lines. All command names, parameter names, and parameter values are not case-sensitive. An exception to this is the 'ssid' parameter in many commands, which is case-sensitive.

The following special characters should not be used:

```
$#&*()[]{};:.,^<>?|`~+=
```

Some APIs list the dependant commands that must be called prior to calling that API.

3.2 Response syntax

A command response consists of one or more Response Elements. Each Response Element consists of an Element Name and Element Value. These tokens are also separated by commas. The general response format is:

```
<Response Element 1>: <Status Value>
```

```
<Response Element 2>: <Element Value1>, <Element Value2>, [,]
```

The first Response Element token is always the status. For some commands, the status element is followed by other command-specific response elements. The status can have the values given in Table 2.

Table 2. Defined status values

Status Value	Description
RUNNING	The previous command was well-formed and the Control Agent is busy executing the command.
INVALID	The previous command was either malformed, or a parameter value was invalid or not understood. The Control Agent should use the errorCode Response Element to return more details on the error.
ERROR	The previous command produced errors when it was executed on the DUT. The Control Agent should use the errorCode Response Element to return more details on the error.
COMPLETE	The previous command was well-formed and the Control Agent has finished executing the command. The results of executing the command are also returned using zero or more command-specific response elements.

Commands and response strings will be less than WIFI_DUT_CONTROL_AGENT_MAX_LINE_LENGTH characters long. This maximum length will be 2048 bytes.

3.3 Response timeouts

After the UCC sends a command to the Control Agent, it waits for a response with a timeout length of one second.

The Control Agent must parse the command to verify that it is well-formed and return a response element within this timeout with a status 'RUNNING'.



If a response element with a status of RUNNING is received, the UCC will then block for a follow-up response from the Control Agent. The Control Agent shall return a status 'ERROR' or 'COMPLETE', or 'INVALID'. The timeout threshold for the Control Agent is 120 seconds unless specified otherwise in a particular command definition.

4 Control Agent state machine

Figure 1 illustrates the Control Agent State machine.

Once a connection has been made, the Control Agent waits until a new-line terminated text string is received. After a line of text is received, the values are parsed to ensure that the command name, parameter names, and parameter values are syntactically correct. If the command is syntactically correct, then a status of RUNNING is reported, and the vendor proprietary method of communication with the Device begins. Once the Device has finished executing the function, the appropriate response is sent, and the Control Agent waits to read the next line of input.

It is important that the Control Agent does not read additional lines from the TCP socket until each command completes in turn. This will allow several commands that are pasted into a telnet window at once to be executed sequentially as expected.

The communication between the Control Agent and the device should use an out-of-band mechanism because in-band communication such as using the wireless interface under test may not allow communication before the Device is associated.

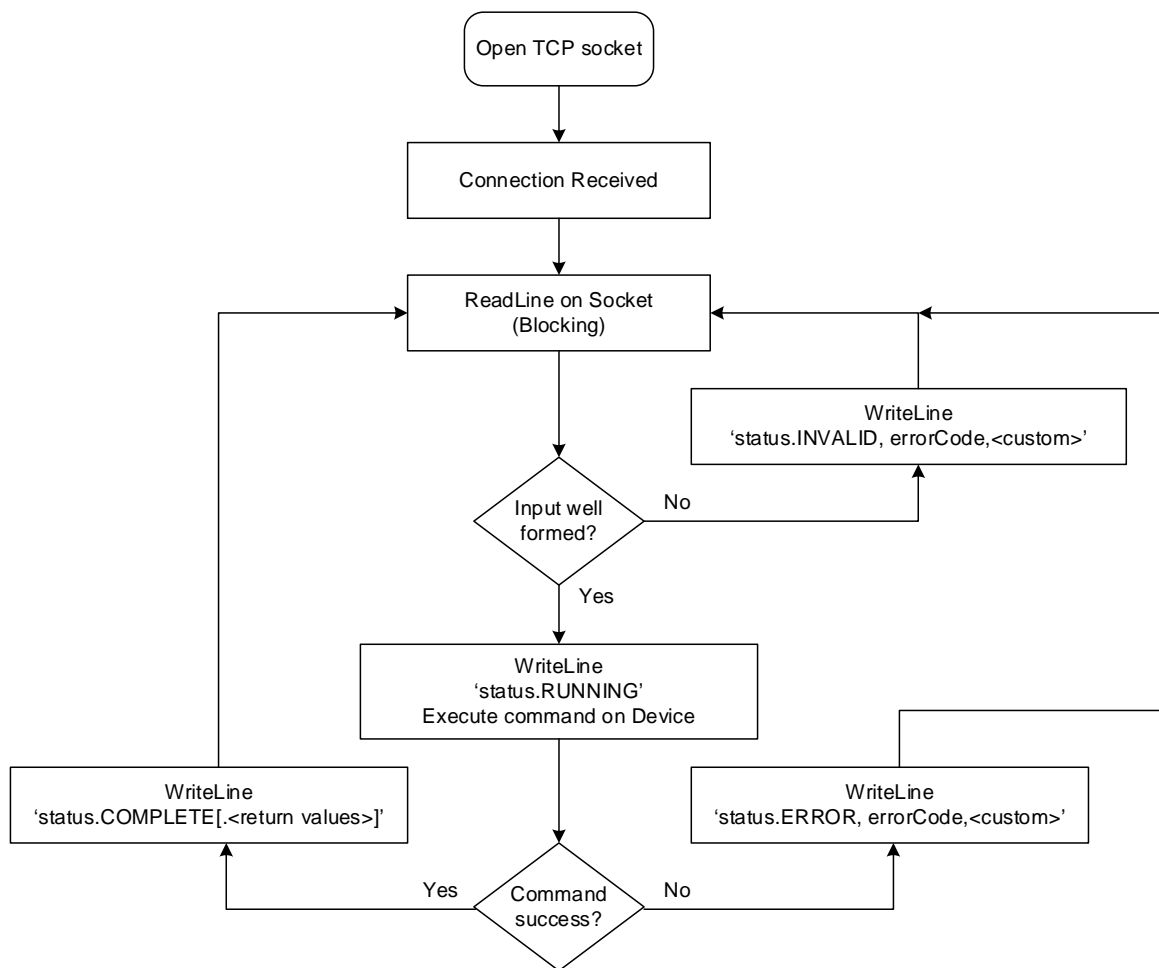


Figure 1. Control Agent state machine

5 Data types

All tokens, including command names, parameter names, parameter values, response element names and response element values, are ASCII strings. Common data types are represented using the conventions in Table 3.

Table 3. Wi-Fi Test Suite data types

Data Type	Representation
Boolean	ASCII Character '1' indicates True/Enabled, '0' indicates False/Disabled
Short Integer	Decimal String, such as 234 or -123
Hexadecimal String	String using the 0-9 and A-F characters, e.g 1B234A3467
IP Address	Dotted notation, such as 192.168.1.1
MAC Address	IEEE notation, such as 11:22:33:44:55:AA
String	String (any ASCII character)

List elements will be separated by a space and must not consist of any white-space characters themselves.

6 Unified CAPI Console (UCC)

Built on top of the Wi-Fi Test Suite architecture, the UCC provides another level of abstraction and automation to the Test Engine. It expands the CAPI capability to use the editable script command files as input to control both DUT and standard test-bed device test sequences. The extension greatly enhances the flexibility and usability of CAPI commands. This will benefit users who intend to adopt the Wi-Fi Test Suite in both test and development environments.

The UCC framework includes two key components: the UCC Core and UCC Command Scripts.

6.1 UCC core

The UCC core configuration is shown in Figure 2.

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 in order to drive test execution. The core consists of CAPI interpretation, command delivery, and result handling that stacks return information for use in other command parameters. It also processes final test returns and determines whether the test passes or fails based on pre-defined criteria.

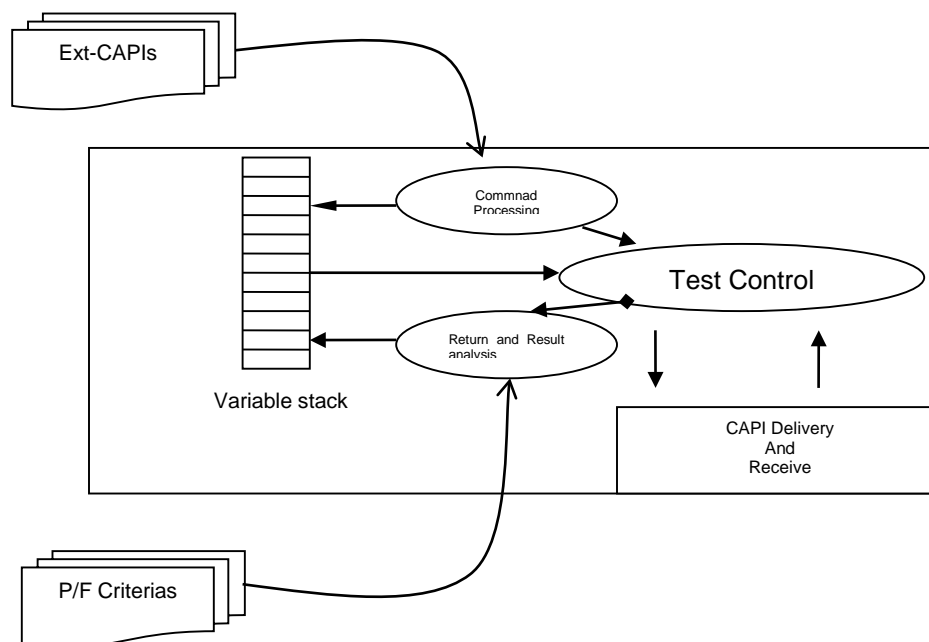


Figure 2. UCC standard configuration

The architecture gives flexible extensibility to “Command Processing” and “Result Analysis” by creating and adding new PYTHON object modules while “Test Control” remains unchanged.

UCC CORE normally takes two text files for test execution. The Ext-CAPI files define the initial setup for a particular test environment such as aliases for IP addresses or macros of frame rates that will be used throughout the entire test execution. The P/F criteria file(s) contains the necessary UCC commands to instruct the test bed devices and DUT to perform the specific test cases.

6.2 UCC command scripts

UCC command scripts are text based extended CAPI commands that are input to the UCC core. With an editable text format, test sequences can be easily changed and adjusted. New tests can be added through simple editing. Learning a small set of keywords is all that is required for command definition and test sequence control.

7 Station control API

The following is a list of API Commands required to configure and control a Station under Test (STAUT) or test bed STA for Wi-Fi Alliance certification.

7.1 CA_GET_VERSION

This command is used to obtain the version of the Control Agent software. The command does not attempt to perform any action on the DUT itself. This command may be used to verify basic connectivity to the Control Agent.

Parameters

None.

Return Values

Name	Description	Value
Version	Control Agent software version	String

Dependencies

None.

Example

```
UCC: CA_GET_VERSION, TestInfo, P2P-4.1.1_Jun-11-2010__12-07-38
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,version,1.0
```

7.2 CONNECT_GO_START_WFD

This command is used to connect to a specified device ID and group ID as a P2P Client and starts a WFD session. This command returns the WFD session ID (Part of the M6 message).

The command shall block until the STA is connected to the given GroupID and establishes the WFD session or returns with a valid result. The command shall implicitly run the Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit discovery is run then the STA shall go back to its previous state (for example Listen) after it discovers the required P2P device.

The command shall implicitly send the provisional discovery request if needed.

The command shall implicitly perform a capability negotiation and session establishment as needed.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PdevID and ssid <P2PdevID ssid>	String
Interface	Interface ID	String
P2PdevID	P2P Device ID The value of DevID shall be case insensitive.	String

Return Values

Name	Description	Value
WFDSessionID	WFD session ID. Returns the WFD session ID when the session is established or -1 if the session is not available on the other device.	String

Examples

```
UCC: connect_go_start_wfd,interface,wlan0,p2pdevid,00:11:22:33:44:55:66,groupid,00:11:22:33:44:55
DIRECT-9U-XYZ
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,WFDSessionId,9876543298
```


7.3 DEVICE_GET_INFO

This command returns the vendor, model and software version of the DUT.

The returned model and version will be truncated by the UCC when it creates the JSON log for the Test Management System.

Dependencies

None.

Parameters

Name	Description	Value
Name	AP device name This parameter is required in a VHT R2 AP Control agent implementation.	String AP Name Example: Broadcom11n/Atheros11n

Return Values

Name	Description	Value
Model	The device model number The string should not contain any blank spaces or special characters.	String
Vendor	Name of vendor The string should not contain any blank spaces or special characters.	String
Version	Version of the DUT's software interface that uniquely identifies the device software The string should not contain any blank spaces or special characters.	String

Example

UCC: device_get_info

CA: status,RUNNING

CA: status,COMPLETE,vendor,MyVendor,model,DutModel,version,1.16

UCC: device_get_info,NAME,abc11n

CA: status,RUNNING

CA: status,COMPLETE,vendor,APVendor,model,APModel,version,5.50

CCG: device_get_info

CA: status,RUNNING

CA: status,COMPLETE,vendor,QualcommAtheros,model,QCA123456_AR8905,version,SixtyGig_1234

7.4 DEVICE_LIST_INTERFACES

This command returns the interface type and a space separated list of interface identifiers.

Parameters

Name	Description	Value
InterfaceType	Type of interface	String For example, "802.11"

Return Values

Name	Description	Value
InterfaceID	Space separated list of interface IDs	String
InterfaceType	Type of interface	String For example, "802.11"

Example

```
UCC: device_list_interfaces,interfaceType,802.11
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,interfaceType,802.11,interfaceID,interfaceId_1 interfaceId_2
```

7.5 REINVOKE_WFD_SESSION

This command is used to re-invoke the WFD session that was previously established.

If the WFD session is using Wi-Fi Direct and the Invitation Action is send:

1. The command shall implicitly run Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit Discovery is run, then the STA shall go back to its previous state (for example. Listen) after it discovers the required P2Pdevice.
2. The command shall block until it receives the P2P Invitation Response from the other STA.
3. If re-invoking the session is successful, the WFD session shall be established before the command returns.

If the WFD session is using Wi-Fi Direct and the Invitation Action is accept:

1. The command shall configure the STA to accept the P2P Invitation Request.
2. [Optional] - The command is optional if device does not need manual intervention to accept the P2Pinvitation request.
3. If the command is implemented, it shall prepare the STA to accept the P2P Invitation Request and then return immediately. It shall not block while waiting for the P2P Invitation Request.
4. The command shall implicitly run Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit Discovery is run then the STA shall return to its previous state (for example Listen) after it discovers the required P2Pdevice.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PdevID and ssid <P2PdevID ssid>	String
Interface	Interface ID	String
InvitationAction	Invitation can be a send or accept.	String Send Accept
PeerAddress	Device address for P2P and station MAC address for TDLS	String

Return Values

None.

Examples

```
UCC: reinvite_wfd_session,interface,wlan0,groupid,00:11:22:33:44:55 DIRECT-9U-XYZ,peeraddress,22:33:44:55:66:77,invitationaction,send
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.6 STA_ADD_ARP_TABLE_ENTRY

This command is used to add the static Address Resolution Protocol (ARP) table entry. This command is only relevant for a test bed STA.

Note – Any configuration changes made by this command shall be cleared by the STA_P2P_RESET command.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid> GroupID is optional if the command is given to the Legacy (non-P2P) station.	String
Interface	Interface ID	String
IPAddress	IP address	String
MACAddress	MAC address	String

Return Values

None.

Example

```
UCC:
sta_add_arp_table_entry,interface,wlan0,macaddress,00:11:22:33:44:55,ipaddress,192.168.42.109,group
id, aa:bb:cc:dd:ee:ff DIRECT-9UABCD
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.7 STA_ADD_CREDENTIAL

This command is used to add credentials to the STA.

Parameters

Name	Description	Value
ClientCertificate	Name of the client (user) certificate configured on the STA	String
Home_FQDN	FQDN of Home Service Provider	String
IMSI	IMSI value	String
Interface	Interface ID	String
Password	Password	String
PLMN_MCC	PLMN MCC For multiple values with same SIM card, the format is a list of MNCs separated by semi-colon.	String
PLMN_MNC	PLMN MNC For multiple values with same SIM card, the format is a list of MNCs separated by semi-colon.	String
Prefer	Indicates if given credentials should be preferred by the STA	Integer 1 = Yes 0 = No (Default)
Realm	Realm name for corresponding credentials	String
Root_CA	Trusted root CA certificate	String
Type	Type of credentials	String uname_pwd sim cert rootcert
Username	Username	String

Return Values

None.

Examples

```
UCC:
sta_add_credential,interface,wlan0,type,uname_pwd,realm,mail.example.com,username,wifiuser,password
,test%11,root_ca,ca.pem
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_add_credential,interface,wlan0,type,sim,plmn_mcc,310,plmn_mnc,026,imsi,
131002600000000000,password,
90dca4eda45b53cf0f12d7c9c3bc6a89:cb9cccc4b9258e6dca4760379fb82581:000000000123
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

UCC:
sta_add_credential,interface,wlan0,type,cert,realm,mail.example.com,username,user,ROOT_CA,ca.pem,clientCertificate,user.pem,home_fqdn,example.com

CA:status,RUNNING

CA:status,COMPLETE

CCG: sta_add_credential,interface,wlan0,type,rootcert, ROOT_CA,ca.pem

CA:status,RUNNING

CA:status,COMPLETE

7.8 STA_ASSOCIATE

This command attempts to associate the DUT's specified wireless interface to the specified SSID. This command must always be called after a corresponding call to any of the station security configuration command STA_SET_ENCRYPTION, STA_SET_PSK, STA_SET_EAPTLS, STA_SET_EAPSIM, STA_SET_PEAP or STA_SET_IBSS. This command starts the association process and does not block until the DUT is associated.

Dependencies

Commands	Description
STA_SET_ENCRYPTION, STA_SET_PSK, STA_SET_EAPTLS/EAPTTLS/SIM/PEAP	Set the security before associating.
If WPS is on – STA_WPS_READ_PIN STA_WPS_ENTER_PIN	

Parameters

Name	Description	Value
BSSID	Basic service set identifier	String
Interface	Interface ID	String
SSID	Service set identifier of BSS or ESS	String
WPS	Indicates if WPS shall be used in association	String WPS

Return Values

None.

Example

```
UCC: sta_associate,interface,interfaceId_1,ssid,MyNetwork,wps,1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.9 STA_BLOCK_ICMP_RESPONSE

This command is used to block the ICMP responses for the requests coming from the given IP Address. This command is only relevant for a test bed STA.

Note – Any configuration change made by this command shall be cleared by the STA_P2P_RESET command.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid> GroupID is optional if the command is given to the Legacy (non-P2P) station.	String
Interface	InterfaceID	String
IPAddress	IP address for which ICMP responses shall be blocked	String

Return Values

None.

Example

```
UCC: sta_block_icmp_response,interface,wlan0,ipaddress,192.168.42.109,groupid, aa:bb:cc:dd:ee:ff
DIRECT-9UABCD
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```


7.10 STA_BSSID_POOL

This command is used to add a list of BSSIDs to the STA pool. The STA shall only use the BSSIDs specified in this command when performing any queries (GAS/ANQP).

Parameters

Name	Description	Value
BSSID_Filter	Enables or disables the BSSID filter	Integer 1 = Enabled 0 = Disabled
BSSID_List	Space separated list of BSSIDs	String
Interface	Interface ID	String

Return Values

None.

Examples

```
UCC: sta_bssid_pool,interface,wlan0,bssid_filter,1,bssid_list,00:11:22:33:44:55 11:22:33:44:55:66
CA:status,RUNNING
CA:status,COMPLETE,
```

7.11 STA_DISCONNECT

This command is used to disconnect the station with an AP.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
Interface	Interface ID	String

Return Values

None.

Examples

```
UCC: sta_disconnect, interface,wlan
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.12 STA_EXEC_ACTION

This command is used to specify the test parameters and turns on/off the functionality of the Station/Device.

Parameters

Name	Description	Value
5GOnly	Configures test bed STA to transmit beacons in 5 GHz band only. This is for test bed STA only.	Integer 1 = 5GHz only 0 = Normal operation
Band	Configures test bed STA to operate in the specified band. This is for test bed STA only.	String 24G = 2.4GHz only 5G = 5GHz only Dual = OOB band operation
FurtherAvailInd	Configures NAN device to transmit or receive further availability indication attributes in SDFs	String Tx Rx
HighTSF	Configures the NAN device to set its TSF to a value >= 0x00000001FF000000	String On Off
HopCount	4-bit hex value, hop count to Anchor Master	Hex value: 0x0 – 0xF
IncludeBit	Set include bit	Integer 0 = Not included 1 = Included
Interface	Interface ID	String
LocalInstanceID	Local ID	Integer
MAC	List of space separated MAC addresses, up to maximum of 10.	String
MasterPref	8-bit hex value that sets the device Master Preference.	Hex value: 0x00 – 0xFF
NANOp	Configure the NAN device to start or stop NAN operation.	String On Off
Prog	Certification program name	String NAN
RandFactor	8-bit hex uniformly distributed random number. NOTE: the STA must not change the random factor after the setting. This setting does NOT apply to any DUT.	Hex value: 0x00 – 0xFF
MethodType	Method type	String Publish Subscribe Followup
	Mandatory Parameters for MethodType = Publish	
	Name	Description
	PublishType	Configures publish type or cancels publish
	ServiceName	Name of the service
		String

Name	Description	Value
Optional Parameters for MethodType = Publish		
Name	Description	Value
SDFDelay	Configure the device to delay sending the SDF frame 3- 4.5ms into the DW. This parameter is for a test bed STA only.	Integer 1 =Delay 0 =No delay
RxMatchFilter	String for positive matches, random string for failed matches used for a solicited publish Each string represents the value of the match filter. Wildcard value is represented with an asterisk (*) with a length of zero.	String One or more colon separated strings
TxMatchFilter	String for positive matches, random string for failed matches used for an unsolicited publish Each string represents the value of the match filter. Wildcard value is represented with an asterisk (*) with a length of zero.	String One or more colon separated strings
DiscRangeLtd	Configures the discovery range limitation for a given Publish function	Integer 0 = Without 1 = With
DiscRangeIgnore	Configures the device to ignore or respond to the Discovery Range Limited bit setting in a received SDF. This parameter is for a test bed STA only.	Integer 0 = Respond 1 = Ignore
Mandatory Parameters for MethodType = Subscribe		
Name	Description	Value
SubscribeType	Configures subscribe type or cancels subscribe	String Active Passive Cancel
ServiceName	Name of the service	String
Optional Parameters for MethodType = Subscribe		
Name	Description	Value
SDFDelay	Configure device to delay sending SDF frame 3- 4.5ms into the DW, 1=delay,0=no delay; this is for test bed STA only	Integer
RxMatchFilter	String for positive matches, random string for failed matches, used for passive subscribe Each string represents the value of the match filter. Wildcard value is represented with an asterisk (*) with a length of zero.	String One or more colon separated strings
TxMatchFilter	String for positive matches, random string for failed matches, used for active subscribe Each string represents the value of the match filter. Wildcard value is represented with an asterisk (*) with a length of zero.	String One or more colon separated strings
IncludeBit	Set include bit	Integer 0 = Not included 1 = Included
SRFType	Set SRF type	Integer 0 = MAC List 1 = Bloom Filter

Name	Description		Value
	DiscRangeLtd	Configures the discovery range limitation for a given subscribe function	Integer 0 = Without 1 = With
	DiscRangeIgnore	Configures device to ignore or respond to the Discovery Range Limited bit setting in a received SDF. This parameter is for test bed STA only.	Integer 0 = Respond 1 = Ignore
	Mandatory Parameters for MethodType = Followup		
	Name	Description	Value
	RemoteInstanceID	Remote ID tra	Integer
	LocalInstanceID	Local ID	Integer
	MAC	NAN Interface address to follow up with	String

Return Values

Name	Description	Value
MAC	MAC address of the wireless interface after enabling/turning on NAN	MAC address

Examples

```
UCC:sta_exec_action,interface,wlan0,prog,nan,nanop,on
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE,mac,11:22:33:44:55:66
```

Example with 3 length value pairs in rxMatchFilter:

```
UCC: sta_exec_action,interface,wlan0,prog,nan,servicename,org.wi-fi.nan.test,methodtype,publish,publishtype,solicited,rxmatchfilter,A:BBB:CC
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

Example with wildcard as a length value pair in txMatchFilter:

```
UCC: sta_exec_action,interface,wlan0,prog,nan,servicename,org.wi-fi.nan.test,methodtype,publish,publishtype,solicited,txmatchfilter,A:*:CCC
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

Example follow up with RemoteInstanceID LocalInstance MAC:

```
UCC: sta_exec_action,interface,wlan0,prog,nan,methodtype,followup 13 2 aa:bb:cc:dd:11:22
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

Example list of 6 MAC addresses for SRF filters

```
UCC: sta_exec_action,interface,wlan0,prog,nan,servicename,org.wi-fi.nan.test,methodtype,subscribe,subscribetype,active,includebit,1,mac, 68:f7:28:65:c8:97
5c:c5:d4:a3:04:9e 5c:26:0a:51:0a:5c 5e:c5:d4:a3:04:9e fc:f8:ae:51:57:df 00:24:d7:a4:de:29,srftype,0
```

CA:status, RUNNING

CA:status, COMPLETE

7.13 STA_GENERATE_EVENT

This command is used to generate an event that may include multiple steps as defined by the passed parameters.

This command generates a single motion input event or multi touch event on the test bed device when type is UIBC. This command is mandatory for test bed devices and DUT sinks that implement the UIBC feature.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
FrameSkip	Start Frame Skipping. This may be done by changing the input content to non-moving content.	String Start
I2C_Struct	I2C command structure data as specified in test plan.	String Ex: 0x01500100
InputContentType	Specifies the input content that is to be rendered during WFD session. This option is only valid for a source. Protected –Audio and Video encryption. Unprotected – No encryption required. ProtectedVideoOnly – only video encryption is required.	String Protected Unprotected ProtectedVideoOnly
Interface	InterfaceID	String
Program	Certification program name	String WFD
Type	Event type. Generic or HID UIBC, Video frame skipping, IChange input content, I2C Read request over TCP port, I2C write request, Cause IDR request to be sent	String UIBC_Gen UIBC_HID FrameSkip InputContent I2CRead I2cWrite IdrReq
UIBC_Prepare	This option is only valid for a source. When received, the source shall prepare to receive the specified UIBC Generic/HID event from the sink.	String KeyBoard Mouse
UIBCEventType	UIBC event type for Generic or HID Input devices. When UIBCEventType with KeyBoard is received by sink, it shall generate event with 100 characters. When UIBCEventType with Mouse is received by sink, it shall generate event where the ending coordinates are different from starting position.	String KeyBoard Mouse

Return Values

None.

Example

UCC: sta_generate_event,interface,wlan0,type,UIBC,UibcEventType,SingleTouch

CA: status,RUNNING

CA: status,COMPLETE

7.14 STA_GET_BSSID

This command is used to obtain the BSSID of the Ad Hoc or Infrastructure network with which the DUT's specified wireless interface is currently associated, or 00:00:00:00:00:00 if the station is not currently associated.

Dependencies

Commands	Description
STA_IS_CONNECTED	Check whether the device is associated

Parameters

Name	Description	Value
Interface	Interface ID	String

Return Value

Name	Description	Value
BSSID	Basic service set identification	MAC address

Example

```
UCC: sta_get_bssid,interface,interfaceId_1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,bssid,11:22:33:44:55:66
```


7.15 STA_GET_EVENT_DETAILS

This command is used to retrieve details about the requested event.

Parameters

Name	Description	Value
EventName	This optional parameter is mandatory for the WFDs program.	String SearchResult SearchTerminated AdvertiseStatus SessionRequest ConnectStatus SessionStatus PortStatus SessionConfigRequest
Interface	Interface ID	String
Program	Program name	String WFDS

Return Values per Event Type

EventType	Name	Description	Value
SearchResult	SearchId		Hex Ex. 0000000A
	Service_MAC	MAC address of remote peer device	String Ex: 00:11:22:33:44:55
	AdvID	Advertisement ID of service found on remote peer	Hex Ex. 0000000A
	Service_Name	Exact name of service discovered on remote peer	String Ex: org.wi-fi.wfds.send.rx
	Service_Status	Service status	Integer 0 = Service not available 1 = Service available
SearchTerminated	SearchID	Identifier of which search is terminated	Hex Ex. 00000005
AdvertiseStatus	AdvID	Identifier of advertisement	Hex Ex. 0000000A
	Status	Status of the advertisement	String Advertised NotAdvertised
SessionRequest	AdvID	Advertisement ID of service found on remote peer	Hex Ex. 0000000A
	Session_MAC	MAC address of remote peer device	String Ex: 00:11:22:33:44:55
	Session_ID	Session identifier assigned by remote peer	Hex Ex. 0000000A
	GetNwCfgPIN	Get_network_config_PIN flag (boolean value)	Integer 0 = False 1 = True

EventType	Name	Description	Value
SessionStatus	NwCfgPIN	PIN that ASP generates to enter on the peer device	String
	Session_MAC	MAC address of session initiator	String Ex: 00:11:22:33:44:55
	Session_ID	Session identifier assigned by session initiator	Hex Ex. 0000000A
	State	State of the session	String OPEN INITIATED REQUESTED CLOSED
ConnectStatus	Session_MAC	MAC address of session initiator	String Ex: 00:11:22:33:44:55
	Session_ID	Session identifier assigned by session initiator	Hex Ex. 0000000A
	Status	Status of group formation and service request	String NetworkRoleRejected ServiceRequestReceived ServiceRequestDeferred ServiceRequestAccepted ServiceRequestFailed GroupFormationStarted GroupFormationComplete GroupFormationFailed
PortStatus	Session_MAC	MAC address of session initiator	String Ex: 00:11:22:33:44:55
	Session_ID	Session identifier assigned by session initiator	Hex Ex. 0000000A
	Port	Port number	Integer
	Status	Status of this port	String LocalPortAllowed LocalPortBlocked Failure RemotePortAllowed
SessionConfigRequest	Session_ID	Session identifier assigned by remote peer	Hex Ex. 0000000A
	GetNwCfgPIN	Get_network_config_PIN flag (boolean value)	Integer 0 = False 1 = True
	NwCfgPIN	PIN that ASP generates to enter on the peer device	String

Examples

```
CCG: sta_get_event_details, interface, wlan0, program, wfds, eventname, SearchResult
```

```
CA: status, RUNNING
```

```
CA:
status, COMPLETE, SearchId, 0000000b, Service_mac, 00:11:22:33:44:55, AdvId, 0000000a, Service_name, org.wi-fi.wfds.send.rx, Service_status, 1
```

CCG: sta_get_event_details,interface,wlan0,program,wfds,eventname,SessionStatus

CA: status,RUNNING

CA: status,COMPLETE,Session_id,0000000c,Session_mac,00:11:22:33:44:56,State,INITIATED

7.16 STA_GET_EVENTS

This command is used to retrieve the latest events from the device. The total size of the report is limited to no more than 2000 bytes.

Parameters

Name	Description	Value
Action	Start, Stop or Get collection of events. This optional parameter is required for the NAN program. Start tells the device to start the logging of events, Stop tells the device to stop logging events, and Get tells the device to return a space separated list of NAN events as described below.	String Start Stop Get
Interface	Interface ID	String
Program	Program name. When Program name is WFDS - Retrieve the latest WFDS events that are reported from ASP When Program name is NAN – Retrieve the latest NAN events that are reported by the NAN Engine	String WFDS NAN

Return Values

Name	Description	Value
EventName	Name of the reported NAN event	String DiscoveryResult Replied PublishTerminated SubscribeTerminated FollowUpReceive
EventList	Space separated list of WFDS ASP events received since reset or most recent STA_GET_EVENTS command	String SearchResult SearchTerminated AdvertiseStatus SessionRequest ConnectStatus SessionStatus PortStatus SessionConfigRequest
RemoteInstanceID	Requestor Instance ID of the remote device	Integer
LocalInstanceID	Device's Instance ID	Integer
MAC	MAC address of device	String Ex: 00:11:22:33:44:55

Examples

Start logging events

```
CCG: sta_get_events, interface, $DUT_IF, program, NAN, Action, Start
```

```
CA: status, RUNNING
```

```
CA: status, COMPLETE
```

No events to report

CCG: sta_get_events,interface,wlan0,program,NAN,Action,Get

CA: status,RUNNING

CA: status,COMPLETE

CCG: sta_get_events,interface,\$DUT_IF,program,WFDS

CA: status,RUNNING

CA: status,COMPLETE,EventList,SessionStatus AdvertiseStatus MyAspStatus

Get latest events and report a service discovery with RemoteInstanceID LocalInstanceID MAC

CCG: sta_get_events,interface,wlan0,program,NAN,Action,get

CA:status,RUNNING

CA: status,COMPLETE,EventName,DiscoveryResult,RemoteInstanceID,13, LocalInstanceID,2,mac,
aa:bb:cc:dd:11:22

7.17 STA_GET_INFO

This command is used to return vendor specific information about the DUT's specified wireless interface.

Dependencies

Commands	Description
Device_list_interfaces	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
Interface	Interface ID	String

Return Values

This command returns vendor specific information about the interface (pairs of 'name' and 'value').

Example

```
UCC: sta_get_info,interface,interfaceId_1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,vendorInfo_1,value_1,vendorInfo_2,value_2
```

7.18 STA_GET_IP_CONFIG

This command is used to obtain the IP configuration for the DUT's specified wireless interface.

This command shall block until the STA gets the IP configuration or a command timeout occurs as specified in section 3.3.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
GroupID	Group Identifier This optional parameter is mandatory for the P2P program.	String
Interface	Interface ID	String
Type	If Type value is not specified then the default shall be IPv4. If the Type value is not supported by the STA then the command shall return ERROR.	Integer 1 = IPv4 (Default) 2 = IPv6

Return Values

This command returns the following IP configuration information for IPv4.

Name	Description	Value
DHCP	Specifies if DHCP is enabled	Boolean
IP	The IP address in use	IP address
Mask	The subnet mask in use	IP address
Primary-DNS	The primary DNS address	IP address

This command returns the following IP configuration information for IPv6.

Name	Description	Value
IP	IPv6 IP address	String

Example

```
UCC: sta_get_ip_config,interface,interfaceId_1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,dhcp,0,ip,192.168.1.101,mask,255.255.255.0,
```

```
primary-dns,192.168.1.1,secondary-dns,192.168.1.101
```

7.19 STA_GET_MAC_ADDRESS

This command is used to return the MAC address of the DUT's specified wireless interface.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Retrieve a list of device interface names

Parameters

Name	Description	Value
interface	Interface ID	String

Return Values

Name	Description	Value
MAC	MAC address of the wireless interface	MAC address

Example

```
UCC: sta_get_mac_address,interface,interfaceId_1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,mac,11:22:33:44:55:66
```

7.20 STA_GET_P2P_DEV_ADDRESS

This command retrieves the device address of the Peer-to-Peer (P2P) device. Refer to the Wi-Fi Test Suite Getting Started Guide for the intended Wi-Fi Test Suite command flow for typical P2P test cases.

Parameters

Name	Description	Value
Interface	InterfaceID	String

Return Values

Name	Description	Value
DevID	P2P Device ID The value of DevID shall be case insensitive.	String

Example

```
UCC: sta_get_p2p_dev_address,interface,wlan0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,DevID,00:11:22:33:44:55
```


7.21 STA_GET_P2P_IP_CONFIG

This command retrieves the IP configuration of the P2P interface for the given GroupID. This command shall block until the STA (GO or Client) gets the IP configuration.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	InterfaceID	String

Return Values

Name	Description	Value
DHCP	Specifies if DHCP is enabled	Boolean
IP	The IP address in use	IP address
Mask	The subnet mask in use	IP address
Primary-DNS	The primary DNS address	IP address
P2PInterfaceAddress	The P2P Interface address used by the device for specified P2P group.	String

Example

UCC: sta_get_p2p_ip_config,interface,wlan0,groupid,aa:bb:cc:dd:ee:ff DIRECT-9UABC

CA: status,RUNNING

CA: status,COMPLETE,dhcp,1,ip,192.168.1.101,mask,255.255.255.0,

primary-dns,192.168.1.1,p2pinterfaceaddress,12:34:56:78:90:11

7.22 STA_GET_PARAMETER

This command retrieves the requested parameter information and sends the information in the response.

Parameters

Name	Description	Value
Interface	Interface ID	String
Parameter	<p>AID: the AID assigned to the STA by the PCP/AP</p> <p>DiscoveredDevList: Discovered devices DeviceID or MAC address are framed in space separated list and return</p> <p>MasterRank: retrieve the Master Rank value from the NAN device</p> <p>MasterPref: retrieve the Master Preference value from the NAN device</p> <p>RandFactor: retrieve the random factor value from the NAN device</p> <p>HopCount: retrieve the Hop Count value from the NAN device</p> <p>BeaconTransTime: retrieve the Beacon Transmission Time value from the NAN device</p> <p>ListenChannel: P2P Listen channel of the device</p> <p>NANStatus: retrieve the NAN status (on/off) from the NAN device</p> <p>OpenPorts: List of all open TCP or UDP ports framed in space separated list after running port checker tool for the specified Peer IP. The PortList will be generated by UCC and will have the mandatory ports status as per test plan, well known ports, DUT declared ports along with some random 50 ports in 1-65536.</p> <p>WSCPIN: WSC PIN value for WSC Config method to setup the WFDS session. The ASP shall generate this PIN and provide to service for displaying.</p> <p>P2P_Result: P2P group formation result</p> <p>GroupID: P2P Group ID</p>	<p>String</p> <p>AID</p> <p>DiscoveredDevList</p> <p>MasterRank</p> <p>MasterPref</p> <p>RandFactor</p> <p>HopCount</p> <p>BeaconTransTime</p> <p>ListenChannel</p> <p>NANStatus</p> <p>OpenPorts¹</p> <p>WSCPIN</p> <p>P2P_Result¹</p> <p>GroupID¹</p>
PeerIP	Peer IP address to find the open ports	String
PortList	<p>Space separated list of ports.</p> <p>The PortList will be generated by UCC and will have the mandatory ports status as per test plan, well known ports, DUT declared ports along with some random 50 ports in 1-65536.</p>	
Program	Program name	<p>String</p> <p>WFD</p> <p>WFDS</p> <p>NAN</p> <p>P2PNFC</p> <p>VHT</p> <p>60GHz</p>
<p>Notes:</p> <p>1. This value is only for a test bed device and will block until the device has completed Group formation or had valid results.</p>		

Return Values

Name	Description	Value
AID	The hex value of the AID,	Hex (with 0x)
DeviceList	<p>Return the discovered devices DeviceID or MAC address in a space separated list.</p> <p>DeviceID is only applicable to the WFD program.</p> <p>For WFDS, this the return value from DiscoveredDevList.</p>	String
MasterRank	Return the Master Rank value from the NAN device	<p>Hex value</p> <p>0x00 – 0xFF</p>
MasterPref	Return the Master Preference value from the NAN device	Hex value

Name	Description	Value
		0x00 – 0xFF
RandFactor	Return the Random Factor value from the NAN device	Hex value 0x00 – 0xFF
HopCount	Return the Hop Count value from the NAN device	Hex value 0x00 – 0xFF
NANStatus	Return the state of NAN function on/off from the NAN device	String On Off
ListenChnl	Listen channel of the P2P device	Integer
RemoteOpenPortList	List of all Remote peer device open TCP or UDP ports framed in space separated list and return. For WFDS, this the return value from OpenPorts.	String
WSCPIN	Displayed WSC PIN. For WFDS, this the return value from WSCPIN.	String
P2P_Result	P2P group formation result. For WFDS, this the return value from P2P_Result.	String Go Client FAIL
GroupID	P2P Group ID. For WFDS, this the return value from GroupID.	String

Examples

CCG: sta_get_parameter, interface, wlan0, program, wfd, Parameter, DiscoveredDevListDiscoveredDevListCA:
status, RUNNING

CA: status, RUNNING

CA: status, COMPLETE, DeviceList, 00:11:22:33:44:55 00:01:02:03:04:05

CCG:
sta_get_parameter, interface, wlan0, program, wfd, Parameter, OpenPorts, peerip, 192.168.250.101, portlist,
1 10 20 67 9700 200 9000 9500 65000

CA: status, RUNNING

CA: status, COMPLETE, OpenPortList, 22 139 445 68 9700

7.23 STA_GET_PSK

This command retrieves the passphrase information for WPA2-PSK security.

NOTE- This command is only required by the Group Owner.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	Interface ID	String

Return Values

Name	Description	Value
PassPhrase	The pass phrase used to generate the pre-shared key	String
SSID	Service set identification	String

Example

