Wi-Fi Test Suite
Control API Specification
Version 10.7.0

WI-FI ALLIANCE PROPRIETARY – SUBJECT TO CHANGE WITHOUT NOTICE

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

List of figures

List of tables

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
LOC	Location
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.

Command strings will be less than 4096 bytes.

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>, [,]
```

Response strings must be less than 2048 bytes.

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.

3.3 Response timeouts

After the UCC sends a command to the Control Agent, it waits for a response with a timeout lengthof 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.

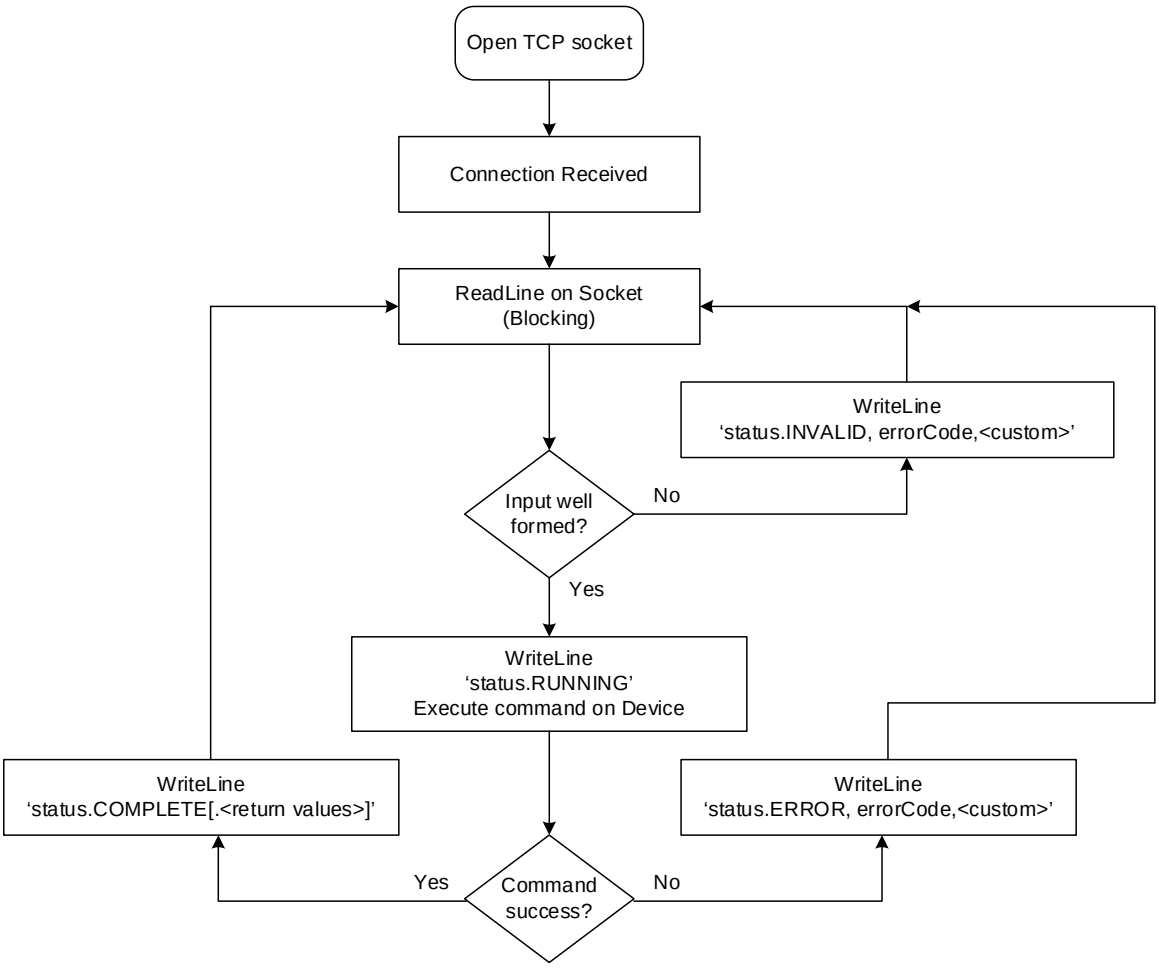

Figure 1. Control Agent state machine

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.

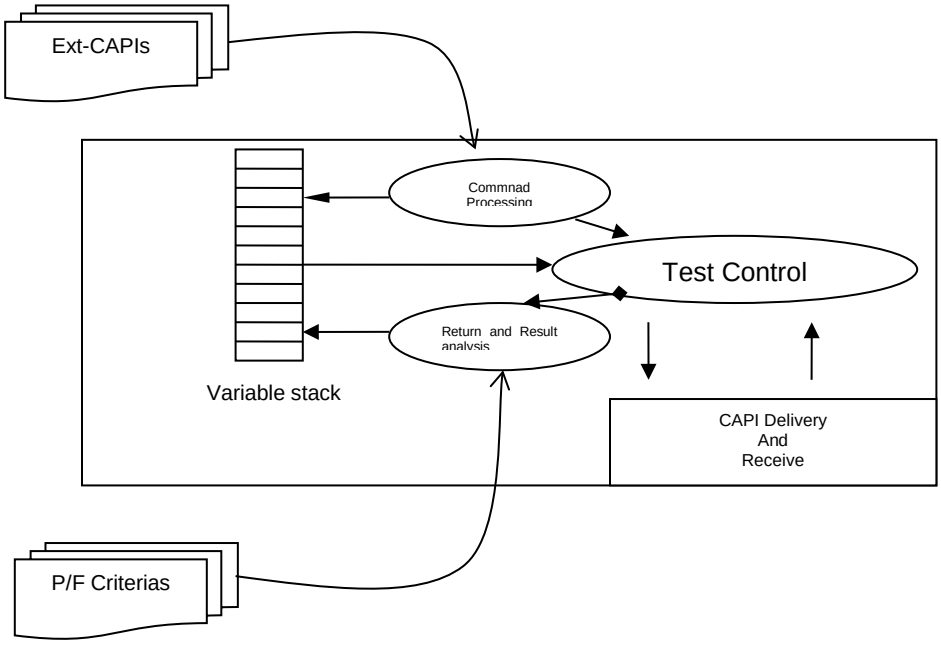

Figure 2. UCC standard configuration

The architecture gives flexible extendibility 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 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
```