```
UCC: sta_get_psk,interface,wlan0,groupid,aa:bb:cc:dd:ee:ff DIRECT-9UABCD
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,passPhrase,123456789,ssid,DIRECT-9UABCD
```

7.24 STA_HS2_ASSOCIATE

This command is used to associate the STA to the appropriate AP after transmitting the required GAS queries per station's implementation. STA should consider updated PPSMO received (if any) from OSU Server for selection of AP. The station shall return the SSID and BSSID of the AP it gets associated to.

The provisioning commands that configure the STA shall be sent before this command, if required.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device
STA_ADD_CREDENTIAL	Set the security before associating

Parameters

Name	Description	Value
Ignore_Blacklist	Ignore blacklist	Integer 1 = STA device tries to associate with AP listed in blacklist 0 = STA device do not try association with blacklisted AP (default)
Interface	Interface ID	String

Return Values

Name	Description	Value
SSID	Service set identification	String
BSSID	Basic service set identification	String

Examples

```
UCC: sta_hs2_associate, interface,wlan0
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE,SSID,HotspotAP1,BSSID,00:11:22:33:44:55
```

7.25 STA_INVOKE_COMMAND

This command sets the required parameters and invokes the WFDS ASP primitive / session commands. This command will return after completion of the primitive and waits for the corresponding event if needed.

Parameters

Name	Description	Value
Command_Type		String Primitive
Interface	Interface ID	String
Primitive_Type	<p>Describes the type of primitive being invoked. This parameter is only mandatory if Command_Type = Primitive.</p> <p>Advertise will wait for an AdvertiseStatus event and returns the Adv_ID.</p> <p>Seek is not a blocking call, it may return immediately even without a SearchResult event.</p> <p>ConnectSession scenarios: Except for enable service when a session uses the WFDS Default Configuration Method, ConnectSession returns with valid session_Id and waits until the sessionstatus event with session state is open/closed, the GO result and role. When session uses WSC PIN Method – ConnectSession returns with a valid session_Id and P2P_Result as PROVISION with when ASP triggers the SessionConfigRequestEvent to service.</p> <p>ConfirmSession scenarios: When a session uses the WSC PIN Method, ConnectSession returns with a valid session_Id and waits until the sessionstatus event with session state is open/closed, the GO result and role.</p>	String Advertise Seek Cancel ConnectSession ConfirmSession SetSessionReady BoundPort ServiceStatusChange CloseSession
Prog	Program name	String WFDS

Parameters per Primitive_Type

Primitive_Type	Name	Description	Value
Advertise	Service_name	Exact service name of the service to be advertised	String org.wi-fi.wfds.send.tx org.wi-fi.wfds.send.rx
	Auto_accept	Auto_accept value	Integer 0 = Deferred PD Exchange 1 = Auto Accept 2 = Default (OOB)
	Service_info	Service information	String Null <Ascii string format of service_info>
	Service_status	Service Status	Integer 0 = Service not available 1 = Service available
Seek	Service_name	ExactService name of the service to be sought.	String Org.wi-fi.wfds.send.rx
	Exact_search	Exact Search for service name If Exact search is false, then the value of service name will be org.wi-fi.wfds	Integer 0 = False 1 = True
	MAC_address	MAC Address (P2P Device Address)	String For example:

Primitive_Type	Name	Description	Value
			Null 00:11:22:33:44:55
	Service_Info	Service information request	String GAS Null
Cancel	CancelMethod	Method name	String Seek
ConfirmSession	Session_MAC	MAC address	String
	Session_ID	Session ID	Hex Ex: 0000000A
	Confirmed		Integer 0 = False 1 = True
ConnectSession	Service_MAC	Advertiser service MAC address	String
	AdvID	Advertisement ID	Hex Ex: 0000000A
	Session_info	Session Information specific to individual service	String
	Network_role	Network Role	Integer 0 = Don't care 1 = Group Owner
	ConnectionCapabilityInfo	GO or Client connection capability	String GO CLI NewGO New CliGO
SetSessionReady	Session_MAC	MAC address	String
	Session_ID	Session ID	Hex Ex: 0000000A
BoundPort	Session_MAC	MAC address	String
	Session_ID	Session ID	Hex Ex: 0000000A
	Port	Port number	Integer
ServiceStatusChange	AdvID	Advertisement ID	Hex Ex: 0000000B
	ServiceStatus	Service Status	Integer 1 = Available 0 = UnAvailable
CloseSession	Session_MAC	MAC address	String
	Session_ID	Session ID	Hex Ex: 0000000A
Advertise	Network_Config	Preferred WSC config method	Integer 1 = WFDS default / PIN 2 = PIN only
Cancel	SearchID	Search ID or List of Search ID's	Hex Ex: 0000000A
ConfirmSession	Network_Config_PIN	WSC PIN value	String

Primitive_Type	Name	Description	Value
ConnectSession	SSID	SSID for WFDS devices if it starts AutoGO	String
	Oper_Chnn	Operating Channel for WFDS devices if it starts AutoGO	Integer
	Network_Config	Preferred WSC config method	Integer 1 2

Return Values per Primitive_Type

Primitive_Type	Name	Description	Value
Advertise	ServName	An exact service name	String
	AdvID	Advertisement ID corresponding to the service defined by serv_name parameter	Hex Ex. 0000000A
	Service_MAC	Service MAC corresponding to the service defined by serv_name parameter	String
Seek	SearchID	Search ID for the seek operation	Hex: Ex. 0000000A
ConfirmSession	Session_ID	The same Session ID that was created by seeker and used as part of ConnectSession method call. Will be returned even if P2P Result is FAIL.	HEX: Ex. 0000000A
	P2P_Result	P2P group formation result	String Go Client FAIL
	GroupID	P2P Group ID	String
ConnectSession	Session_ID	The same Session ID that was created by seeker and used as part of ConnectSession method call. Will be returned even if P2P Result is FAIL.	HEX: 0000000a
	P2P_Result	P2P group formation result	String Go Client PROVISION FAIL
	GroupID	P2P Group ID	String

Examples

CCG:
sta_invoke_command,interface,\$DUT_IF,prog,wfds,command_type,primitive,primitive_type,seek,service_name,\$service_name,service_role,rx,exact_search,1,mac_address,null,service_info,null

CA:status,RUNNING

CA:status,COMPLETE,searchID,<integer>

CCG:
sta_invoke_command,interface,\$STA1_IF,prog,wfds,command_type,primitive,primitive_type,advertise,service_name,\$service_name,auto_accept,default,service_status,\$service_status,service_info,\$service_info

CA: status,RUNNING

CA:status,COMPLETE,servicename, send display print play,advid,00000000 00000001 00000002 00000003

7.26 STA_IS_CONNECTED

This command verifies that the DUT's specified wireless interface is connected. This command returns 'connected' as '1' if the DUT has an 802.11 Association with an Ad hoc or Infrastructure network, regardless of the 802.1x Authenticated state of the connection.

Dependencies

Commands	Description
STA_ASSOCIATE	Device must associate an AP

Parameters

Name	Description	Value
Interface	Interface ID	String

Return Values

Name	Description	Value
Connected	State of the 802.11 association	Integer 0 = Not associated 1 = Associated

Example

```
UCC: sta_is_connected, interface, interfaceId_1
```

```
CA: status, RUNNING
```

```
CA: status, COMPLETE, connected, 1
```

7.27 STA_MANAGE_SERVICE

A seeker device is expected to perform the following on reception of this command:

- Connect Session if there is no existing P2P connection
- P2P Group Formation
- ASP Session establishment including BoundPort and Application socket connection
- Service level operation that follows the sequence of actions as specified in this command

This command returns after all the actions specified in the Manage_Actions parameter are completed.

A receiver device is expected to return after all the actions specified in Manage_Actions parameter are completed.

Parameters

Name	Description	Value
AdvID	Advertisement ID	String Ex: 00000001
ConnectionCapabilityInfo	GO or Client connection capability	String GO CLI NewGO New CliGO
Interface	Interface ID	String
Manage_Actions	Space separated list of actions that shall be performed sequentially for a Session. Each action to be performed is defined below. Transfer: Transfer File/Filelist Pause: Pause Send File Transfer Resume: Resume Send File Transfer Modify: Modify the ongoing Send Session Cancel: Cancel File Send / Transfer / Print amidClose: Close Send Session during file transfer Close: Close theSession Receive: Receive File/FileList/Printfile/play content/Render content Play: Send/Stream file Display: Screen share GetPrinterAttr: invokes the Get-Printer-Attribute PrintJobOperation: Print Job Operation GetJobAttr: Invoke the Get-Job-attributes operation CreateJobOper: Invoke the Create Job Operation SendPrintDoc: Invoke Send Document Operation DoNothing: Wait for > 2 mins to make the device idle	String Examples of Send Tx: Transfer Pause Resume Close Transfer Modify Close Transfer amidClose
Network_Role	Network Role	Integer
Prog	Type of credentials	String WFDS
Send_FileList	Filename or container name. List can be a space separated list of files or container name	String Ex: Icon.jpg<space>abc.txt
SendModify_FileList	Filename or container name to be part of modifySendSession List can be a space separated list of files or container name	String Ex: Icon.jpg<space>abc.txt

Name	Description	Value
Service_MAC	MAC Address (P2P device Address) of remote peer device	String Ex: 00:11:22:33:44:55
Service_Name	Name of service to be managed on device. This optional parameter is required for WFDs.	String Send Play Print Display
Service_Role	Service role. This optional parameter is required for WFDs.	String TX RX
Session_Info	Session Information specific to individual service	String

Return Values

Name	Description	Value
Session_ID	Session ID	String
P2P_Result	P2P group formation result	String Go Client FAIL
GroupID	P2P Group ID	String
Checksum	MD5 hash value (128 bit) of the file(s) in space separated list. When receiving/transmitting multiple files, the CheckSum shall have space separated list of Check sum for all the received/transmitted files in the same order that they were received/transmitted.	String: Ex: 30d3c5cd73d80c8ab9110a1c41cf9e77

Examples

CCG:
sta_manage_service,interface,\$DUT_IF,prog,wfds,service_name,send,service_role,tx,service_mac,00:11:22:33:44:55,advID,00000001,session_info,<String>,network_role,0, connectionCapabilityInfo,new, Manage_Actions,Transfer Pause Resume Close, Send_FileList,icon.jpg abc.txt

CA: status,RUNNING

CA:
status,COMPLETE,Session_id,<String>,P2P_Result,GO,GroupID,<String>,Checksum,30d3c5cd73d80c8ab9110a1c41cf9e77

CCG:
sta_manage_service,interface,\$DUT_IF,prog,wfds,service_name,send,service_role,rx,service_mac,00:11:22:33:44:55,connectionCapabilityInfo,new,Manage_Actions,Receive Pause Resume Close

CA: status,RUNNING

CA:
status,COMPLETE,Session_id,<String>,P2P_Result,GO,GroupID,<String>,Checksum,30d3c5cd73d80c8ab9110a1c41cf9e77

7.28 STA_OSU

This command initiates the OSU procedure described below on the station. The command shall have a timeout value of 120 seconds. If a timeout occurs before the OSU procedure is complete, the command returns with ERROR.

OSU Procedure:

1. STA shall scan and recognize that OSU is available based on ANQP Query. If provided, an OSU friendly name is used to select the provider. Otherwise, the user is asked to select the provider.
2. STA shall associate to OSU ESS.
3. IF ProdESSAssoc,0 parameter is supplied in the command (only for test bed device), the command returns after associating to OSU ESS
4. ELSE, perform the Subscription Registration process and once provisioned with PPSMO, the STA disassociates from the OSU ESS and associates to the Production ESS using credentials (Username:Password OR Certificate) and then returns.

This command returns the SSID and BSSID of the AP BSS it associated to most recently if successful, and with the SSID and BSSID values as empty strings if failed.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
Interface	Interface ID	String
OSUFriendlyName	SP friendly name of the OSU This parameter is for a test bed STA only.	String
ProdESSAssoc	When disabled, subscription registration and association to the production ESS not required. The STA shall return without disconnecting with OSU ESS. (Test bed STA only) When enabled, the STA shall use Credentials from OSU to associate to the Production ESS. If this parameter is not supplied, it defaults to enabled (1).	Integer 0 = Registration not required 1 = Use OSU credentials (Default)

Return Values

Name	Description	Value
BSSID	Basic service set identification	String
SSID	Service set identification	String

Examples

Example 1: (Successful Case - STAUT)

```
CCG: sta_osu, interface,wlan0
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE,SSID,HotspotAP1,BSSID,00:11:22:33:44:55
```

Example 2: (Successful Case – Testbed STA)

CCG: sta_osu, interface,wlan0, osuFriendlyName,Wi-Fi Alliance OSU

CA:status,RUNNING

CA:status,COMPLETE,SSID,HotspotAP1,BSSID,00:11:22:33:44:55

Example 3: (Successful Case)

CCG: sta_osu, interface,wlan0, osuFriendlyName,Wi-Fi Alliance OSU,ProdESSAssoc,0

CA:status,RUNNING

CA:status,COMPLETE,SSID,Legacy-OSU,BSSID,00:11:22:33:44:55

Example 4: (Failure Case)

CCG: sta_osu, interface,wlan0, osuFriendlyName,Wi-Fi Alliance OSU

CA:status,RUNNING

CA:status,COMPLETE,SSID, ,BSSID,

7.29 STA_P2P_CONNECT

This command connects to specified device ID and group ID as a P2P Client.

The command shall block until the STA is connected to the given GroupID.

The command shall implicitly run the Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit discovery is run, then the STA shall go back to its previous state (for example Listen) after it discovers the required P2P device.

The command shall implicitly send the provisional discovery request if needed.

Dependencies

Commands	Description
STA_SET_WPS_PBC	Selects the push button as the configuration method as part of the WPS registration process or P2P group formation process.
STA_WPS_READ_PIN	Selects the keypad as the configuration method as part of the WPS registration process or P2P group formation process.
STA_WPS_ENTER_PIN	Selects the display as the configuration method as part of the WPS registration process or P2P group formation process.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	InterfaceID	String
P2PDevID	P2P Device ID, case insensitive	String

Return Values

None.

Example

```
UCC: sta_p2p_connect,interface,wlan0,p2pdevId,00:11:22:33:44:55:66,groupid,00:11:22:33:44:55
DIRECT-9U-XYZ
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.30 STA_P2P DISSOLVE

This command is used to disconnect from the current P2P group if the device is a client, otherwise disconnect all clients connected to the device and dissolve the P2P group if the device is the group owner.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID SSID>	String
Interface	InterfaceID	String

Return Values

None.

Example

```
UCC: sta_p2p_dissolve,interface,wlan0,groupid,00:aa:bb:cc:dd:ee DIRECT-G1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.31 STA_P2P RESET

This command is used to reset all P2P parameters to device defaults including but not limited to removal of persistent group and stored credentials. It shall also stop any ongoing execution of previous commands.

Refer to the P2P Test Plan for more details on the device default parameter values.

Parameters

Name	Description	Value
Band	The band in which the device should operate.	String 60GHz
Interface	InterfaceID	String

Return Values

None.

Example

```
UCC: sta_p2p_reset,interface,wlan0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.32 STA_P2P_START_GROUP_FORMATION

This command is used to start the P2P group formation with the given intent value.

The command shall implicitly run the Discovery if the given P2PDeviceID is not already discovered by the STA.

Dependencies

Commands	Description
STA_SET_WPS_PBC	Selects the push button as the configuration method as part of the WPS registration process or P2P group formation process.
STA_WPS_READ_PIN	Selects the keypad as the configuration method as part of the WPS registration process or P2P group formation process.
STA_WPS_ENTER_PIN	Selects the display as the configuration method as part of the WPS registration process or P2P group formation process.

Parameters

Name	Description	Value
Init_Go_Neg	Indicates if the STA should initiate the GO Negotiation Request. If INIT_GO_NEG = 0, the command shall prepare the STA to accept a GO Negotiation Request and then return immediately without waiting for the end of group formation. The command return values 'result' and 'groupid' will be null in this case. If INIT_GO_NEG = 1, the STA should initiate GO negotiation by sending GO Negotiation Request and block through the end of group formation. The command returned the result of GO Negotiation and the Group ID.	Integer 0 = Configure but do not send 1 = Send GO request
Intent_Val	Relative value between 0 and 15 used to indicate the desire of the P2P Device to be the P2P Group Owner, with a larger value indicating a higher desire. If Intent_Val is anything other than 0-15, then the station shall use its default value.	Short Integer 0-15
Interface	InterfaceID	String
Oper_Chn	Channel number on which the P2P Device is operating as the P2P Group Owner	Short Integer
P2PDevID	P2P device ID, case insensitive	String

Return Values

Name	Description	Value
Result	Result of GO Negotiation 'FAIL' is returned only in the case where GO Negotiation fails due to two STAs having a GO intent value of 15.	String GO CLIENT FAIL
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String

Example

```
UCC:
sta_p2p_start_group_formation,interface,wlan0,p2pdevid,00:11:22:33:44:55,intent_val,15,init_go_neg,
1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,resut,GO,groupid,00:aa:bb:cc:dd:ee DIRECT-G1
```


7.33 STA_POLICY_UPDATE

This command is used to trigger the Policy Update sequence on the STA device. This is a blocking command and the STA shall return once the status of the procedure is available.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device
HS2_ASSOCIATE	STA HotSpot 2 association
SERVER_SET_PARAMETER	Server set parameter

Parameters

Name	Description	Value
Interface	Interface ID	String
PolicyUpdate	Configures the STA for policy update	Integer 1 = On reception, device immediately triggers policy update procedures with Policy Server. This command overrides the value of UpdateInterval in PPS MO. 0 = Wait for Policy Update Interval in PPSMO to elapse before initiating Policy Update procedure
Timeout	Timeout value for the procedure	Integer

Return Values

Name	Description	Value
PolicyUpdateStatus	Status of Policy Update procedure	String SUCCESS FAIL TIMEOUT

Examples

HS2-R2 Example:

```
CCG: sta_policy_update, interface,wlan0,PolicyUpdate,1,timeout,60
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE,PolicyUpdateStatus,SUCCESS
```

7.34 STA_PRESET_TESTPARAMETERS

This command specifies the test parameters that should be enabled/disabled for the Station/Device.

Program Specific Notes:

Wi-Fi Display

This command will be sent before starting the Wi-Fi Display connection so that the devices can enable the required feature before testing starts.

This command is required for a Wi-Fi Display DUT if it supports HDCP (Source), TDLS, UIBC and Device Discoverability to support the HDCP_KM, TDLS UIBC_Prepare and Device Discoverability parameters.

Hot Spot 2.0

This command is required for HS2-R2 DUT and Test Bed STA devices. This command is used to specify the PPSMO ID and Client Certificates/keys to be configured and installed on the STA device.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
AvChange	Enable or disable Explicit AV format change feature	String Enable Disable
ConnectionCapabilityInfo	GO or Client connection capability	String GO CLI NewGO New CliGO
DeviceDiscoverability	Enable or disable the Device Discoverability feature	String Enable Disable
EDID	Enable or disable the EDID feature	String Enable Disable
FileName	Provides the name of file to be fetched PPSMO: pps-mo-id1-v0.11-vendorA.xml CERT: IDM-xxx (STA device shall append file extension to this value, based on encoding format they support and retrieve the file; for example: IDM-vendorA.pem or IDL-vendorB.p12) Client Certificates are available in following encoding formats: .pem .p12 (password: wifi@123)	String
FilePath	Provides the path from where the File specified by Filename is to be retrieved (This can be achieved in 2 ways using HTTP from external Webserver hosting the file (URL will be supplied) from local database (VendorSpecific)	String Example: http://10.254.11.85 VendorSpecific

Name	Description	Value
FileType	<p>Specifies the type of file to be fetched</p> <p>PPSMO – Per Provider Subscription Management Object</p> <p>CERT – Certificate Credentials (includes Certificate and associated Key file)</p> <p>When the PPSMO uses a client certificate, the certificate files are always downloaded first with FileName, CERT. The following command with FileName, PPSMO indicates for which subscription the certificate files are. The DUT needs to download the trust roots mentioned in the downloaded PPS MO unless they are already present in the device. This may include the trust roots for the subscription server, policy server, and AAA server.</p>	String PPSMO CERT
FrameSkip	Enable or disable Video Frame Skip feature	String Enable Disable
Frgmnt	Fragmentation threshold	Short integer
FT_DS	Configure the station with over-the-DS fast transition enabled or disabled	String Enable Disable
FT_OA	Configure the station with over-the-air fast transition enabled or disabled	String Enable Disable
HDCP	Enable or disable HDCP feature	String Enable Disable
HDCP_Km	HDCP Master Key. This parameter is only applicable to sources that implement HDCP 2.2 and later. This parameter must be supported by the DUT as well as test bed device.	String Enable = enable the stored Km Disable = disable the stored Km feature or erase the stored Km feature
HDCP_Version	Enables only the requested HDCP version. This parameter is only required for source and sink test bed devices.	String HDCP2.1 = enable only version 2.1 HDCP2.2 = enable versions 2.2 and later
HT	Enable/disable the station's HT mode	String On Off
HT_TKIP	Enable/disable TKIP in HT mode	String Enable Disable
HT_WEP	Enable/disable WEP in HT mode	String Enable Disable
I2C	Enable or disable the I2C feature support	String Enable Disable
InputContent	Specifies the input content that is to be rendered when WFD session established. This is valid for Source only.	String Protected Unprotected ProtectedVideoOnly
Interface	Interface ID	String

Name	Description	Value
ManagementTreeURI	Provides information to client device on the exact location of the PPSMO in the management tree. This attribute is to be supplied in CAPI command if FileType is set to PPSMO ./Wi-Fi/wi-fi.org/PerProviderSubscription	String
MNCLength	Number of digits comprising MNC.	Integer 2 3
Mode	Wireless operating mode	String 11b 11g 11a 11n 11ng 11nl (nabg) 11ac 11na
NFC	Enable or Disable NFC interface. If NFC is enabled, the NFC interface should be activated and ready for the following NFC action. This uses the P2P device discovery.	String Enable Disable
NoAck	No ACK setting for each QoS categories. String format is ordered list of Enable or Disable for BE_Policy:BK_Policy:VI_Policy:VO_Policy	String Enable Disable
Oper_Chnn	Operating channel, any valid Wi-Fi channel number	Integer
OptionalFeature	Disable all the program specific optional features	String DisableAll
PDLType	Preset the PDL type to the specified format. A DUT does not need to support this parameter. The device will only enable the specified format.	String PCLM = PCLm PDL PWG Raster PDL
PinConfigMethod	WSC PIN configuration type that will be enabled on the devices as part of connection establishment. The test bed must support this parameter. When acting as Advertizer, the test bed device shall only indicate this PIN configuration method in the probe response. When acting as the Seeker, the ASP chooses the requested configuration method to connect with the peer device.	String Display Keypad
Powersave	Powersave mode legacy	String off on (PSPoll) Fast/PSNonPoll
Preamble	Specified preamble	String Long Short
Prog	Program Name	String WFDS
Program	Program Name	String PMF TDLS VOE NAN WFD P2PNFC

Name	Description	Value
		HS2 HS2-R2 WFDS WPS 60GHz
Reset	Preset 11n parameters	String 11n
RTS	RTS threshold	Short integer
SessionAvailability	Set the session availability bit to available or not available	Integer 0 = Not Available 1 = Available
SSID	Service set identification	String
Standby	Enable or disable Standby-Resume feature	String Enable Disable
Supplicant	Name of the supplicant to be used for the test	String WPA_Supplicant Cisco ZeroConfig Open1x Marvell Default
TDLS	The parameter also effects the value for the PC bit. PC = 1 for Enabled PC = 0 for Disabled	String Enabled Disabled
TDLSMode	Default – default STA mode for TDLS HiLoMac – STA should respond with TDLS setup requirement for a TDLS setup requirement ExistLink – STA should send TDLS setup requirement even if direct link already exists APProhibit – STA should send TDLS setup requirement even if AP prohibits TDLS WeakSecurity – STA should enable all the weak security modes (open/WEP/WPA)	String Default HiLoMac ExistLink APProhibit WeakSecurity
Type	Type of action to be taken on device. AcceptPD - User accepts the request RejectPD - User Rejects the request IgnorePD - User Ignores the request RejectASP – Accept the PD and during the ASP session establishment device ASP responds to the peer device with REJECTED_SESSION when it receives ADDED_SESSION.	String AcceptPD RejectPD IgnorePD RejectASP
UI_Input	This parameter will be passed when any UIBC is enabled. If more than one is listed, they shall be space separated.	String Keyboard Mouse
UIBC_Gen	Enable or disable UIBC generic use input feature	String Enable Disable
UIBC_HID	Enable or disable UIBC HID input feature	String Enable Disable
UIBC_Prepare	Valid on source devices to start the Wi-Fi Display session including UIBC operation	String Enable Disable

Name	Description	Value
VideoFormat	For a sink, this parameter sets or enables the video formats supported in the M3 message. Space separated list If more than one mode mentioned. For a source, this parameter specifies the video format in the M4 message. The format is CEA-<Index> or VESA-<Index> or HH-<Index>. Index is as defined in the specification.	String
VideoRecovery	Enable or disable Video Recovery	String Enable Disable
WFDDevType	Set the device to Source/Sink/Dual source or sink	String Source P-Sink S-Sink Dual
WMM	Enable/disable WMM	String on off
WPS	Used to enable or disable WPS feature on the device.	String Enable Disable
WPSTestAttributes	If enabled, test bed devices will add vendor specific attributes and attributes with zero length.	String Enable Disable
WPSVersion	WPS Version to be advertised as part of the version attribute	String
WSCEapFragment	If enabled, test bed devices will enable fragmentation of the EAP message.	String Enable Disable
WSCIEFragment	If enabled, test bed devices will enable to fragmentation of the WSC IE.	String Enable Disable
WSCState	WSC state of the device set to unconfigure.	String unconfigure

Return Values

None.

Example

Example 1:

```
UCC: sta_preset_testparameters,interface,ath0,supplicant,Cisco,RTS,500
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

Example 2:

```
UCC: sta_preset_testparameters,interface,ath0,noack:disable:disable:enable:disable,mode,11n1
```

Example 3 WFD:

```
UCC: sta_preset_testparameters,interface,ath0,program,wfd,wfddevtype,sink
```

```
CA: status,RUNNING
```

CA: status,COMPLETE

Example 4 2P-NFC:

CCG: sta_preset_testparameters,interface,ath0,program,p2pnfc,nfc,enable

CA: status,RUNNING

CA: status,COMPLETE

Example 5 HS2-R2:

CCG: sta_preset_testparameters,interface,wlan0,program,HS2-R2,FileType,PPSMO,FileName,pps-mo-id1-v0.11-vendorA.xml,FilePath,http://10.254.11.85,managementTreeURI,./Wi-Fi/wi-fi.org/PerProviderSubscription

CA: status,RUNNING

CA: status,COMPLETE

Example 6 HS2-R2:

CCG: sta_preset_testparameters,interface,wlan0,program,HS2-R2,FileType,PPSMO,FileName,pps-mo-id1-v0.11-vendorB.xml,FilePath,VendorSpecific,managementTreeURI,./Wi-Fi/wi-fi.org/PerProviderSubscription

CA: status,RUNNING

CA: status,COMPLETE

Example 7 HS2-R2:

CCG: sta_preset_testparameters,interface,wlan0,program,HS2-R2,FileType,CERT,FileName,IDM-CERT-xxx,FilePath,http://10.254.11.85

CA: status,RUNNING

CA: status,COMPLETE

Example 8 (setting NAN device to use 5GHz channel 44):

UCC: sta_preset_testparameters,interface,wlan0,program,nan,oper_chn,44

CA: status,RUNNING

CA: status,COMPLETE

7.35 STA_REASSOC

This command is used to re-associate the STA with a given AP.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device
STA_ASSOCIATE	STA association

Parameters

Name	Description	Value
BSSID	BSSID	String
Interface	Interface	String
InterfaceID	Interface ID	String

Return Values

None.

Examples

```
UCC: sta_reassoc, interface,wlan,bssid,aa:bb:cc:dd:ee:ff
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```


7.36 STA_REASSOCIATE

This command is used to re-associate the STA with a given AP.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device
HS2_ASSOCIATE	STA HotSpot 2 association

Parameters

Name	Description	Value
BSSID	Basic service set identification	String
Channel	Channel number	String
Interface	Interface ID	String
SSID	Service set identification	String

Return Values

None.

Examples

```
UCC: sta_reassociate,interface,wlan,bssid,aa:bb:cc:dd:ee:ff,ssid,a0:b0:c0:d0:e0:f0,channel,36
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.37 STA_RESET_DEFAULT

This command is used to reset the station to its default program specific configuration.

Parameters

Name	Description	Value
Band	The band in which the device should operate. The device should continue to operate in the specified band for all subsequent commands until it receives another STA_RESET_DEFAULT command.	String 60GHz
DevRole	Role of the device. The device will remain in this state for subsequent commands until it receives STA_RESET_DEFAULT command with a different DevRole. For DevRole STA, clears all existing profiles.	String PCP STA
Interface	Interface ID	String
Prog	Program	String PMF WFD1 VHT HS2 HS2-R2 ² WFDS ³ NAN WMMPS TDLS P2P 60GHz WPS
Type	STA configuration type – DUT or test bed Refer the program specific Test Plan for the default settings.	String DUT Testbed

Notes:

1. For Wi-Fi Display the default settings of the station is same as OOB and the interface state is idle as defined in Wi-Fi Direct. All optional features are set to disable by testbed devices by default.
2. For HS2.0 Release 2, the default settings of station are set to OOB. This includes clearing all previously cached profiles, if any.
3. For WFDS the default settings of the station is same as OOB and the interface state is idle as defined in Wi-Fi Direct.

Examples

```
UCC: sta_reset_default, interface,wlan0,prog,PMF
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_reset_default, interface,wlan0,prog,VHT,type,Testbed
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
CCG: sta_reset_default, interface,wlan0,prog,HS2-R2
```



CA:status,RUNNING

CA:status,COMPLETE

UCC: sta_reset_default,interface,wlan0,prog,60GHz,DevRole,PCP

CA: status,RUNNING

CA: status,COMPLETE

7.38 STA_RESET_PARM

This command is used to reset a specific parameter value(s) within a device.

Parameters

Name	Description	Value
ARP	“All” entries or a specific IP address. The IP address must be shown as: “00:11:22:33:44:55”.	String
Interface	Interface ID	String

Examples

```
UCC: sta_reset_parm, interface,wlan0,arp,all
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_reset_parm, interface,wlan0,arp,192.168.1.10
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.39 STA_SCAN

This command is used to initiates active scanning on the station. The command shall return the status after initiating the scan and shall not block until the scan is finished.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
Accs_Net_Type	Access network type	Integer 0-15
HESSID	HESSID	String
Interface	Interface ID	String

Return Values

None.

Examples

```
UCC: sta_scan, interface,wlan0
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.40 STA_SEND_ADDDBA

This command is used to send an ADDDBA request to the associated AP for a specific Traffic Identifier (TID).

Parameters

Name	Description	Value															
Dest_MAC	Destination MAC address Example: aa:bb:cc:dd:ee:ff	String															
Interface	Interface ID	String															
TID	<p>Traffic identifier</p> <p>For values 0,1,4,6, the number represents the Access Category and not the absolute number. The device can choose a User Priority (based on implementation) corresponding to that AC.</p> <table border="1"> <thead> <tr> <th>Value</th><th>Corresponding AC</th><th>Device can choose UP between</th></tr> </thead> <tbody> <tr> <td>0</td><td>BE</td><td>0 or 3</td></tr> <tr> <td>1</td><td>BK</td><td>1 or 2</td></tr> <tr> <td>4</td><td>VI</td><td>4 or 5</td></tr> <tr> <td>6</td><td>VO</td><td>6 or 7</td></tr> </tbody> </table>	Value	Corresponding AC	Device can choose UP between	0	BE	0 or 3	1	BK	1 or 2	4	VI	4 or 5	6	VO	6 or 7	Short integer 0-15
Value	Corresponding AC	Device can choose UP between															
0	BE	0 or 3															
1	BK	1 or 2															
4	VI	4 or 5															
6	VO	6 or 7															

Return Values

None.

Examples

```
UCC: sta_send_adddba, interface,wlan,tid,5,Dest_mac,aa:bb:cc:dd:ee:ff
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.41 STA_SEND_COEXIST_MGMT

This command is used to send a 20/40 BSS Co-existence management Frame to the associated AP.

Parameters

Name	Description	Value
Interface	Interface ID	String
Type	MHz – coexist mgmt frame with 40Mz intolerant BSS – coexist mgmt frame with 20 Mhz BSS width request ChnlRepo - coexist mgmt frame with 20/40 BSS intolerant channel report	String MHz BSS ChnlRepo
Value		String 0 = MHz 1 – BSS Period separated channel list for ChnlRepo

Return Values

None.

Examples

```
UCC: sta_send_coexist_mgmt, interface,wlan,type,Mhz,value,1
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.42 STA_SEND_FRAME

This command is used to send a protocol frames such as action frames or other layer two frames.

Notes:

1. This command has been obsoleted and replaced with the command DEV_SEND_FRAME. This command is documented for the Voice Enterprise program.
2. This command applies to test bed STAs only.
3. To send a PMF frame, use the WLANTEST commands from UCC.

Parameters

Name	Description	Value
Frame	Frame groups	String General TDLS VENT
Interface	InterfaceID	String
SSID	SSID used in BSS Transition Mgmt Query	
Type	Frame type	String NeigReq TransMgmtQuery

Return Values

None.

Examples

```
sta_send_frame /interface wlan0 /frame vent /type neigreq
```

7.43 STA_SEND_P2P_INVITATION_REQ

This command is used to send a P2P Invitation Request.

This command shall implicitly run the Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit discovery is run then the STA shall go back to its previous state (for example Listen) after it discovers the required P2P device.

The command shall block till it gets the P2P Invitation Response back from the other STA.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	InterfaceID	String
P2PDevID	P2P Device ID	String
Reinvoke	Indicates if the invitation is for reinvoking the persistent group	Integer 0 = No 1 = Yes

Return Values

None.

Example

UCC:

```
sta_send_p2p_invitation_req,interface,wlan0,p2pdevId,00:11:22:33:44:55,groupid,aa:bb:cc:dd:ee:ff  
DIRECT-9UABC,reinvoke,1
```

CA: status,RUNNING

CA: status,COMPLETE

7.44 STA_SEND_P2P_PRESENCE_REQ

This command is used to send a P2P presence request.

Parameters

Name	Description	Value
Duration	Duration in micro-seconds	Double
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	InterfaceID	String
Interval	Interval in micro-seconds	Double

Return Values

None.

Example

```
UCC: sta_send_p2p_presence_req,interface,wlan0,duration,51200,interval,102400
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.45 STA_SEND_P2P_PROVISION_DIS_REQ

This command is used to send the P2P Provision Discovery Request.

The command shall implicitly run the Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit discovery is run then the STA shall go back to its previous state (for exampleListen) after it discovers the required P2P device.

Parameters

Name	Description	Value
ConfigMethod	WPS configuration method	String Display Label Keypad
Interface	InterfaceID	String
P2PDevID	P2P device ID	String

Return Values

None.

Example

```
UCC: sta_send_p2p_provision_dis_req,interface,p2pdeviceid,00:11:22:33:44:55, configmethod,display
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.46 STA_SEND_SERVICE_DISCOVERY_REQ

This command is used to send a P2P Service Discovery Request to the given peer address.

The command shall implicitly run the Discovery if the given P2PDeviceID is not already discovered by the STA. If implicit discovery is run then the STA shall return to its previous state (for example Listen) after it discovers the required P2P device.

Parameters

Name	Description	Value
Interface	Interface ID	String
P2PDevID	P2P device ID to which service discovery request shall be sent.	String

Return Values

None.

Example

```
UCC: sta_send_service_discovery_req,interface,wlan0,p2pdevId,00:11:22:33:44:55
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.47 STA_SET_11N

This command is used to set the 11n STA settings. The STA Control Agent should return the “ERROR” response element for unsupported features only if it cannot produce the effective setting. For example, a device that does not implement the HT Greenfield feature should return “ERROR” if Greenfield is set to “Enable”, but return “COMPLETE” if Greenfield is set to “Disable”.