7.2 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
```

7.3 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: reinvoke_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
```

7.4 STA_ACCEPT_P2P_INVITATION_REQ

This command is used to configure the STA to accept the P2P Invitation Request.

This command is optional if the device does not need manual intervention to accept the P2P invitation request.

If the command is implemented, it shall prepare the STA to accept the P2P Invitation Request and return immediately after that. It should not block waiting for P2P Invitation 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 example Listen) after it discovers the required P2P device.

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 of the sender of P2P Invitation request	String
Reinvoke	Indicate if the invitation is for reinvoking the persistent group	Integer 0 = No 1 = Yes

Return Values

None.

Example

```
UCC:
sta_accept_p2p_inviation_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
```

7.5 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
```

7.6 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,cl
ientCertificate,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
```

7.7 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
Network_mode	Network mode to set	String BSS
SSID	Service set identifier of BSS or ESS	String
WPS	Indicates if WPS shall be used in association	String WPS
Channel	Channel number of the target BSS Note: This parameter is only for CTT/test bed devices	Integer

Return Values

None.

Example

```
UCC: sta_associate,interface,interfaceId_1,ssid,MyNetwork,wps,1
CA: status,RUNNING
CA: status,COMPLETE

UCC: sta_associate,interface,interfaceId_1,ssid,MyNetwork,Channel,11
CA: status,RUNNING
CA: status,COMPLETE
```

7.8 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
```

7.9 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,
```

7.10 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
Maintain_profile	Maintain the network profile and the associated security keys on disconnect when the value is set to 1	Integer 0 1

Return Values

None.

Examples

```
UCC: sta_disconnect, interface,wlan
CA:status,RUNNING
CA:status,COMPLETE
```

7.11 STA_EXEC_ACTION

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

For Location, the command is used to configure a station/device to send an initial FTM request (iFTMR) with the specified parameters to the AP within 30 seconds after receiving the configuration command.

Parameters

Name	Description	Value
ASAP	Configure ASAP field for the request	Integer 0 = Non ASAP 1 = ASAP
AskForLCI	Requests the LCI of the AP	Integer 0 = Do not include 1 = Include
AskForLocCivic	Requests the Civic Location of the AP	Integer 0 = Do not include 1 = Include
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
BurstDuration	Configures the Burst Duration field	Integer 2 – 11 15
BurstsExponent	Configure the Bursts Exponent field	Integer 0 - 15
DestMac	Destination MAC address of the AP in aa:bb:cc:dd:ee:ff format	String
FormatBwFTM	Configures the Format and Bandwidth field	Integer For unassociated state: 9–13 For associated state: 0 9–13
FTMsperBurst	Configures the FTM's per burst field	Integer 0 – 31
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:

Name	Description	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 LOC
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 DataRequest DataResponse DataEnd RangeRequest CancelRange SchedUpdate
	Mandatory Parameters for MethodType = Publish	
	Name / **Description**	**Value**
	PublishType / Configures publish type or cancels publish	String Unsolicited Solicited Cancel
	ServiceName / Name of the service	String
	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.	String One or more colon separated strings

Name	Description		Value
		Wildcard value is represented with an asterisk (*) with a length of zero.	
	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
	DataPathFlag	Configures the publish message with or without data path required in the publish message	String enable disable
	DataPathType	Configures the data path type in publish message	String NDP NMSG
	DataPathSecurity	Configures security for the data path in the publish message	String open secure
	MulticastType	Only present if DataPathType = 1, configures type of multicast	Integer 0 = one-to-many 1 = many-to-many
	QoS	Configures if QoS requirements are included in the publish message. Applicable to both testbed and DUT.	Integer 1 = QoS requirements present
	AwakeDWInt	Configures the Awake DW Interval in the publish message, indicates the interval between two DW's when the supporting service is awake to tx or rx SDFs, in units of 512TU	Integer 1,2,4,8,16
	RangeRequired	Configure Publish message with Ranging Required or not	String enable disable
	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.	String One or more colon separated strings

Name	Description		Value
		Wildcard value is represented with an asterisk (*) with a length of zero.	
	IncludeBit	Set include bit	Integer 0 = Not included 1 = Included
	SRFType	Set SRF type	Integer 0 = MAC List 1 = Bloom Filter
	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
	AwakeDWInt	Configures the Awake DW Interval in the publish message, indicates the interval between two DW's when the supporting service is awake to tx or rx SDFs, in units of 512TU	Integer 1,2,4,8,16
	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
	Mandatory Parameters for MethodType = DataRequest		
	Name	**Description**	**Value**
	DataRequestType	Configures DataRequest type	String ndp
	RespNanMac	The MAC address for the Responder device	String EX:11:22:33:44:55:66
	DataPathSecurity	Configures security for the data path in the publish message	String open secure
	Optional Parameters for MethodType = DataRequest		
	Name	**Description**	**Value**
	IncludeImmutable	Tells test bed STA to include an immutable NDL schedule in NDP Request to DUT	Integer 1 2 = 2 immutables in 2 channels
	AvoidChannel	Tells the test bed STA to NOT use this channel during this test, as in NDP counter	Integer
	InvalidNANSchedule	Tells the test bed STA one of the invalid cases for NDP request	Integer 1. = Invalid NAN availability with overlapping time slots with different channels 2 = NDC outside NAN availability

Name	Description		Value
			3 = Immutable outside NAN availability
	maporder	Tells the test bed wich map order to use for test case 5.3.11	1 = map 1 followed by map 2 2 = map 2 followed by map 1
	QoS	Configures if QoS requirements are included in the NDP request. Applicable to testbed only	Integer 1 = QoS requirements present
	Mandatory Parameters for MethodType = DataResponse		
	Name	**Description**	**Value**
	NDLResponse	Configures the test bed how to respond to NDL requests, not part of the publish message, Auto means the normal behavior of the device, not controlled by UCC, Reject means always reject, Accept means always accept, Counter means always counter	String Auto Reject Accept Counter
	DatapathID	NAN Data path ID	Integer
	M4ResponseType	Test bed only, type of M4 message to send	String Accept Reject BadMic
	Mandatory Parameters for MethodType = DataEnd		
	Name	**Description**	**Value**
	NDPID	The NDP ID value, 1-255	Integer
	InitiatorNDI	NDP Initiator's Data Interface Address	String EX:11:22:33:44:55:66
	Optional Parameters for MethodType = DataEnd		
	Name	**Description**	**Value**
	Security	Tells test bed device to send the data terminate in clear after channel has been encrypted	String Open
	Mandatory Parameters for MethodType = RangeRequest		
	Name	**Description**	**Value**
	Range_Report	Device need to include range report in Range request message	String Open
	Mandatory Parameters for MethodType = SchedUpdate		
	Name	**Description**	**Value**
	type	For unaligned schedule, it is notification only. For aligned schedule update, specify if it is negotiation or notification	String ULWnotify NDLnegotiate NDLnotify
	Optional Parameters for MethodType = SchedUpdate		
	Name	**Description**	**Value**
	ChannelAvailability	Indicating if channel is available or not during ULWs, When this parameter is not present, it means device is not available on any channels during ULWs.	Integer 1 = available 0 = not available

Name	Description		Value
	ResponderNMI	Responder's NMI for SchedUpdate type	String EX:11:22:33:44:55:66
Trigger	Triggers the specified action on the device		String FTMsession ANQPQuery

Return Values

Name	Description	Value
ErrMsg	Message describing the reason for the configuration error. The message should not exceed 200 bytes in length.	String
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
```

Example for successful executionof for Location

```
UCC: sta_exec_action,prog,loc,interface,wlan0,destmac,AA:BB:CC:DD:EE:FF, trigger,FTMsession,
BurstsExponent,0,asap,0,FormatBwFTM,13,askForLocCivic,0,askForLCI,0
AP: status,RUNNING
AP: status,COMPLETE
```

Example for error condition for Location

```
UCC: sta_exec_action,prog,loc,interface,wlan0,destmac,AA:BB:CC:DD:EE:FF, trigger,FTMsession,
BurstsExponent,0,asap,0,FormatBwFTM,9,askForLocCivic,0,askForLCI,0
AP: status,RUNNING
AP: status,ERROR,errmsg,Unable to configure FTM format and BW for sending iFTMR
```

Example publish with NAN R2 Data Path setup unicast with open security

```
UCC: sta_exec_action,interface,wlan0,prog,nan,servicename,org.wi-
fi.nan.test,methodtype,publish,publishtype,solicited,datapathflag,enable,datapathtype,ndp,datapaths
ecurity,open
CA:status,RUNNING
CA:status,COMPLETE
```

Example DataRequest to start NDP:

```
UCC: sta_exec_action,interface,wlan0,prog,nan,servicename,org.wi-
fi.nan.test,methodtype,datarequest,datarequesttype,ndp
CA:status,RUNNING
CA:status,COMPLETE
```

Example DataResponse to reply to DataRequest:

```
sta_exec_action,interface,wlan0,prog,nan,methodtype,dataresponse,ndlresponse,accpet
CA:status,RUNNING
CA:status,COMPLETE
```

Example DataEnd to terminate NAN R2 data path:

```
UCC:
sta_exec_action,interface,wlan0,prog,nan,methodtype,dataend,ndpid,39,initiatorndi,11:22:33:44:55:66
CA:status,RUNNING
CA:status,COMPLETE
```

Example setting NAN Availability attribute

```
UCC:
sta_exec_action,interface,wlan0,program,nan,nanavailability,0x12120000010e000260180004fcffff7f1151f
f0700
CA: status,RUNNING
CA: status,COMPLETE
```

Example DataRequest to start NDP with an avoid channel

```
UCC:
sta_exec_action,interface,wlan0,prog,nan,methodtype,datarequest,datarequesttype,ndp,avoidchannel,11
CA:status,RUNNING
CA:status,COMPLETE
```

7.12 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 encrypton 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
```