The STA_SET_RIFS_TEST command modifies the following STA_SET_11N parameters: MCS_Supported, AMPDU, AMSDU, Greenfield, and SGI20. The STA_SET_11N command must not be used to modify these parameters during RIFS testing.

NOTE: This command will eventually become obsolete. An alternative command, STA_SET_WIRELESS shall be used instead.

Parameters

Name	Description	Value
40_Intolerant	Enable or disable the 40 MHz Intolerant feature	String Enable Disable
ADDBA_Reject	Enable or disable the rejecting any ADDBA request by sending ADDBA response with the status “decline”	String Enable Disable
AMPDU	Enable or disable the AMPDU Aggregation feature	String Enable Disable
AMSDU	Enable or disable the AMSDU Aggregation feature	String Enable Disable
Greenfield	Enable or disable the HT Greenfield feature	String Enable Disable
Interface	Interface ID	String
MCS_FixedRate	Fixed MCS rate	Short integer 0-31
MCS32	Enable or disable HT Duplicate Mode	String Enable Disable
Reset_Default	Reset the station to program's default configuration	String 11n
RIFS	Enable/disable the RIFS feature	String Enable Disable
RXSP_Stream	Rx spatial stream	String 1SS 2SS 3SS
SGI20	Enable or disable the Short Guard Interval feature	String Enable Disable
SMPS	SM Power Save mode	Short integer 0 = Dynamic 1 = Static

Name	Description	Value
		2 = No Limit
STBC_RX	STBC receive streams	Short integer
STBC_TX	STBC transmit streams	Short integer
TXSP_Stream	Tx spatial stream	String 1SS 2SS 3SS
Width	802.11n channel width	String 20 40 Auto

Return Values

None.

Examples

```
UCC: sta_set_11n, interface,wlan,40_intolerant,disable
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.48 STA_SET_EAPAKA

This command is used to configure the DUT's specified wireless interface for EAP-AKA authentication with WPA or WPA2 key management. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption cipher	String TKIP AES-CCMP AES-CCMP-TKIP
Interface	Interface ID	String
KeyMgmtType	Key management type	String WPA WPA2 WPA2-FT WPA2-WPA-ENT
Password	Password to be used with the specified username	String
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
SSID	Service set identification	String
Username	Username for authentication According to RFC 5448, if the leading character in Username is "6", "7" or "8" then the station shall use EAP-AKA'.	String

Return Values

None.

Example

```
UCC: sta_set_eapaka, interface, interfaceId_1, ssid, MyNetwork, username, console, password, azimuth,
encpType, tkip, keyMgmtType, wpa
```

```
CA:status, RUNNING
```

```
CA:status, COMPLETE
```

7.49 STA_SET_EAPFAST

This command is used to configure the DUT's specified wireless interface for EAP-FAST authentication with WPA or WPA2 key management. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption cipher	String TKIP AES-CCMP AES-CCMP-TKIP
InnerEAP	Inner EAP authentication method	Valid values: MSCHAPv2 GTC
Interface	Interface ID	String
KeyMgmtType	Key management type	Valid values: WPA WPA2 WPA2-FT WPA2-WPA-ENT
PACFile	PAC file name, where the generated PAC file is saved	String
Password	Password to be used with the specified username	String
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
SSID	Service set identification	String
TrustedRootCA	Name of the Trusted Root CA, whose certificate is configured on the specified STA	String
Username	Username for authentication	String

Return Values

None.

Example

```
UCC:sta_set_eapfast,interface,interfaceId,ssid,MySsid,username,wifiuser,password,password,encpType,
tkip,keyMgmtType,wpa,innerEAP,GTC,validateServer,no,pacFile,wifiuserPAC
```

```
CA:status, RUNNING
```

```
CA:status, COMPLETE
```

7.50 STA_SET_EAPSIM

This command is used to configure the DUT's specified wireless interface for EAP-SIM authentication with WPA or WPA2 key management. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption cipher	Valid values: TKIP AES-CCMP
Interface	Interface ID	String
KeyMgmtType	Key management type	Valid values: WPA WPA2 WPA2-FT
Password	Password to be used with the specified username	String
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
SSID	Service set identification	String
Username	Username for authentication	String

Return Values

None.

Example

```
UCC:
sta_set_eapsim,interface,interfaceId_1,ssid,MyNetwork,username,console,password,azimuth,encpType,tk
ip,keyMgmtType,wpa
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.51 STA_SET_EAPTLS

This command is used to configure the DUT's specified wireless interface for EAP-TLS authentication with WPA or WPA2 key management. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
ClientCertificate	Name of the client (user) certificate configured on the specified STA	String
EncpType	Encryption cipher	Valid values: TKIP AES-CCMP
Interface	Interface ID	String
KeyMgmtType	Key management type	Valid values: WPA WPA2 WPA2-FT
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
SSID	Service set identification	String
TrustedRootCA	Name of the Trusted Root CA whose certificate is configured on the specified STA	String
Username	Username for authentication.	String

Return Values

None.

Example

```
UCC: sta_set_eaptls,interface,interfaceId_1,ssid,MyNetwork,trustedRootCA,Azimuth CA,
clientCertificate,console-cert,encpType,tkip,keyMgmtType,wpa
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```


7.52 STA_SET_EAPTTLS

This command is used to configure the DUT's specified wireless interface for EAP-TTLS authentication with WPA or WPA2 key management. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption cipher	Valid values: TKIP AES-CCMP
Interface	Interface ID	String
KeyMgmtType	Key management type	String WPA WPA2 WPA2-FT
Password	Password to be used with the specified username	String
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
Prefer	This parameter if set to 1 indicates that the SSID provided in this command is the user-preferred SSID. When multiple credentials/profiles are configured on the STA, the STA shall select the network based on this preference. For example: If the STA is configured with 2 profiles: SSID=ABC with EAP-TTLS with Prefer=1 SSID=XYZ with WPA2-PSK and Prefer=0 Then the STA shall consider ABC as the user-preferred SSID.	Boolean 1 = Yes 0 = No (Default)
Prog	Program name	String HS2 HS2-R2
SSID	Service Set Identifier	String
TrustedRootCA	Name of the Trusted Root CA, whose certificate is configured on the specified STA	String
Username	Username for authentication	String

Return Values

None.

Example

```
UCC: sta_set_eapttls,interface,interfaceId_1,ssid,MyNetwork,username,test,password,test123,
```

trustedRootCA,cas, encpType,tkip,keyMgmtType,wpa

CA: status,RUNNING

CA: status,COMPLETE

UCC: sta_set_eapttls,interface,interfaceId_1,ssid,MyNetwork,username,test,password,test123,

trustedRootCA,cas,encpType,aes-ccmp,keymgmttype,wpa2,prefer,1,prog,hs2

CA: status,RUNNING

CA: status,COMPLETE

7.53 STA_SET_ENCRYPTION

This command is used to configure the DUT's specified wireless interface to operate with WEP or no encryption, with Open authentication. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption cipher	Valid values: None WEP
Interface	Interface ID	String
Key1	WEP Key 1	5 or 13 Hex bytes
SSID	Service set identification	String

Return Values

None.

Example

```
UCC: sta_set_encryption,interface,interfaceId_1,ssid,wifi,encpType,wep,  
    key1,12345678901234567890123456
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.54 STA_SET_IP_CONFIG

This command is used to set the IP configuration for the DUT's specified wireless interface.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
DefaultGateway	The default gateway	IP address
DHCP	Specifies if DHCP is enabled If DHCP is set to 1 for Type IPv6 then the STA shall configure its IPv6 address using stateless address auto-configuration.	Boolean
Interface	Interface ID	String
IP	The IP address to be used. This parameter must be specified if 'DHCP' is disabled.	IP address
Mask	The subnet mask to be used. This parameter must be specified if 'DHCP' is disabled.	IP address
Primary-DNS	The primary DNS address	IP address
Type	If Type value is not specified, then the default shall be IPv4. If the Type value is not supported by the STA, then the command shall return ERROR.	Integer 1 = IPv4 (Default) 2 = IPv6

Return Values

None.

Example

```
UCC: sta_set_ip_config,interface,interfaceId_1,dhcp,0,ip,192.168.1.101,mask,255.255.255.0,
primary-dns,192.168.1.1,secondary-dns,192.168.1.101
CA: status,RUNNING
CA: status,COMPLETE
UCC: sta_set_ip_config,interface,interfaceId_1,dhcp,1,type,2
CA: status,RUNNING
CA: status,COMPLETE
```

7.55 STA_SET_MACADDR

This command is used to set a station's wireless interface to a specified MAC address.

Parameters

Name	Description	Value
Interface	Interface ID	String
MAC	The temporary MAC address or default. The format must be shown as: "00:11:22:33:44:55".	String

Return Values

None

Examples

```
UCC: sta_set_macaddr, interface,wlan,mac,00:11:22:33:44:55
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

7.56 STA_SET_OPPORTUNISTIC_PS

This command is used to enable opportunistic power save on the device with the specified CTWindow value.

Parameters

Name	Description	Value
CTWindow	CTWindow time in milli-seconds	Integer
GroupID	P2P Group ID. The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	Interface ID	String

Return Values

None.

Example

```
UCC: sta_set_opportunistic_ps,interface,wlan0,ctwindow,10,groupid, aa:bb:cc:dd:ee:ff DIRECT-9UABCD
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.57 STA_SET_P2P

This command is used to configure the Peer-to-Peer (P2P) parameters of the P2P Device.

Parameters

Name	Description	Value
Concurrency	Capability to support concurrent mode	Integer 0 = No 1 = Yes
CrossConnection	Cross Connection bit value	Integer 0 = No 1 = Yes
Discoverability	P2P Client Discoverability bit set	Integer 0 = No 1 = Yes
DiscoverType	<p>Informs the station the type of device discovery. It can use P2P device discovery mechanism or TDLS tunneled device discover mechanism when specified. When WFD device discover is specified, it is device implementation that it can use any kind of discover.</p> <p>If DiscoverType is set for WFD, it can use any discovery mechanisms and it is implementation specific.</p>	String WFD P2P TDLS
Ext_Listen_Time_Interval	Extended Listen time interval in milli-seconds	Integer
Ext_Listen_Time_Period	Extended Listen time period in milli-seconds	Integer
GO_APSD	Enable or disable WMM power save functionality for P2P Group Owner. This parameter is only applicable for Group Owner.	Integer 0 = Disable 1 = Enable
Interface	Interface ID	String
Intra_BSS	Capability bit to support Intra BSS Transition	Integer 0 = No 1 = Yes
Listen_Chn	Channel number on which the P2P Device is operating as the P2P Device in the Listen State	Short integer
NoA_Count	<p>Number of absence intervals</p> <p>If NoA count is less than 255 (which implies that it will be non-periodic NoA) then the start offset for NoA shall be configured to 50 milliseconds after the beacon advertising the NoA.</p>	Integer
NoA_Duration	Duration of the Absence in TU units	Integer
NoA_Interval	<p>Interval between two absences in TU units</p> <p>If NoA Interval is specified as zero then the STA shall configure the NoA Interval equal to the Beacon Interval.</p>	Integer
P2P_Mode, P2PMODE	Set the P2P mode	String Listen Discover Idle Disable
P2PInvitation	Capability to support P2P Invitation	Integer 0 = No 1 = Yes
P2PManaged	Support for P2P Managed	Integer 0 = No 1 = Yes

Name	Description	Value
Persistent	Capability to support persistent group	Integer 0 = No 1 = Yes
Service_Discovery	Service Discovery functionality If Service_Discovery = 1, enable the Service Discovery functionality. If Service_Discovery = 2 (Test bed STA only), enable the Service Discovery functionality and configure large enough set of local services to cause service discovery responses to be fragmented	Integer 0 = Disable 1 = Enable 2 = Enable STA

Return Values

None.

Example

Example 1:

```
UCC: sta_set_p2p,interface,wlan0,p2p_mode,listen,listen_chn,11
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

Example 2:

```
UCC: sta_set_p2p,interface,wlan0,concurrency,1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

Example 3:

```
UCC: sta_set_p2p,interface,wlan0,discoverability,1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.58 STA_SET_PEAP

This command is used to configure the DUT's specified wireless interface for PEAP authentication with WPA or WPA2 key management. This command must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption cipher	String TKIP AES-CCMP
InnerEAP	Inner EAP authentication method	String MSCHAPv2 GTC
Interface	Interface ID	String
KeyMgmtType	Key management type	String WPA WPA2 WPA2-FT
Password	Password to be used with the specified username	String
PEAPVersion	PEAP protocol version	Integer 0 1
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
SSID	Service set identification	String
TrustedRootCA	Name of the Trusted Root CA, whose certificate is configured on the specified STA	String
Username	Username for authentication	String

Return Values

None.

Example

```
UCC: sta_set_peap,interface,interfaceId_1,ssid,MyNetwork,username,console,password,azimuth,
    trustedRootCA,Azimuth CA, encpType,tkip,keyMgmtType,wpa,innerEAP,MSCHAPv2,
    peapVersion,0
CA: status,RUNNING
CA: status,COMPLETE
```


7.59 STA_SET_POWER_SAVE

Configures the legacy Power Save mode. Device must operate in the powersave mode specified.

Dependencies

None.

Parameters

Name	Description	Value
Interface	Interface ID	String
Powersave	Powersave legacy mode	String Off PSPoll(on) Fast PSNonPoll unscheduled
Program		String 60GHz

Return Values

None.

Example

```
sta_set_power_save /interface wlan0 /powersave PSNonPoll
```

```
sta_set_power_save /interface wlan0 /program 60GHz powersave unscheduled
```

7.60 STA_SET_PSK

This command is used to configure the DUT's specified wireless interface for Pre-Shared Key (PSK) security. Must be called before calling STA_ASSOCIATE.

Dependencies

Commands	Description
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from device

Parameters

Name	Description	Value
EncpType	Encryption type	Valid values: TKIP AES-CCMP AES-CCMP-TKIP
Interface	Interface ID	String
KeyMgmtType	Key management type	Valid values: WPA WPA2 WPA-PSK WPA2-PSK WPA2-FT WPA2-WPA-PSK
MICAlg	MIC algorithm	String SHA-1 (Default) SHA-256
Mode	Wireless operating mode	String 11a 11g 11b 11n
PassPhrase	Pass phrase used to generate the pre-shared key	String
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
Prefer	This parameter if set to 1 indicates that the SSID provided in this command is the user-preferred SSID. When multiple credentials/profiles are configured on the STA, the STA shall select the network based on this preference. For example: If the STA is configured with 2 profiles: SSID=ABC with EAP-TTLS with Prefer=0 SSID=XYZ with WPA2-PSK and Prefer=1, Then the STA shall consider XYZ as the user-preferred SSID.	Boolean 1 = Yes 0 = No (Default)
Prog	Program name	String HS2 HS2-R2

Name	Description	Value
SSID	Service Set Identifier	String

Return Values

None.

Example

```
UCC:
sta_set_psk,interface,interfaceId_1,ssid,MyNetwork,passPhrase,123456789,keyMgmtType,wpa2,encpType,a
es-ccmp
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

```
UCC: sta_set_psk,interface,interfaceId_1,ssid,MyNetwork,passphrase,9876543210,keymgmttype,wpa2,
encpType,aes-ccmp,prog,hs2,prefer,1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.61 STA_SET_PWRSAVE

Configures the station Power Save Mode (On/Off) for the DUT's specified wireless interface.

Dependencies

None.

Parameters

Name	Description	Value
Interface	Interface ID	String
Mode	Enables or disables power save	String On Off (Default)

Return Values

None.

Example

```
UCC: sta_set_pwrsave, interface, interfaceId_1, mode, on
```

```
CA: status, RUNNING
```

```
CA: status, COMPLETE
```

7.62 STA_SET_RADIO

This command is used to turn the radio ON or OFF per interface(s). When the radio is disabled, the station shall not send any notification frames (disassociation, nullfunc data with PwrMgt=1, etc.) to the AP or direct link peers.

Parameters

Name	Description	Value
Interface	Interface ID	String
Mode	Enables or disables the radio	String On Off

Examples

```
UCC: sta_set_radio, interface, wlan0, mode, off
```

```
CA: status, RUNNING
```

```
CA: status, COMPLETE
```

7.63 STA_SET_RFEATURE

This command is used to set a run-time functional feature either generally or per program. This command is also used to allocate multiple CBAP allocations at one time.

Parameters

Name	Description	Value
AllocID	Indicates the Allocation ID used.	Integer 1-n
AllocType	Indicates the type of allocation used.	String CBAP SP
CBAPOnly	The CBAPOnly field indicates the type of link access provided by the STA sending the DMG Beacon frame in the data transfer interval (DTI) of the beacon interval. The CBAPOnly field is set to 1 when the entirety of the DTI portion of the beacon interval is allocated as a CBAP. The CBAPOnly field is set to 0 when the allocation ???	Integer 1 0
ChSwitchMode	Channel Switch Mode. If set to Initiate, channel switching will be enabled as per the TDLS Test Plan. If set to Passive, channel switching will be disabled but will still accept incoming channel switch requests as per the TDLSTest Plan. . This is the default mode for a STA. If set to RejReq, channel switching is disabled and the STA will reject all incoming channel switch requests. If set to UnSolResp, channel switching is enabled using unsolicited channel switch response to return to the base. Once the STA has returned to the base channel using an unsolicited channel switch response, the STA shall not send any further channel switch requests.	String Valid values for STAUT or test bed STA: Initiate Passive (Default) Valid values test bed STA only: RejReq UnSolResp
DestAID	The Destination AID field indicates the AID of a STA that is expected to communicate with the source DMG STA during the allocation or broadcast.	Hex
ExtSchIE	If enabled, the AP or PCP can split the Allocation fields into more than one Extended Schedule element entry in the same DMG Beacon or Announce frame.	String Enable Disable
Interface	Interface ID	String
NSS_MCS_Opt	Refers to nss_operating_mode and mcs_operating_mode. This parameter defines the spatial stream and mcs rate for the STA. Example: If a STA needs to operate in 1SS with MCS 9, then: nss_mcs_opt,1;9 If a STA needs to operate in 4SS with MCS 7, then: nss_mcs_opt,4;7 or to set it to the device's default capability, then: nss_mcs_opt,def;def – to	String
OffChNum	The off channel number in GHz. This parameter is used in conjunction with the “ChSwitchMode – Enable/UnSolResp” parameter.	Integer 2.4 5
Peer	Peer MAC address. This parameter is mandatory for a test bed STA and a DUT if it is a PU sleep STA. A STA informs the peer about its powersave state according to the ‘Uapsd’ parameter.	String
PercentBI	Percentage of beacon interval used.	Integer 0-100

Name	Description	Value
Prog	Program	String PMF TDLS VHT 60GHz
SecChOffset	This parameter is used in conjunction with the “ChSwitchMode – Enable/UnSolResp” parameter.	String 20 40above 40below
SrcAID	The Source AID field is set to the AID of the STA that initiates channel access during the SP or CBAP allocation.	Hex
TPKTimer	This parameter is only valid for test beds. If enabled, the STA can send a TDLS setup request or teardown after a timeout. If disabled, the STA should not send any TDLS setup request or tear down after timeout.	String Enable (Default) Disable
UAPSD	This parameter is mandatory for a test bed STA and a DUT if it is a PU sleep STA. When enabled, the STA goes into powersave and informs the AP. When disabled, the STA comes out of powersave and informs the AP.	String Enable Disable (Default)

Examples

```
UCC: sta_set_rfeature, interface,wlan0,prog,general,uapsd,enable
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_set_rfeature, interface,wlan0,prog,TDLS,uapsd,enable,peer,00:01:02:03:04:05
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_set_rfeature, interface,wlan0,prog,TDLS,uapsd,disable,peer,00:01:02:03:04:05
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_set_rfeature, interface,wlan0,prog,TDLS,tpktimer,disable
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta_set_rfeature,  
interface,wlan0,prog,TDLS,ChSwitchMode,Enable,OffChNum,60,SecChOffset,40above
```

```
CA:status,COMPLETE
```

UCC: sta_set_rfeature, interface,wlan0,prog,TDLS,ChSwitchMode,Disable

CA:status,COMPLETE

UCC:
sta_set_rfeature,interface,STA1_IF,prog,60GHz,CBAPOnly,0,ExtSchIE,Enable,AllocType,CBAP,AllocID,1,PercentBI,5,SrcAID,0xff,DestAID,0xff, AllocID,2,PercentBI,16,SrcAID,0xff, DestAID,0xff,
AllocID,3,PercentBI,16,SrcAID,0xff,DestAID,0xff

CA:status,COMPLETE

7.64 STA_SET_SECURITY

This command is used to set the optional security formats for the STA.

Parameters

Name	Description	Value
EncpType	Encryption type	String TKIP AES-CCMP AES-GCMP
Interface	Interface ID	String
KeyMgmtType	Key management type	String WPA WPA2 WPA2-FT WPA2-PMF
Passphrase	InterfaceID for PSK, or encryption key for 60GHz	String
PMF	Configures the PMF settings	String Required Optional Disable (Default) Forced_Required Forced_Disabled
SSID	Service set identification	String
Type	Security type	String PSK EAPTLS EAPTTLS EAPPEAP EAPSIM EAPFAST EAPAKA OPEN

Return Values

None.

Example

```
UCC: sta_set_security,type,eaptls,interface,interfaceId_1,ssid,MyNetwork,trustedRootCA,Azimuth CA,
clientCertificate,console-cert,encpType,tkip,keyMgmtType,wpa,pmf,enable
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```


7.65 STA_SET_SLEEP

This command puts the STA into legacy sleep mode. This command shall return after the STA goes into sleep mode.

Parameters

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PdevID ssid> GroupID is optional if the command is given to the Legacy (non-P2P) station.	String
Interface	InterfaceID	String

Return Values

None.

Example

```
UCC: sta_set_sleep,interface,wlan0,groupid,aa:bb:cc:dd:ee:ff DIRECT-9UABCD
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.66 STA_SET_SYSTIME

This command is used to set the date and time for the test bed components. It is currently used to set the date in the DUT for Negative EAP tests.

Parameters

Name	Description	Value
Date	Date	Integer 1-31
Hours	Hours (24 hour representation)	Integer 0-23
Minutes	Minutes	Integer 0-59
Month	Month	Integer 1-12
Seconds	Seconds	Integer 0-59
Year	Year	Integer Four digit value: XXXX

Return Values

None.

Example

```
UCC:sta_set_systime,month,06,date,04,year,2009,hours,0,minutes,0,seconds,0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.67 STA_SET_UAPSD

This command is used to configure the Automatic Power Save Delivery Mode for the DUT's specified wireless interface. This command must be called before calling STA_ASSOCIATE.

Dependencies

None.

Parameters

Name	Description	Value
ACBE	Enable or disable APSD mode for the Best Effort Access Category	Boolean 1 = Enable (Default) 0 = Disable
ACBK	Enable or disable APSD mode for the Background Access Category	Boolean 1 = Enable (Default) 0 = Disable
ACVI	Enable or disable APSD mode for the Video Access Category	Boolean 1 = Enable (Default) 0 = Disable
ACVO	Enable or disable APSD mode for the Voice Access Category	Boolean 1 = Enable (Default) 0 = Disable
Interface	Interface ID	String
MaxSPLength	Maximum service period length	Integer 0 1 2 4 (Default)
SSID	Service set identification	String

Return Values

None.

Example

```
UCC: sta_set_uapsd,interface,interfaceId_1,ssid,wifi,maxSPLength,2,acBK,0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.68 STA_SET_WIRELESS

This command is used to configure the STA settings for the VHT and 11n programs. The STA Control Agent should return the “ERROR” response element for unsupported features only if it cannot produce the effective setting. For example, a device that does not implement the HT Greenfield feature should return “ERROR” if Greenfield is set to Enable, but returns “COMPLETE” if Greenfield is set to Disable.

NOTE: The STA_SET_RIFS_TEST command modifies the following STA_SET_WIRELESS parameters: MCS_SUPPORTED, AMPDU, AMSDU, GREENFIELD, and SGI20. The STA_SET_WIRELESS command must not be used to modify these parameters during RIFS testing.

Parameters

Name	Description	Value
ABFTLRang	Specifies the minimum value of the range. Does not specify the maximum value of the range.	String Gt1 = Greater than or equal to 1
ADDBA_Reject	Enable or disable the rejecting any ADDBA request by sending ADDBA response with the status “decline”	String Enable Disable
AllocID	Indicates the Allocation ID used	Integer 1-n
AllocType	Allocation Type.	String CBAP
AMPDU	Enable or disable the AMPDU Aggregation feature	String Enable Disable
AMSDU	Enable or disable the AMSDU Aggregation feature	String Enable Disable
BAckRcvBuf	BlockAck receive buffer size in terms of number of MPDUs.	Integer 2 4
Band	Band	String 2.4 5
BcnInt	Beacon Interval	Integer
BW_Sgnl	Enable or disable the ability to send out RTS with bandwidth signaling	String Enable Disable
CBAPOnly	CBAP only.	Boolean 1 0
Channel	Channel number for scanning for a STA. Channel number in which the PCP should transmit beacons,	Integer
CTS_Width	Sets the CTS bandwidth that is sent from the test bed	String 20 40 80
DevRole	Device role. A PCP device can start transmission of DMG beacons after this command.	String PCP STA

Name	Description	Value
	A STA device with DiscoveryMode set to 1 should not initiate active scan until the device receives the STA_SCAN command.	
DYN_BW_Sgnl	If enabled then the STA sends the RTS frame with dynamic bandwidth signaling, otherwise the STA sends RTS with static bandwidth signaling. BW signaling is also enabled when DYN_BW_SGNL is enabled.	String Enable Disable
Encrypt	Encryption Type.	String AES-GCMP
ExtSchIE	Extended Schedule IE.	String Enable Disable
Heartbeat	Indicates whether or not the STA expects to receive a frame from the PCP/AP during the ATI and expects to receive a frame with the DMG control modulation from a source DMG STA at the beginning of an SP or TXOP.	Boolean 1 = Yes 0 = No
Interface	Interface ID	String
LDPC	Enable or disable the use of LDPC code at the physical layer for both Tx and Rx side	String Enable Disable
MCS_FixedRate	Fixed MCS rate	Short integer 0-31
MSDUSize	Size of each MSDU in bytes.	Integer
MU_TxBF	Enables or disables Multi User (MU) TxBF beamformee capability with explicit feedback	String Enable Disable
NSS_MCS_Cap	This parameter gives a description of the supported spatial streams and mcs rate capabilities of the STA. Examples – For setting a STA to support 2SS with MCS 0-9: nss_mcs_cap,2;0-9 For setting a STA to support 4SS with MCS 0-9: nss_mcs_cap,4;0-9	String
Opt_Md_Notif_IE	Sets the operating mode notification elements NSS (number of spatial streams) and channel bandwidth. Example - For setting the operating mode notification element with NSS=1 & BW=20Mhz: Opt_md_notif_ie,1;20	String NSS Channel bandwidth in MHz
PercentBI	Percentage of Beacon Interval.	Integer 0-100
Program	Program name	String VHT 11n 60GHz WPS
PSK	Encryption Passphrase	String
Radio	Turns the device Radio on and off. This is used start and stop the device frame transmission/reception.	String ON OFF
RTS_Force	Forces a STA to send RTS	String Enable Disable
RTS_Width	Sets the bandwidth in the RTS request initiated by the DUT	String

Name	Description	Value
		20 40 80
RXSP_Stream	Rx Spatial Stream	Integer 1 2 3
Security	Security mode	String Open WPA2-PSK
SGI80	Enable or disable the short guard interval at 80 MHz	String Enable Disable
SSID	SSID string for the BSS.	String
STBC_RX	STBC receive streams	Short Integer
TxBandwidth	Sets the STA transmission bandwidth	String 20 40 80 160
Tx_CTS	This command allows a STA to send 80 or 40MHz CTS.	String Enable Disable
Tx_LGI_Rate	Sets the Tx Highest Supported Long Gi Data Rate subfield	Integer
Tx_RTS	Initialize the STA to transmit RTS frame with specified bandwidth indicated in the subsequent field	String Enable Disable
TxBF	Enable or disable SU TxBF beamformee capability with explicit feedback	String Enable Disable
TXSP_Stream	Tx Spatial Stream	Integer 1 2 3
VHT_TKIP	Enable or disable TKIP in VHT mode	String Enable Disable
VHT_WEP	Enable or disable WEP in VHT mode	String Enable Disable
Width	802.11n channel width Use this parameter to set a STA channel width to 160 Mhz.	String 20 40 80 160 Auto
Zero_CRC	Sets the CRC field to all zeroes	String Enable Disable

Return Values

None.

Examples

```
UCC: sta_set_wireless, interface,wlan,40,width,80
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

```
UCC: sta-  
set_wireless,interface,wlan,Program,60GHz,DevRole,PCP,SSID,MyNetwork,Channel,2,BcnInt,100,Security,  
Open
```

```
CA: status,RUNNING
```

```
CA: status, COMPLETE
```

```
UCC: sta-set_wireless,interface,wlan,Program,60GHz,DevRole,STA,DiscoveryMode,1,ScanMode,active
```

```
CA: status,RUNNING
```

```
CA: status, COMPLETE
```

7.69 STA_SET_WMM

This command is used to configure the station WMM related parameters for the specified wireless interface.

Parameters

Name	Description	Value
Action	Command performs WMM-AC TSPEC ADD or DELETE	String addts delts DMGADDTS PTPADDTS
BurstSize		Short Integer 1-0xFFFFFFFF
DelayBound		Short Integer 1-0xFFFFFFFF
Direction		String uplink downlink bidirectional
Fixed		String True False
Group	One of the parameter groups	String WMMAC 60GHz
Inactivity		Short Integer 1-0xFFFFFFFF
Interface	Interface ID	String
Max_Srvc_Intrvl		Short Integer 1-0xFFFFFFFF
MaxSize		Short Integer 1-0xFFFF
MeanDataRate		Short Integer 1-0xFFFFFFFF
Medium_Time		Short Integer 1-0xFFFF
MinDataRate		Short Integer 1-0xFFFFFFFF
Min_Srvc_Intrvl		Short Integer 1-0xFFFFFFFF
PeakDataRate		Short Integer 1-0xFFFFFFFF
PHYRate		Short Integer 1-0xFFFFFFFF
PSB		String Legacy UAPSD
SBA		Double or Short Integer 1.0 – 8.0

Name	Description	Value
		or 0x2001 – 0xFFFF
Size	MSDU size	Short Integer 0x0 – 0x7FFF
SrvcStartTime		Short Integer 1-0xFFFFFFFF
Suspension		Short Integer 1-0xFFFFFFFF
TID		Short Integer 0-7
UP		0-7

Return Values

None.

Examples

```
TM:sta_set_wmm,interface,wlan0,GROUP,wmmac,ACTION,addts,DIALOG_TOKEN,10,TID,2,DIRECTION,uplink,PSB,
UAPSD,UP,7,Fixed,true,SIZE,208,MAXSIZE,512,MIN_SRVC_INTRVL,3000,MAX_SRVC_INTRVL,3001,INACTIVITY,100
0,SUSPENSION,20000,SRVCSTARTTIME,300,MINDATARATE,83200,MEANDATARATE,80000,PEAKDATARATE,83201,PHYRAT
E,6000000,BURSTSIZE,100000,DELAYBOUND,12100,SBA,1.5,MEDIUM_TIME,3567,ACCESSCAT,VO
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

```
UCC:sta_set_wmm,interface,wlan0,GROUP,wmmac,ACTION,delts,TID,2
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.70 STA_SET_WPS_PBC

This command selects the push button as the configuration method as part of the WPS registration process or P2P group formation process. This command does not start the WPS registration process or the P2P Group negotiation process.

If the device is already acting as a P2P GO and the optional Group ID is supplied, then it will authorize incoming client connections on this specific P2P GO.

This command may be issued to all STAs including the PCP.

Parameters

Name	Description	Value
Band	The band in which the device should operate.	String 60GHz
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid> GroupID is only required if this command is issued to the Group Owner.	String
Interface	Interface ID	String

Return Values

None.

Example

```
UCC: sta_set_wps_pbc,interface,wlan0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.71 STA_START_AUTONOMOUS_GO

This command is used to start the autonomous P2P group.

Parameters

Name	Description	Value
Interface	InterfaceID	String
Oper_Chn	Channel number on which the P2P Device is operating as the P2P group owner	Short integer
RTSP	Prepare RTSP as applicable by the type of device. If the device is Source start RTSP server. If device is Sink prepare for RTSP session (implementation specific)	Integer 0 = Source 1 = Sink
SSID	SSID postfix used by autonomous GO	String

Return Values

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String

Example

```
UCC: sta_start_autonomous_go,interface,wlan0,oper_chn,11,SSID,-XYZ
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,groupid,00:11:22:33:44:55 DIRECT-9U-XYZ
```

7.72 STA_WPS_ENTER_PIN

This command is used to enter the WPS PIN into the STA.

This command selects the display on the STA as the configuration method as part of the WPS registration process or the P2P group formation process.

This command may also be used to set the WPS configuration method as keypad. This command will not start the WPS registration process or the P2P Group negotiation process.

This command may be issued to all STAs including the PCP.

Parameters

Name	Description	Value
Band	The band in which the device should operate.	String 60GHz
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid> This will indicate the GO to authorize incoming client connection on this specific P2P GO. GroupID is only required if this command is issued to the Group Owner.	String
Interface	Interface ID	String
Pin	WPS PIN value	String

Return Values

None.

Example

```
UCC: sta_wps_enter_pin,interface,wlan0,pin,1234ABCD
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.73 STA_WPS_READ_LABEL

This command is used to read the label when using the WPS configuration method 'Label'.

This command selects the label on the STA as the configuration method as part of the WPS registration process or the P2P group formation process.

This command will not start the WPS registration process or the P2P Group negotiation process.

This command may be issued to all STAs including the PCP.

Parameters

Name	Description	Value
Band	The band in which the device should operate.	String 60GHz
GroupID	P2P Group ID GroupID is used only if this command is issued to the group owner. The format is space separated P2PDevID and ssid <P2PDevID ssid> This will indicate the GO to authorize incoming client connection on this specific P2P GO.	String
Interface	Interface ID	String

Return Values

Name	Description	Value
Label	WPS label	String

Example

```
UCC: sta wps_read_label,interface,wlan0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,label,1234ABCD
```

7.74 STA_WPS_READ_PIN

This command is used to generate the WPS PIN (if required) and returns the WPS PIN value.

This command selects the keypad on the STA as the configuration method as part of the WPS registration process or the P2P group formation process.

This command does not start the WPS registration process or the P2P Group negotiation process.

This command may be issued to all STAs including the PCP.

Parameters

Name	Description	Value
Band	The band in which the device should operate.	String 60GHz
GroupID	P2P Group ID GroupID is used only if this command is issued to the group owner. This field will indicate the GO to authorize incoming client connection on this specific P2P GO. The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
Interface	Interface ID	String

Return Values

Name	Description	Value
PIN	WPS PIN Value	String

Example

```
UCC: sta_wps_read_pin,interface,wlan0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,PIN,1234ABCD
```

7.75 START_WFD_CONNECTION

This command is used to initiate the Group negotiation/TDLS link, WFD capability negotiation and establishes the WFD session. WPS Provision is set before issuing this command if the link is P2P. This command returns the WFD session ID (Part of the M6 message) if it initializes the group negotiation or TDLS link setup. Once a session is established, streaming shall be started with a default content unless it is explicitly set using other CAPI command.

Parameters

Name	Description	Value
Init_WFD	<p>If P2P is used as the preferred connection, this parameter indicates if the STA should initiate the GO Negotiation Request.</p> <p>0 = The command will prepare the STA to accept the GO Negotiation and then return immediately without waiting for the end of group formation. The command will not initiate the GO Negotiation Request. The GO Negotiation Request will be sent from another STA. The command return values Result, Group ID and WFDSessionID will be null.</p> <p>1 = The STA will initiate the GO Negotiation and block through the end of WFD session establishment. The command will return the Result, Group ID and WFDSessionID.</p> <p>If TDLS is used as the preferred connection, this parameter indicates if the STA should initiate a TDLS connection.</p> <p>0 = The command shall prepare the device for TDLS and then return immediately without the SessionID (implementation specific).</p> <p>1 = The device shall initiate the TDLS setup and block through the end of WFD session establishment. The command shall then return with the WFDSessionID.</p>	<p>Integer</p> <p>Valid values:</p> <p>0 = Disabled</p> <p>1 = Enabled</p>
Intent_Val	<p>Intent value for becoming group owner</p> <p>If Intent value is anything other than 0-15 then station shall use its default value.</p>	Integer
Interface	Interface ID	String
Oper_Chnl	Operating channel	Integer
PeerAddress	<p>DeviceID for Wi-Fi direct or Station MAC address for TDLS.</p> <p>Space separated list of DeviceID and Station MAC address, allowing the device to choose the connection type. The value of P2P DevID shall be case insensitive.</p>	String

Return Values

Name	Description	Value
Result	<p>Result of GO Negotiation.</p> <p>'FAIL' is returned only in the case where GO Negotiation fails due to two STAs having the GO intent value 15 or P2P group is not established.</p> <p>The response will be NULL if TDLS is used.</p>	<p>String</p> <p>GO</p> <p>CLIENT</p> <p>FAIL</p>
GroupID	<p>Wi-Fi Direct Group ID</p> <p>The format is space separated P2PDevID and ssid <P2PDevID ssid></p>	String or NULL if TDLS is used
WFDSessionID	<p>WFD session ID.</p> <p>Will be set to NULL if the WFD session is not established because of TDLS setup failure or P2P group formation failure.</p> <p>Will be set to '1' if the WFD session is not available on the peer device.</p>	<p>String</p> <p>NULL</p> <p>1.20</p>

Examples

```
UCC:start_wfd_connection,interface,wlan,init_go_neg,1,intent_val,15,peeraddress,00:01:22:33:44:55
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE,result,GO, GroupID,00:12:13:14:15:16-DIRECT-CA-LAB, WFDSessionId,98765432abcd
```

7.76 START_WPS_REGISTRATION

This command starts the WPS Registration process as enrollee or registrar as indicated by the 'WPSRole' parameter. This command is applicable to WPS capable APs, STAs and PCPs. This command is a blocking call and should return after successful WPS Provisioning, i.e after M8 message.