7.13 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
```

7.14 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
	NwCfgPIN	PIN that ASP generates to enter on the peer device	String
SessionStatus	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,000000b,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
```

7.15 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 optiona 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 Progrm 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
```

7.16 STA_GET_INFO

This command is used to returns 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
```

7.17 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
```

7.18 STA_GET_MAC_ADDRESS

This command is used to returns 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
```

7.19 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
```

7.20 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
```

7.21 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	AID: the AID assigned to the STA by the PCP/AP DiscoveredDevList: Discovered devices DeviceID or MAC address are framed in space separated list and return MasterRank: retrieve the Master Rank value from the NAN device MasterPref: retrieve the Master Preference value from the NAN device RandFactor: retrieve the random factor value from the NAN device HopCount: retrieve the Hop Count value from the NAN device BeaconTransTime: retrieve the Beacom Transmission Time value from the NAN device ListenChannel: P2P Listen channel of the device NANStatus: retrieve the NAN status (on/off) from the NAN device 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. 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. P2P_Result: P2P group formation result GroupID: P2P Group ID NDPChannel: final NAN NDP channels negotiated by devices SchedUpdateChannel: One future schedule update negotiation channel. Applicable testbed only	String AID DiscoveredDevList MasterRank MasterPref RandFactor HopCount BeaconTransTime ListenChannel NANStatus OpenPorts[1] WSCPIN P2P_Result[1] GroupID[1] NDPChannel SchedUpdateChannel
PeerIP	Peer IP address to find the open ports	String
PortList	Space separated list of ports. 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.	
Program	Program name	String WFD WFDS NAN P2PNFC VHT 60GHz DisplayR2
Notes: 1. This value is only for a test bed device and will block until the device has completed Group formation or had valid results.		

Return Values

Name	Description	Value
AID	The hex value of the AID,	Hex (with 0x)
DeviceList	Return the discovered devices DeviceID or MAC address in a space separated list. DeviceID is only applicable to the WFD program.	String

Name	Description	Value
	For WFDS, this the return value from DiscoveredDevList.	
MasterRank	Return the Master Rank value from the NAN device	Hex value 0x00 – 0xFF
MasterPref	Return the Master Preference value from the NAN device	Hex 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
NDPChannel	Return the final negotiated NAN channels, up to 2, as a comma separted list	Integer
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
SchedUpdateChannel	Return one future schedule update NAN channel by testbed	Interger
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,wfds,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
```

NAN get hop count example

```
UCC: sta_get_parameter,interface,wlan0,program,nan,parameter,hopcount
CA: status, running
CA: status,COMPLETE,hopcount,0x02
```

Get final negotiated NDP channels (2)

```
sta_get_parameter,interface,wlan0,program,nan,parameter,ndpchannel
CA: status, running
CA: status,COMPLETE,ndpchannel,6,ndpchannel,149
```

7.22 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 identiffication	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
```

7.23 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 identiffication	String
BSSID	Basic service set identiffication	String

Examples

```
UCC: sta_hs2_associate, interface,wlan0
CA:status,RUNNING
CA:status,COMPLETE,SSID,HotspotAP1,BSSID,00:11:22:33:44:55
```

7.24 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	Describes the type of primitive being invoked. This parameter is only mandatory if Command_Type = Primitive. Advertise will wait for an AdvertiseStatus event and returns the Adv_ID. Seek is not a blocking call, it may return immediately even without a SearchResult event. 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 PROVISON with when ASP triggers the SessionConfigRequestEvent to service. 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.	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 seeked.	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

Primitive_Type	Name	Description	Value
	MAC_address	MAC Address (P2P Device Address)	String For example: 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

Primitive_Type	Name	Description	Value
			Ex: 0000000A
ConfirmSession	Network_Config_PIN	WSC PIN value	String
ConnectSession	SSID	SSID for WFDS devices if it starts AutoGO	String
	Oper_Chn	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
```

7.25 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
```

7.26 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	Determines the network role for the station	Integer 0 = Don't care 1 = Shall be group owner All other values are undefined and reserved.
Prog	Program	String ASP2 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

Name	Description	Value
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
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	A checksum is returned only when Service_Name=Send. 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,30d3c5cd73d80c8ab9110a1
c41cf9e77

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,30d3c5cd73d80c8ab9110a1
c41cf9e77
```

7.27 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,
```

7.28 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
```

7.29 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
```

7.30 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
```

7.31 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 returnd 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
```

7.32 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
```

7.33 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
ANPQ	Enable or disable ANQP query	Integer 0 = Disable 1 = Enable
Assoc_Disallow	This parameter is only used for the MBO test bed. Enable: Station will follow implementation while the AP advertises assoc disallowed Disable: Station proceed sending association request while AP advertises assoc disallowed	String Enable Disable
AvChange	Enable or disable Explicit AV format change feature	String Enable Disable
BSS_Transition	Reject or Accept BSS transition request, reason code is implentation specific	String Reject Accept
BTMsupt	Enable or disable BTM support.	String Enable Disable
Cellular_Data_Cap	Sets the STA's cellular data capability to the given value	Integer
Ch_Op_Class	Sets the preferred channel operating class	Integer
Ch_Pref	Sets the preferred channel number preference value. Clear: Removes the channel preference	String 0 1 255 Clear
Ch_Pref_Num	Sets the preferred channel number	Integer
Ch_Reason_Code	Sets the non-Preferred channel reason code	Integer 0,1,2

Name	Description	Value
ConnectionCapabilityInfo	GO or Client connection capability	String GO CLI NewGO New CliGO
DeviceDiscoverability	Enable or disable the Device Discoverability feature	String Enable Disable
DMS	Enables/disables directed multicast service (DMS).	Integer 0 = Disabled 1 = Enabled
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
FileType	Specifies the type of file to be fetched PPSMO – Per Provider Subscription Management Object CERT – Certificate Credentials (includes Certificate and associated Key file) 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.	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.	String

Name	Description	Value
	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.	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
Interworking	Enable or disable interworking element	Integer 0 = Disable 1 = Enable
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
MaxTCPSegmentSize	Set the Maximum TCP Segment Size	String For example: Min Max A value, for example, 1200
mDNS_Disc	Configures the mDNS Infrastructure Discovery functionality.	String Enable Disable
mDNS_Role	Configures the mDNS role for a device. • Browser - acts as an mDNS seeker • Responder - acts as an mDNS advertiser	String Browser Responder
MNCLength	Number of digits comprising MNC.	Integer 2 3
Mode	Wireless operating mode	String

Name	Description	Value
		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_Chn	Operating channel, any valid Wi-Fi channel number For DPP, this is the channel number for STA to associate with AP after completion of DPP protocol exchange	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 HS2 HS2-R2 WFDS WPS 60GHz LOC DisplayR2

Name	Description	Value
		IoTLP MBO
RadioMsnt	Enable/disable radio measurement report functionality	Integer 0 = Disable LCI and CivicLOC 1 = Enable LCI 2 = Enable CivicLOC 3 = Enable LCI and CivicLOC
Reset	Preset 11n parameters	String 11n
RMEnabledCapBitmap	Provides a bitmap containing a semicolon separated list of bit position:value (colon separated) in the RM enabled capabilities field that indicates the corresponding capability is enabled or disabled. **Example**: 12:0;34:1 is interpreted as: LCI measurement capability enabled bit is set to disabled FTM range report capability enabled bit is set to enabled	String 12 = LCI measurement capability enabled 3 = Repeated measurements capability enabled 34 = FTM range report capability enabled
Roaming	Enable/Disable Station Romaing.	String Enable Disable
RSN_IE	In WPS 2.0, if enabled, testbed STA will include RSN IE in association request	String Enable Disable
RTS	RTS threshold	Short integer
SessionAvailability	Set the session availability bit to available or not available	Integer 0 = Not Available 1 = Available
SessionState	Expected state of a WFD session, when an action is required to be performed as specified in CTT_sequence<#>. See the table below for more information.	String PAUSE PLAY
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	String Default HiLoMac ExistLink APProhibit WeakSecurity

Name	Description	Value
	APProhibit – STA should send TDLS setup requirement even if AP prohibits TDLS WeakSecurity – STA should enable all the weak security modes (open/WEP/WPA)	
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
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

Name	Description	Value
WSCState	WSC state of the device set to unconfigure.	String unconfigure

Parameters for CTT support only

Name	Description		Value
CTT_Sequence<#>	Sequence of action numbers to take Where #: Integer beween 1 to 20 incrementally assigned by the scripts Format: CTT_Sequence#,<key name>:<value>;<key name 2>:<value 2> Colon (:) is used to separate between key and value, while semi colon (;) is used to separate key/value pairs.		String
	Key Name	**Description**	**Value**
	CTT_Action	CTT Action to take	String SKIP SEND DELAY
	CTT_AudioFormat	Audio Format	String LPCM_2 LPCM_4 LPCM_6 ALL NONE
	CTT_AuxCodec	Codec used to encode and decode the content in auxiliary stream	String PNG JPEG H264
	CTT_AuxOverlay	Maximium number of overlay layers a sink device can handle at a time	Integer
	CTT_Count	Number of times to run the sequence	Integer
	CTT_CodecProfile	Codec and profile. The format is H264-<profile> and H265-<profile>	String Example: H264-0 means CBP
	CTT_ExcludeParam	Exclude parameters	String For Example: wfd_av_format_change_timing
	CTT_Message	Type of RTSP message	String For Example: RTSP_M5_REQ_PAUSE RTSP_M5_RESP
	CTT_NonTransSupport	Non transcoding support	Integer 0 = Disabled 1 = Enabled
	CTT_RandomLength	Send RTSP message that includes random length	Integer 0 = Disabled 1 = Enabled
	CTT_RandomText	Send RTSP message that includes random text	Integer 0 = Disabled 1 = Enabled
	CTT_TCPSYN	TCP SYN packet to recover the TCP connection	String

			Yes No
	CTT_Time	Amount of time to delay	Integer
	CTT_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 WFD_CEA-<index> WFD_VESA-<index> WFD_HH-<index> WFD_ALL-<index> WFD2_CEA-<index> WFD2_VESA-<index> WFD2_HH-<index> WFD2_ALL-<index> NONE

Return Values

None.

Examples

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,11nl
```

Example 3 WFD:

```
UCC: sta_preset_testparameters,interface,ath0,program,wfd,wfddevtype,P-Sink
CA: status,RUNNING
CA: status,COMPLETE
```

Example 4 P2P-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
```

Example 9 Location:

```
UCC: sta_preset_testparameters,interface,ath0,program,LOC, RMEnabledCapBitmap,3:0;34:1
CA: status,RUNNING
CA: status,COMPLETE
```

Example 10 Configure the CTT to not transmit M3 response:

```
CCG: sta_preset_testparameters,program,DisplayR2,ctt_sequence1,Message:RTSP_M3_RESP;
CTT_Action:Skip
CA: status,RUNNING
CA: status,COMPLETE
```

Example 11 Configure the CTT to skip M6 request after transmitting M5 response:

```
CCG: sta_preset_testparameters,program,DisplayR2,ctt_sequence1,CTT_Message:RTSP_M6_REQ;
CTT_Action:Skip
CA: status,RUNNING
CA: status,COMPLETE
```

Example 12 Configure the CTT to transmit M9 request in response to M5 request with trigger teardown:

```
CCG: sta_preset_testparameters,program,DisplayR2,ctt_sequence1,CTT_Message:RTSP_M8_REQ;
CTT_Action:Skip,ctt_sequence2,CTT_Message:RTSP_M9_REQ; CTT_Action:Send
CA: status,RUNNING
CA: status,COMPLETE
```

Example 13 Configure the CTT to delay transmission of M6 request by more than 6 seconds:

```
CCG:sta_preset_testparameters,program,DisplayR2,ctt_sequence1,CTT_Message:RTSP_M6_REQ;
CTT_Action:Delay; CTT_Time:6
CA: status,RUNNING
CA: status,COMPLETE
```

Example 14 Configure the CTT to transmit multiple M5 trigger PAUSE:

```
CCG: sta_preset_testparameters,program,DisplayR2,ctt_sequence1,CTT_Message:RTSP_M5_REQ;
CTT_Action:Delay; CTT_Count:6
CA: status,RUNNING
CA: status,COMPLETE
```

Example 15 Configure the CTT to include wfd-video-formats in M4 request message:

```
CCG: sta_preset_testparameters,program,DisplayR2,ctt_sequence1,CTT_Message:RTSP_M4_REQ;
CTT_Action:Send; CTT_VideoFormat:CEA-0
CA: status,RUNNING
CA: status,COMPLETE
```

Example 16 Codec Change: Send M4 SET_PARAMETER Request to pick all combinations and profiles.

```
CCG: sta_preset_testparameters,program,DisplayR2,ctt_sequence1,
CTT_Message:RTSP_M4_REQ_SET_PARAMETER; CTT_Action:Send; CTT_VideoFormat:All
CA: status,RUNNING
CA: status,COMPLETE
```

Example 17 WFD R2:

```
UCC: sta_preset_testparameters,interface,ath0,program,DisplayR2,wfddevtype,P-
Sink,mDNS_Disc,enable,mDNS_Role,responder
CA: status,RUNNING
CA: status,COMPLETE
```

Example 18 MBO:

```
UCC: sta_preset_testparameters,interface,wlan0,program,MBO, Cellular_Data_Cap,1
CA: status,RUNNING
CA: status,COMPLETE
```

7.34 STA_REASSOC

This command is used to re-associate the STA with a given AP. The previously added profile shall be used if a new profile is not specified.

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
Channel	This parameter is only for CTT/Test bed devices. Used to specify the channel number of the target BSS.	Integer

Return Values

None.

Examples

```
UCC: sta_reassoc, interface,wlan,bssid,aa:bb:cc:dd:ee:ff
CA:status,RUNNING
CA:status,COMPLETE

UCC: sta_associate,interface,interfaceId_1,ssid,MyNetwork,Channel,11
CA: status,RUNNING
CA: status,COMPLETE
```

7.35 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
```

7.36 STA_RESET_DEFAULT

This command is used to reset the station to its default program specific configuration, as well as remove any cached profiles, keys and credentials.

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 P2P
Interface	Interface ID	String
Prog/Program	Program specific settings to be enabled Enables the program functionality based on the specified String value. Notes applicable only to DPP: 1. A device shall delete all its configuration objects/connectors that it received as an Enrollee. 2. A Configurator may only change its Configurator signing key when executing this command.	String PMF WFD1 VHT HS2 HS2-R2[2] WFDS[3] NAN WMMPS TDLS P2P 60GHz WPS LOC[4] IoTLP TM DPP WPA3
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 are the 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 are the same as OOB and the interface state is idle as defined in Wi-Fi Direct.
4. For Location, refer to the Test Plan for the STA default configuration.

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

CCG: sta_reset_default,interface,wlan0,prog,IoTLP
CA:status,RUNNING
CA:status,COMPLETE

CCG: sta_reset_default,interface,wlan0,prog,TM,DevRole,P2P
CA:status,RUNNING
CA:status,COMPLETE
```

7.37 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
```

7.38 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
```

7.39 STA_SCAN_BSS

This command is used to initiates active scanning on the provided BSSID. The command shall return the channel and the SSID information.

Dependencies

Commands	Description
AP_GET_MAC_ADDRESS	Obtain MAC address of the wireless interface
DEVICE_LIST_INTERFACES	Obtain a list of network interfaces from the device

Parameters

Name	Description	Value
BSSID	Basic service set identification	String
Interface	Interface ID	String

Return Value

Name	Description	Value
BSSChannel	Channel number on which AP is beaconing	Integer
SSID	Service set identification	String

Examples