Dependencies

The WPS provisioning configuration method must have been set before issuing this command.

Commands	Description
STA_SET_WPS_PBC	
STA_WPS_READ_PIN	
STA_WPS_ENTER_PIN	

Parameters

Name	Description	Value
Interface	Interface ID	String
WPSConfigMethod	Represents the WPS config method that should be used in the registration process.	String PBC Label Keypad Display
WPSRole	This parameter indicates what role of the device will be as part of the registration process.	String Enrollee Registrar

Return Values

Name	Description	Value
WpsState	InSession: If this command received again during an existing session, the device should return with InSession. OverlapSession: If device detects overlap session, device should return with OverlapSession. NoPeer: If device detects no peer is ready to start registration process (ex: station finds no AP with SR=1), device should return with NoPeer. Successful: Successfully completed the registration process.	String InSession OverlapSession NoPeer Successful

Example

```
UCC: start_wps_registration,interface,$DUT_IF,WpsRole,enrollee,WpsConfigMethod,pbc
```

```
CA: status, RUNNING
```

```
CA: status, COMPLETE,WpsState,successful
```


7.77 TRAFFIC_AGENT_CONFIG

This command is used to configure the traffic profile on the traffic agent.

Parameters

Name	Description	Value
Profile	Specifies the traffic profile	String File_Transfer Multicast IPTV Transaction Start_Sync UAPSD
Direction	Specifies whether the agent is the sender or receiver	String Send Receive

There are four basic traffic profiles. The tables below specify the parameters required for each profile in each direction.

Parameters required for File_Transfer send

Name	Description	Value
Destination	Specifies the IP address of the receive traffic agent	IP address
DestinationPort	Specifies the transport destination port to be used	Short integer
Duration	Number of seconds to send packets	Short integer
FrameRate	The number of frames per second to send Note: While framerate equals to 0, the STA may pick the best frame rate and payload size combination to achieve its best performance for the implementation. Currently PC-ENDPOINT payload size is set to 1470 to match the maximum Ethernet frame size if frame rate equals to 0.	Short integer
PayloadSize	The length in bytes of the transport layer payload in each packet	Short integer
SourcePort	Specifies the transport source port to be used	Short integer
StartDelay	Specifies the delay after which the traffic is to be started after receiving a traffic_agent_send command	Short integer 0 (Default)

Parameters required for File_Transfer receive

Name	Description	Value
DestinationPort	Specifies the transport destination port to be used	Short integer
Source	Specifies the IP address of the source traffic agent	IP address
SourcePort	Specifies the transport source port to be used	Short integer

Parameters required for Multicast send

Name	Description	Value
Destination	Specifies the Multicast group IP address of the receive traffic agent	IP address
DestinationPort	Specifies the transport destination port to be used	Short integer

Name	Description	Value
Duration	Number of seconds to send packets	Short integer
FrameRate	The number of frames per second to send	Short integer
PayloadSize	The length in bytes of the transport layer payload in each packet	Short integer
SourcePort	Specifies the transport source port to be used	Short integer
StartDelay	Specifies the delay after which the traffic is to be started after receiving a traffic_agent_send command	Short integer 0 (Default)

Parameters required for Multicast receive

Name	Description	Value
DdestinationPort	Specifies the transport destination port to be used	Short integer
Destination	Specifies the Multicast group IP address of the receive traffic agent	IP address
Source	Specifies the IP address of the source traffic agent	IP address
SourcePort	Specifies the transport source port to be used	Short integer

Parameters required for IPTV send

Name	Description	Value
Destination	Specifies the IP address of the receive traffic agent	IP address
DestinationPort	Specifies the transport destination port to be used	Short integer
Duration	Number of seconds to send packets	Short integer
FrameRate	The number of frames per second to send Note: While frame rate equals to 0, the STA could pick the best frame rate and payload size combination to achieve its best performance for that implementation. Currently PC-ENDPOINT payload size is set to 1470 to match the maximum Ethernet frame size if framerate equals to 0.	Short integer
HTI	HT throughput flag. On – Enables HT payload size of 16384 bytes Off (DEFAULT value) – Enables NonHT payload size of 1450 bytes	String On Off
PayloadSize	The length in bytes of the transport layer payload in each packet	Short integer
SourcePort	Specifies the transport source port to be used	Short integer
StartDelay	Specifies the delay after which the traffic is to be started after receiving a traffic_agent_send command	Short integer 0 (Default)
TrafficClass	The traffic class used to send this stream's traffic	String Voice Video Background BestEffort
TransProtoType	If set to 1, TCP data transfer. If set to 0, UDP data transfer.	Short integer 0 (default) 1

Parameters required for IPTV receive

Name	Description	Value
DestinationPort	Specifies the transport destination port to be used	Short integer
Source	Specifies the IP address of the source traffic agent	IP address
SourcePort	Specifies the transport source port to be used	Short integer
TransProtoType	Sets the data transfer type. If set to 1, TCP data transfer. If set to 0, UDP data transfer.	Short integer 0 (default) 1

Parameters required for Transaction send

Name	Description	Value
Destination	Specifies the IP address of the receive traffic agent	IP address
DestinationPort	Specifies the transport destination port to be used	Short integer
Duration	Number of seconds to send packets	Short integer
FrameRate	The number of frames per second to send	Short integer
MaxCnt	Packets to be sent before the “send” stops	Integer 0 – 70,000
PayloadSize	The length in bytes of the transport layer payload in each packet	Short integer
SourcePort	Specifies the transport source port to be used	Short integer
StartDelay	Specifies the delay after which the traffic is to be started after receiving a traffic_agent_send command)	Short integer 0 (Default)

Parameters required for Transaction receive

Name	Description	Value
DestinationPort	Specifies the transport destination port to be used	Short integer
PayloadSize	The length in bytes of the transport layer payload in each packet	Short integer
Source	Specifies the IP address of the source traffic agent	IP address
SourcePort	Specifies the transport source port to be used	Short integer

Parameters required for UAPSD

Name	Description	Value
TagName	WMMPs test case tag; M.D, B.B, B.W,...	String

Return Values

Name	Description	Value
StreamID	Unique identifier for this traffic stream	Short integer

Example

```
UCC: traffic_agent_config,profile,File_transfer,direction,send,  
     destination,192.168.1.1, destinationPort,5600,sourcePort,5700,frameRate,100,  
     duration,100,payloadSize,64
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,streamID,34
```

7.78 TRAFFIC_AGENT_RECEIVE_START

This command is used to start traffic reception on specified streams. The command may return before the reception of packets has started.

The streams to be started are specified as a space-separated list of streamIDs.

Dependencies

Commands	Description
TRAFFIC_AGENT_CONFIG	Configure traffic profile
STA_ASSOCIATE	Associate to a network
STA_SET_IP_CONFIG	Set IP address

Parameters

Name	Description	Value
StreamID	Specifies the traffic streams on which to start receiving traffic	List of short integers

Return Values

None.

Example

```
UCC: traffic_agent_receive_start,streamID,23 24
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.79 TRAFFIC_AGENT_RECEIVE_STOP

This command is used to stop traffic reception on the specified streams and returns traffic statistics associated with each stream. The Traffic Agent will include only those traffic statistics that it supports and will not include the traffic statistics (for example outOfSequenceFrames) that it does not support.

The streams to be stopped are specified as a space-separated list of streamIDs.

The results are returned as space-separated lists of streamIDs and traffic statistics.

Dependencies

Commands	Description
TRAFFIC_AGENT_CONFIG	Configure traffic profile
TRAFFIC_AGENT_START	Send traffic
STA_SET_IP_CONFIG	Set IP address
STA_ASSOCIATE	Associate to a network

Parameters

Name	Description	Value
StreamID	Specifies the traffic streams on which to stop receiving traffic	List of Short Integers

Return Values

Name	Description	Value
TxActFrames	Number of frames generated by the traffic generator application	List of Short Integers
TxFrames	Number of non-duplicate UDP frames transmitted by the application	List of short integers
RxFrames	Number of non-duplicate UDP frames received with a valid checksum	List of short Integers
TxPayloadBytes	Number of UDP data bytes transmitted in non-duplicate frames	List of short integers
RxPayloadBytes	Number of UDP data bytes received in non-duplicate frames with a valid checksum	List of short integers
OutOfSequenceFrames	Number of non-duplicate frames with a valid checksum received out of sequence	List of short integers

Example

```
UCC: traffic_agent_receive_stop,streamID,23 24
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE, streamID,23 24,txFrames,1045 533,rxFrames,1032 472,
    txPayloadBytes,50455 19003,rxPayloadBytes,50101 24667,outOfSequenceFrames,12 5
```

7.80 TRAFFIC_AGENT_RESET

This command deletes all streams on the traffic agent. If there are streams on which traffic reception is in progress, the traffic reception is aborted and the traffic agent is brought to a clean state.

Parameters

None.

Return Values

None.

Example

```
UCC: traffic_agent_reset
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

7.81 TRAFFIC_AGENT_SEND

This command is used to start traffic transmission on the specified streams. The command will block (for example, will not return status COMPLETE) until all traffic is sent, at which point the command returns the transmit statistics for each stream. The Traffic Agent will include only those traffic statistics that it supports and will not include the traffic statistics (for example outOfSequenceFrames) that it does not support.

The streams to be started are specified as a space-separated list of streamIDs.

The results are returned as space-separated lists of streamIDs and traffic statistics.

Dependencies

Commands	Description
TRAFFIC_AGENT_CONFIG	Configure traffic profile
STA_ASSOCIATE	Associate to a network
STA_SET_IP_CONFIG	Set IP address

Parameters

Name	Description	Value
StreamID	Specifies the traffic streams to use for sending traffic	List of short integers

Return Values

Name	Description	Value
TxActFrames	Number of frames generated by the traffic generator application	List of Short Integers
TxFrames	Number of non-duplicate UDP frames transmitted by the application	List of short integers
RxFrames	Number of non-duplicate UDP frames received with a valid checksum	List of short integers
TxPayloadBytes	Number of UDP data bytes transmitted in non-duplicate frames	List of long integers
RxPayloadBytes	Number of UDP data bytes received in non-duplicate frames with a valid checksum	List of short integers
OutOfSequenceFrames	Number of non-duplicate frames with a valid checksum received out of sequence	List of short integers

Example

```
UCC: traffic_agent_send,streamID,23 24
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE, streamID,23 24, txFrames,1045 508,rxFrames,1032 700, txPayloadBytes,50455
23544, rxPayloadBytes,50101 22055,outOfSequenceFrames,12 7
```


7.82 TRAFFIC_AGENT_VERSION

This command is used to obtain the version of the traffic generator running on the device.

Parameters

None.

Return Values

Name	Description	Value
Version	Version of the traffic generator	String

Example

```
UCC: traffic_agent_version,  
CA: status, RUNNING  
CA: status, COMPLETE, version, TG-1.0
```

7.83 TRAFFIC_SEND_PING

This command is used to start traffic by sending pings to the specified destination at a specified frame rate.

The timeout to wait for a ping response is implied based on the frame rate. It is the intention of this command to allow pings to be sent and received in a single thread.

The returned streamID is used in subsequent calls to TRAFFIC_STOP_PING.

Dependencies

Commands	Description
STA_ASSOCIATE	Associate to a network
STA_SET_IP_CONFIG	Set the IP address

Parameters

Name	Description	Value
Destination	IP address to send the pings	IP address
DSCP	Differentiated Services Code Point value for IP package priority	Integer
Duration	Number of seconds to ping	Short integer 0 = continuous
FrameRate	Number of pings to send each second	Short integer
FrameSize	Frame size in bytes	Short integer
IPType		Integer 1 = IPv4 (Default) 2 = IPv6
TOS	Type of Service value for IP packet priority	Integer

Return Values

Name	Description	Value
StreamID	Unique identifier for this traffic stream	Short Integer

Example

```
UCC: traffic_send_ping,destination,192.168.1.1,frameSize,64,frameRate,1,duration,20
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,streamID,34
```

7.84 TRAFFIC_STOP_PING

This command is used to stop the ping traffic and returns the ping results. The ping traffic may have already stopped before calling this command, based on the duration specified when the pings were started.

Dependencies

Commands	Description
TRAFFIC_SEND_PING	Send ping

Parameters

Name	Description	Value
StreamID	Identifies a ping traffic stream started by a prior call to traffic_send_ping	Short integer

Return Values

Name	Description	Value
Sent	The number of ICMP ping requests sent in the specified stream	Short integer
Replies	The number of the ICMP ping requests which were replied to	Short integer

Example

```
UCC: traffic_stop_ping,streamID,34
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,sent,20,replies,15
```

8 Access Point Configuration API

8.1 Access Point Commands

Similar to STAs, Access Point (AP) commands consist of the commands to set and get the wireless interface parameters and AP information. These commands can be either production, draft, optional, or not applicable depending on the platform component and certification program. The CAPI Program matrix classifies the commands based on component and program.

8.2 AP_CA_VERSION

This command is used to retrieve the AP Control Agent version number.

Parameters

Name	Description	Value
Name	A name such as ABC 11n AP	String

Return Values

Name	Description	Return Value
Version	Control agent version	String

Example

```
UCC: ap_ca_version,NAME,11n
AP: status,RUNNING
AP: status,COMPLETE,version,00.00.05
```

8.3 AP_CONFIG_COMMIT

This command commits the configuration changes. The AP shall not return “COMPLETE” until all configuration changes take effect. Some device vendors may require a delay function to allow their device to complete the write to memory.

Parameters

Name	Description	Value
Name	Name such as ABC 11n AP	String

Return Values

None.

Example

```
UCC: ap_config_commit,NAME,abc11n
AP: status,RUNNING
AP: status,COMPLETE
```

8.4 AP_DEAUTH_STA

This command is used to send the de-authentication frame to the provided station.

Parameters

Name	Description	Value
Name	A name such as ABC 11n AP	String
STA_MAC_Address	MAC address of STA	String

Return Values

None.

Example

```
UCC: ap_deauth_sta,NAME,abc11n,interface,24g,sta_mac_address,aa:bb:cc:dd:ee:ff,minorcode,2
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.5 AP_GET_MAC_ADDRESS

This command returns the MAC address of the specified AP wireless interface.

Parameters

Name	Description	Value
Interface	The radio hardware interface. Space separated	String 2.4 5.0 60
Name	Name such as ABC 11n AP	String

Return Values

Name	Description	Value
MAC	MAC address of the wireless interface	MAC address

Example

```
UCC: ap_get_mac_address,interface,interfaceId_1
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,mac,11:22:33:44:55:66
```

```
UCC: ap_get_mac_address,interface,60
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,mac,11:22:33:44:55:66
```

8.6 AP_RESET_DEFAULT

This command is used to reset the AP to its default program specific configuration.

Parameters

Name	Description	Value
Name	Name such as ABC 11n AP	String
Program	Program name	String WPA2 WMM HS2 HS2-R2 VHT 11n 60GHz
Type	Type of the device – Test bed or DUT For VHT, refer to Annex A of the Test Plan for the DUT and Annex E for test bed. For 60GHz, refer to Annex F in the Test Plan for DUT and AnnexE for test bed.	String Test bed DUT

Return Values

None.

Example

```
UCC: ap_reset_default,NAME,abc11n,interface,24g,program,wmm
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
UCC: ap_reset_default,NAME,abc,program,VHT,type,Testbed
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
CCG: ap_reset_default,NAME,abc11n,interface,24g,program,wmm
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
CCG: ap_reset_default,NAME,xyz-AP,program,HS2-R2
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
UCC: ap_reset_default,NAME,abc,Program,60GHz,Type,DUT
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.7 AP_SEND_ADDBA_REQ

This command is used to send an ADDBA request from the AP to the STA device.

Parameters

Name	Description	Value															
Name	A name such as ABC 11n AP	String															
STA_MAC_Address	MAC address of station device	String Example: 00:11:22:33:44:55															
TID	<p>Traffic identifier</p> <p>For values 0,1,4,6, the number represents the Access Category and not the absolute number. The device can choose a User Priority (based on implementation) corresponding to that AC.</p> <table> <tr> <th>Value</th><th>Corresponding AC</th><th>Device can choose UP between</th></tr> <tr> <td>0</td><td>BE</td><td>0 or 3</td></tr> <tr> <td>1</td><td>BK</td><td>1 or 2</td></tr> <tr> <td>4</td><td>VI</td><td>4 or 5</td></tr> <tr> <td>6</td><td>VO</td><td>6 or 7</td></tr> </table>	Value	Corresponding AC	Device can choose UP between	0	BE	0 or 3	1	BK	1 or 2	4	VI	4 or 5	6	VO	6 or 7	Integer 0-15
Value	Corresponding AC	Device can choose UP between															
0	BE	0 or 3															
1	BK	1 or 2															
4	VI	4 or 5															
6	VO	6 or 7															

Return Values

None.

Example

```
UCC: ap_send_addba_req,NAME,xyz11n,sta_mac_address,00:11:22:33:44:55,tid,0
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.8 AP_SEND_BCNRPRT_REQ

This command is used to trigger an AP to send a Beacon Report Request.

Parameters

Name	Description	Value
APChanRpt	AP Channel Report	???
BSSID	BSSID	String
Channel	Channel number	Short Integer 0- 255
DestMAC	Destination MAC Address	String
MeaDur	Measurement Duration in milliseconds. The minimum is 20 ms.	Integer
MeaDurMand	Measurement Duration Mandatory	Boolean
MeaMode	Measurement Mode	String ACTIVE TABLE PASSIVE
Name	A name such as VOEAP	String
RandInt	Randomization Interval	String ANY Or a number
RegClass	Regulatory Class	Short Integer 12 1
ReqInfo	Information Codes	String For example: 0/48/54...
RptCond	Reporting Condition Code	Short Integer
RptDet	Reporting Detail	Boolean 0 1
SSID	SSID	String

Return Values

None.

Example

CCG:
 ap_send_bcnrpt_req,NAME,VOE1, DestMac,11:22:33:44:55:66,REGCLASS,12,CHANNEL,255,RANDINT,ANY,MEADUR,20,MEAMODE,ACTIVE,RPTCOND,0,RPTDET,0,MEADURMAND,1,APCHANRPT,(???) ,REQINFO,0/48/54,

AP: status,RUNNING

AP: status,COMPLETE

8.9 AP_SEND_BSSTRANS_MGMT_REQ

This command is used to send a BSS Transition Management Request to a STA.

Parameters

Name	Description	Value
BSSID	BSSID in aa:bb:cc:dd:ee:ff format	String
Candidate_List	BSS Transition Candidate List Entries which are Neighbor Report Elements, SSID/BSSID of other AP	String
DestAddr	Destination MAC address	String
Dis_Assoc_Time	Disassociation Timer	Integer
Name	A name such as ABC 11n AP	String
Req_Mode	Request mode value	Integer 1,2,3,4,5,6 and 7
Validity_Int	Validity Interval	Integer Any non zero number

Return Values

None.

Example

```
ap_send_bsstrans_mgmt_req,NAME,CiscoVE1AP,BSSID,08:D0:9F:16:2D:6F,DESTADDR,00:10:18:96:32:D9,REQ_MODE,1,DIS_ASSOC_TIME,0,VALIDITY_INT,200,CANDIDATE_LIST,WiFi1
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.10 AP_SEND_LINK_MEA_REQ

This command is used to send a Link Measurement Request to a STA.

Parameters

Name	Description	Value
BSSID	Basic service set identifier	String
Count	Number of Link Measurement requests to be sent	Integer
DestAddr	Destination MAC address	String
Interval	Time (in seconds) between two Link Measurement Requests	Integer
Name	A name such as ABC 11n AP	String

Return Values

None.

Example

```
CCG: ap_send_link_mea_req,NAME,abc,DESTADDR,aa:bb:cc:dd:ee:ff
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.11 AP_SEND_TSMRPT_REQ

This command is used to initiate or force (unless specified otherwise in the command description) an AP to send Traffic Stream Metrics (TSM) report. If frame not supported or not allowed due to some reason, the command will return ERROR.

This command may be applicable to a DUT based on the type of program.

Parameters

Name	Description	Value
Avg_Err_Thr	Average Error Threshold	Integer
BRang	Bin 0 Range	Short Integer
BSSID	BSSID in aa:bb:cc:dd:ee:ff format	String
Con_Err_Thr	Cons Error Threshold	Integer
Del_MSDU_Cnt	Del MSDU Count	Integer
DestAddr	Destination MAC address (STA MAC address)	String
Duration	Duration	Boolean Yes No
Mea_Dur	Measurement duration in TU units (~10s)	Integer
Name	A name such as VOEAP	String
PeerAddr	Peer Station MAC Address	String
RandInt	Randomization Interval	String ANY Or a number
Repetition	Number of repetitions	Short Integer 0-0xFFFF
TID	Traffic Identifier	Integer
TrigRpt	Triggered Reporting, normal (0) or triggered reporting requests	Boolean
TrigTimeOut	Trigger Time Out in milliseconds	Integer

Return Values

None.

Example

```
CCG: ap_send_tsmrpt_req,NAME,VOE1,DURATION,Yes,TID,93,PEERADDR,aa:bb:cc:dd:ee:ff
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.12 AP_SET_11D

This command is used to set the AP's 802.11d related parameters.

Parameters

Name	Description	Value
CountryCode	Two-character country code. Example: United States = US China = CN	String
Name	Name of AP	String
Regulatory_Mode	Regulatory mode	String 11d

Return Values

None.

Example

```
UCC: ap_set_11d,NAME,xyz,regulatory_mode,11d,CountryCode,US
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.13 AP_SET_11H

This command is used to set the AP's 802.11h related parameters.

Parameters

Name	Description	Value
DFS_Chan	Channel for operation	Integer
DFS_Mode	Enables/disables dynamic frequency selection mode	String Enable Disable
Name	Name of AP	String
Regulatory_Mode	Regulatory mode	String 11d

Return Values

None.

Example

```
UCC: ap_set_11h,NAME,xyz,dfs_mode,enable,dfs_chan,56,regulatory_mode,11h
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.14 AP_SET_11N_WIRELESS

This command is used to set the AP's 802.11n related parameters. The AP Control Agent should return the “ERROR” response element for unsupported features only if it cannot produce the effective setting.

Parameters

Name	Description	Value
AMPDU	Enables/disables AMPDU Aggregation	Enable Disable
Interface	A radio interface	String
Name	Name of AP	String
Sgi20	Short Guard Interval	String Enable Disable
Spatial_Rx_Stream	Sets the Rx spatial streams of the AP and which means the Rx MCS Rates capability	String 1SS – MCS 0-7 2SS – MCS 0-15 3SS – MCS 0-23
Spatial_Tx_Stream	Sets the Tx spatial streams of the AP and which reflects the Tx MCS Rates capability	String 1SS – MCS 0-7 2SS – MCS 0-15 3SS – MCS 0-23
Width	Channel width for 11n channel bonding	Integer 20 40 Auto

Return Values

None.

Examples

```
UCC:
ap_set_11n_wireless,NAME,abc,interface,2.4g,sgi20,disable,SPATIAL_TX_STREAM,2SS,SPATIAL_RX_STREAM,2SS,width,20
```

```
AP:status,RUNNING
```

```
AP:status,COMPLETE
```

8.15 AP_SET_APQOS

This command is used to set the AP WMM parameters.

Parameters

Name	Description	Value
ACM_<actype>	ACM VO/VI/BE/BK (on/off)	String
AIFS_<actype>	AIFS VO/VI/BE/BK	Short integer
CWMax_<actype>	CWMAX VO/VI/BE/BK	Short integer
CWMin_<actype>	CWMIN VO/VI/BE/BK	Short integer
Name	Name such as ABC 11n AP	String
TxOp_<actype>	TXOP VO/VI/BE/BK	Short integer

Return Values

None.

Example

```
UCC:ap_set_apqos,NAME,abc_11g,INTERFACE,2.4g,cwmin_vo,3,cwmax_vo,7,AIFS_vo,32,TXOP_VO,32,ACM_VO,on
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.16 AP_SET_HS2

This command is used to configure the AP's Hotspot 2.0 parameters.

Parameters

Name	Description	Value
Accs_Net_Type	Access network type1	Integer 0-15
ANQP	ANQP Query	Integer 0 = Disabled 1 = Enabled
BSS_LOAD	ID number. Refer HS2.0 test plan Appdex B.1 for details. Value of 0 shall be used if QBSS IE has to be disabled on AP (Testbed AP only)	Integer
Conn_Cap	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
DGAF_Disable	Downstream group-addressed forwarding disabled bit	Integer 0 1
Domain_List	Semicolon separated list of domain names	String
GAS_CB_Delay	GAS comeback delay in TUs This parameter only applies to an AP that supports 4-frame GAS exchange. This parameter only applies test bed devices.	Integer
HESSID	HESSID	String
ICMPv4_Echo	Filter function for ICMPv4 echo requests Allows ICMP Echo request when enabled. Denies ICMP echo request when disabled.	Integer 0 = Disabled 1 = Enabled
Interworking		Integer 0 = Disabled 1 = Enabled
IP_Add_Type_Avail	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
L2_Traffic_Inspect	L2 traffic inspection and filtering This parameter applies to APs that support the built-in inspection and filtering function.	Integer 0 = Disabled 1 = Enabled
NAI_Realm_List	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
Name	Name such as ABC 11n AP	String
Net_Auth_Type	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
Oper_Class	ID number. Refer HS2.0-R2 test plan Appdex B.1.12 for details. Applicable only for Testbed AP and not applicable to DUT (APUT).	Integer
Oper_Name	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
OSU_METHOD	Provisioning Protocol The value shall be a space separated List of OSU Method List where multiple OSU providers are present with orderly mapping of OSU providers.	String OMADM SOAP
OSU_ICON_TAG	Request the AP to use a different content of an icon file without changing the file name or any other information in the OSU Providers List.	Integer 1 = Default 2 = Content of the image file wifi-abgn-logo_270x73.png is sent instead of the image file indicated in the OSU_PROVIDER_LIST.

Name	Description	Value
OSU_PROVIDER_LIST	ID number. Refer HS2.0 test plan Appdex B.1 for details	Integer
OSU_SERVER_URI	OSU Server URI The value shall be a space separated List of URIs where multiple OSU providers are present with orderly mapping of OSU providers.	String
OSU_SSID	SSID of OSU ESS	String
PLMN_MCC	MCC For multiple values, list the MCCs separated by a semicolon.	String
PLMN_MNC	MNC For multiple values, list the MCCs separated by a semicolon.	String
Proxy_ARP		Integer 0 = Disabled 1 = Enabled
QoS_MAP_SET	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
Roaming_Cons	Semicolon separated list of Roaming Consortium OI in Hex String value 'Disabled' will disable the Roaming Consortium OI.	String
STA_MAC	Station MAC Address Valid only as an optional parameter with QoS_MAP_SET, to enable corresponding QoS Map Set parameters for an associated STA	String
Venue_Name	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
WAN_Metrics	ID number. Refer HS2.0 test plan Appdex B.1 for details.	Integer
WLAN_TAG	To link HS2.0 parameters to specific WLAN	Integer

Return Values

None.

Example

```
UCC: ap_set_hs2,NAME,abc_11a, Interworking,1,Accs_Net_Type,2,Internet,0,
Venue_Grp,2,Venue_Type,8,HESSID,00:11:22:33:44:55,Roaming_Cons,506F9A,
DGAF_Disable,0,ANQP,1,L2_Traffic_Inspect,1
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
UCC: ap_set_hs2,NAME,abc_11a,PLMN_MCC,310;234PLMN_MNC026;56
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
UCC: ap_set_hs2,NAME,abc_11a, L2_TRAFFIC_INSPECT,1
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```



CCG: ap_set_hs2,NAME,abc_11a,OSU_PROVIDER_LIST,1,OSU_SERVER_URI,https://osu-server.r2-testbed.wi-fi.org,OSU_SSID,OSU,OSU_ICON_TAG,1

AP: status,RUNNING

AP: status,COMPLETE

CCG: ap_set_hs2,NAME,abc_11a,QoS_MAP_SET,2,STA_MAC,00:11:22:33:44:55

AP: status,RUNNING

AP: status,COMPLETE

CCG: ap_set_hs2,NAME,abc_11a ,OSU_SERVER_URI, https://osu-server.r2-testbed-red.wi-fi.org
https://osu-server.r2-testbed-blue.wi-fi.org https://osu-server.r2-testbed-green.wi-fi.org
https://osu-server.r2-testbed-orange.wi-fi.org,OSU_SSID,OSU,OSU_METHOD,SOAP SOAP SOAP
SOAP,OSU_PROVIDER_LIST,10

AP: status,RUNNING

AP: status,COMPLETE

8.17 AP_SET_PMF

This command is used to configure the AP PMF setting. .If an AP device already handles PMF setting through AP_SET_SECURITY, this command shall be ignored.

Parameters

Name	Description	Value
Interface	Radio hardware interface	String
Name	Name such as ABC 11n AP	String
PMF	Configures the PMF settings	String Required Optiona Disabled

Return Values

None.

Example

```
UCC:ap_set_pmf,NAME,abc_11g,INTERFACE,2,pmf,disabled
```

```
AP:status,RUNNING
```

```
AP:status,COMPLETE
```

8.18 AP_SET_RADIUS

This command is used to set the AP's RADIUS server information.

Parameters

Name	Description	Value
IPAddr	IP Address of RADIUS server	String
Name	Name such as ABC 11n AP	String
Password	Shared secret between RADIUS server and AP	String
Port	Port number of RADIUS service	String
WLAN_TAG	To link Radius Server to specific WLAN	Integer

Return Values

None.

Example

Example 1:

```
UCC: ap_set_radius, NAME, abc11n, INTERFACE,2.4g,IPADDR, 192.168.1.13, PORT, 1812, PASSWORD, pass
AP: status, RUNNING
AP: status, COMPLETE
```

Example 2:

```
CCG: ap_set_radius,NAME,abc11n,WLAN_TAG,1,IPADDR,192.165.100.207,PORT,1812, PASSWORD,12345678
AP: status, RUNNING
AP: status, COMPLETE
```

Example 3:

```
CCG: ap_set_radius,NAME,abc11n,WLAN_TAG,2,IPADDR,192.165.140.10,PORT,1813, PASSWORD,radius
AP: status, RUNNING
AP: status, COMPLETE
```

8.19 AP_SET_RFEATURE

This command is used to set a specific feature during runtime.

Parameters

Name	Description	Value
ChNum_Band	Used to change the channel in which the AP is operating and the bandwidth simulatenously. For example: Chnum_band,100;80	String (channel number;bandwidth)
CTS_Width	Sets the bandwidth of CTS from the test bed	Integer 20 40 80
Name	Name of the AP	String
NDPA_STAinfo_MAC	Configures the AP in real time to a specific MAC address to solicit immediate feed back from the STA	String MAC address of STA aa:bb:cc:dd:ee:ff
NSS_MCS_Opt	Refers to nss_operating_mode; mcs_operating_mode. This parameter tells what spatial stream and mcs rate the AP will operate in from the supported set of capabilities. For example, To set an AP to operate in 1SS with MCS 9, then nss_mcs_opt,1;9 To set an AP to operate in 4SS with MCS 7, then nss_mcs_opt,4;7 To set it to device's default capability nss_mcs_opt,def;def	String
Opt_Md_Notif_IE	Used to set the operating mode notification element for NSS (number of spatial streams) and channel width. For example, to set the operating mode notification element with NSS=1 and BW=20Mhz, Opt_md_notif_ie,1;20	String (NSS;channel width)
Program	Program name	String
RTS_Force	Forces the test bed AP to send RTS during dynamic /static BW Signaling in real time without commit command	String Enable Disable
TxBandwidth	Sets the AP transmission bandwidth	Integer 20 40 80 160
Type	Program name	String VHT

Return Values

None.

Example

UCC: ap_set_rfeature,NAME,xyz,type,VHT,nss_mcs_opt,1;9

AP: status,RUNNING

AP: status,COMPLETE

8.20 AP_SET_RRM

This command is used to configure the AP Radio Resource Management parameters.

Parameters

Name	Description	Value
BAE	Enables/disables BSS Available Admission Control Element(BAE)	String Enable Disable
BLE	Enables/disables BSS Load Element(BLE)	String Enable Disable
BSSTrans	Enables/disables BSS transition	String Enable Disable
CE	Enables/disables Country Element (CNE)	String Enable Disable
Name	Name such as ABC 11n AP	String
PCE	Enables/disables Power Constraint Element(PCE)	String Enable Disable
QTE	Enables/disables Quiet Element(QTE)	String Enable Disable

Return Values

None.

Example

```
UCC: ap_set_rrm,NAME,VOE1,BAE,Enable,QTE,Enable
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.21 AP_SET_SECURITY

This command is used to configure an AP security test setting.

Parameters

Name	Description	Value
Encrypt	Encryption cipher configuration	String WEP TKIP AES AES-GCMP
Interface	Radio hardware interface	String
KeyMgmt		String NONE WPA-PSK WPA2-PSK WPA-ENT WPA2-ENT WPA2-PSK-Mixed WPA2-Mixed OSEN
Name	A name such as ABC 11n AP	String
PMF	Robust management frame protection configuration	String Required Optional Disabled
PreAuthentication	Enables/disables Preauthentication in the AP	String Enabled Disabled
PSK	Passphrase	String
SHA256AD	Enables/Disables SHA-256 Advertisement. This parameter works with PMF "Optional" setting to publish both SHA1 and SHA256.	String Enable Disable
SSID	A SSID to the security profile to distinguish it from others if needed	String
WEKey	WEP key	String
WLAN_TAG	To link KEYMGNT (security profile) to specific WLAN	Integer

Return Values

None.

Example

```
UCC: ap_set_security,NAME,abc_11a,INTERFACE,2.4g,SSID,wifi,KEYMGNT,WPA2-PSK,ENCRYPT,TKIP,PSK,12345678
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```




CCG: ap_set_security,NAME,abc_11a,INTERFACE,2.4g,SSID,wifi,KEYMGNT,WPA2-PSK,ENCRYPT,TKIP,PSK,12345678

AP: status,RUNNING

AP: status,COMPLETE

CCG: ap_set_security,NAME,abc_HS2R2,WLAN_TAG,1,KEYMGNT,WPA2-ENT

AP: status,RUNNING

AP: status,COMPLETE

CCG: ap_set_security,NAME,abc_HS2R2,WLAN_TAG,2,KEYMGNT,NONE

AP: status,RUNNING

AP: status,COMPLETE

CCG: ap_set_security,NAME,abc_HS2R2,WLAN_TAG,2,KEYMGNT,OSEN

AP: status,RUNNING

AP: status,COMPLETE

8.22 AP_SET_STAQOS

This command is used to set the STA WMM parameters.

Parameters

Name	Description	Value
ACM_<actype>	ACM VO/VI/BE/BK (on/off)	String
AIFS_<actype>	AIFS VO/VI/BE/BK	Short integer
CWMax_<actype>	CWMAX VO/VI/BE/BK	Short integer
CWMin_<actype>	CWMIN VO/VI/BE/BK	Short integer
Name	Name such as ABC 11n AP	String
TxOp_<actype>	TxOP VO/VI/BE/BK	Short integer

Return Values

None.

Example

```
UCC:ap_set_sta qos,NAME,abc_11g,INTERFACE,2.4g,cwmin_vo,3,cwmax_vo,7,AIFS_vo,32,TXOP_VO,32,ACM_VO,on
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

8.23 AP_SET_WIRELESS

This command configures an AP with the provided test parameters. The AP Control Agent shall return an “ERROR” response element for unsupported features only if it cannot produce the effective setting. For example, a device that does not implement the HT Greenfield feature shall return “ERROR” if Greenfield is set to “Enable”, but return “COMPLETE” if Greenfield is set to “Disable”.

Parameters

Name	Description	Value
40_Intolerant	Enables/disables 40 Mhz Intolerant	Enable Disable
ADDBA_Reject	Enables/disables rejecting any ADDBA request by sending ADDBA response with status “decline”	Enable Disable
AMPDU	Enables/disables AMPDU Aggregation	Enable Disable
AMSDU	Enables/disables AMSDU Aggregation	Enable Disable
BCNInt	Beacon Interval	String
BW_Sgnl	Enables/disables the ability to send out RTS with bandwidth signaling	String Enable Disable
Channel	Channel number For example: 36 for a single band operation 36;6 for a dual band operation 64,100,140,144,165 for extended channel operation	String
ChannelUsage	Set the channel usage	String
CountryCode	Two-character country code in Country Information Element	String For example, ‘US’
Domain	Mobility Domain	String
DTIM	DTIM Count	Integer
Dyn_BW_Sgnl	If ‘DYN_BW_SGNL’ is enabled then the AP sends the RTS frame with dynamic bandwidth signaling, otherwise the AP sends RTS with static bandwidth signaling. BW signaling is enabled on the use of this command.	Enable Disable
Frgmnt	Fragmentation	Short integer
FT_DS	Enables/disables Fast BSS Transition Over the DS	String Enable Disable
FT_OA	Enables/disables Fast BSS Transition Over the Air	String Enable Disable
GAS_Frag_Thr	GAS Fragmentation Threshold value in octets. If the Query reponse length is greater than the GAS fragmentation threshold value, the responding STA shall use GAS fragmentation to deliver the query response. The GAS fragments are transmitted using the GAS Comeback Request and GAS Comeback Response frame exchange as defined in IEEE 802.11-2012 specification section 10.24.3.1.4. For example, 100 - If query response is greater than 100 octets, GAS fragmentation is used. 0 - Use default threshold value for GAS fragmentation.	Integer

Name	Description	Value
Greenfield	Enables/disables HT Greenfield	Enable Disable
HS2	Enables/disables HS 2.0 Indication element	Integer 0 = Disabled 1 = Enabled
HT_TKIP	Enable/disables HT with TKIP	String Enable Disable
HTC-VHT	Enables/disables support for receiving a VHT variant HT control field	String Enable Disable
Interface	A radio interface for multiple radio AP	String
LDPC	Enables/disables the use of LDPC code at the physical layer for both Tx and Rx side	String Enable Disable
MCS_32, MCS32	Enables/disables HT Duplicate Mode	Enable Disable
MCS_FixedRate	Fixed MCS rate. Single value. For VHT, If 'MODE' = 11na, MCS rate varies from 0 to 31 If 'MODE' = 11ac, MCS rate varies from 0 to 9	Short Integer 0-32
MCS_Mandatory	MCS mandatory streams	Integer 4 7 12 15 23
MCS_Supported	Number of MCS supported streams	Integer 4 7 12 15 23
Mode	Mode. For example: 11a for single band operation 11ac; 11ng for dual band operation	String 11b 11bg 11bgn 11a 11g 11na 11ng 11ac 11ad
MU_TxBF	Enables or disables Multi User (MU) TxBF beamformer capability with explicit feedback	String Enable Disable
MU_NDPA_FrameFormat	NDPA Frame Format Setting Set to 0 for VHT, TA=0 Set to 1 for Non-HT, TA=1 Set to 2 for Non-HT,TA=0	Integer 0 1 2

Name	Description	Value
	Refer to IEEE 11ac 8.3.1.20 and 9.7.6.6 for implementation.	
Name	A name such as ABC 11n AP	String
NEIBRPT		String Enable Disable
NoAck	Turns on off Ack	String On Off
NSS_MCS_Cap	This parameter gives a description of the supported spatial streams and mcs rate capabilities of the STA. Example – For setting a STA to support 2SS with MCS 0-9: nss_mcs_cap,2;0-9	String
Offset	Secondary channel offset	Above Below
P2PCoexistence	Checks for the P2PCoexistence bit	Integer 1 0
P2PMgmtBit	Set P2P Management bit	Boolean
PreAuthentication	Enables/disables Preauthentication in AP	String Enabled Disabled
Prog	Program name	String
Program	A certification program currently supported	String VHT HS2 HS2-R2 60GHz
Pwr_Const	AP Power Contstrain (in dBm)	Integer
Radio	Turn On/Off the radio of given interface	Boolean
RIFS_Test	Enables/disables the Set Up (Tear Down) RIFS Transmission Test as instructed within Appendix H of the 11n Test Plan	Enable Disable
RRM		String Enable Disable
RTS	Threshold	Short integer
SGI_20	Enables/disables the Short Guard Interval in 20MHz	String Enable Disable
SGI20	Enables/disables the Short Guard Interval	Enable Disable
SGI40	Enables/disables the Short Guard Interval	String Enable Disable
SGI80	Enables/disables the Short guard interval at 80 Mhz	String Enable Disable
Spatial_RX_Stream	Sets the Rx spacial streams of the AP which relates to the Rx MCS Rates capability.	String

Name	Description	Value
	<p>For VHT, If 'MODE' = 11na, sets the Rx spacial streams of the AP which relates to the Rx MCS Rates capability. If 'MODE' = 11ac, sets the Rx spacial streams of the AP. No inter-dependency of number of spatial streams and MCS rates.</p>	<p>For 11n: 1SS – MCS 0-7 2SS – MCS 0-15 3SS – MCS 0-23</p> <p>For 11ac: 1SS 2SS 3SS 4SS</p>
Spatial_TX_Stream	<p>Sets the Tx spacial streams of the AP and which reflects the Tx MCS Rates capability. For VHT, If 'MODE' = 11na, sets the Tx spacial streams of the AP which relates to the Tx MCS Rates capability. If 'MODE' = 11ac, sets the Tx spacial streams of the AP. No inter-dependency of number of spatial streams and MCS rates.</p>	<p>String</p> <p>For 11n: 1SS – MCS 0-7 2SS – MCS 0-15 3SS – MCS 0-23</p> <p>For 11ac: 1SS 2SS 3SS 4SS</p>
SpectrumMgt	Enables/disables the spectrum management feature with minimum number of beacons with switch announcement IE = 10, channel switch mode = 1	<p>String</p> <p>Enable Disable</p>
SSID	SSID	String
STBC_Rx	Enables/disables Rx STBC in the AP	<p>Integer</p> <p>0 1</p>
STBC_TX	<p>STBC Transmit Streams</p> <p>Comma separated list of the number of spatial streams and the number of space time streams.</p> <p>For example – 1spatial stream and 2 space time streams = 1;2</p>	String
TDLSChswitchProhibit		<p>String</p> <p>Enable Disable</p>
TDLSProhibit		<p>String</p> <p>Enable Disable</p>
TxBF	Enables/disables the SU TxBF beamformer capability with explicit feedback	<p>String</p> <p>Enable Disable</p>
VHT_TKIP	Enables/disables TKIP in VHT mode	<p>String</p> <p>Enable Disable</p>
VHT_WEP	Enables/disables WEP in VHT mode	<p>String</p> <p>Enable Disable</p>
Width	<p>Channel width for 11n channel bonding</p> <p>Used to configure the AP channel width to 160 MHz</p>	<p>20 40 60 80 160 Auto</p>

Name	Description	Value
Width_Scan	BSS channel width trigger scan interval in seconds	Short integer
WLAN_TAG	This shall be used to differentiate between multiple BSSs on an AP. (Example: WLAN_TAG value 1 is for first BSS. WLAN_TAG 2 is for second BSS.) If not present, WLAN_TAG shall default to 1	Integer
WME	Enables/disables WME	String On Off
WMMPS	Disable/enable APSD	String On Off
Zero_CRC	Clears the CRC field	String Enable Disable

Return Values

None.

Examples

```
UCC: ap_set_wireless,NAME,abc_11n,INTERFACE,2.4g,SSID,MyNetwork,MODE,11g,CHANNEL,11, PWRSAVE,off,
WME,on,WMMPS,off,RTS,12FRGMNT,1000
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
UCC: ap_set_wireless, NAME, abc_11n, INTERFACE, 2.4g, SSID, MyNetwork, MODE,11g, CHANNEL,11,
PWRSAVE,off, WME,on, WMMPS,off, RTS,12, FRGMNT,1000, 40_INTOLERANT,DISABLE
```

```
AP:status,RUNNING
```

```
AP:status,COMPLETE
```

```
CCG: ap_set_wireless,NAME,abc_11n,INTERFACE,2.4g,SSID,MyNetwork,MODE,11g,CHANNEL,11
PWRSAVE,off, WME,on,WMMPS,off,RTS,12FRGMNT,1000
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
CCG: ap_set_wireless,NAME,abc_HS2,WLAN_TAG,1,SSID,MyNetwork_1,MODE,11g,CHANNEL,11
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE
```

```
CCG: ap_set_wireless,NAME,abc_HS2,WLAN_TAG,2,SSID,MyNetwork_2,MODE,11a,CHANNEL,36
```

```
AP: status,RUNNING
```



AP: status,COMPLETE

CCG: ap_set_wireless,NAME,abc_HS2,WLAN_TAG,1,SSID,MyNetwork_1,MODE,11g,CHANNEL,11,GAS_Frag_thr,100

AP: status,RUNNING

AP: status,COMPLETE

9 Wi-Fi Test Suite Sniffer Control API

9.1 Sniffer Control Commands

This section defines the commands for controlling the Wi-Fi Test Suite sniffer. These commands can be either production, draft, optional, or not applicable depending on the platform component and certification program.

9.2 SNIFFER_CONTROL_START

This command is used to start the Wi-Fi Test Suite sniffer to capture packets.

Parameters

Name	Description	Value
Bandwidth	20/40/80 MHz bandwidth	Integer 20 40 80
Channel	A wireless channel	Short integer
DualChnl	A wireless channel used by secondary sniffer interface	Short Integer
ExtnChnl	Extension Channel for 40 Mhz	Integer
Filename	Name of the Wi-Fi Test Suite sniffer trace file	String
Interface	Interface on which the Wi-Fi Test Suite sniffer will capture data. The value "Both" will start the Wi-Fi Test Suite sniffer capture on both the wired and wireless interfaces.	String Both Wired Wireless
Name	Name used to identify the Wi-Fi Test Suite sniffer	String
Passphrase	WPA2-PSK passphrase to generate the security keys	Integer 1-8
Program	Program Name	String General PMF TDLS VOE WFD 60GHz
Reset	Flushes the existing SA data and counters for the AP and STA	Integer 1 = Flush 0 = No action
SharedSecret	The key used as a credential for authentication purposes. The value may be 8 or 64 characters in length. The 11n test plan specifies a 64 char key. ex: 1234567890123456789012345678901234567890123456789012345678901234	String
StartDelay	Time in seconds	Integer
WiredIF	The name of the test network interface that is connected between the server and the Sniffer via the hub/mirror switch.	String ex: eth1

Return Values

None.

Examples

```
UCC:  
sniffer_control_start,filename,_capturefilename,channel,6,Program,PMF,Passphrase,12345678,Reset,1!D  
EFAULT
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE
```

9.3 SNIFFER_CONTROL_STOP

This command is used to stop the Wi-Fi Test Suite sniffer capture.

Parameters

Name	Description	Value
Name	Name used to identify the Wi-Fi Test Suite sniffer	String
Program	Program Name	String General PMF TDLS VOE WFD 60GHz

Return Values

None.

Examples

```
UCC: sniffer_control_stop,name,wireShark_wlan
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE
```

9.4 SNIFFER_CONTROL_UPLOAD

This command uploads the capture log (pcap/media/text) to the control console.

Parameters

Name	Description	Value
DestpPath	UCC Log folder name created for test execution in \bin\log\<Log Folder Name>	String
Filename	List of colon separated files names to be uploaded including the file extension if applicable.	String
Filetype	Type of file	String media text pcap
Name	Name used to identify the Wi-Fi Test Suite sniffer	String

Return Values

None.

Examples

```
UCC: sniffer_control_upload,filename,5113_Step10_Video.mp4,filetype,media,destpath, WFD-4.1.1_Aug-02-2012__12-55-23
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE
```

9.5 SNIFFER_CONTROL_FIELD_CHECK

This command verifies the specified fields or bit values that are specified through the parameters in the trace file. The input parameters give the fields and their expected values. The implementation will check the Wi-Fi Test Suite sniffer trace file and return “SUCCESS” if the values match or return “FAIL” if the values does not match. Only once check may be executed at a time.

Parameters

Name	Description	Value
11u_AdvProto	Confirms the IEEE 802.11u Advertisement Protocol element	Integer 0 – Should not exist 1 – Should exist
11u_AdvProtoID	Confirms if the IEEE 802.11u Advertisement Protocol ID element matches the given value	Integer value
11u_AdvProtoID	Confirms if the 11u Advertisement protocol ID matches the given value	Integer
11u_IW	Confirms the IEEE 802.11u IW element	Integer 0 = Should not exist 1 = Should exist
11u_IW_ANT	Confirms the IEEE 802.11u Access Network Type value	Integer
11u_IW_HESSID	Confirms the IEEE 802.11u HESSID value	String
11u_IW_HESSID_present	Check for HESSID element	Integer 0 1
11u_IW_INT	Confirms the IEEE 802.11u Internet bit value	Integer
11u_IW_VenueGRP	Confirms the IEEE 802.11u Venue Group value	Integer
11u_IW_VenueType	Confirms the IEEE 802.11u Venue Type value	Integer
11u_RC	Confirms the IEEE 802.11u Roaming Consortium element	Integer 0 – Should not exist 1 – Should exist
11u_RC_numOIs	Confirms if the number of OIs in 11u Roaming consortium matches the given value	Integer
11u_RC_OI1_Len	Confirms if the Roaming consortium OI1 length matches the given value	Integer
11u_RC_OI2_Len	Confirms if the Roaming consortium OI2 length matches the given value	Integer
11u_RC_Value	Confirms the IEEE 802.11u Roaming Consortium element value	String
11u_RC_Value1	Confirms if the OI number 1 in 11u Roaming consortium matches the give value	String Hex value separated by a colon
11u_RC_Value2	Confirms if the OI number 2 in 11u Roaming consortium matches the give value	String Hex value separated by a colon
2040BSSIntolerantChnlRep	Confirms the action frame 20/40 Coex management frame with 20/40 Bss Intolerant Channel Report field	0 = Not present 1 = Should present -1 = Ignore
20MhzBSSWidthReq	Confirms the action frame 20/40 Coex management frame with a 20Mhz Bss Width Request field	0 = Not set 1 = Set -1 = Ignore
3GPP_GUD	Confirms if the GUD value in 3GPP information matches the given value	Integer

Name	Description	Value
3GPP_IEI	Confirms if the IEI value in 3GPP information matches the given value	Integer
3GPP_NumPLMN	Confirms if the number of PLMNs matches the given value	Integer
3GPP_PLMN	Confirms if the PLMN value in 3GPP information matches the given value	Hex with 0x prefix
3GPP_PLMNLen	Confirms if the length of PLMN matches the given value	Integer
3GPP_UDHL	Confirms if the UDHL values in 3GPP information matches the given value	Integer
40MhzCheck	Confirms the data frame transmitted at 20 or 40 Mhz rate	Integer 0 = Not present 1 = Should be present -1 = Ignore
802.11 Management Action frames dialog token	Confirms the P2P Action Management Action frames dialog token	Integer
ACKFrame	Confirms whether an Ack frame follows the data / qosdata	1= Ack frame present after the data 0 = Ack frame not present after data / qosdata
AckPolicy	Acknowledge policy bits set or not set	Integer 0 = Ack Policy not set 1 = Ack Policy set
Action_Status		
ActionCode	Confirms the action frame code of the current action frame	20-40BssCoEx GONegReq = Group Owner Negotiation Request GONegRes = Group Owner Negotiation Response GONegCon = Group Owner Negotiation Response ProvDisReq = Provision Discovery Request P2PInvReq = P2P Invitation Request P2PInvResp = P2P Invitation Response AddBaResp = ADDBA Response VHTCmpBmForm = VHT compressed beam forming frame Provdresp P2PPresenceResp GasInitReq GasinitResp GasComebackResp DLSReq
ActionStatus	Confirms the Status code in the Action frame	Integer 0 = Disabled 1 = Enabled
ACType	Type of stream and the UP field. UP field is a 3 bit value. The function will match the other valid values for the specified AcType. (For example 4 or 5 for a value of 5)	Integer 0 = BE 1 = BK 5 = VI 6 = VO

Name	Description	Value
Ad_Proto_ID	Confirms if the 11u Advertisement protocol ID matches the given value.	String anqp
ADDBA_Resp	Confirms the the ADD Block Ack Response value	0 = Decline 1 = Success -1= Ignore
AM_BcnTxtime	Confirms the Anchor Master Beacon Transmission Time of NAN device	Hex value
AMI_Verify_Addr	Confirms the octets of the NAN interface address	Hex value
AMI_Verify_MasterPref	Confirms the Master Preference value by checking the Anchor Master Info element	Hex value: 0x00 – 0xFF
AMPDU	Confirms if AMPDU is enabled in the data/qos data frames	0 = No AMPDU frames 1 = AMPDU frames -1 = Ignore
ANQP_Cap_List	Confirms if the ANQP capability list element is present	Integer 0 – Should not exist 1 – Should exist
ANQP_Info_ID	Confirms if the Info ID of ANQP element matches the given value	String RoamConsortList = Roaming Consortium List
ANQP_Info_Len	Confirms if the Length of ANQP information element matches the given value	Integer value
ANQP_Query_Info_ID	Confirms the ANQP information ID.	String NAIRealmLst DomainNameLst VenueNameInfo 3GPP AnqpCapList
ANQP_RC_OI_Len	Confirms if the Roaming consortium OI length matches the given value	Integer value
ANQP_RC_OI_Val	Confirms if the Roaming consortium OI value in ANQP frame matches the given value	Hex value separated by colon
ARP_HW_Addr_Len	ARP hardware address length	Integer 6
ARP_HW_Type	Checks for ARP hardware type	Integer 1
ARP_IP_Addr_Len	Checks for the ARP IP address length	Integer 4
ARP_OpCode	Checks for ARP OpCode	Integer 1 2
ARP_Proto_Type	Checks for ARP protocol type	Hex 0x800
ARP_Sender_IP	ARP Sender IP	String 192.165.x.x 0.0.0.0
ARP_Sender_MAC	Checks for ARP sender MAC address	String aa:bb:cc:dd:ee:ff
ARP_Target_IP	ARP target IP address	String 192.165.x.x
ARP_Target_MAC	Checks for the ARP target MAC address	String aa:bb:cc:dd:ee:ff

Name	Description	Value
AssocStatus	Confirms the Association Response status	Integer 0 = Success 1 = Fail
Auth_Param_ID	Authentication parameter ID	Integer 2 5
BA_Param_Amsdu_Support	Confirms the A-MSDU Supported subfield in the Block Ack Parameter Set field in ADDBA Response	Integer 0 = Not set 1 = Set
BCN_Int	Confirms the beacon interval in seconds matches the given range. The range is separated with a dash.	String Example: 0.02-1.0
BSS_Load_IE	Checks BSS load Information Element	Integer 0 = Disabled 1 = Enabled
BSSID	Confirms the BSSID matches the given string	String
BssLoad_AAC	Confirms the BssLoad Available Admission Capacity subfield	Integer n = value of AAC -2 = Check for sub field -1 = ignore
BSSLoad_AAC	Confirms if the AAC field is present within BSS Load element	Integer 0 – Should not exist 1 – Should exist
BssLoad_CU	Confirms the BssLoad Channel Utilization subfield	Integer n = value of CU -2 = Check for sub field -1 = ignore
BSSLoad_CU	Confirms if the CU field is present within BSS Load element	Integer 0 – Should not exist 1 – Should exist
BssLoad_IE	Confirms the QBSS Load IE in the frame	Integer 0 = Not present 1 = Should be present -1 = Ignore
BssLoad_SC	Confirms the Associated STA count in QBSS load IE in beacon frame	Integer n = value of Associated STAs -2 = Check for the subfield -1 = ignore
BSSLoad_STA_Count	Confirms the BSS Load element for STA Count Field	Integer
Cap_Qos_Bit	Confirms if the QoS capability bit value matches the given value	Integer 0 = Disabled 1 = Enabled
Capability_ESS	Confirms the ESS bit against the given value	Integer 0 = Disabled 1 = Enabled
Capability_IBSS	Confirms the IBSS bit against the given value	Integer 0 = Disabled 1 = Enabled
Category_Code	Category code	Integer 10 = BSS Transition Management Request frame

Name	Description	Value
CCMP_AD	Confirms the IE 48 in frames in the frame	Integer 0 = Not present 1 = Should be present -1 = Ignore
CertStatus_EAPAlert	Checks for certificate status in EAP messages	Integer 1 0
ChannelWidth	Confirms the Channel Width field in the HT Capabilities IE	Integer 20 = 20 Mhz only 40 = 20 and 40 Mhz -1 = Ignore
ClientHello	Checks if ClientHello message is included	Integer 0 1
ClientKeyExchange	Client key exchange	Integer 1 0
ClusterId	Confirms the Cluster Id of the NAN cluster	String
ClusterId	Confirms the Cluster Id of the NAN cluster	String
ClusterId_Format	Confirms if the cluster ID in NAN beacon is in the correct format	String
Conn_CapElem	Confirms if the Connection capability element is present	Integer 1 = Should be present 0 = Should not be present
Conn_CapElem_ProtolInfo	Confirms if the Protocol info field in connection capability element matches the given value	Integer
ConnCap_PortNo	Checks for the connection capability port number	Integer 0 20 80 443
ConnCap_PortStatus	Checks for connection capability port status	Integer 1 0
ConnCap_Proto	Connection capability protocol information	Integer 6 = HTTP 17 = VoIP 50
Country_IE	Confirms for Country IE	Integer 0 = Should not exist 1 = Should exist
CountryIE_CN	Confirms the country name	String Example, 'US'
CountryIE_Sub_Trip	Confirms if the Subband triplet exists	Integer 0 = Should not exist 1 = Should exist
CrossConBit	Confirms the Cross Connection bit in Device Capability bitmap	Integer 0 = Disabled 1 = Enabled

Name	Description	Value
CSA_IE	Confirms the Channel Switch Announcement IE	Integer 0 = Should not exist 1 = Should exist
CSW_IE	Confirms the r Channel Switch Wrapper IE	Integer 0 = Should not exist 1 = Should exist
CSW_IE_VHT_TPE	Confirms the VHT Transit Power Envelope sub-element in Channel Switch Wrapper IE	Integer 0 = Should not exist 1 = Should exist
CSW_IE_WB	Confirms the Wide Bandwidth sub-element in Channel Switch Wrapper IE	Integer 0 = Should not exist 1 = Should exist
CTS_Service_BW	Confirms the bandwidth bits of the PPDU carrying the CTS	String 80Mhz 40Mhz 20Mhz
Data_Length	Confirms the length of the packet is equal or greater than specified	Integer -1 = Ignore
Data_Rate	Confirms the data transmission rate of the NAN beacon frames	String
DataLen	Confirms the data length of the data frame. If null frame check confirm the QosNull (44 bytes)	String NULL / <Value>
DataRate	Confirms the data rate of the packet is equal or greater than specified	Integer value in Mbps. Example: 2, 54 -1 = Ignore
Deauth_Reason_Code	Deauthentication reason code	Integer 0 = Reason code not present 1 = Reason code present
Deauth_Reason_URL	Checks for Deauthentication reason URL	String http://www.r2-testbed.wi-fi.org/DeauthImminentNotice.html
Deauth_Reason_URL_Len	Checks for deauthentication reason URL length	Integer 0
Deauth_Type	Deauthentication type ID	Integer 1
DestMAC	Destination MAC address in aa:bb:cc:dd:ee:ff format	String
DGAF_Disabled	Confirms if the DGAF Disabled bit in Hotspot 2.0 Indication element is set	Integer 0 = Disabled 1 = Enabled
DialogToken	Confirms the 802.11 Management Action frames dialog token	Integer
DLS_Status	Confirms if the DLS status code matches the given value	Integer
Domain_Nlist_Len	Confirms if the length of Domain Name List matches the given value	Integer
Domain_Nlist_Value	Confirms if the Domain Name list matches the given value	String
DS_ParamSet	Confirms the DS parameter matches the supplied listen channel	Integer
EAP_Identity	Confirms if the EAP identity field matches the given value	String Hex value separated by a colon
EAP_Method_Count	Checks for the EAP method count	Integer 1

Name	Description	Value
EAP_SUCCESS	Checks for EAP Success message	Integer 0 = Not present 1 = present
EAP_Type_SIM	Checks for EAP-SIM method type	Integer 18 = EAP-SIM
EAP_Type_TTLS	Checks for EAP-TTLS method type	Integer 21 = EAP-TTLS
EC_IE	Confirms the Extended Capabilities element	Integer 0 – Should not exist 1 – Should exist
EC_IE_Bit12	Confirms if bit 12 of extended capabilities is set	Integer 0 = Disabled 1 = Enabled
EC_IE_Bit31	Confirms if bit 31 of extended capabilities is set	Integer 0 = Disabled 1 = Enabled
EC_IE_Bit32	Confirms if bit 32 of extended capabilities is set	Integer 0 = Disabled 1 = Enabled
EC_IE_Bit46	Confirms if bit 46 of extended capabilities is set	Integer 0 = Disabled 1 = Enabled
EOSPBit	1-Bit EOSP	Integer
Excap_OpMode_Notif	Confirms the Operating Mode Notification bit in the Extended Capabilities element	Integer 0 = Lack of support to receive Operating Mode Notification element and frame 1 = Support to receive Operating Mode Notification element and frame
ExtCap_b19	Confirms the r Extended Capabilities IE – bit 19	Integer 0 = Not present 1 = Should be present -1 = Ignore
ExtCap_IE	Confirms the Extended Capabilities IE in the frame	Integer 0 = Not present 1 = Should be present -1 = Ignore
Fama_Verify	Confirms the further availability map attribute and availability interval bitmaps (process up to 6 further availability map/channel values)	String
Filename	Wi-Fi Test Suite sniffer trace filename	String
FN_Lang_Code	Checks for FN language code	String eng kor
FrameLength	Confirms if the length of captured frame is less than the specified value in bytes	Short Integer

Name	Description	Value
FrameName	802.11 frame type to be checked	String ProbeReq = probe request frame ProbeResp = Probe response frame AssoReq = Association request frame AssoResp = Association response frame Data = Data frame QosData = QOS data frame Action = Action frame Beacon = Beacon frame Ack = Acknowledgment frame Eapolkey = EAPOL key frames Eapolkey2 = EAPOL key 2 frames Assocreassocreq = Association or reassociation request frames BcnRepoReq = Beacon report request frame BcnRepoResp = Beacon report response frame NbrRepoReq = Neighbor report request NbrRepoResp = Neighbor report response BssTransMgmtReq = BSS transition management request frame BssTransMgmtQuery = BSS transition management query frame BssTransMgmtResp = BSS transition management response frame TSMRepoReq = TSM report request frame TSMRepoResp = TSM report response frame Eaprespiden = EAP response identity Reassocreq = Reassociation request frame CTS = CTS frame RTS = RTS frame VhtNDPA = VHT NDPA frame VhtAction = VHT action frame RTSP = Real Time Streaming Protocol GasinitResp GasComebackResp eapol icmp
FrameTxRate	Confirms the Transmitted DataRate of the frame	String Not11bRate
Friendly_Name	Checks for the friendly name	String
Further_NAN_ServiceDiscoveryAttribute	Confirms the Further NAN Service Discovery attribute	Short integer
FurtherAvailMapAttr	Confirms the Further Availability Map attribute	Short integer

Name	Description	Value
GAS_Comeback_Delay	Confirms the GAS Comeback Delay value (in milliseconds)	Integer
GAS_Comeback_Delay_NE	Confirms if the GAS comeback delay is not equal to the given value	Integer
GAS_ComebackRespStatus	Confirms if the Status code of GAS comeback response frame matches the given value	Integer value
GAS_InitialRespStatus	Confirms the GAS initial Response status code files	Integer 0 = Disabled 1 = Enabled
GAS_MoreFrag	Confirms the More GAS frames bit in GAS action frames	Integer 0 = Disabled 1 = Enabled
GO_Intent	Confirms the Group Owner Intent attribute is present	Integer 0 = Disabled 1 = Enabled
GO_IntentValue	Confirms the GO Intent value that is present in the frame is same as the supplied value	Integer
Greenfield	Confirms the HT GreenField bit in the HT Capabilities IE	0= Not supported 1= Supported -1 = Ignore
HopCount	Confirms the Hop count to Anchor Master of the NAN device	Hex value: 0x00 – 0xFF
HS_CapList_Cap	Confirms the Hotspot capability list	String HSCapLst OperFriendName WANMet ConnCap
HS_CapList_Elem	Confirms if the Hotspot Capability list element is present	Integer 1 = Should be present 0 = should not be present
HS_IE	Confirms the Hotspot 2.0 Indication element	Integer 0 = Should not exist 1 = Should exist
HS_Query_Elem_Subtype	Confirms the Hotspot query element subtype.	String HSCapLst OperFriendName WANMet OperClassind
HS_Query_HomeRealm_Name	Confirms if the Home realm name in Hotspot query matches the given value.	String
HS_Query_Subtype	Confirms the Hotspot query subtype.	String HomeRealm
HS_Vendor_OUI	Confirms if the Hotspot vendor OUI matches the given value	Hex with 0x prefix
HS_Vendor_OUI_type	Confirms if the Hotspot vendor OUI type matches the given value	Integer
HS_Vendor_OUI_Type	HS vendor OUT type	Integer 16
HS2_Release	Checks for the HS2 release number in beacon frames	Integer 1 = Release 2.0 0 = Release 1.0

Name	Description	Value
HT_Info_IE_Ch_Width	Confirms the channel width that is set in HT Information element	Integer 20 40
HT_Opt_Prim_Ch	Confirms the primary channel parameter from the HT Operation element	Integer
HT_Opt_Protection	Confirms the HT Protection Mode in HT Operation element	Integer 0 - 3
IBSS_Attr	Confirms the IBSS attribute	Short integer
ICON_FName	Checks for Icon filename	String
ICON_Height	Check for the Icon height	String
ICON_TYPE	Icon type	String image/png
ICON_Width	Width of icon	Integer
IE_45	Confirms the HT capabilities element in the frame	Integer 0 = Not present 1 = Should be present -1 = Ignore
IE_61	Confirms the HT Information element in the frame	Integer 0 = Not present 1 = Should be present -1 = Ignore
Intended_P2P_Interface_Add	Confirms the Intended P2P Interface Address attribute is present	Integer 0 = Disabled 1 = Enabled
IntolerantChnlRepChannel	Confirms the channel is present in the 2040BSS Intolerant Channel Report in Action frame 20/40 Coex management frame	Integer -1 = Ignore
IPAdd_Type_IPv4	Confirms the IPv4 value in IP address type	String SingNATed
IPAdd_Type_IPv6	Confirms the IPv6 value in IP address type	String NotAvail
IPv6_Dest	IPv6 destination address	String Multicast
IPv6_Src	IPv6 source address	String D00:0000:0000:0000:02AA:00FF:FE3F:2A1C
IPv6_TargetAddr_NA	IPv6 target address	String 192.165.x.x
IPv6_TargetAddr_NS	IPv6 target address	String 192.165.x.x
Key_Desc_Version	Checks for key description version in EAPOL key	Integer 0
Lang_Code	Language code	String eng zxx
Legacy_OSU_SSID	Legacy OSU SSID	String SSID_2
LL_TargetAddr	Link layer target address	String aa:bb:cc:dd:ee:ff

Name	Description	Value
MasterPref	Confirms the Master preference field of the NAN device	Hex value: 0x00 – 0xFF
MatchFilter	Confirms the status of the match filter	Short Integer 0 = Set 1 = Reset
MatchFilterLen	Confirms the length of the match filter string	Short integer
MatchFilterVal	Confirms the value of the match filter	String
MCS_Rate	Confirms the MCS rate of the PPDU which is transmitted	Integer
MCS32	Confirms the MCS 32 capability in the HT Capabilities IE	0 = Not supported 1 = Supported -1 = Ignore
MCSSet	Confirms the supported MCS Set array in the HT Capabilities IE	1 = 1 SS 2 = 2 SS 3 = 3 SS -1 = Ignore
MeshAttr	Confirms the Mesh attribute	Short integer
MoreData_Bit	Confirms the More Data bit is set or not in FC	Integer 0 = Disabled 1 = Enabled
NAI_Credential_Info	NAI credential information	String
NAI_Realm_EAP_Method	NAI Realm EAP methods	String EAP-TLS EAP-TTLS
NAI_Realm_List	Confirms if the NAI Realm list is present	Integer 0 = Should not exist 1 = Should exist
NAI_Realm_Value	Confirms if the NAI Realm value matches the given value	String Hex value separated by a colon
Name	Name used to identify the Wi-Fi Test Suite sniffer	String
Net_AuthType_Ind	Confirms the Network Authentication Type indication.	String AcceptTermCond
Net_AuthType_URL	Confirms if the Network Authentication type URL matches the given value	String Hex value separated by a colon
Notes: 1. For multiple streams of traffic.		
NSS_Used	Confirms the number of spatial streams used	Integer 0.5 3
OCSP_StatusReq	Check for OCSP certificate status	Integer 0 = do not check for OCSP status 1 = check for OCSP status
OCSPCertStatus_good	Checks for the OCSP certificate status Good	Integer 1 0
OCSPCertStatus_Revoked	Check for OCSP certificate status revoked	Integer 0 = Not revoked 1 = revoked
Oper_Friend_Lang	Confirms if the Operator friendly language matches the given value	String

Name	Description	Value
Oper_Friend_Name	Confirms if the Operator Friendly name element is present	Integer 1 = Should be present 0 = Should not be present
Oper_Friend_Value	Confirms if the Operator friendly value matches the given value	String Hex value separated by a colon
OSEN_AKM_Suite_Count	Checks for the OSEN AKM suite count	Integer 1
OSEN_AKM_Suite_List	OSEN AKM suite list	Hex 0x506F9A01
OSEN_Group_Data_Cipher_Suite	OSEN group data cipher suite	Hex 0x000FAC
OSEN_IE	Checks if OSEN_IS is present in beacon frames	Integer 1 = Present 0 = Not present
OSEN_Pairwise_Cipher_Suite_Count	OSEN pairwise cipher suite count	Integer 1 = Present 0 = Not present
OSEN_Pairwise_Cipher_Suite_List	OSEN pairwise cipher suite list	Hex 0x000FAC04
OSU_MList		Integer 01
OSU_NAI	OSU NAI	String test-anonymous@wi-fi.org
OSU_NAI_Len	OSU NAI length	Integer 0
OSU_Prov_Len	OSU provider list length	Integer 0 1
OSU_Server_URI	OSU server URI	String
P2P_Cap_IpAddrAllocBit		
P2P_CapAttr	Confirms the P2P Capability attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_Chan_List	Confirms the Channel List attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_Config_GO_TimeOut	Confirms the GO time out value in the Config Timeout attribute	Integer
P2P_Config_TimeOut	Confirms the Configuration Time Out attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_CrossConBit	Confirms the Cross Connection Bit in Grp Capability Bitmap	Integer 0 = Disabled 1 = Enabled
P2P_Dev_ID	Confirms the P2P Device ID with the given string	String
P2P_Dev_Info	Confirms the P2P Device Info attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_DialogToken	Confirms the P2P public Management Action frames dialog token	Integer

Name	Description	Value
P2P_ExtnListenTimingAttr	Confirms the Extended Listen Timing attribute is present in the P2P IE	Integer 0 = Disabled 1 = Enabled
P2P_Group_Limit	Confirms the group limit bit	Integer 0 = Disabled 1 = Enabled
P2P_GroupOwnerBit	Confirms the Group Owner Bit in Grp Capability Bitmap	Integer 0 = Disabled 1 = Enabled
P2P_GrpBssid	Confirms the Group BSSID attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_GrpCapBitMap	Confirms the Group Capability Bitmap in the P2P IE	Integer 0 = Should not exist 1 = Should exist
P2P_GrpFormBit	Confirms the Group Formation Bit in the Grp Capability Bitmap	Integer 0 = Disabled 1 = Enabled
P2P_GrpIdAttr	Confirms the Group BSSID attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_GrpInfoClientDiscBit	Confirms the Client discovery bit in Group Info attribute	Integer 0 = Disabled 1 = Enabled
P2P_IE	Confirms the P2P IE is present	Integer 0 = Should not exist 1 = Should exist
P2P_IE_ListenChnl	Confirms the existence of this field in the IE	Integer 0 = Should not exist 1 = Should exist
P2P_IE_New		
P2P_InfraManagedBit	Confirms the infrastructure managed bit in Device Capability bitmap inside P2P capability of P2P IE	Integer 0 = Disabled 1 = Enabled
P2P_Intr_Attr	Confirms the Interface attribute	Integer 0 = Disabled 1 = Enabled
P2P_Intr_List	Confirms the Interface list for the given ID	String
P2P_IntraBssDistBit	Confirms the Intra-BSS Distribution bit in Group Capability Bitmap of P2P IE	Integer 0 = Disabled 1 = Enabled
P2P_InvFlagBit	Confirms the P2P Invitation Flag in Invitation Flag attribute	Integer 0 = Disabled 1 = Enabled
P2P_InvProcBit	Confirms the Invitation Procedure bit in Device Capability bitmap	Integer 0 = Disabled 1 = Enabled
P2P_InvTypeBit	Confirms the Invitation type bit in Invitation flags bitmap	Integer
P2P_IP_Addr_KDE_ReqIP		

Name	Description	Value
P2P_MgeAttr_CrconPer	Confirms if the Cross connection permitted bit in P2P Manageability attribute is set	Integer 0 = Disabled 1 = Enabled
P2P_MgeAttr_CrconPer	P2P cross connection attribute	Integer 0 = Not present 1 = present
P2P_MgeAttr_DevMgmt	Confirms if the Device management bit in P2P manageability attribute is set	Integer 0 = Disabled 1 = Enabled
P2P_MgeAttr_DevMgmt	P2P device management attribute	Integer 1 0
P2P_NoA_Attr	Confirms the NoA attribute is present	Integer 0 = Disabled 1 = Enabled 0/1
P2P_NoA_Descriptor	Confirms the presence of NoA descriptor in NoA attribute	Integer 0 = Disabled 1 = Enabled
P2P_OperatingChnlAttr	Confirms the Operating Channel Attribute is present	Integer 0 = Disabled 1 = Enabled
P2P_OperAttr	Confirms the P2P attribute	Short integer
P2P_OperChnl	Confirms the P2P operating channel number is same as passed	Integer
P2P_OpprPsBit	Confirms the Opportunistic power save bit in NoA attribute	Integer 0 = Disabled 1 = Enabled
P2P_PerGrpBit	Confirms the Group Formation Bit in Grp Capability Bitmap	Integer 0 = Disabled 1 = Enabled
P2P_ServiceDiscBit	Confirms the Service Discovery Bit in Device Capability attribute of P2P IE	Integer 0 = Disabled 1 = Enabled
P2P_SSID	Confirms the SSID contains the given string	String
P2P_SSID_Verify	Confirms the SSID is properly formatted as per the specification	String Postfix of SSID
PortNum ¹	Destination port number of the traffic	Integer
PPDU_BmformBit	Confirms the beamformed bit of the PPDU	Integer
PPDU_Ch_Bandwidth	Confirms the channel width of the transmitted PPDU	String CBW20 for 20MHz CBW40 for 40MHz CBW80 for 80MHz CBW160 for 160MHz CBW80+80 for 80+80 MHz
PPDU_Format	Confirms the format of the PPDU	String VHT HT-GT HT-MF Non-HT

Name	Description	Value
PPSMO_ID	Checks for the PPSMO identifier	Integer 1 2 3
PPSMO_ID_ne	Checks if PPSMO Id is not present in beacon frames	Integer 0 = Not present 1 = Present
PPSMO_ID_Present	Checks if PPSMO ID is present in beacon frames	Integer 1 = Present 0 = Not present
PVB_Bit	Confirms the PVT bit set	Integer 0 = Disabled 1 = Enabled
Pwr_Cons_IE	Confirms the Power Constraint Information Element	Integer 0 = Should not exist 1 = Should exist
QBSS_AAC	Confirms if the AAC field is present (value not verified)	Integer 1 = Should be present 0 = Should not be present
QBSS_CU	Confirms if the Channel utilization field is present (value not verified)	Integer 1 = Should be present 0 = Should not be present
QBSS_StaCount	Confirms if the QBSS station count value matches the given value	Integer
QoS_Map_DSCP_Exception	QoS map DSCP exception	Hex 0x3502 0x1606
QoS_Map_DSCP_Exception_DSCP_Val	QoS map DSCP value	Integer 53 22
QoS_Map_DSCP_Exception_UP	Check for the QoS map DSCP exception user priority	Integer 2 6
QoS_Map_DSCP_Range	Checks for the DSCP range in QoS map frame	Hex 0x080f 0x0007 0xffff 0x101f 0x2027 0x282f
QoS_Map_Tag_Num	Checks for the QoS map tag number	Integer 110
QoS_Priority	Checks for the QoS priority	Integer 3 = Best Effort 6 = Voice
Query_Req_Len	Query request length	Integer 25
Query_Resp_Len	Confirms if the length of Query Response in ANQP response frame matches the given value	Integer value
QueryResp_OUI_Subtyp	Confirms the OUI subtype in Service discovery Query response action frame	Integer

Name	Description	Value
QueryResp_OUI_Value	Confirms the OUI value in Service discovery Query response action frame	String Example: 50:6F:9a
RandFactor	Confirms the Random Factor field of the NAN device	Hex value: 0x00 – 0xFF
RangingAttr	Confirms the ranging attribute	Short integer
Reauth_Delay	Checks for reauthentication delay	Integer 60
ReqMode_Disassoc_Imminent_Bit2	Checks for disassoc imminent bit 2 in disassoc imminent action frame.	Integer 0 = Not present 1 = present
ReqMode_ESS_Disassoc_Imminent_Bit4	Checks for disassoc imminent bit 4 in disassoc imminent action frame.	Integer 0 = Not present 1 = present
RrmCap_B0	Confirms the Rrm capabilities IE- bit 0	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B1	Confirms the Rrm capabilities IE- bit 1	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B14	Confirms the Rrm capabilities IE- bit 14	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B15	Confirms the Rrm capabilities IE- bit 15	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B32	Confirms the Rrm capabilities IE- bit 32	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B4	Confirms the Rrm capabilities IE- bit 4	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B5	Confirms the Rrm capabilities IE- bit 5	Integer 0 = Not present 1 = Should be present -1 = Ignore
RrmCap_B6	Confirms the Rrm capabilities IE- bit 6	Integer 0 = Not present 1 = Should be present -1 = Ignore
RSN_AKM_Type_NE	Confirms if the AKM type field in RSN information is not equal to the given value	Integer
RSN_IE	Confirms the RSN IE is present	Integer 0 = Should not exist 1 = Should exist

Name	Description	Value
RSN_IE_AKM	Confirms if the AKM suit in RSN IE contains the given suite type	Integer 2 = PSK 6 = PSK with SHA256
RSN_IE_Pairwise	Confirms if the Pairwise Cipher Suite List in RSN IE contains given value	String WEP-40 WEP-104 TKIP
RSN_PCS_Type_NE	Confirms if the PCS type field in RSN information is not equal to the given value	Integer
RSNCap_MFPC	Management Frame Protection Capable	Integer 1 = Enable 0 = Disable
RTS_Service_BW	Confirms the bandwidth bits of the PPDU carrying RTS	String 80Mhz 40Mhz
RTS_Service_BW_Type	Confirms the bandwidth type of the RTS frame	String Static Dynamic
RxMAC	Receiver MAC address in aa:bb:cc:dd:ee:ff format	String
SD_Lang_Code	Checks for service description language code	String eng kor
SdaID	Confirms the Service ID in the NAN SDF	String
Server_URL	Checks for the server URL	String https
ServerCert_EKU	Server Certificate Extended Key Usage	String ClientAuth ServerAuth
ServerCert_SubjectAltName	Checks for the server certificate subject name	String wi-fi-spoof.org
ServerCertStatus	Checks the status of the server certificate	Integer 0 1
Service_Desc	Service Description	String
ServiceCtrlType	Confirms the type of NAN service discovery frame	Short Integer 0 = Publish 1 = Subscribe 2 = FollowUp
ServiceDescAttr	Confirms the service descriptor attribute	String
ServiceId	Confirms the Service identification for the supported service	String
Session_Info_URL	Checks for session information URL	String http://www.r2-testbed.wi-fi.org/DisassocImminent.html
SGI20	Confirms the Short GI for 20 Mhz capability in the HT Capabilities IE	0 = SGI 20 not supported 1 = SGI 20 supported -1 = Ignore
SGI40	Confirms the Short GI for 40 Mhz capability in the HT Capabilities IE	0 = SGI 40 not supported 1 = SGI 40 supported -1 = Ignore

Name	Description	Value
SpectrumMgmt	Confirms the spectrum management capability	Integer 0 = Spectrum 94gmt not supported 1 = Spectrum 94gmt supported
SrcMAC	Source MAC address in aa:bb:cc:dd:ee:ff format Not mandatory for data /qos data frame packet check.	String
SSID	Confirms the SSID matches the given string	String
STA_count	Confirms the Associated STA count	Integer 0
Stachwidth	Confirms the STA channel width	Integer
STBC_Rx	Confirms the STBC Rx field in the HT Capability IE is supported	0 = Not supported 1 = supported -1 = Ignore
SupportedRates	Confirms the Supported Rates IE data	String 11gRates No11bRates
Tag_Len	Tag length	Integer 16 = action frame 20 = assoresp
TAG_No	Checks for tag number	Integer 221
TIM_Element	Confirms the TIM element is zero or non-zero.	Integer 0 = Disabled 1 = Enabled
TKIP_AD	Confirms the IE221 in frames in framename	Integer 0 = Not present 1 = Should be present -1 = Ignore
TSF_Init	Confirms if the TSF of the first beacon transmitted is less than the specified value as per test plan	Hex value
Venue_Group	Confirms if the Venue group matches the given value	Integer
Venue_Lang_Code	Confirms if the Venue language code matches the given value	string
Venue_Name	Confirms if the Venue Name matches the given value	String Hex value separated by a colon
Venue_Type	Confirms if the Venue Type matches the given value	Integer
Verify_Serviceld	Confirms if the number of service IDs included in beacons is less than the specified value	Short Integer
VHT_Cap_IE	Confirms the VHT Capability IE	Integer 0 = Should not exist 1 = Should exist
VHT_Cap_NumSndDim	Confirms the Number of sounding dimensions under VHT Capability IE	Integer

Name	Description	Value
VHT_Cap_RxMcsSet	Confirms the Rx MCS Set capability in the Rx MCS Map field	String 1ss = 1SS MCS supported 2ss = 2SS MCS supported 3ss = 3SS MCS supported 1ss_mcs8 = MCS8 rate 1ss_mcs9 = MCS 9 rate 2ss_mcs8 = MCS8 rate 2ss_mcs9 = MCS 9 rate 3ss_mcs8 = MCS8 rate 3ss_mcs9 = MCS 9 rate 1ss_NA = 1SS Not supported 2ss_NA = 2SS Not supported 3ss_NA = 3SS Not supported
VHT_Cap_RxSTBC	Confirms the VHT Capability RX STBC	Integer 0 = Should not exist 1 = Should exist
VHT_Cap_Sgi80	Confirms the SGI 80 bit in VHT capability IE	Integer 0 = Should not exist 1 = Should exist
VHT_Cap_SuppCW	Confirms the Supported Channel Width in the VHT Capability IE	Integer
VHT_Cap_TxMcsSet	Confirms the Tx MCS Set capability in the Tx MCS Map field	String 1ss = 1SS MCS supported 2ss = 2SS MCS supported 3ss = 3SS MCS supported 1ss_mcs8 = MCS8 rate 1ss_mcs9 = MCS 9 rate 2ss_mcs8 = MCS8 rate 2ss_mcs9 = MCS 9 rate 3ss_mcs8 = MCS8 rate 3ss_mcs9 = MCS 9 rate 1ss_NA = 1SS Not supported 2ss_NA = 2SS Not supported 3ss_NA = 3SS Not supported
VHT_Cap_TxSTBC	Confirms the VHT Capability TX STBC	Integer 0 = Should not exist 1 = Should exist
VHT_Capb_IE_Rx_LDPC	Confirms the Rx LDPC bit in the VHT Capabilities Info field that indicates support for receiving LDPC encoded packets	Integer 0 = Not supported 1 = Supported
VHT_Capb_IE_Su_Bm_Former	Confirms the field indicates support for SU Beamformer	Integer 0 = If not supported 1 = If supported
VHT_Opt_IE	Confirms the VHT Operation IE	Integer 0 = Should not exist 1 = Should exist
VHT_Opt_IE_Ch_Width	Confirms the Channel Width that is set within VHT Operation Information field of VHT Operation IE The field together with the HT Operating element STA channel width field, defines the BSS operating channel width.	Integer 0 = 20/40Mhz operating channel width 1 = 80 Mhz operating channel width 2 = 160 Mhz operating channel width 3 = 80+80Mhz operating channel width

Name	Description	Value
VHT_Tx_Pwr_Env_IE	Confirms the VHT Transmit Power Envelope element	Integer 0 = Should not exist 1 = Should exist
VhtCmpBmForm_CW	Confirms the channel width in the VHT Compressed Beamforming VHT action frame	String 80Mhz 40Mhz 20Mhz
VHTCmpBmForm_FbType	Confirms the Feedback Type in the VHT Compressed Beamforming VHT action frame	Integer
VHTCmpBmForm_FirFeedSeg	Confirms the First Feedback Segment in the VHT compressed beam forming frame	Integer
VHTCmpBmForm_RemFeedSeg	Confirms the Remaining feedback segment in the VHT compressed beamforming frame	Integer
VHT-SIG-A1_BW	Confirms the BW bits (B0 and B1) in VHT-SIG-A1 field in the PPDU header.	String 20Mhz 40Mhz 80Mhz 160Mhz & 80+80Mhz
VHT-SIG-A1_GID	Confirms for bits B4-B9 in the VHT-SIG-A1 field. The bits indicate the Group ID	Integer 0 or 63 for SU PPDU 1<=n<=62 for MU PPDU
VHT-SIG-A1_STBC	Confirms the B3 bit in the VHT-SIG-A1 field	Integer 0 = No spatial stream of any user uses STBC 1 = All spatial streams of all users have STBC
VHT-SIG-A2_SU_Coding	Confirms the B2 bit in the VHT-SIG-A2 field for a SU PPDU	Integer 0 = BCC 1 = LDPC
WAN_At_Capacity	Confirms if the At Capacity field in WAN metrics matches the given value	Integer
WAN_Dnlink_Load	Confirms if the Downlink Load field matches the given value	Integer
WAN_Dnlink_Speed	Confirms if the Downlink speed field in WAN metrics matches the given value	Integer or Hex with 0x prefix
WAN_Link_Status	Confirms if the WAN link status matches the given value	Integer
WAN_Met_Elem	Confirms if the WAN metrics element is present	Integer 1 = Should be present 0 = Should not be present
WAN_Uplink_Load	Confirms if the Uplink Load field matches the given value	Integer
WAN_Uplink_Speed	Confirms if the Uplink speed field in WAN metrics matches the given value	Integer or Hex with 0x prefix
WLAN_InfraStructAttr	Confirms the WLAN Infrastructure attribute	Short integer
WLAN_SSID	Wireless LAN SSID	String SSID_2
WMM_Param_Element	Confirms the presence of WMM parameter information element	Integer 0 = Disabled 1 = Enabled

Name	Description	Value
WMMIE	Confirms if the IE exists or is not in the frame	Integer 0 = Should not exist 1 = Exists
WMMmmle	Confirms the WMM IE is present	Integer 1 = Should exist 0 = Should not exist
WMMPe_BE_Def_Val	Confirms if the AC parameter values for BE in WMM parameter element matches with the default value in the test plan	Integer 1 = Should match 0 = Should not match
WMMPe_BK_Def_Val	Confirms if the AC parameter values for BE in WMM parameter element matches the default value in the test plan	Integer 1 = Should match 0 = Should not match
WMMPe_VI_Def_Val	Confirms if the AC parameter values for BE in WMM parameter element matches the default value in the test plan	Integer 1 = Should match 0 = Should not match
WMMPe_VO_Def_Val	Confirms if the AC parameter values for BE in WMM parameter element matches the default value in the test plan	Integer 1 = Should match 0 = Should not match
WNMNotification_Type	Checks for WNM notification type	Integer 1 0
WPAIE	Confirms the WPA Information Element in EAPOL KEY packets	Integer 0 = Not present 1 = Should be present
WPS_ConfMethod	Confirms the Config method in the Config Method Attribute of WPS IE	String WPS_Display WPS_Keypad WPS_Label WPS_PushButton
WPS_DevPass_IDAttr		
WPS_DevPassIDAttr	Confirms the Device Password ID Attribute is present	Integer 0 = Disabled 1 = Enabled
WPS_DevPwld	Confirms the Device Password ID matches the given mode	String Registrar User PIN PushButton
WPS_IE	Confirms the WPS IE is present	Integer 0 = Should not exist 1 = Should exist
WPS_IE_PrimaryDevType	Confirms the Existence of Primary device type in WPS IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_DevicePasswordID	Confirms the Existence of Device Password ID	Integer 0 = Should not exist 1 = Should exist
WPS_IE_DevName	Confirms the Existence of Device name in WPS IE	Integer 0 = Should not exist 1 = Should exist

Name	Description	Value
WPS_IE_New		

Return Values

Name	Description	Value
CheckResult	Result of the requested Wi-Fi Test Suite sniffer check	Success Fail

Examples