```
UCC: sta_scan_bss,interface,wlan0,bssid,11:22:33:44:55:66
CA:status,RUNNING
CA:status,COMPLETE,ssid,DPPNET01,bsschannel,6
```

7.40 STA_SEND_ADDBA

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

Parameters

<table>
<tr><th>Name</th><th>Description</th><th>Value</th></tr>
<tr><td>Dest_MAC</td><td>Destination MAC address
Example: aa:bb:cc:dd:ee:ff</td><td>String</td></tr>
<tr><td>Interface</td><td>Interface ID</td><td>String</td></tr>
<tr><td>TID</td><td>Traffic identifier
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.
<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></td><td>Short integer
0-15</td></tr>
</table>

Return Values

None.

Examples

```
UCC: sta_send_addba, 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 exampleListen) 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_inviation_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,p2pdevid,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

Name	Description	Value
		1 = Static 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,
     key1,1234567890123456789012345
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	Informs the station the type of device discovery. It can use P2P device discory 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. If DiscoverType is set for WFD, it can use any discovery mechanisms and it is implementation specific.	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	Number of absence intervals 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 adeversiting the NoA.	Integer
NoA_Duration	Duration of the Absence in TU units	Integer
NoA_Interval	Interval between two absences in TU units If NoA Interval is specified as zero, then the STA shall configure the NoA Interval equal to the Beacon Interval.	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

Name	Description	Value
		1 = Yes
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 IoTLP

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 or configure 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 CBAP Only subfield is set to 0 when the allocation of the DTI portion of the beacon interval is provided through the Extended Schedule element	Integer 1 0
Ch_Op_Class	Preferred channel operating class	Integer
Ch_Pref_Num	Preferred channel number	integer
Ch_Pref	Preferred channel number preference Clear: Remove the existing channel preferenceap	String 0 1 255 Clear
Ch_Reason_Code	Non-Preferred channel reason code	Integer 0,1,2
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 disabledand 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
FrameName	CTT only parameter. When used in conjunction with KeepAlive as mgmt, will enable usage of SAQuery for PMF networks as a KeepAlive.	String Default SAQuery
Interface	Interface ID	String

Name	Description	Value
KeepAlive	CTT only parameter. When set to one of the options, will then only allow use of that type of frame as a KeepAlive frame.	String Data Mgmt PSpoll None
MCS_FixedRate	Fixed MCS rate	Short integer 0-31
Mgmt_Data_TX_Resp_Frame	CTT only parameter. When enabled, the CTT will transmit management and data frames or responses. When disabled, the CTT will not transmit management and data frames or responses.	String Enable Disable
Nebor_BSSID	BSSID of neighbor AP	String Mac seprated by ':'
Nebor_Op_Class	Operating Class of the neighbor AP	Integer
Nebor_Op_Ch	Channel Number of the neighbor AP	Integer
Nebor_Pref	Preference value of Neigbor AP for BSS transition candidate list (Nebor_BSSID)	Integer
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
Prog	Program	String PMF TDLS VHT 60GHz IoTLP MBO
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

Name	Description	Value
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 resquest 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,P
ercentBI,5,SrcAID,0xff,DestAID,0xff, AllocID,2,PercentBI,16,SrcAID,0xff, DestAID,0xff,
AllocID,3,PercentBI,16,SrcAID,0xff,DestAID,0xff
CA:status,COMPLETE

UCC: sta_set_rfeature, interface,wlan0,prog,IoTLP,IoTLP_No_Tx_Frame,1
CA:status,COMPLETE