```
UCC:sniffer_control_field_check,filename,sniff_capture,srcmac,00:21:06:e8:ae:72,framefilename,probereq,wmmie,1
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE,CheckResult,SUCCESS
```

9.6 SNIFFER_CHECK_PMK_ID

This command verifies the PMK ID from the association / re-association frame and compares the PMK ID that occurs in the first EAPOL frame. Returns SUCCESS if the PMK ID matches and FAIL if the comparison fails or the frames do not exist.

Parameters

Name	Description	Value
BSSID	BSSID in aa:bb:cc:dd:ee:ff format	String
Filename	Wi-Fi Test Suite sniffer trace file	String
Name	Name used to identify the Wi-Fi Test Suite sniffer	String
STAMAC	MAC address of STA in aa:bb:cc:dd:ee:ff format	String

Return Values

Name	Description	Value
CheckResult	Result of the requested Wi-Fi Test Suite sniffer check	Success Fail

Examples

```
UCC:sniffer_check_pmk_id,name,wireShark_wlan,FileName,capture_file,StaMac,aa:bb:cc:dd:ee:ff,bssid,  
aa:bb:cc:dd:ee:ff
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, CheckResult,SUCCESS
```

9.7 SNIFFER_CHECK_RETRY

This command verifies the PMK ID from the association / re-association frame and compares the PMK ID that is occurs in the first EAPOL frame. Returns SUCCESS if the PMK ID matches and FAIL if the comparison fails or the frames do not exist.

Parameters

Name	Description	Value
DataLen	Data length of data frames	String / Integer NULL / <value>
Filename	Input capture filename	String
FrameName	Frame name	String

Return Values

Name	Description	Value
ID	Retry count	Integer

Examples

```
UCC:wfa_sniffer_check_retry,filename,P2P_5111,datalen,framename,anydata
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE
```

9.8 SNIFFER_FRAME_CHECK

This command is used to verify if the specified frames exist in the specified capture file.

Parameters

Name	Description	Value
AfterTimeStamp	Timestamp in seconds	Integer
BSSID	BSSID	String aa:bb:cc:dd:ee:ff
ClientMAC	Client MAC address	String 00:11:22:33:44:55
DataLen	Date length	Integer 1516
DestMAC	Destination MAC address	String aa:bb:cc:dd:ee:ff
DstMC ¹	MAC address of destination in aa:bb:cc:dd:ee:ff format	String
Filename	Wi-Fi Test Suite sniffer trace file	String
FrameName	Frame name. The Protection frames are RTS/CTS or CTS to self. BA checks for Block Acks.	String EAPOL_START EAPOL_REQUEST Protection_Frames BA Beacon
Name	Name used to identify the Wi-Fi Test Suite sniffer	String
Present	Confirms the frame exists	String Yes No
RxMAC ¹	MAC address of receiver in aa:bb:cc:dd:ee:ff format	String
SrcMAC ¹	MAC address of source in aa:bb:cc:dd:ee:ff format	String
TxMAC ¹	MAC address of transmitter in aa:bb:cc:dd:ee:ff format	String
Type	Type of data	String AC_Video
Notes:		
1. Use any of these fields as required to identify the frame.		

Return Values

Name	Description	Value
CheckResult	Success / Failure	String

Examples

```
UCC:sniffer_frame_check,name,wireShark_wlan,FileName,capture_file,SrcMac,aa:bb:cc:dd:ee:ff,DstMac,aa:bb:cc:dd:ee:ff,FrameName,EAPOL_START, Present,No
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, CheckResult,SUCCESS
```

9.9 SNIFFER_CHECK_CHNI_SWITCH

This command verifies the channel switch announcement from the AP and verifies that the station did not send any wireless packets on this channel after the channel switch announcement beacon frames was sent.

Parameters

Name	Description	Value
BcnFrames	Number of beacon frames, after which STA should not send any wireless packets	Integer
BSSID	BSSID in aa:bb:cc:dd:ee:ff format	String
DeviceType	Supported device type	String STAUT (Default) APUT
Filename	Wi-Fi Test Suite sniffer trace file	String
Name	Name to identify the Wi-Fi Test Suite sniffer	String
ProgName	Program name	String VHT 11n (Default)
STAMAC	MAC address of STA in aa:bb:cc:dd:ee:ff format	String
VHTTxPwrEnv	Conform the VHT Transmit Power Envelope is present in the channel switch announcement	Integer 1 0

Return Values

Name	Description	Value
CheckResult	Result of the requested Wi-Fi Test Suite check	Success Fail

Examples

```
UCC:sniffer_check_chnl_switch,name,wireShark_wlan,FileName,capture_file,StaMac,aa:bb:cc:dd:ee:ff,bsid, aa:bb:cc:dd:ee:ff,BcnFrames,6
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, CheckResult,SUCCESS
```

9.10 SNIFFER_GET_FIELD_VALUE

This command verifies the channel switch announcement from the AP and verifies that the station does not send any wireless packets on this channel after the specified channel switch announcement beacon frames was sent.

Parameters

Name	Description	Value
ActionCode	Checks for Action frame code of the referenced Action frame	For 20-40BssCoEx GONegReq = Group Owner Negotiation Request GONegResp = Group Owner Negotiation Response GONegCon = Group Owner Negotiation Response ProvDisReq = Provision Discovery Request P2PInvReq = P2P Invitation Request P2PInvResp = P2P Invitation Response P2PDevDiscReq = Device Discovery req P2PDevDiscResp = Device Discovery response GASInitReq = GAS initial request GASInitResp = GAS initial response GASComebackReq = GAS come back request GASComebackResp = GAS comeback response P2PPresenceReq = P2P presence Request P2PPresenceResp = P2P presence Response P2P_GoDiscReq = P2P Go Discovery Req BSSTransMgmtReq -1 = Ignore
BSSID	BSSID address in aa:bb:cc:dd:ee:ff format	String
DataLen	Frame data length	Integer
DestMAC	Destination MAC address in aa:bb:cc:dd:ee:ff format	String

FieldName	Field that is to be read	String
		<p>CrossConBit = returns the Cross Connection Bit value</p> <p>P2P_ActionStatus, P2P_Status = returns the P2P Action Status value</p> <p>SnifferTimeStamp = returns the relative of the frame from the beginning of the trace in seconds</p> <p>WPS_ConfMethod = Returns the WPS Config methods as</p> <ul style="list-style-type: none"> WPS_Display WPS_Label WPS_Keypad WPS_PushButton <p>DialogToken = 802.11 Action frames dialog token</p> <ul style="list-style-type: none"> P2P_DialogToken P2PAction_DialogToken CTWindow DsParam_ListenChnl GAS_ComebackDelay ExtLstnTime_Period ExtLstnTime_Interval NoA_Duration NoA_StartTime P2P_SSID P2P_OperChnl Key_Mgmt RSN_Capab Valid_Bip_Mmie Invalid_Bip_Mmie Missing_Bip_Mmie Bip_Deauth Bip_Disassoc Auth_Tx Auth_Rx Assocreq_Tx Reassocreq_Tx Ptk_Learned Valid_Deauth_Tx Valid_Deauth_Rx Invalid_Deauth_Tx Invalid_Deauth_Rx Valid_Disassoc_Tx Valid_Disassoc_Rx Invalid_Disassoc_Tx Invalid_Disassoc_Rx Valid_Saqueryreq_Tx Valid_Saqueryreq_Rx Invalid_Saqueryreq_Tx Invalid_Saqueryreq_Rx Valid_Saqueryresp_Tx Valid_Saqueryresp_Rx Invalid_Saqueryresp_Tx Invalid_Saqueryresp_Rx Ping_Ok Assocresp_Comeback Reassocresp_Comeback Deauth_rx_rc6

Name	Description	Value
		Deauth_rx_rc7 Disassoc_rx_rc6 Disassoc_rx_rc7 Deauth_rx_asleep Disassoc_rx_asleep Disassoc_rx_awake Deauth_rx_awake Valid_deauth_rx_ack Valid_disassoc_rx_ack Invalid_deauth_rx_ack Invalid_disassoc_rx_ack BssLoad_AAC BssLoad_CU FrameNumber CountryIE_CN VHT_Cap_Rx_Lgi_Rate = Rx Highest Supported Long GI data rate VhtCmpBmForm_Nr VhtCmpBmForm_Nc VhtCap_SterBmFormAnt MasterPref = Master preference of NAN device RandFactor = Random factor of NAN device HopCount = Hop count of NAN device ClusterId = Cluster identification of NAN cluster TSF = Timing Synchronization function value for the frame Sdald = Service ID in NAN SDF RequestorInstId = Requestor Instance ID of the remote device InstId = Device's own Instance ID AM_Rank = Anchor Master Rank of the device BaseCSeq P2PIE_ConfMethod P2P_NoA_Descriptor Framecount WNM_DisassocTimer BSSLoad_CU SeqNo
Filename	Wi-Fi Test Suite sniffer trace file	String

Name	Description	Value
FrameName	802.11 frame type to be checked	String ProbeReq = probe request frame AssoReq = Association Request frame Data = Data frame QosData = QoS Data frame Action = Action frame Beacon = Beacon frame ProbeResp = Probe Response frame Ack = Acknowledge Assoresp = Association response EapolKey = EAPOL Key frame VhtAction = VHT Action frame HDCCP_AKE_TxInfo HDCCP_AKE_Init eapfailure eapclienthello assocreassocreq
Message		String RTSP_M3_RESP RTSP_M3_REQ
Name	Name to identify the Wi-Fi Test Suite sniffer	String
Program	Program Name	String General PMF TDLS VOE WFD NAN 60GHz
QOS_Type	Access category	String Voice Video
SrcMAC	MAC address of Source in aa:bb:cc:dd:ee:ff format	String
StationID	The STA MAC address which is in the BSS of the AP.	String Ex: 00:11:22:33:44:55

Return Values

Name	Description	Value
CheckResult	Result of the requested Wi-Fi Test Suite sniffer check	Success Fail
ReturnValue	Return value for the requested FieldName	String

Examples

```
UCC:sniffer_get_field_value,name,wireShark_wlan,FileName,capture_file,StaMac,aa:bb:cc:dd:ee:ff,bssid, aa:bb:cc:dd:ee:ff,frameName,Acton,ActionCode,GoNegResp,FieldName,P2P_ActionStatus
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, CheckResult,SUCCESS,ReturnValue,0
```

9.11 SNIFFER_CHECK_TIME_DIFFERENCE

This command is used to verify the time difference between two frames which were present in the input files.

Parameters

Name	Description	Value
FirstFrameFile	Input capture filename	String
GTE	Checks that the time difference in micro-seconds is greater than or equal to the time difference between the first frame and then second frame	Integer
LTE	Checks that the time difference in micro-seconds is less than or equal to the time difference between the first frame and the second frame	Integer
SecondFrameFile	Output capture filename with the specified frames	String

Files should have only one frame.

Return Values

Name	Description	Value
TimeDiff	Time difference in micro-seconds between the first and second frames	Integer

Examples

```
UCC:sniffer_check_time_difference,FirstFrameFile,file1,SecondFrameFile,file2,LTE,100
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.12 SNIFFER_CHECK_FRAME_FIELD

This command is used to verify that the field values are the same in the supplied frames. The files shall only have one frame.

Parameters

Name	Description	Value
FileName1	Input capture filename	String
FileName2	Output capture filename that includes the specified frames	String
FrameField	Field name of the WMM_PE	String

Return Values

None.

Examples

```
UCC:sniffer_check_frame_fields,FileName1,file1,FileName2,file2,FrameField,WMM_PE
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

```
sniffer_check_pvb_pspoll_data
```

9.13 SNIFFER_CHECK_PVB_PSPOLL_DATA

This command verifies the requirements for test case 6.1.11 (pspoll section) of the Wi-Fi Direct Test plan.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
DataLen	Data packet that is send from GO	Integer
Filename	Input capture filename	String
STAMAC	P2P client MAC address	String

Return Values

None.

Examples

```
UCC:sniffer_check_pvb_pspoll_data,FileName,file1,stamac,aa:bb:cc:dd:ee:ff,bssid,aa:bb:cc:dd:ee:ff,datalen,1444
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.14 SNIFFER_CHECK_PVB_PSNONPOLL_DATA

This command verifies the requirements for test case 6.1.11 (psnonpoll section) of the Wi-Fi Direct Test plan.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
DataLen	Data packet that is send from GO	Integer
Filename	Input capture filename	String
STAMAC	P2P client MAC address	String

Return Values

None.

Examples

```
UCC:sniffer_check_pvb_psnonpoll_data,FileName,file1,stamac,aa:bb:cc:dd:ee:ff,bssid,aa:bb:cc:dd:ee:ff,dataLen,1444
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.15 SNIFFER_CHECK_P2P_NOA_DESCRIPTOR

This command verifies the requirements for test case 6.1.9 step 5 of the Wi-Fi Direct Test plan.

Parameters

Name	Description	Value
File1	Input capture filename containing only the presence response	String
File2	Input capture filename that contains the presen response and beacon	String
FrameName	Frame name	String
SrcMAC	P2P GO MAC address	String

Return Values

None.

Examples

```
UCC:sniffer_check_p2p_NoA_Descriptor,file1,P2P_619_2,file2,P2P_619,srcmac,aa:bb:cc:dd:ee:ff
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.16 SNIFFER_CHECK_P2P_OPPTS_CLIENT

This command verifies the requirements for test case 6.1.8 or 7.1.3 of the Wi-Fi Direct Test plan.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
CTwindow	Ctwindow in milli-seconds	Integer
DataLen	Data length of ping traffic	Integer
FileName	Input capture filename	String
SrcMAC	Source MAC of station from which data is originating	String
STAMAC	Client MAC address which is in powersave	String

Return Values

None.

Examples

```
UCC:sniffer_check_p2p_opps_client,filename,P2P_618_A,bssid,aa:bb:cc:dd:ee:ff,srcmac,aa:bb:cc:dd:ee:ff,stamac,aa:bb:cc:dd:ee:ff,datalen,52,ctwindow,1024
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.17 SNIFFER_CHECK_P2P_OPPTS_GO

This command verifies the requirements for test case 6.1.8 step 8 of the Wi-Fi Direct Test plan.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
CTwindow	Ctwindow in milli-seconds	Integer
DataLen	Data length of ping traffic	Integer
DstMAC	Destination MAC of station to which data is transferred	String
FileName	Input capture filename	String

Return Values

None.

Examples

```
UCC:sniffer_check_p2p_opps_go,filename,P2P_618_A,bssid,aa:bb:cc:dd:ee:ff,destmac,aa:bb:cc:dd:ee:ff,datalen,52,ctwindow,1024
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

```
sniffer_check_p2p_NoA_start_time
```

9.18 SNIFFER_CHECK_P2P_NOA_START_TIME

This command verifies the requirement that the NoA Start time is relative to the beacon as per its announcement.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
DataLen	Data length of ping traffic	Integer
DestMAC	destination MAC of station to which data is transferred	String
FrameName	Input capture filename	String
NoA_Duration	NoA duration in micro-seconds	Integer
SrcMAC	Source MAC of data transmission	String

Return Values

None.

Examples

```
UCC:sniffer_check_p2p_NoA_start_time,filename,P2P_617,srcmac,aa:bb:cc:dd:ee:ff,bssid,aa:bb:cc:dd:ee:ff,destmac,aa:bb:cc:dd:ee:ff,DataLen,52,NoA_duration,51200
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.19 SNIFFER_CHECK_P2P_NOA_WMMPS_RETRIGGER

This command verifies the requirement for checking EOSP, and 'More bit' values whenever the NoA interrupts the Group Owner under Test (GOUT) sniffer transmission. The command also verifies that GOUT transmits only after the retrigger frame reception.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
ClientMAC	Client MAC address that initiates the retrigger	String
DataLen	Data length of ping traffic	Integer
DestMAC	Destination MAC of station to which data is transferred	String
Filename	Input capture filename	String
NoA_Duration	NoA duration in micro-seconds	Integer
SrcMAC	Source MAC of data transmission	String
TrigDataLen	Trigger Data frame length	Integer

Return Values

Name	Description	Value
CheckResult	Result of the Wi-Fi Test Suite sniffer check	String Success Fail
RefDataFrameTime	Time reference of data packet from GO which passes all the verification according to the test plan	String
Data_Between_NoA_ReTrigger	Time reference of data packet from GO after NoA and before retrigger frame from client	String

Examples

```
UCC:sniffer_check_p2p_noa_wmmpps_retrigger,filename,P2P_6113,srcmac, aa:bb:cc:dd:ee:ff,clientmac,
aa:bb:cc:dd:ee:ff,bssid, aa:bb:cc:dd:ee:ff,datalen,1516,TrigDataLen,144,noa_duration,51200,destmac,
aa:bb:cc:dd:ee:ff
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE,CheckResult,SUCCESS,RefDataFrameTime,1374006463.064553000
```


9.20 SNIFFER_CHECK_P2P_CLIENT_PS_RETRIGGER

This command verifies the requirement for retrigger frames in section 7.1.5 of the Wi-Fi Direct test plan.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
ClientMAC	Client MAC address that initiates the retrigger	String
DataLen	Data length of ping traffic	Integer
FileName	Input capture filename	String
NullDataLen	Null Frame data length	Integer
TrigDataLen	Trigger Data frame length	Integer

Return Values

None.

Examples

```
UCC:sniffer_check_p2p_client_ps_retrigger,filename,P2P_715,bssid,aa:bb:cc:dd:ee:ff,clientmac,aa:bb:cc:dd:ee:ff,DataLen,1444,NullDataLen,NUL,TrigDataLen,1044
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.21 SNIFFER_CHECK_P2P_NOA_DURATION

This command verifies the requirement for NoA duration and that no data is transmitted as per test cases 7.1.2, 7.1.4, 6.1.7 and 6.1.13 of the Wi-Fi Direct test plan.

Parameters

Name	Description	Value
BSSID	P2P GO MAC address	String
DataLen	Data length of ping traffic	Integer
DestMAC	Destination MAC address for data	Integer
FrameName	Input capture filename	String
NoADuration	NoA Duration in micro seconds	Integer
SrcMAC	Source MAC address of station which sends the data	String

Return Values

None.

Examples

```
UCC:sniffer_check_p2p_NoA_duration,filename,P2P_712_1,bssid,aa:bb:cc:dd:ee:ff_STA1,srcmac,aa:bb:cc:dd:ee:ff,destmac,aa:cc:dd:ee:ff:gg,NoA_duration,51200,DataLen,52Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.22 SNIFFER_CONTROL_FIELD_CHECK_ALL

This command verifies that the specified field is the same in all frames.

Parameters

Name	Description	Value
ClusterId	Cluster Id of NAN cluster	String
DialogToken	Field name	Integer
Filename	Input capture filename	String
FrameName	802.11 frame type to be checked For example, Name of the 802.11 frame	String ProbeReq = probe request frame AssoReq = Association Request frame Data = Data frame QosData = QoS Data frame Action = Action frame Beacon = Beacon frame ProbeResp = Probe Response frame Ack = Acknowledge Assoresp = Association response EapolKey = EAPOL Key frame VhtAction = VHT Action frame
GAS_ComebackRespStatus	Field name	Integer
HopCount	Hop count to Anchor Master of NAN device	Hex value: 0x00 – 0xFF
MasterPref	Master preference field of NAN device	Hex value: 0x00 – 0xFF
P2P_DialogToken	Field name	Integer
P2PAction_DialogToken	Field name	Integer
RandFactor	Random Factor field of NAN device	Hex value: 0x00 – 0xFF
Relative_Time_Space	Relative Time difference	Integer

Return Values

None.

Examples

```
UCC:sniffer_control_field_check_all,filename,P2P_5120_4,GAS_comebackRespStatus,0
Sniffer:status,RUNNING
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.23 SNIFFER_CONTROL_FILTER_CAPTURE / WFA_SNIFFER_CONTROL_FILTER_CAPTURE

This command is used to filter the specified frame from the existing capture file. This command will reduce the processing time when running multiple tests on a single frame.

Parameters

Name	Description	Value
ActionCode	Checks for Action frame code of the referenced Action frame	For 20-40BssCoEx: GONegReq = Group Owner Negotiation Request GONegResp = Group Owner Negotiation Response GONegCon = Group Owner Negotiation Response ProvDisReq = Provision Discovery Request ProvDisResp = Provision Discovery Response P2PInvReq = P2P Invitation Request P2PInvResp = P2P Invitation Response P2PDevDiscReq = Device Discovery req P2PDevDiscResp = Device Discovery response GASInitReq = GAS initial request GASInitResp = GAS initial response GASComebackReq = GAS come back request GASComebackResp = GAS comeback response P2PPresenceReq = P2P presence Request P2PPresenceResp = P2P presence Response P2P_GoDiscReq = P2P Go Discovery Req ANQP_Req = ANQP Request ANQP_Resp = ANQP Response ARP_Req = ARP request ARP_Reply = ARP Reply NeighSolicit = Neighbor Solicitation message UnSolicitNeigh = Unsolicited Neighbor advertisement message PingReq = Ping Request packets PingReply = Ping Reply packets TDLSSetupReq = TDLS Setup Request frame TDLSDiscReq = TDLS Discovery Request frame DLSReq = DLS Request frame

Name	Description	Value
		DLSResp = DLS Response frame AddBaResp = ADDBA Response BSSTransMgmtReq QoSMapConf NeighAdvert PingReq -1 = Ignore
ANQP_InfolD	Confirms if the Info ID of ANQP element matches the given value	Integer 56797 260 263
ANQP_SubType	ANQP Subtype value	Integer 2 = HS Capability list 8 = OSU Provider List
Band	Filter packets based on the specified band	String 24G 5G
BCN_Int	Beacon Interval (in milli-seconds)	Integer
bidMAC	MAC Address	String AA:BB:CC:DD:EE:FF
BSSID	BSSID address in aa:bb:cc:dd:ee:ff format	String
DataLen	Data length of data frames	String / Integer NULL / <value>
DestMAC	MAC address of destination in aa:bb:cc:dd:ee:ff format	String
DestMACCon	Filter packets with destination MAC address containing the given string	String
DSValue	Distribution system value	Integer 1 2
EthDestIP	Ethernet destination IP address	String
EthDestMAC	Ethernet destination MAC address in aa:bb:cc:dd:ee:ff format	String
EthSrcMAC	Ethernet source MAC address in aa:bb:cc:dd:ee:ff format	String
EthSrcIP	Ethernet source IP address	String
Ethv6DesIPCon	Filter packets with Ethernet IPv6 destination IP address containing the given string	String
Ethv6DestIP	Ethernet IPv6 destination IP address	String
FrameName	802.11 frame type to be checked	String ProbeReq = Probe request frame AssocReq = Association Request frame Data = Data frame QoSData = QoS Data frame Qosnull = QoS Null AnyData = QoS / non QoS Data Action = Action frame Beacon = Beacon frame ProbeResp = Probe Response frame Ack = Acknowledge

Name	Description	Value
		Assorep = Association response Assocreassocreq = Association Re-Association Req Assocreassocresp = Association Re-Association Resp EapolKey = EAPOL Key frame BcnRepoReq = Beacon Report Request BcnRepoResp = Beacon Report response Udp = UDP protocol Icmp = ICMP protocol Arp = Arp request or Arp response packets Garp = Gratuitous ARP Eaprespind = EAP response identity VHT_NDPA = VHT NDPA frame CTS = CTS frame RTS = RTS frame HDCP2x = High-Bandwidth Digital Content Protection version 2.x RTSP = Real Time Streaming Protocol DHCP eapol 80211_arp 80211_garp deauth disassoc eapfailure eap_clienthello eaprespinden anqpresp
FrameNum_GT	Reference Frame number	Integer
HasField	Identifies the field specified as the allowable strings	String NoA
InFile	Input capture filename	String
NFrames	Number of frames to filter from the original filter	Integer
OutFile	Output capture file with the specified frames	String
RxMAC	MAC address of receiver in aa:bb:cc:dd:ee:ff format	String
RxMAC1	MAC address of receiver in aa:bb:cc:dd:ee:ff format	String
SnifferTimeGTE	Epoch time greater than the specified time in seconds	Integer
SnifferTimeGTE1	Epoch time greater than the specified time in seconds	Integer
SnifferTimeLTE1	Epoch time lesser than the specified time in seconds	Integer
SrcMAC1	MAC address of Source in aa:bb:cc:dd:ee:ff format	String
TxMAC	MAC address of transmitter in aa:bb:cc:dd:ee:ff format	String
udpdstport		
Notes:		

Name	Description	Value
1. Use any of these fields as required to identify the frame.		

Return Values

Name	Description	Value
FilterStatus		String Success NoPacketsFound
FrameCount	Number of actual frames filtered	Integer

Examples

```
UCC:sniffer_control_filter_capture,InFile,capture_file,OutFile,output_file,SrcMac,aa:bb:cc:dd:ee:ff  
,DstMac, aa:bb:cc:dd:ee:ff,FrameName,assocreq, nframes,1
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.24 SNIFFER_INJECT_FRAME

This command is used to inject the frame data on the current Wi-Fi Test Suite sniffer channel.

Parameters

Name	Description	Value
Count	Number of times the packets to be setn	Integer
FrameData	The frame contents to be injected starting from the 802.11 header to the end of the frame body not including the FCS	String
Interval	Interval between the two frames in milli-seconds	Integer

Return Values

None.

Examples

```
UCC:sniffer_inject_frame,data,<frame data>,count,10,interval,50
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE
```

9.25 SNIFFER_CLEAR_COUNTERS

This command is used to clear AP and STA Wi-Fi Test Suite sniffer counters

Parameters

Name	Description	Value
BSSID	BSSID in 00:01:02:03:04:05 format	String
Program	Program Name	String General PMF TDLS VOE WFD
StationID	MAC address of the STA in the BSS in 00:01:02:03:04:05 format	String

Return Value

None.

Examples

For clearing AP counters

```
UCC: sniffer_clear_counters,Program,PMF,bssid,00:01:02:03:04:05
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE
```

For clearing STA counters

```
UCC: sniffer_clear_counters,Program,PMF,bssid,00:01:02:03:04:05,stationID,00:11:22:33:44:55
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE
```


9.26 SNIFFER_DECRYPT_TRACE

This command is used to decrypt the Wi-Fi Test Suite sniffer frames in given Wi-Fi Test Suite sniffer trace file based on the specified list of PMK values and corresponding STA MAC addresses.

Parameters

Name	Description	Value
InFile	Input file name	String
MSKFile	MSK file generated by Hostapd Radius server	String
OutFile	Output file name	String

Return Values

None.

Examples

```
UCC:sniffer_decrypt_trace,infile,_HS2_46,outfile,HS2_46,PMKFile,pmkdump_46 Sniffer:status,RUNNING
Sniffer:status,COMPLETE
```

9.27 SNIFFER_FETCH_FILE

This command is used to fetch a file (MSK file for Hotspot2.0) from a given IP address/path of the Radius server and copies the file to a destination path on the local machine. This command implementation uses the Linux utility 'scp' to fetch the file from the remote machine.

Parameters

Name	Description	Value
DstPath	Destination path on local machine	String
IPAddr	IP Address	String
Password	Password	String
SrcPath	Full path including file name on source (Radius server)	String
TransFile	Method to transfer file from AAA server to sniffer.	String scp wget
User	User name	String

Return Values

None.

Examples

```
UCC:sniffer_fetch_pmk_file,IPAddr,192.165.100.10,SrcPath,/tmp/pmkdump,DstPath,/WFASniffer/pmkdump_46
```

```
Sniffer:status,RUNNING
```

```
Sniffer:status,COMPLETE
```

9.28 WFA_SNIFFER_CONTROL_CAPTURE_DECRYPT

This command is used to derive the PTK from the base capture file that contains a 4-way handshake. The PTK file is then used to decrypt the encrypted data file.

Parameters

Name	Description	Value
BSSID	BSSID (MAC address of AP/GO)	String
Fourway_File	Four way capture file containing the derived PTK	String
Infile	Input Wi-Fi Test Suite sniffer capture file	String
Outfile	Output Wi-Fi Test Suite sniffer result file	String
Passphrase	Passphrase used to derive PTK	String
SrcMAC	Source MAC address	String

Return Values

Name	Description	Value
FourwayFile	Returned four-way handshake filename	String

Examples

```
UCC:wfa_sniffer_control_capture_decrypt,infile,_WFD_511,outfile,_WFD_511_A,srcmac,$P2P_IF_ADDR_STA1,bssid,$P2P_IF_ADDR_DUT,passphrase,$PASSPHRASE
```

```
Sniffer:status,RUNNING
```

```
Sniffer: status,COMPLETE,FourwayFile,fourway
```

9.29 WFA_AV_CAPTURE

This command controls the camera connected to the Wi-Fi Test Suite sniffer PC to capture the specified media type (audio/video/picture). The resulting output file is saved in the specific format (.mp4/.jpeg/.mp3) on the Wi-Fi Test Suite sniffer PC.

Parameters

Name	Description	Value
AudioIF	Audio interface name of the camera/webcam	String Example: hw:1
Capture_Type	Media type of the capture	String Example: Photo/ Video/ Audio
Duration	Duration in seconds for Video/Audio capture	Integer
Filename	Filename of the Media Capture file	String Example: video.mp4 photo.jpeg
MicType	Built-in MIC details of the camera/webcam	String Example: Single/Dual
VideoIF	Video interface name of the camera/webcam	String Example: video0

Return Values

None.

Examples

```
UCC:wfa_av_capture,Videoif,video0,Audioif,hw:1,micType,Single,capture_type,video,duration,15,filename,511_Step25_Video.mp4
```

```
Sniffer:status,RUNNING
```

```
Sniffer: status,COMPLETE,CheckResult,SUCCESS
```

9.30 SNIFFER_MEDIA_CHECK

This command is used to trigger dvbsnoop (an external software tool) to analyze the audio/video streams for correct H.264 codecs and Audio codecs.

Parameters

Name	Description	Value
CheckType	Specific check to be performed by dvbsnoop	String Example: PES_Header/ LPCM_RegDesc
Infile	Input Wi-Fi Test Suite sniffer capture filename	String
Mediatype	Type of media to be checked by dvbsnoop	String Example: audio
Outfile	Output Wi-Fi Test Suite sniffer result filename	String
PES_Header_Field	Specific PES header field check to be performed by dvbsnoop	String Example: Stuffing_Bytes

Return Values

None.

Examples

```
UCC:sniffer_media_check,infile,WFD511_DVBout2,outfile,WFD511_LPCMDesc,mediaType,audio,checkType,LPCM_RegDesc
```

```
Sniffer:status,RUNNING
```

```
Sniffer: status,COMPLETE,CheckResult,SUCCESS
```

9.31 SNIFFER_PTS_CALC

This command is used to extract the Presentation Time Stamps of the required stream (audio/video) PES header from the specified Wi-Fi Test Suite sniffer capture file and calculates the PTS delta.

Parameters

Name	Description	Value
Count	Number of PTS values to be considered for PTS Delta Calculation	Integer
DestMAC	Destination MAC address	String
Infile	Input Wi-Fi Test Suite sniffer capture file	String
SrcMAC	Source MAC address	String
Stream	Stream type	String Example: audio/video

Return Values

Name	Description	Value
PTS Delta Average	Returns the average (based on count) for PTS delta values	Float

Examples

```
UCC:sniffer_pts_calc,infile,WFD_5113_BB,srcmac,00:11:22:33:44:55,destmac,00:33:44:55:66:77,stream,v
ideo,count,10
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE,CheckResult,SUCCESS,PTS_Val,0.016
```

9.32 WFA_MERGE_TRACE

This command merges two sniffer capture files (libpcap/pcapng) and returns the merged output file.

Parameters

Name	Description	Value
Infile1	Input Wi-Fi Test Suite sniffer capture file - 1	String
Infile2	Input Wi-Fi Test Suite sniffer capture file - 2	String
Outfile	Merged output file	String

Return Value

None.

Example

```
UCC:wfa_merge_trace,infile1,_WFD_514_0,infile2,_WFD_514_A,outfile,WFD_514_Merged_Temp1
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE,MergeStatus,SUCCESS
```

9.33 SNIFFER_CHECK_DSCV_WINDOW

This command verifies the discovery window parameters in a Wi-Fi Test Suite sniffer capture file.

Parameters

Name	Description	Value
Duration	Duration in seconds for the Wi-Fi Test Suite sniffer capture	Integer
FieldName	Field that is to be checked bcn_int = Beacon interval for calculating average RandFactor = Verify whether the Random Factor changes in discovery windows as per requirement TSF = Timing synchronization function checks in discovery windows AMI = Anchor Master Information	String bcn_int RandFactor TSF AMI
FieldnName	Field that is to be checked TSF = Device TSF AMI = Anchor Master Info BCN_INT = Beacon interval RandFactor = Random factor of the device SnifferTimeStamp = Relative Tx time of the frame from the beginning of the trace	String TSF AMI BCN_INT RandFactor SnifferTimeStamp
Filename	Name of the Wi-Fi Test Suite sniffer trace file	String
Operation	Type of Wi-Fi Test Suite sniffer check to be performed for the NAN discovery windows chkVal = Check beacon TU values and deviations chkDrift = Sample beacons and check the drifts in TSFs match = Check if beacons from AM and another NAN device are separated by less than 16TUs within discovery windows sequence = Check the sequence of sync beacons from AM and a NAN device within discovery windows offset = Check the time offset between NAN beacon frames in the 2.4GHz and 5GHz bands offsetDevs = Check the time offset between NAN beacon frames in the 2.4GHz (DUT) and 5GHz (STA) bands chkTrans_AM = Check whether the device transitions to Anchor Master role within 2 DW chkTrans_NAM = Check whether device transitions from Anchor Master role to any other role within 1 DW chkContMitigation = Check whether service discovery frames are transmitted around 2.3ms after the sync beacon transmission	String chkVal chkDrift match sequence offset offsetDevs chkTrans_AM chkTrans_NAM chkContMitigation
PeriodDW	Periodicity of SDF transmission in DWs	Integer
SrcMAC	MAC address of Source in aa:bb:cc:dd:ee:ff format	String
SrcMAC2	MAC address of Source in aa:bb:cc:dd:ee:ff format	String
TmFilter	The Wi-Fi Test Suite sniffer field (TSF or MAC timestamps) to use for time based analysis	String tsf = Use TSFs to perform sniffer analysis mac = Use MAC timestamps to perform Wi-Fi Test Suite sniffer analysis
TU	TU for beacons to be checked	Integer

Return Value

None.

Example

```
UCC:
sniffer_check_Dscv_Window,filename,NAN_521_2,srcmac,11:22:33:44:55:66,fieldname,bcn_int,TU,512,duration,300
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE,CheckResult,SUCCESS
```


9.34 SNIFFER_GENERATE_HASH

This command is used to generate a SHA-256 hash from the service name that is supported by the device

Parameters

Name	Description	Value
Service_name	Name of the Wi-Fi Test Suite sniffer trace file	String

Return Value

Name	Description	Value
ServiceHash	Returns the SHA-256 hash of the service name	String

Example

```
UCC: sniffer_generate_hash,service_name,org.wifi.nan.test
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE,ServiceHash,241201081206
```

9.35 SNIFFER_GENERATE_RAND_MACS

This command is used to generate a list of random MAC addresses.

Parameters

Name	Description	Value
Count	Number of space separated random MAC address to generate	String
IncludeMacs	MAC address to include at a random position in the space separated list of random MAC address	String

Return Value

Name	Description	Value
MACs	List of space separated MAC addresses	String

Example

```
UCC: sniffer_generate_rand_macs,count,5,includeMacs,11:22:33:44:55:66
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE,macs,AA:BB:CC:DD:EE:FF 12:34:56:78:90:51 11:22:33:44:55:66
FF:FF:FF:FF:FF:FF 0A:1B:2C:3D:4E:5F
```

9.36 SNIFFER_CONTROL_SUBTASK

This command is used to instruct the Sniffer Control Perl program to call the local executable software using the passed parameters.

Parameters

Name	Description	Value
Command	Command related shell script or executable software	String
Param1	First parameter for above executable command	String
Param2	Second parameter for above executable command	String
Param3	Third parameter for above executable command	String
Param4	Fourth parameter for above executable command	String
Param5	Fifth parameter for above executable command	String

Return Value

None.

Example

```
sniffer_control_subtask,command,/usr/bin/WMMPS/parseCap,param1,A.U,param2,$STA1_MACAddress,param3,$
TGMacAddress,param4,$DutMacAddress
```

10 Common commands for AP/STA/PCP/Sniffer-Injector

10.1 DEV_SEND_FRAME

This command is used to initiate or force (unless specified otherwise in the command description) a device to send a specific frame. The device can be an AP or STA or Injector Tool (Hosted by Wi-Fi Alliance Wi-Fi Test Suite Sniffer). If any of the frames are not supported or not allowed due to some reason, the command will return ERROR.

This command may be applicable to a DUT based on the type of program.

Parameters

For the HS2 and HS2-R2 Optional parameters, the STA/Injector sends an ANQP query to the AP based on given parameters.

Name	Description	Value
3GPP_Info	3GPP Cellular Network information This parameter is only required if the STAUT has SIM or USIM credentials.	Integer 0 = Don't include (Default) 1 = Include
ANQP_Cap_List	GAS ANQP for the Capability list.	Integer 0 = Don't include (Default) 1 = Include
BSSID	AP MAC address used for BSS in 00:01:02:03:04:05 format	String
	AP MAC address	String
	MAC address	String
CapReNegotiateParam	Renegotiate with the requested video format. The format is CEA-<Index> or VESA-<Index> or HH-<Index>. Index as defined in the technical specification.	String Example: CEA-0 means 640x48060fps
Case	If set to Normal, the device sends the frame with correct parameters. If set to WrongBssid, the device sends the frame with an incorrect BSSID "00:01:02:03:04:05" This parameter is required for an AP Control agent implementation.	String Normal WrongBssid
ChannelWidth	Channel width in MHz	String 20 40
	The operating channel width the sending STA is able to receive	Short integer Example, 20/40/80/160.
Dest	Destination MAC address (BSSID or STA MAC address)	String
Dest_MAC. DestMAC	Destination MAC address (BSSID or STA MAC address) For 60GHz, specifies the destination MAC address to send the frame.	String
Destination	Destination MAC address	String
DestIP	Destination IP Address for generating ARP Reply (Only for testbed devices)	String
Domain_List	Domain Name list	Integer 0 = Don't include (Default) 1 = Include
FrameName	For HS2-R2: Gratuitous ARP Request and Gratuitous ARP Response frames must be implemented by a Wi-Fi Test Suite PC Endpoint and are optional for test bed STA.	String ANQPQuery DLSrequest GARPreq GARPres NeighAdv

Name	Description	Value
		ARPProbe ARPAnnounce NeighSolicitReq
	For WFD: Frame template to be used for injector or transmission that contains the necessary frame data	String WFD_ProbeReq WFD_ProbeReqTds RTSP WFD_ServDiscReq3 11v_TimingMsrReq
	For PMF: Frame to be injected from the DEV	String assocreq reassocreq auth disassoc deauth saquery
	For VHT: Operating Mode Notification frame	String Op_md_notif_frm
	For 60GHz: Beam refinement protocol frame, Sector Sweep frame, Announce frame	String Brp Ssw Announce
HS_Cap_List	HS Capability list	Integer 0 = Don't include (Default) 1 = Include
Icon_Request	Icon Request includes the filename to be requested	String Ex: 1234_wifi.png
Interface	Interface ID	String
L-RX	This field indicates the compressed number of TRN-R subfields requested by the transmitting STA as part of beam refinement.	Integer. For example, 16
Message	Type of RTSP message	String RTSP_M14_Req RTSP_M15_Req
Mode	If set to Protected, the frame is sent in protected mode. If set to UnProtected, the frame is sent in open security mode. This parameter is required for an AP Control agent implementation.	String Protected UnProtected
NAI_Home_Realm_List	HS NAI Home Realm The value of Home Realm is mail.example.com.	Integer 0 = Don't include (Default) 1 = Include
NAI_Realm_List	GAS ANQP NAI Realm list	Integer 0 = Don't include (Default) 1 = Include
NSS	Indicates the number of spatial streams the sending STA is able to receive	Short integer
OP_Class	Operating Class Indication	Integer 0 = Don't include (Default) 1 = Include
Oper_Name	HS Operator Friendly Name	Integer 0 = Don't include (Default) 1 = Include

Name	Description	Value
OSU_Provider_List	OSU provider list Include the element in query	Integer 0 1
Peer	Peer MAC address	String
Program	Program Name	String PMF TDLS WFD HS2 HS2-R2 VHT 60GHz
Reason	Reason code for TDLS teardown frame. This option is used with frame type 'Teardown'.	Integer
RTSPMsgType	Type of RTSP message	String PAUSE PLAY TEARDOWN TRIGGER-PAUSE TRIGGER-PLAY TRIGGER-TEARDOWN SET_PARAMETER
SenderIP	Sender IP Address for generating ARP Reply (Only for testbed devices)	String
SetParameter	RTSP set parameter Value	String CapUibcKeyBoard CapUibcMouse CapReNego Standby uibcSettingEnable UibcSettingDisable Standby route_audio 3dVideoParam 2dVideoParam
Source	Source MAC address	String
SrcMAC	Source MAC address with which the frame has to be transmitted This parameter is required for an AP Control agent implementation.	String
StationID	STA MAC or Broadcast MAC. Required if command issued from an AP. This parameter must be always the last parameter on the command list.	String Example: 00:11:22:33:44:55 or: ff:ff:ff:ff:ff:ff
Status	For TDLS, the status code for TDLS Setup confirmation frame. This option is used with frame type 'Setup'. Support for non-zero values is only required for test bed STAs. For HS2, this is the status code.	Integer
Timeout	Timeout value. This value will be always greater than 300 seconds per the test plan definition. Support for this parameter is required only for test bed STAs and only the value of 301 is required to be supported.	Integer
Type	Frame type to be sent	String discovery setup teardown chswitchreq2

Name	Description	Value
Venue_Name	HS Venue Name Information	Integer 0 = Don't include (Default) 1 = Include
WAN_Mat	WAN Metrics	Integer 0 = Don't include (Default) 1 = Include

Return Values

None.

Examples

For Wi-Fi Test Suite Sniffer Injector:

```
UCC:dev_send_frame,Name,<AP_Name>,interface,wlan0,Program,PMF,FrameName,saquery,Protected,CorrectKey,sender,sta,bssid,00:01:02:03:04:05,stationID,00:11:22:33:44:55
```

```
Sniffer: status,RUNNING
```

```
Sniffer: status,COMPLETE
```

WFD example:

```
UCC:dev_send_frame,interface,wlan,Program,WFD,frameName,WFD_ProbeReq,source,00:01:02:03:04:05,destination,ff:ff:ff:ff:ff:ff,devtype,sink
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2 example 1:

```
UCC: dev_send_frame, program, HS2,interface,wlan0, dest,00:11:22:33:44:55, FrameName, ANQPQuery, NAI_Realm_List, 1, 3GPP_Info,1,Domain_List,1
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2 example 2:

```
UCC: dev_send_frame, program, HS2, interface, wlan0, dest, 00:11:22:33:44:55, FrameName, ANQPQuery, HS_Cap_List,1
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2 Example 3:

```
UCC: dev_send_frame, program, HS2, interface, wlan0, dest, 00:11:22:33:44:55, Type, BTM_Req, ESS_DISASSOC_IMM,1, SESS_INFO_URL http://www.wi-fi.org/
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

60GHz example:

```
UCC: dev_send_frame, program, 60GHz, interface, wlan0, frameName,brp,L-RX,16
```



CA:status,RUNNING

CA:status,COMPLETE

11 STA command line interface utilities

The Station Command Line Interface (STA CLI) utilities provide common interfaces for interacting with a local STA device. Each command is implemented as an independent utility program (typically windows batch file or executable). These commands can be either production, draft, optional, or not applicable depending on the platform component and certification program. The CAPI Program matrix classifies commands based on component and program.

The Interface ID used in the following definitions must match the Interface ID returned from the CAPI command `DEVICE_LIST_INTERFACES`.

Note The commands in this section are only applicable for STAs utilizing WinXP or Win7 Wi-Fi Test Suite binary distributed by Wi-Fi Alliance. All other platform STAs should refer to section 0.

11.1 Error codes and return values

Each CLI shall return an error code status through the environment variable: `WFA_CLI_STATUS`.

`WFA_CLI_STATUS` Code Definition:

`WFA_CLI_COMPLETE(0)`

`WFA_CLI_ERROR(1)`

`WFA_CLI_INVALID(2)`

`WFA_CLI_RETURN`:

If there is any return value from the CLI, it shall be returned through the environment variable `WFA_CLI_RETURN` (String). The return value format is `<var1_name, var1_value, var2_name, var2_value>`.

For example, if CLI 'STA_P2P_START_GROUP_FORMATION' returns two values "result" and "groupid" then the environment variable `WFA_CLI_RETURN` should be set to `"result,client,groupid,00:11:22:33:44:55 DIRECT-S2"`.

11.2 RESET_DEFAULT

This command is used to reset the STA parameters to their default values per the definition of each individual program certification test plan.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Set	Test Program	String 11n P2P TDLS PMF

Return Values

Pass/Fail error codes through the environment variables.

Example

```
reset_default /interface wlan /set 11n
```


11.3 SET_11N_CHANNEL_WIDTH

This command is used to set the supported 11n channel width.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Width	Supported channel width	20 40 Auto

Return Values

None.

Example

```
set_11n_channel_width /interface wlan /width auto
```

11.4 SET_40_INTOLERANT

This command is used to set the 40MHz intolerant feature.

Parameters

Name	Description	Value
/Action	Enable or disable the 40 Mhz intolerant feature	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_40_intolerant /interface wlan /action enable
```

11.5 SET_ADDBA_REJECT

This command is used to set the device to reject any ADDBA request by sending the ADDBA response with status “decline.”

Parameters

Name	Description	Value
/Action	Enable or disable the ADDBA reject feature	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_addba_reject /interface wlan /action enable
```

11.6 SET_AMPDU

This command is used to set the device AMPDU aggregation.

Parameters

Name	Description	Value
/Action	Enable or disable the AMPDU aggregation feature	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_ampdu /interface wlan /action enable
```

11.7 SET_AMSDU

This command is used to set the device AMSDU aggregation.

Parameters

Name	Description	Value
/Action	Enable or disable the AMSDU aggregation feature	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_amsdu /interface wlan /action enable
```

11.8 SET_GREENFIELD

This command is used to set the device HT Greenfield support.

Parameters

Name	Description	Value
/Action	Enable or disable the HT Greenfield feature	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_greenfield /interface wlan /action enable
```

11.9 SET_MCS

This command is used to set the device MCS settings.

Parameters

Name	Description	Value
/FixedRate	The MCS rate	Integer 0-23 Null
/Interface	Interface ID	String
/MCS32	Enable or disable the MCS feature	Enable Disable

Return Values

None.

Example

```
set_mcs /interface wlan /fixedRate 7 /mcs32 disable
```

11.10 SET_RIFS_TEST

This command is used to set up or tear down the RIFS testing configuration as instructed in the 11n Test Plan.

Parameters

Name	Description	Value
/Action	Enable or disable RIFS testing	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_rifs_test /interface wlan /action enable
```

11.11 SET_SGI20

This command is used to set device Short Guard Interval setting.

Parameters

Name	Description	Value
/Action	Enable or disable the Short Guard Interval feature	Enable Disable
/Interface	Interface ID	String

Return Values

None.

Example

```
set_sgi20 /interface wlan /action enable
```

11.12 SET_STBC_RX

This command is used to set the device STBC Receive stream support settings.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Streams	Number of supported streams	Short integer 1-4

Return Values

None.

Example

```
set_stbc_rx /interface wlan /streams 2
```

11.13SET_SMPS

This command is used to set the device Spatial Multiplexing Power Save settings. Note that this command generates the applicable action frame.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Mode	SM Power Save Mode	Dynamic Static NoLimit

Return Values

None.

Example

```
set_smps /interface wlan /mode dynamic
```

11.14SEND_ADDBA

This command is used to send an ADDBA request to the associated AP for a specific TID.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/TID	Traffic Identifier	Short integer

Return Values

None.

Example

```
send_addba /interface wlan /tid 5
```

11.15SEND_COEXIST_MGMT

This command is used to send a 20/40 BSS Coexistence Management Frame to the associated AP.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Type	If set to Mhz, send a coexistence management frame with 40Mhz intolerant If set to BSS, send a coexistence management frame with 20 MHz BSS width request If set to ChnlRepo, send a coexistence management frame with 20/40 BSS intolerant channel report	String Mhz BSS ChnlRepo
/Value		String 0 = Mhz 1 = BSS Period separated list for ChnlRepo

Return Values

None.

Example

```
send_coexist_mgmt /interface wlan /type Mhz /value 1
send_coexist_mgmt /interface wlan /type chnlrepo /value 3
```

11.16SET_NOACK

This command is used to set a NOACK on the specific QoS categories.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Mode	QoS Policy (BE BK VI VO in order)	Short Integer

Return Values

None.

Examples

```
Set_noack /interface wlan /mode 0 1 0 1
```

11.17SET_TXSP_STREAM

This command is used to set the transmit spatial stream.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Value	Number of streams	Short integer

Return Values

None.

Examples

```
Set_txsp_stream /interface wlan /value 2
```

11.18SET_RXSP_STREAM

This command is used to set the receive spatial stream.

Parameters

Name	Description	Value
/Interface	Interface ID	String
/Value	Number of streams	Short integer

Return Values

None.

Examples

```
Set_rxsp_stream /interface wlan /value 2
```


12 Remote power switch control API

To enable or disable a device, a remote power switch accepts external commands through telnet, http or serial line interfaces to turn on/off the power at the specified power port.

12.1 POWER_SWITCH_CTRL

This command is used to control the power switch.

Parameters

Name	Description	Value
Device	Device name	String
Name	Switch name	String
State	Device state	String

Return Values

None.

Examples

```
UCC: power_switch_ctrl,name,powerSwitch_1,device,AP_1,state,ON
```

```
PSW:status,RUNNING
```

```
PSW:status,COMPLETE
```

12.2 POWER_SWITCH_RESET

This command is used to reset the power switch by turning off all ports.

Parameters

Name	Description	Value
Name	Switch name	String

Return Values

None.

Examples

```
UCC: power_switch_ctrl,name,powerSwitch_1
```

```
PSW:status,RUNNING
```

```
PSW:status,COMPLETE
```

13 Wi-Fi Alliance Extended MAC Tester CAPI commands

This section describes the CAPI commands to control the Wi-Fi Alliance Extended MAC Tester (EMT).

13.1 WFAEMT_CONFIG_NAV

This command is used to configure the NAV test parameters. This command shall configure the EMT device in monitor mode and configure the channel as per the given inputs. The command will then verify that the device was set to monitor mode in the specified channel and return COMPLETE.

Parameters

Name	Description	Value
Band	Radio band	String
Channel	Channel number	Integer
Interface	Interface name	String

Return Values

None.

Examples

```
UCC: wfaemt_config_nav,interface,wlan0,channel,11,band,11g
```

```
EMT:status,RUNNING
```

```
EMT:status, COMPLETE
```

13.2 WFAEMT_START_NAV_TEST

This command is used to start a NAV test.

Parameters

Name	Description	Value
Interface	Interface name	String

Return Values

None.

Examples

```
UCC: wfaemt_start_nav_test,interface,wlan0
```

```
EMT:status,RUNNING
```

```
EMT:status,COMPLETE
```

13.3 WFAEMT_STOP_NAV_TEST

This command is used to stop a NAV test.

Parameters

Name	Description	Value
Interface	Interface name	String

Return Values

None.

Examples

```
UCC: wfaemt_stop_nav_test,interface,wlan0
```

```
EMT:status,RUNNING
```

```
EMT:status,COMPLETE
```

13.4 WFAEMT_CONFIG_STAUT_MIC

This command is used to configure the MIC test parameters. This command shall configure the EMT device in Access Point mode and set the channel, band and security as per the given input.

Parameters

Name	Description	Value
Band	Radio band	String
Channel	Channel number	Integer
Interface	Interface name	String
IPAddress	IP address	String
Netmask	Netmask	String
Passphrase	Passphrase	String
SecuMode	Security mode	String WPA-PSK WPA2-Mixed-PSK
SSID	SSID	String

Return Values

None.

Examples

```
UCC: wfaemt_config_staut_mic,interface,wlan0,channel,11,band,11g,secumode,WPA-PSK,passphrase,12345678,ssid,STAUT-MIC,ipaddress,192.168.250.100,netmask,255.255.255.0
```

```
EMT:status,RUNNING
```

```
EMT:status, COMPLETE
```

13.5 WFAEMT_START_STAUT_MIC_TEST

This command is used to start a STAUT MIC test. This command will return after the MIC testing has completed. This may take 3-5 minutes.

Parameters

Name	Description	Value
AttackMAC	MAC address of the station for Pure WPA and Broadcast address (ff:ff:ff:ff:ff:ff) for Mixed mode	String
Interface	Interface name	String
MACAddress	MAC address of the station	String

Return Values

None.

Examples

```
UCC:
wfaemt_start_staut_mic_test,interface,wlan0,macaddress11:22:33:44:55:66,attackmac,00:11:22:33:44:55

EMT:status,RUNNING

EMT:status, COMPLETE
```

13.6 WFAEMT_STOP_STAUT_MIC_TEST

This command is used to stop the STAUT MIC test.

Parameters

Name	Description	Value
Interface	Interface name	String

Return Values

None.

Examples

```
UCC: wfaemt_stop_staut_mic_test,interface,wlan0

EMT:status,RUNNING

EMT:status, COMPLETE
```

13.7 WFAEMT_CONFIG_APUT_MIC

This command is used to configure the APUT MIC test. This command shall configure the EMT device in station mode and associate the EMT device to the specified AP which has the SSID and BSSID as specified in the input parameters by using wpa_supplicant to configure the EMT to act like a station. This command returns COMPLETE after it connects to the specified AP.

Parameters

Name	Description	Value
Band	Radio band	String
BSSID	AP MAC address	String
Channel	Channel number	Integer
Interface	Interface name	String
IPAddress	Wireless station IP address	String
Netmask	Netmask	String
Passphrase	Passphrase	String
SecuMode	Security mode	String WPA-PSK WPA2-Mixed-PSK
SSID	SSID	String

Return Values

None.

Examples

```
UCC: wfaemt_config_aput_mic,interface,wlan0,band,11g,channel,11,secumode,WPA-PSK,passphrase,12345678,ssid,APUT-MIC, bssid,00:11:22:33:44:55, ipaddress, 192.165.100.23, netmask, 255.255.255.0
```

```
EMT:status,RUNNING
```

```
EMT:status,COMPLETE
```

13.8 WFAEMT_START_APUT_MIC_TEST

This command is used to start the APUT MIC test. The command will be return after the MIC testing completes. This may take 3-5 minutes.

Parameters

Name	Description	Value
Interface	Interface name	String
MACAddress	MAC address of the AP sending MIC attack frame	String

Return Values

None.

Examples

```
UCC: wfaemt_start_aput_mic_test,interface,wlan0,macaddress11:22:33:44:55:66
```

```
EMT:status,RUNNING
```

```
EMT:status,COMPLETE
```

13.9 WFAEMT_STOP_APUT_MIC_TEST

This command is used to stop the APUT MIC test.

Parameters

Name	Description	Value
Interface	Interface name	String

Return Values

None.

Examples

```
UCC: wfaemt_stop_aput_mic_test,interface,wlan0
```

```
EMT:status,RUNNING
```

```
EMT:status,COMPLETE
```

14 Server control APIs

14.1 SERVER_SET_PARAMETER

This command is used to set the supplied value for the specified parameter on the server.

Parameters

Name	Description	Value
CertReEnroll	Configures OSU Server with certificate re-enrollment requirement for the certificate referenced by SerialNo. When enabled, certificate re-enrollment is required. When disabled, certificate re-enrollment is not required.	String Enable Disable
Device	Device Type	String AAAServer OSUServer
InterCACert	Certificate ID of Intermediate CA Certificate to be configured/installed on the OSU server	String ID-Z.2 = NetworkFX ID-Z.1 = Comarch ID-Z.3 = Verizon ID-Z.4 = DigiCert
Issuing_Arch	Column 2 (col2) is the intermediate CA is working on behalf of the trust root CA. Column 4 (col4) the intermediate CA is working on behalf of the SP	String col2 col4
Name	OSU server name	String
OSUServerCert	Certificate ID of OSU Server Certificate to be configured/installed on the OSU server	String ID-Q – NetworkFx ID-P – Comarch ID-V – Verizon ID-W – DigiCert
Program	Program Name	String HS2-R2
ProvisioningProto	Provisioing protocol to be communicated to the mobile device by the AAA server,	String OMADM SOAP
SerialNo	Serial number of the certificate	String Example: 100D
Timeout	Timeout in seconds	Integer
TrustRootCACert	Certificate ID of Root CA to be configured/installed on the OSU server	String ID-T = NetworkFX ID-S = Comarch ID-X = Verizon ID-Y = DigiCert

Return Values

None.



Examples

HS2-R2 Example 1:

```
CCG: server_set_parameter,Program,HS2-  
R2,Device,OSUServer,CertReEnroll,enable,SerialNo,100D,timeout,120
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2-R2 Example 2:

```
CCG: server_set_parameter,Program,HS2-R2,Device,OSUServer,Name,xyz,TrustRootCACert,ID-  
T,InterCACert,ID-Z.2,OSUServerCert,ID,ID-Q
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2-R2 Example 3: (Test Case 5.7d and 5.7e)

```
CCG: server_set_parameter,Program,HS2-R2,Device,AAAServer,ProvisioningProto,SOAP
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```


14.2 SERVER_REQUEST_STATUS

The command is used to fetch the status of the specified operation/ procedure from the server. This is a blocking command and the device shall only return when the operation status is available.

Parameters

Name	Description	Value
ClientMACAddr	MAC address of STA wireless interface	String
Device	Device Type	String AAAServer OSUServer
IMSI_Val	IMSI	String
OCSP	OSU server to contact OCSP responders in the test bed and fetch OCSP responses for all server certificates	String fetch
Program	Program Name	String HS2-R2
SerialNo	Client certificate serial number	String Example: 105A
Status	Status of requested procedure /operation The device is expected to block on this command until the status is available.	String Authentication PolicyUpdate PolicyProvisioning Remediation
Timeout	Timeout in seconds	Integer
UserName	User account username	String Example: test01

Return Values

Name	Description	Value
AuthStatus	Status of the 802.1x authentication procedure Returned if Status = Authentication	String SUCCESS FAIL TIMEOUT
MSK	Master Session Key if AuthStatus is 'SUCCESS' "NULL" if AuthStatus is 'FAIL' or 'TIMEOUT' Returned if Status = Authentication	String
PolicyUpdateStatus	Status of policy update procedure Returned if Status = PolicyUpdate	String Example: Permission Denied Command Failed UpdateComplete
PolicyProvStatus	Status of policy provisioning procedure Returned if Status = PolicyProvisioning	String Example: Provisioning Complete
OSUStatus	Status of OSU procedure Returned if Status = OSU	String Example: SUCCESS Unauthorized

Name	Description	Value
SerialNo	Serial number of client certificate if issued	String
RemediationStatus	Status of remediation procedure for a given UserName or SerialNo Returned if Status = Remediation	String

Examples

HS2-R2 Example 1:

```
CCG: server_request_status,Program,HS2-  
R2,Device,AAAServer,UserName,test01,Status,Authentication,Timeout,120
```

CA:status,RUNNING

CA:status,COMPLETE,AuthStatus,SUCCESS,MSK,12345678123456781234567812345678123456781234567812345678
2345678

HS2-R2 Example 2:

```
CCG: server_request_status,Program,HS2-  
R2,Device,AAAServer,UserName,test01,Status,Authentication,Timeout,120
```

CA:status,RUNNING

CA:status, COMPLETE, AuthStatus, FAIL, MSK, NULL

HS2-R2 Example 3:

```
CCG: server_request_status,Program,HS2-  
R2,Device,OSUServer,UserName,test01,Status,PolicyUpdate,Timeout,120
```

CA:status,RUNNING

CA:status,COMPLETE,PolicyUpdateStatus,Permission denied

HS2-R2 Example 4:

```
CCG: server_request_status,Program,HS2-  
R2,Device,OSUServer,ClientMACAddr,00:11:22:33:44:55,Status,OSU,Timeout,120
```

CA:status,RUNNING

CA:status,COMPLETE,OSUStatus,SUCCESS,SerialNo,100D

HS2-R2 Example5: - IMSI example

```
CCG: server_request_status,Program,HS2-  
R2,Device,OSUServer,imsi_val,234564085155155,Status,PolicyUpdate,Timeout,120
```

```
CA:status,RUNNING
```

CA:status,COMPLETE,PolicyUpdateStatus,SUCCESS

HS2-R2 Example 6: - Remediation example

```
CCG: server_request_status,Program,HS2-  
R2,Device,OSUServer,UserName,test01,Status,Remediation,Timeout,60
```

CA:status,RUNNING

CA:status,COMPLETE,RemediationStatus,Remediation Complete

HS2-R2 Example 7: - ReEnrollment example

CCG: server_request_status,Program,HS2-
R2,Device,OSUSever,SerialNo,100C,Status,Remediation,Timeout,60

CA:status,RUNNING

CA:status,COMPLETE,RemediationStatus,Remediation Complete,SerialNo,100F

HS2-R2 Example 8: - OCSP example

CCG: server_request_status,Program,HS2-R2,Device,OSUSever,OCSP,fetch

CA:status,RUNNING

CA:status,COMPLETE

14.3 SERVER_RESET_DEFAULT

The command is used to reset user accounts on the OSU and AAA servers to the preconfigured state. The preconfigured state has to be configured as per the program name and its associated test plan.

Parameters

Name	Description	Value
ClientMACAddr	Client MAC address to uniquely identify a STA attempting to sign up online using a certificate based credential	String
IMSI_Val	IMSI value to identify a client with SIM based credential	String
Program	Program Name	String HS2-R2
SerialNo	Client certificate serial number	String
UserName	User account username	String

Return Values

None.

Examples

HS2-R2 Example 1:

```
CCG: server_reset_default,Program,HS2-R2,UserName,test01
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2-R2 Example 2:

```
CCG: server_reset_default,Program,HS2-R2,SerialNo,100D
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2-R2 Example 3:

```
CCG: server_reset_default,Program,HS2-R2,imsi_val,310026000000001
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

HS2-R2 Example 4:

```
CCG: server_reset_default,Program,HS2-R2,ClientMACAddr,00:11:22:33:44:55
```

```
CA:status,RUNNING
```

```
CA:status,COMPLETE
```

14.4 SERVER_CA_GET_VERSION

This command is used to obtain the version of the Control Agent software. It should not attempt to perform any action on the Server itself. This command may be used to verify basic connectivity to the Control Agent.

Parameters

Name	Description	Value
TestInfo	Test information such as TestID, TimeStamp. This is an optional parameter providing debug information to the Server about the test being performed. The server shall ignore this parameter if it does not use it.	String

Return Values

Name	Description	Value
Version	Control Agent software version	String

Examples

Example 1:

```
CCG: server_ca_get_version
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,version,1.0
```

Example 2:

```
CCG: server_ca_get_version,TestInfo,HS2-4.1-09-45-59
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,version,1.0
```

14.5 SERVER_GET_INFO

This command returns the vendor, model and software version of the server.

Parameters

None.

Return Values

Name	Description	Value
Vendor	Vendor name	String
Model	Server model	String
Version	Server image or software version	String

Examples

CCG: server_get_info

CA: status,RUNNING

CA: status,COMPLETE,vendor,xxxOSU,model,Ubuntu11.10-VM,version,1.16

15 WPS-NFC Commands

15.1 STA_ER_CONFIG

This command configures the external registrar with the given parameters.

Parameters

Name	Description	Value
BSSID	BSSID of the AP to be configured	String
PassPhrase	Passphrase to generate the security keys	String Ex: 12345678
PIN	Personal Identification Number	String Ex: 12345678
Security	Key management type	String WPA2-PSK
SSID	Service set identification	String
UUID	UUID of the AP to be configured	String Ex: 791A5087-A542-4554-969E-21DCC85EB9B1

Return Values

None.

Examples

```
CCG: sta_er_config,default_ssid,dssid,ssid,scaptest,security,wpa2-psk,passphrase,12345678,pin,12345678
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

15.2 STA_NFC_ACTION

This command defines the NFC specific actions. The command returns after configuring the necessary parameters/steps without waiting for the completion of the given actions.

Parameters

Name	Description	Value
Init	Indicates who will initiate the Handover Request when NFC Operation is set to Connection Handover.	Integer 0 = the device must support both WPS connection handover and P2P connection handover 1 = the device will initiate the handover request
Intent_Val	Relative value between 0 and 15 used to indicate the desire of the P2P Device to be the P2P Group Owner, where a larger value indicating a higher desire.	Short Integer
interface	Interface ID	String
Oper_Chn	Channel number on which the P2P Device is operating as the P2P group owner	Short Integer
Operation	<p>NFC action.</p> <p>WRITE_SELECT generates the P2P Connection handover select and writes onto the NFC Tag when the user puts the tag on the NFC interface. The return values will be null for this option except for PauseFlag.</p> <p>WRITE_CONFIG writes WPS configuration token onto NFC Tag when the user put the tag on the NFC interface.</p> <p>WRITE_PASSWD writes the WPS Password token onto NFC Tag when the user put the tag on the NFC interface. A DUT does not need to support this operation.</p> <p>READ_TAG reads the information from the NFC Tag when the user puts the tag on the NFC interface. This will automatically initiate the connection procedures through association.</p> <p>WPS_READ_CONFIG reads the read WPS config token information from the NFC Tag when the user puts the tag on the NFC interface. This will automatically initiate the WPS connection procedures through association.</p> <p>WPS_READ_PASSWD reads the WPS password token information from the NFC Tag when the user puts the tag on the NFC interface and follow the specific actions as per the WpsStaAction.</p> <p>CONN_HNDOVR generates a connection handover when the user puts the tag near the peer NFC interface. This will automatically initiate the connection procedures through association. If a device operating as Group Owner, it will return immediately.</p> <p>WPS_CONN_HNDOVR results in a WPS Connection handover when the user brings near to the peer NFC interface. This will automatically initiate the connection procedures through association.</p>	String WRITE_SELECT WRITE_CONFIG WRITE_PASSWD READ_TAG WPS_READ_CONFIG WPS_READ_PASSWD CONN_HNDOVR WPS_CONN_HNDOVR

Return Values

4. If Operation is Write then all return values will be NULL/blank (zero length string).
5. If Operation is READ_TAG, then the return value must have a valid Result and GroupID.
6. If Operation is CONN_HNDOVR,
 - a. If the device is running as the Group Owner, then return immediately with all returns values set to NULL/blank (zero length string).

- b. If the role indication in the connection handover message received from the peer is 1 or if the role indication sent by the device is 1 (P2P client in a Group), then return immediately and return the PeerRole parameter.
 - c. Otherwise wait for the connection completion and return the valid Result and GroupID.
7. If Operation is WPS_CONN_HNDOVR, the return values will be NULL/blank (zero length string). If a station receives this command with init=1, then it will return after successful connection.
 8. If Operation is WPS_READ_PASSWD and WPSStaAction is ConfigureAP, then the station returns NULL/blank after successful AP configuration and successful association to the given BSSID.
 9. If Operation is WPS_READ_PASSWD and WPSStaAction is Enroll, then the station returns immediately with return parameters as NULL/blank (zero length string).
 10. If Operation is WPS_READ_CONFIG, the the station returns after successful association to the AP and the return value will be NULL/blank (zero length string).

Name	Description	Value
GroupID	P2P Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String
PauseFlag	PauseFlag indicates that the device requires a pause in the test execution after the requested NFC action because the device has no control over the completion time.	Integer 0 = Disable pause 1 = Enable pause
PeerRole	A device returns the peer role indication after the static/dynamic connection handover,	Integer 0 = P2P device is not in a group 1 = P2P client is in a group 2 = GO in the group -1 = WPS device connection handover
Result	Result of GO Negotiation for P2P-NFC. 'FAIL' is returned only in the case where the GO Negotiation fails due to two STAs having a GO intent value 15. 'ERROR' is returned when a requested WPS-NFC action failed. For example, if a STA_NFC_ACTION command failed.	String GO CLIENT FAIL ERROR

Example

CCG: sta_nfc_action,interface,wlan0,operation,write_select

CA: status,RUNNING

CA: status,COMPLETE,result,,GroupId,,PeerRole,,PauseFlag,,

CCG: sta_nfc_action,interface,wlan0,operation,WPS_READ_CONFIG

CA: status,RUNNING

CA: status,COMPLETE,result,ERROR

CCG:

sta_nfc_action,interface,wlan0,operation,WPS_READ_PASSWD,bssid,aa:bb:cc:dd:ee:ff,UUID,791A5087-A542-4554-969E-21DCC85EB9B1



CA: status,RUNNING

CA: status,COMPLETE

15.3 AP_NFC_ACTION

This command defines the AP NFC specific actions. The command returns after configuring the necessary parameters/steps without waiting for the completion of the given actions.

Parameters

Name	Description	Value
Interface	Interface ID	String
Name	Name of the AP, such as AP-4	String
Operation	<p>NFC action.</p> <p>WRITE_CONFIG writes WPS configuration token onto NFC Tag when the user puts the tag on the NFC interface.</p> <p>WRITE_PASSWD writes the WPS Password token onto the NFC Tag when the user puts the tag on the NFC interface. A DUT does not need to support this operation.</p> <p>WPS_READ_CONFIG reads the read WPS config token information from the NFC Tag when the user puts the tag on the NFC interface. This will automatically initiate the WPS connection procedures through association.</p> <p>WPS_READ_PASSWD reads the WPS password token information from the NFC Tag when the user puts the tag on the NFC interface and follow the specific actions as per the WpsStaAction.</p> <p>WPS_CONN_HNDOVR results in a WPS Connection handover when the user brings near to the peer NFC interface. This will automatically initiate the connection procedures through association.</p>	<p>String</p> <p>WRITE_CONFIG</p> <p>WRITE_PASSWD</p> <p>WPS_READ_CONFIG</p> <p>WPS_READ_PASSWD</p> <p>WPS_CONN_HNDOVR</p>

Return Values

The return value is NULL.

Example

```
CCG: ap_nfc_action,name,ap-4,interface,wlan0,operation,write_config
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE
```

15.4 AP_WPS_READ_PIN

This command generates the WPS PIN, if required, and returns the PIN value. This command may set the WPS configuration method as Display on the AP's internal registrar, but should not start the registration process (the Selected Registrar bit should not be set 1). This command can be issued to only APs.

Parameters

Name	Description	Value
Name	Name of the AP -such as ABC 11n AP	String
Interface	A radio hardware interface or InterfaceID	String

Return Values

Name	Description	Value
PIN	WPS PIN value	String

Example

```
CCG: ap_wps_read_pin,NAME,abc11n,interface,24g
```

```
AP: status,RUNNING
```

```
AP: status,COMPLETE,pin,12345670
```

```
UCC: ap_wps_read_pin,name,11nAP,interface,wlan0
```

```
CA: status,RUNNING
```

```
CA: status,COMPLETE,PIN,1234ABCD
```

16 Command sequence for traffic generation and testing

The following flow-chart shows the sequence in which the UCC will use the traffic commands for traffic generation and testing.

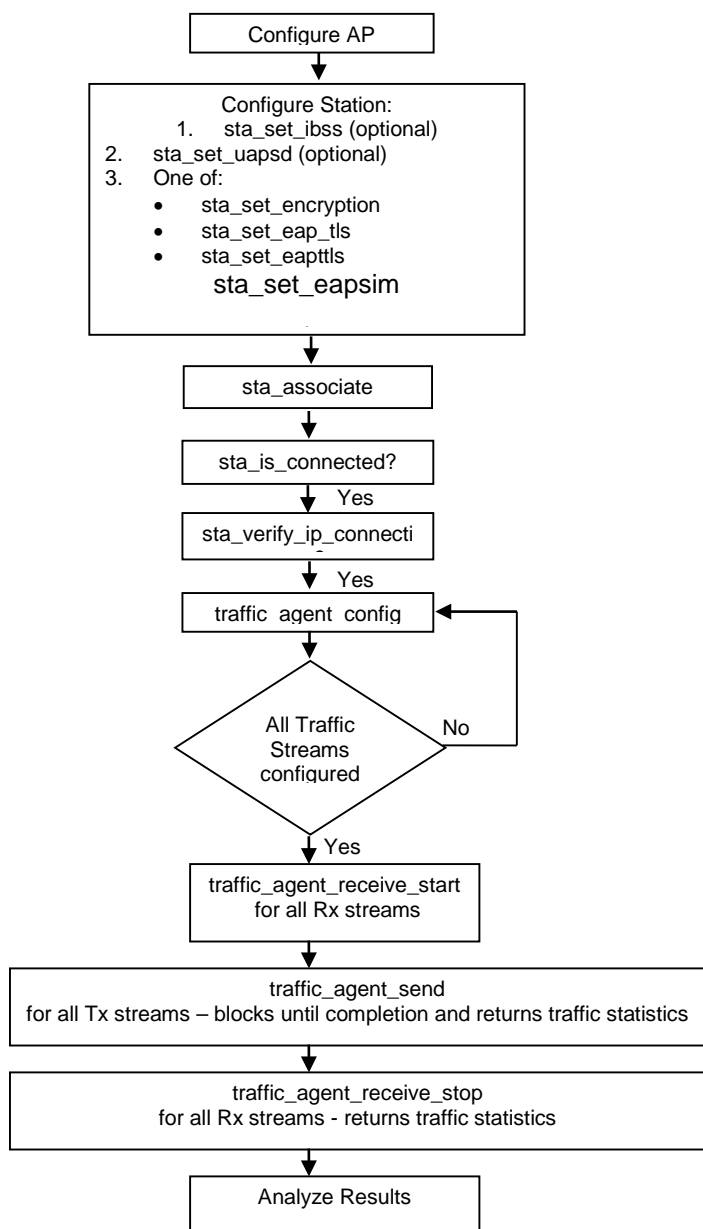


Figure 3. UCC command sequence flow diagram

Appendix A Document revision history

Table 4. Document revision history

Revision Number	Date	Changes
0.1	02/20/2006	Creation
0.2	02/22/2006	Protocol and API revisions
0.3	02/28/2006	<p>Changed 'ASD' to 'DUT' throughout the document.</p> <p>Specified that the maximum command or response length is 2048 characters.</p> <p>Specified that the 'ssid' parameter is case-sensitive.</p> <p>Specified that the 'String' data type is represented as a String.</p> <p>Removed command 'noop' and added 'ca_get_version'</p> <p>Added <i>sta_get_driver_version</i>, <i>sta_set_eaptls</i>, <i>sta_set_eapsim</i>, <i>sta_set_peap</i>, <i>sta_set_ibss</i>.</p> <p>Removed the 'ping' profile in <i>traffic_agent_config</i>.</p>
0.4	03/02/2006	<p>Removed <i>traffic_agent_start</i> and added <i>traffic_agent_send</i> and <i>traffic_agent_receive_start</i>.</p> <p>Removed <i>traffic_agent_stop</i> and added <i>traffic_agent_receive_stop</i>.</p> <p>Removed 'protocol' parameter from <i>traffic_agent_config</i>.</p> <p>Added <i>traffic_agent_reset</i>.</p> <p>Added Appendix B that has a flow chart to show the sequence in which the traffic commands are used.</p> <p>Added a note that communication between the Control Agent and DUT should be out-of-band rather than in-band.</p>
0.5	03/03/2006	<p>Removed <i>traffic_agent_status</i></p> <p>Removed <i>traffic_agent_get_results</i>.</p> <p>Modified <i>traffic_agent_receive_start</i> and <i>traffic_agent_receive_stop</i> to accept multiple streamIDs using a space-separated list.</p> <p>Modified <i>traffic_agent_send</i> to return results for multiple streamIDs using a space-separated list.</p> <p>Renamed <i>sta_get_driver_version</i> to <i>sta_get_device_info</i>. Modified it to return vendor, model and software version.</p>
0.6	03/08/2006	<p>Added optional parameter 'startDelay' in <i>traffic_agent_config</i> for 'IPTV send'.</p> <p>Added command table in Appendix A for PC Traffic Agent.</p>
0.7	03/09/2006	Changes for Version 0.6 were not on the updated document.
0.8	03/20/2006	Added optional parameter 'startDelay' in <i>traffic_agent_config</i> for 'send' profiles 'File_Transfer', 'Multicast' and 'Transaction' too.
0.9	03/22/2006	Added mandatory parameter 'source' in <i>traffic_agent_config</i> for 'receive' profiles 'File_Transfer', 'Multicast', 'IPTV' and 'Transaction'.
1.0	03/23/2006	Renamed mandatory parameter 'port' to 'destinationPort' and added 'sourcePort' in <i>traffic_agent_config</i> for both 'send' and 'receive' of all profiles.
1.1	04/04/2006	<p>Renamed <i>sta_get_device_info</i> to <i>device_get_info</i>.</p> <p>Added <i>device_list_interfaces</i>.</p> <p>Added mandatory parameter 'interface' parameter to all the sta_* functions.</p> <p>Added <i>sta_get_info</i> to get vendor specific information about a specified interface.</p> <p>Added <i>sta_set_ip_config</i>.</p>
1.2	04/04/2006	<p>Changed column heading from 'Interface Status' to 'Device Status' in table of API commands.</p> <p>In <i>sta_set_ip_config</i> changed parameters 'ip', 'mask' and 'dns' from mandatory to optional.</p> <p>Added optional parameter 'defaultGateway'</p>
1.2a	05/04/2006	Editorial changes – header and footer

Revision Number	Date	Changes
1.3a	05/08/2006	Additional edits for pre-beta release
1.4	08/17/2007	Wi-Fi Alliance Logo Update Add a new traffic class "UAPSD" to the command "traffic_agent_config" Add "sta_set_mode" for IBSS
1.5	11/07/2007	Add new traffic profiles "Start_Sync" and "Finish_Sync" to command "traffic_agent_config"
1.6	02/15/2008	Add new command "sta_up_load". Remove profile "Finish_Sync" from command "traffic_agent_config" that is no long a requirement for Voice test
1.7	04/03/2008	In <i>sta_set_ip_config</i> removed parameter 'dns' and added parameters 'primary-dns' and 'secondary-dns' to match implementation
1.8	04/30/2008	Added UCC Extensions Added Draft AP Commands
2.0	05/30/2008	Add Appendix C for UCC commands and examples Add Appendix D for sniffer control and examples Add Appendix E for power switch control
2.0.1	7/02/2008	Replaced 'Integer' with 'Short Integer' Clarified content and corrected typos Removed 8.2 – TESTBED STATION COMMANDS, 8.11 – STA_GET_STATS
2.1	7/31/2008	Fixed content errors. [Issue(s): 21, 22, 42] Appendix B – Added command ap_config_commit Appendix E – Added keywords 'define' and ResultWMM; Renamed aliases 'wfa_testbed_sta_tg_ethernet' and 'wfa_testbed_sta_tg_wireless'
2.2	12/22/2008	Corrected revision history numbering. [Issue(s): 47] Appendix C – Added: sniffer_control_field_check; Removed: sniffer_control_set; Modified: sniffer_control_start Appendix E: UCC Aliases – Added: DutMacAddress, TestbedAPConfigServer, wfa_sniffer, PowerSwitchIPAddress, max_throughput, PayloadValue, stream[x]; Removed: dut_default_gateway, wfa_control_agent_sniffer, wfa_control_agent_psw, wfa_control_agent_tb Appendix E: Command Keywords – Added: sleep, info, CheckThroughput, wfa_test_commands, exit, wfa_test_commands_init, DisplayName, resultWMM_1; Removed: ResultIBSS
2.2.1	01/14/2009	Modified: sniffer_control_field_check
WPA2 DRAFT	05/13/2009	Added: ap_set_radius_server, ap_reset_default, ap_ca_version, ap_get_info, ap_set_security (ssid), ap_set_wireless (BCNINT), ap_get_wirelesaps (BCNINT) Removed ap_get_info, ap_set_security (RADIUS) Added "INTERFACE" parameter to most AP commands.
11n DRAFT		Added AP CAPI Commands: ap_set_11n, ap_set_11n_channel, ap_set_mcs, ap_set_rifs_test Added Appendix D – STA CLI Utility Commands: reset_default, set_11n_channel_width, set_40_intolerant, set_addba_reject, set_ampdu, set_amsdu, set_band, set_greenfield, set_mcs, set_rifs_test, set_sgi20, set_stbc_rx, set_smps, send_addba, send_coexist_mgmt
	06/16/2009	Merged few STA 11n CAPI commands Modified AP_GET_INFO command Added parameter RADIO to AP_SET_WIRELESS command
	06/18/2009	Added the parameter INTERFACEID for new AP CAPI commands
	06/25/2009	Consolidated ap_set_11n, ap_set_11n_channel, ap_set_mcs, and ap_set_rifs_test into ap_set_11n_wireless.

Revision Number	Date	Changes
		Added parameters from ap_set_wireless to ap_set_11n_wireless. Designated additional ap_set_wireless and ap_set_11n_wireless parameters as optional.
	08/06/2009	Add "sta_disconnect" command Update/add parameters to sta_set_11n_wireless
	08/27/2009	Added 11n MIMO Sniffer checks. Updated other WPA2 sniffer commands
2.3.1 11n	09/11/2009	Merged the required changes from WP2 CAPI Added commands – sta_preset, sta_set_sys_time, sta_set_eapfast, sta_set_eapaka
2.3.2 11n		Added ADEPT CAPI commands
		Added NoACK to preset command, rename mode setting command from b,g,a,an,gn to 11b,11g,11a,11na,11ng,11nl
2.3.3 11n		Added ADEPT CAPI commands for APUT MIC
2.3.4 11n	10/13/2009	- Removed all the comments and yellow tags. SET_BAND – Removed SET_SMPS – Comments cleared SEND_COEXIST_MGMG- Comments cleared Sta_preset_parameters – mode- 11abg removed
2.3.5 11n	12/11/09	- Sta_preset_parameters – added values off, PSpoll(on), Fast to powersave parameter.
2.3.6 11n	12/15/09	sta_reassoc command added. Ref – 11n APUT – PMK caching.
2.3.7 P2P	03/03/10	First draft for Wi-Fi Direct CAPIs – STA and Sniffer.
2.3.8 P2P	03/17/10	Updated the CAPI matrix in Appendix H
2.3.9 P2P	03/26/10	Updated CAPI commands based on TG feedback.
2.3.10 P2P	4/12/10	P2P CAPI updates for group formation and persistent group reinvite commands based on TG feedback.
2.3.11 P2P	4/22/10	Added Appendix J – To indicate the intended Sigma command flow for typical P2P cases (Autonomous GO, Group Formation and Persistent Group re-invoke) Added separate command for starting autonomous GO Added sta_p2p_reset command Added the parameter (init_go_neg) in group_formation command to control the initiation of GO Negotiation Request. Added notes in few commands to clarify the command intention. Corrected few examples. CLI return value method for Windows STA updated(section 12.1)
2.3.12 P2P	5/11/2010	New commands added for P2P PF3 cases- STA_SEND_PROBE_REQ and STA_SEND_P2P_PRESENCE_REQ STA_SET_P2P: added parameters to support PF3 cases STA_START_AUTONOMOUS_GO:added Group ID as return value STA_P2P_CONNECT:added groupID as an parameter STA_P2P_START_GROUP_FORMATION:added optional parameter of operating channel number STA_SEND_P2P_INVITATION_REQ/STA_ACCEPT_P2P_INVITATION_REQ: added group ID and reinvite bit in parameter STA_GET_PSK:added group ID as an parameter Added STA_GET_P2P_IP_CONFIG to support multiple P2P groups

Revision Number	Date	Changes
2.3.13 P2P	05/17/2010	Calirification added to the commands STA_P2P_CONNECT, STA_P2P_START_GROUP_FORMATION, STA_SEND_P2P_INVITATION_REQ/STA_ACCEPT_P2P_INVITATION_REQ New command STA_WPS_READ_LABEL added to support WPS Config method 'Label' Optional parameter 'GroupID' added to all the WPS commands Updated appendix J: Sigma Command Sequence
2.3.14 P2P	06/29/2010	New P2P commands added: sta_set_sleep, sta_set_opportunistic_ps Sniffer command added-sniffer_inject_frame P2PInterfaceAddress added in sta_get_p2p_ip_config return values sta_set_p2p: service discovery description added One command removed: sta_send_probe_req Sta_send_service_discovery_req command added More description added to few P2P commands regarding discovery state.
2.3.15 P2P	07/23/2010	Updated section 16 to mark the following command required for P2P Test bed STA- sta_preset_testparameters, sta_set_11n, sta_set_uapsd Added P2Pmanaged, GO_APSD parameters to sta_set_p2p_command
2.3.16 P2P	08/05/2010	Added two new commands – sta_add_arp_table_entry and sta_block_icmp_response sta_set_p2p – NoA configuration changes sta_set_sleep – Group ID parameter made optional
4.0.0	10/14/2010	Changing version number scheme to match the overall Sigma release Correct a few naming on this revision comment table Add a few words to clearly explain sta_preset_testparameters used for. sta_set_eapfast modified for “validateserver” parameter moved to optional parameter. traffic_agent_config added optional parameter for transaction send – “maxcnt” SNIFFER_CONTROL_FIELD_CHECK added optional parameters ‘wpaie’, ‘40MhzCheck’, ‘TKIP_Ad’, ‘CCMP_Ad’, ‘IE_45, IE_61’
4.1.0	1/21/2011	Changing version number to match the Sigma release v4.1.0 [no change in the document]
4.2.0	6/24/2011	Added WPS_IE_DevicePasswordID in section 10.5[Sniffer] Added WFA-EMT CAPI commands
6.0.0-RC1	8/31/2011	Merged TDLS and PMF CAPI specifications to trunk CAPI specification document.
6.0.2	6/19/12	Merged Voice-Ent and Hotspot 2.0 changes
7.0.0	8/14/12	Merged Display-0.6 version Moved section 16 (UCC Aliases and Command Keywords) to UCC User Manual
7.1.0	10/23/12	Added missing IPv6 parameters for get_ip, set_ip and send_ping commands Clarification added in dev_send_frame
8.0.0	07/09/2013	Merged the VHT CAPI specification 6.0.2-VHT08
8.1.0	03/31/2013	STA_SET_PSK Added description for "Ssid" Added "wpa2-wpa-psk" to keyMgmtType Added "aes-ccmp-tkip" to encpType Added "Prog" and "Prefer" parameters STA_SET_EAPTTL Added description for "Ssid" Added "Prog" and "Prefer" parameters Corrected the example

Revision Number	Date	Changes
		STA_PRESET_TESTPARAMETERS Removed "Supplicant" parameter from Mandatory parameters Removed Note Corrected the example for "REINVOKE_WFD_SESSION" Added "AP_SET_11D" and "AP_SET_11H" commands Updated Appendix J: CAPI Program matrix for STA_PRESET_TESTPARAMETERS, AP_SET_11D and AP_SET_11H commands. Added "SNIFFER_CHECK_P2P_NoA_WMMPS_RETRIGGER" command STA_SET_EAPTLS 1. Added "Username" as an optional parameter
8.1.1	07/31/2014	Updated specification terms of use
8.2.0	03/30/2015	Added note for ap_set_11n used for test bed APs Modifications to CAPI program matrix
8.3.0		Added support for NAN.
9.0.0	10/15/2015	1. Added PeerIP and PortList parameters to STA_GET_PARAMETER. 2. Added commands STA_ER_CONFIG, STA_SET_POWER_SAVE, STA_INVOKE_COMMAND, STA_GET_EVENT_DETAILS, STA_NFC_ACTION, STA_POLICY_UPDATE, STA_MANAGE_SERVICE, STA_REASSOCIATE, STA_OSU, SNIFFER_CHECK_RETRY, SNIFFER_CONTROL_SUBTASK. 3. Added NFC, Oper_Chn, Type, ConnectionCapabilityInfo, PDLType, PinConfigMethod, FileType, FileName, FilePath, MNCLength, ManagementTreeURI, HDCP_Km, HDCP_Version, HT, FT_OA, FT_DS, HT_WEP, HT_TKIP, Reset, WMM parameters and examples 4-7 to STA_PRESET_TESTPARAMETERS, AP_SET_11N, AP_SET_11N_WIRELESS, AP_NFC_ACTION. 4. Added Reset_Default and RIFS parameters to STA_SET_11N. 5. Added VHT, HS2, HS2-R2, WFDS, and WMMPS as string values to Prog parameter in the command STA_RESET_DEFAULT. Also added the HS2-R2 example. 6. Added UAPSD, Peer, TPKTimer, ChSwitchMode, OffChNum, SecChOffset, NSS_MCS_Opt parameters to STA_SET_RFEATURE. 7. Removed SDFTxDW parameters from STA_EXEC_ACTION. 8. Added Service_Name, Service_Role, Service_MAC, AdvID, Session_Info, Network_Role, ConnectionCapabilityInfo, Manage_Actions, Send_FileList, SendModify_FileList parameters to STA_MANAGE_SERVICE. Added return values Session_ID, P2P_Result, GroupID and new examples. 9. Added dependencies to STA_SCAN, STA_DISCONNECT, STA_REASSOC. 10. Added dependencies and Type parameter to STA_GENERATE_EVENT. 11. Added EventList parameter, WFDS to parameter Program and new examples to STA_GET_EVENTS. 12. Added GroupID parameter to STA_GET_IP_CONFIG. 13. Added Mode parameter to STA_SET_PSK. 14. Added Prefer parameter to STA_SET_EAPTTLS. 15. Moved Triplet1, Triplet2, Triplet3, parameters to Optional for STA_SET_EAPAKA. 16. Added DYNyn_BW_Sgnl, SGI80, TxBF, LDPC, Opt_Md_Notif_IE, NSS_MCS_Cap, Tx_LGI_Rate, Zero_crc, VHT_TKIP, VHT_WEP, BW_Sgnl, WPSNFC, FakePubKey parameters to STA_SET_WIRELESS. 17. Added dependencies and Ignore_Blacklist parameter to STA_HS2_ASSOCIATE. 18. Added TOS, DSCP parameters to TRAFFIC_SEND_PING. 19. Added new parameters and examples to AP_SET_WIRELESS. 20. Added WLAN_TAG, PreAuthentication parameters and examples to AP_SET_SECURITY.

Revision Number	Date	Changes
		<p>21. Added WLAN_TAG parameter and examples to AP_SET_RADIUS.</p> <p>22. Added WMM and HS2-R2 examples to AP_RESET_DEFAULT.</p> <p>23. Added Minor_Code parameter to AP_DEAUTH_STA.</p> <p>24. Added OSU_PROVIDER_LIST, OSU_SERVER_URI, OSU_METHOD, OSU_SSID, OSU_ICON_TAG, QoS_MAP_SET, STA_MAC, BSS_LOAD, WLAN_TAG, Oper_Class, 4_Frame_Gas parameters and new examples to AP_SET_H2.</p> <p>25. Moved parameter Name to Optional for SNIFFER_CONTROL_START. Added parameters WiredIF, SharedSecret, StartDelay. 26. Moved parameter Name to Optional for SNIFFER_CONTROL_STOP.</p> <p>27. Added RxMAC, P2P_IE_New, WPS_IE_New, Action_Status, P2P_IP_Addr_KDE_ReqIP, P2P_Cap_IpAddrAllocBit, WPS_DevPass_IDAttr mandatory parameters to SNIFFER_CONTROL_FIELD_CHECK. Added RTSP, GasinitResp, GasComebackResp, eapol, icmp as values to FrameName. Added optional parameters Category_Code, BSS_Load_IE, OSU_MList, OSU_Prov_Len, OCSPCertStatus_revoked, OCSP_StatusReq, OSEN_AKM_Suite_List, LL_TargetAddr, EAP_Type_TTLS, Deauth_Reason_URL, EAP_Type_SIM, NAI_Realm_EAP_Method, Tag_Len, Legacy_OSU_SSID Deauth_Type, OSEN_Pairwise_Cipher_Suite_List, OSU_NAI_Len ICON_Width, OSEN_Group_Data_Cipher_Suite ICON_TYPE, ARP_Sender_IP NAI_Credential_Info Service_Desc, ServerCert_EKU Deauth_Reason_Code, Lang_Code IPv6_TargetAddr_NS, EC_IE_Bit46, IPv6_TargetAddr_NA ClientHello, OSEN_Pairwise_Cipher_Suite_Count, ARP_HW_Addr_Len, RSNCap_MFPC, Auth_Param_ID, PPSMO_ID, P2P_MgeAttr_CrconPer, QoS_Priority, QoS_Map_DSCP_Exception_UP WLAN_SSID, ReqMode_Disassoc_Imminent_Bit2, OSU_NAI, QoS_Map_DSCP_Exception, OSU_Server_URI ClientKeyExchange, HS_Vendor_OUI_Type, ConnCap_PortStatus, Reauth_Delay, ICON_FName, Deauth_Reason_URL_Len, IPv6_Dest, Query_Req_Len, WNMNotification_Type, STA_count, QoS_Map_DSCP_Range, ARP_Target_IP, Key_Desc_Version ARP_OpCode, QoS_Map_DSCP_Exception_DSCP_Val, IPv6_Src P2P_MgeAttr_DevMgmt, ConnCap_Proto, FN_Lang_Code, CertStatus_EAPAlert, Session_Info_URL, SD_Lang_Code, ARP_HW_Type, ReqMode_ESS_Disassoc_Imminent_Bit4, TAG_No, PPSMO_ID_ne, EAP_SUCCESS, Server_URL OSEN_IE OCSPCertStatus_good, HS2_Release 11u_IW_HESSID_present, EAP_Method_Count, Friendly_Name, ICON_Height, ARP_Sender_MAC, ServerCertStatus, QoS_Map_Tag_Num, PPSMO_ID_Present, EC_IE_Bit32, OSEN_AKM_Suite_Count, ARP_Proto_Type, ARP_IP_Addr_Len ARP_Target_MAC, erverCert_SubjectAltName, ConnCap_PortNo.</p> <p>28. Added optional parameters BSSID, DestMAC, Type, AfterTimeStamp, ClientMAC, DataLen to SNIFFER_FRAME_CHECK.</p> <p>29. Added HDCP_AKE_TxInfo, HDCP_AKE_Init, eapfailure, eapclienthello, assocreassocreq values to FrameName parameter for SNIFFER_GET_FIELD_VALUE. Add values BaseCSeq, P2PIE_ConfMethod, P2P_NoA_Descriptor, Framcount, WNM_DisassocTimer, BSSLoad_CU, SeqNo to FieldName. Added optional parameters Message, StationID, Name.</p> <p>30. Added alternate name WFA_SNIFFER_CONTROL_FILTER_CAPTURE to SNIFFER_CONTROL_FILTER_CAPTURE. Added values HDCP2x, RTSP, DHCP, eapol, 80211_arp, 80211_garp, deauth, disassoc, eapfailure, eap_clienthello, eaprespiden, anqpresp to parameter FrameName. Added optional parameters udpdstport, ANQP_Infold, ANQP_SubType, bidiMAC, DSValue.</p> <p>31. Added parameter TransFile to SINFFER_FETCH_FILE.</p> <p>32. Added optional parameters CertReEnroll, SerialNo, ProvisioningProto, TrustRootCACert, InterCACert, OSUServerCert, Name, Issuing_Arch, Timeout to SERVER_SET_PARAMETER.</p> <p>33. Added Server APIs for HS2.</p> <p>34. Replaced command matrix with Command Matrix spreadsheet.</p> <p>35. Removed obsolete commands: AP_SEND_FRAME, AP_SET_FEATURE, STA_SET_EAPAKAPRIME, STA_GET_KEY</p>

Revision Number	Date	Changes
9.1.0	02/24/2016	<ol style="list-style-type: none"> 1. Corrected AP_GET_INFO return value Interface to InterfaceID. 2. Added the dependencies for STA_P2P_CONNECT, STA_P2P_START_GROUP_FORMATION 3. Updated the API description for STA_SET_WPS_PBC, STA_WPS_READ_PIN, STA_WPS_READ_LABEL, and STA_WPS_ENTER_PIN. 4. For STA_SET_EAPFAST and STA_SET_EAP_AKA, added the value wpa2-wpa-ent to the parameter keyMgmtType, and aes-ccmp-tkip to encpType. 5. Corrected AP_SET_WIRELESS HS2 examples. 6. Added command AP_SET_RFEATURE.
9.1.0 ac R2 update	06/14/2016	<ol style="list-style-type: none"> 1. Added Name parameter and examples to DEVICE_GET_INFO. Added limitations on strings for TMS. 2. Added VHT to parameter Program in command STA_GET_PARAMETER. 3. Added example of 4SS to parameter NSS_MCS_Opt for command STA_SET_RFEATURE and STA_SET_WIRELESS. 4. Added parameters CTS_Width, MU_TxBF, RTS_Force, TxBandwidth, TX_CTS to STA_SET_WIRELESS. Added values of 60 and 80 to parameter Width. 5. Added command AP_SET_RFEATURE. 6. Added extended channel operation example to parameter Channel for AP_SET_WIRELESS. Added new parameters GroupID, MU_TxBF, MU_NDPA_FrameFormat, Tx_CTS. Added 11ac values for parameters Spatial_RX_Stream and Spatial_TX_Stream. Added to example of 160 MHz channel to parameter Width.
9.2.0		<ol style="list-style-type: none"> 1. Corrected time units for NoA parameters in STA-SET_P2P. 2. For STA_SET_WMM, added the value 60Hz to Group, and DMGADDTS, PTADDTS to Action. Added new parameters. 3. For TRAFFIC_AGENT_CONFIG, added new parameter transProtoType. 4. For TRAFFIC_AGENT_SEND, added new parameter txActFrames. Corrected txPayloadBytes to long integer. 5. For TRAFFIC_AGENT_RECEIVE_STOP, added new parameter txActFrames. 6. For STA_PRESET_TESTPARAMETERS, added values 60GHz, WPS to Program. Added new parameters WPSVersion, WPSTestAttributes, WSCIEFragment, WSCIEapFragment, WSCState, WPS. Added value "on" for parameter Powersave. 7. For STA_SET_WIRELESS, added values 60GHz, WPS to Program. Added new parameters DevRole, BAcKrcvBuf, MSDUSize, Channel, Radio, BcnInt, Security, Encrypt, PSK, ExtSchIE, AllocID, AllocType, PercentBI, CBAPOnly, ABFTLRang, Heartbeat, and new examples. 8. For STA_SEND_ADDDBA, added more detail for parameter TID. 9. For STA_SET_SECURITY, added OPEN as a value for Type, added value AES-GCMP for EncpType. 10. For STA_RESET_DEFAULT, added values 60GHz, WPS to Prog. Added new parameters Type, DevRole, Band and new examples. 11. For STA_SET_RF_FEATURE, added value 60GHz to Prog. Added new parameters CBAPOnly, ExtSchIE, AllocType, AllocID, PercentBI, SrcAID, DestAID and new examples. 12. For STA_GET_PARAMETERS, added value 60GHz to Program. Added value AID to the parameters Parameter, added new return value AID. 13. For STA_SET_POWERSAVE, added new value uncscheduled to parameter Powersave, added new parameter Program, and new example. 14. Added new commands TRAFFIC_AGENT_VERSION. 15. Added depreciated command STA_SEND_FRAME for VE. 16. Added commands AP_SEND_BCNRPRT_REQ, AP_SEND_TSMRPT_REQ, AP_SEND_LINK_MEA_REQ, AP_SEND_BSSTRANS_MGMT_REQ for VE. 17. For AP_SET_WIRELESS, added value 60GHz to Program. Added new parameters ExtSchIE, AllocType, PercentBI, CBAPOnly, BAcKrcvBuf, GAS_Frag_Thr, NumMSDU, MSDUSize, the value 11ad to the parameter Mode, and new examples. 18. For AP_SET_SECURITY, added AES-GCMP as a value for Encrypt.

Revision Number	Date	Changes
		<ul style="list-style-type: none"> 19. For AP_RESET_DEFAULT, added value 60GHz to Program. Added 60GHz text to parameters Type and new examples. 20. For AP_GET_MAC_ADDRESS, added values to parameter Interface, and new examples. 21. For AP_SEND_ADDDBA, added more detail for parameter TID. 22. Added new text and examples to AP_WPS_READ_PIN. 23. For SINFFER_CONTROL_START and SINFFER_CONTROL_STOP and SNIFFER_GET_FIELD_VALUE, added value 60GHz to Program. 24. For DEV_SEND_FRAME, added value 60GHz to Program, and added new 60GHz value for FrameName, new parameters L-RX, TX-TRN-REQ, TRN-T-SubFlds, and new examples. 25. Added new command START_WPS_REGISTRATION. 26. For STA_SET_11n, changed values for TXSP_Stream and RXST_Stream from integer to string. 27. Removed unused commands: STA_SET_IBSS, STA_SET_MODE, STA_UP_LOAD, STA_VERIFY_IP_CONNECTION, AP_REBOOT, AP_SET_11N, STA_SET_RIFS_TEST, AP_GET_INFO. 28. Removed unused parameters from STA_GENERATE_EVENT: sessionid, STA_INVOKE_COMMAND: proto, STA_P2P_START_GROUP_FORMATION: .SSID, STA_SET_S, ECURITY: innereap, micalgo, peapversion, trustedrootca, username, AP_SET_RRM: tre, AP_SET_HS2: internet. 29. Moved WPS-NFC commands to a separate chapter.