UCC: will set one neighbour AP in list
sta_set_rfeature, interface,wlan0,prog,MBO, Nebor_BSSID, 00:01:02:03:04:05, Nebor_Op_Class,<class>,
Nebor_Op_Ch,<channel>,Nebor_Pref,<PreferenceValue>
CA:status,RUNNING
CA:status,COMPLETE
```

7.64 STA_SET_SECURITY

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

Note. When WPA3(SuiteB/OWE/SAE) is set as the key management type, PMF shall be set to Required as default.

Parameters

Name	Description	Value
CertType	Client certificate type (i.e. Suite B)	String RSA ECC
ECGroupID	Specifies the (EC)DH group ID A list of group IDs may be specifcified as a space separated list	Integer Examples: 19 20 21 19 20 21
EncpType	Encryption type Note. The encryption type will set both pairwise and group cipher.	String TKIP AES-CCMP AES-GCMP
GroupCipher	Group Cipher type A list of multiple ciphers may be specifcified as a space separated list.	String AES-GCMP-256 AES-CCMP-256 AES-GCMP-128 AES-CCMP-128
GroupMgntCipher	Group Management Cipher type A list of multiple ciphers may be specifcified as a space separated list.	String BIP-GMAC-256 BIP-CMAC-256 BIP-GMAC-128 BIP-CMAC-128
Interface	Interface ID	String
InvalidSAEElement	Instructs the test bed to send an SAE Commit message with an invalid element. This parameter should only be used for negative tests. (hex dump)	String 00xxxxxxxxx
KeyMgmtType	Key management type	String WPA WPA2 WPA2-FT WPA2-PMF WPA2-WPA-ENT SuiteB OWE
Netowrk_Mode	Network mode to set	String BSS
OWE	Enables/disables OWE capability on the test bed device	String Enable Disable
PairwiseCipher	Specifies the pairwise cipher. A list of multiple ciphers may be specifcified as a space separated list	String AES-GCMP-256 AES-CCMP-256

Name	Description	Value
		AES-GCMP-128 AES-CCMP-128
Passphrase	InterfaceID for PSK, or encryption key for 60GHz	String
PMF	Configures the PMF settings	String Required Optional Disable (Default)
SSID	Service set identification	String
Type	Security type Notes: PSK-SAE: Transition compatibility mode where PSK and SAE are enabled	String PSK EAPTLS EAPTTLS EAPPEAP EAPSIM EAPFAST EAPAKA OPEN SAE PSK-SAE
TLSCipher	Suite B TLS Cipher types	String TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 Note: The value below is only for negative test cases. TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

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

UCC:
sta_set_security,type,SAE,interface,interfaceId_1,ssid,MyNetwork,KeyMgmtType,WPA2,EncpType,AES-
CCMP,passphrase,12345678
CA: status,RUNNING
CA: status,COMPLETE

UCC: sta_set_security,type,eaptls,interface interfaceId_1,ssid
MyNetwork,usernametest,trustedRootCA,Azimuth CA,clientCertificate,console-
cert,keymgmtType,SuiteB,PairwiseCipher,AES-GCMP-256,GroupCipher,AES-GCMP-256,GroupMgntCipher,BIP-
GCMP-256
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 maxium 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

Name	Description	Value
	A STA device with DiscoveryMode set to1 should not initiate active scan until the device receives the STA_SCAN command.	STA P2P
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. **Example**s – 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 TM
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

Name	Description	Value
RTS_Width	Sets the bandwidth in the RTS request initiated by the DUT	String 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
TM_Support	Sets or resets Extendend capability (bit 23) in the Timing Measurement Action frame	String 1= Enable 0 = Disable
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 bandwith 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

Name	Description	Value
WPS4PIN	Enable or disable the generation of 4-digit PIN	String Enable Disable
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
Dialog_Token	The Dialog_Token field is used for matching action responses with action requests when there are multiple, concurrent action requests. The length of the Dialog_Token field is 1 octet.	Short Integer 1-255
Direction		String uplink downlink bidi
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

Name	Description	Value
		Legacy UAPSD
SBA		Double or Short Integer 1.0 – 8.0 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 negation/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	If P2P is used as the preferred connection, this parameter indicates if the STA should initiate the GO Negotiation Request. 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. 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. If TDLS is used as the preferred connection, this parameter indicates if the STA should initiate a TDLS connection. 0 = The command shall prepare the device for TDLS and then return immediately without the SessionID (implementation specific). 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. If R2ConnectionType is supplied and has the value "Infrastructure", this parameter indicates if the STA should initiate the mDNS query. 0 = The command will prepare the STA to accept the mDNS query and then return immediately without waiting for the end of session establishement. The command will not initiate mDNS query. The mDNS query will be sent from another STA. The command will return the Result, GroupID, and WFDSessionID will be null. 1 = The STA will initiate mDNS query acting as browser (seeker) and block through the end of WFD session establishment. The command will return the Result, GroupID and WFDSessionID.	Integer Valid values: 0 = Disabled 1 = Enabled
InstanceName	Instance name of the service used to perform mDNS discovery and establish a TCP connection	String
Intent_Val	Intent value for becoming group owner If Intent value is anything other than 0-15 then station shall use its default value.	Integer
Interface	Interface ID	String
Oper_Chn	Operating channel	Integer
PeerAddress	DeviceID for Wi-Fi direct or Station MAC address for TDLS. 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.	String
R2ConnectionType	Specifies the type of connection used for WFD Session establishment	String P2P Infrastructure
ServiceType	Service type used to perform PTR lookup in the mDNS Query	String _display._tcp _displaysrc._tcp

Return Values

Name	Description	Value
Result	Result of GO Negotiation.	String

Name	Description	Value
	'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. The response will be NULL if TDLS is used.	GO CLIENT FAIL
GroupID	Wi-Fi Direct Group ID The format is space separated P2PDevID and ssid <P2PDevID ssid>	String or NULL if TDLS is used
WFDSessionID	WFD session ID. Will be set to NULL if the WFD session is not established because of TDLS setup failure or P2P group formation failure. Will be set to '1' if the WFD session is not available on the peer device.	String NULL 1.20

Examples

Example 1:

```
UCC:start_wfd_connection,interface,wlan,init_wfd,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
```

Example 2: Supplied to mDNS Browser

```
UCC:start_wfd_connection,interface,wlan,init_wfd,1,intent_val,15,peeraddress,00:01:22:33:44:55,R2Co
nnectionType,Infrastructure,ServiceType,_display._tcp,InstanceName,XYZ-Sink
CA:status,RUNNING
CA:status,COMPLETE,result,GO, GroupID,00:12:13:14:15:16-DIRECT-CA-LAB, WFDSessionId,98765432abcd
```

Example 3: Supplied to mDNS Responder

```
UCC:start_wfd_connection,interface,wlan,init_wfd,0,intent_val,15,peeraddress,00:01:22:33:44:55,R2Co
nnectionType,Infrastructure
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.

For a STA, this command is a blocking call and should return after successful WPS provisioning, i.e after the M8 message.

For an AP, this command is non-blocking. The AP shall return immediately after initiating WPS registration.

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
Band	A RF band to perform PBC.	String 24g 5gl 5gh
Interface	Interface ID	String
Name	A name such as AP1	String
SSID	SSID	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	This is required for STAs only. 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. Failure: Unsuccessfully completed the registration process.	String InSession OverlapSession NoPeer Successful Failure

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 atraffic_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 Acess 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, sta_mac_address,aa:bb:cc:dd:ee:ff
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 24G 50G
Name	Name such as ABC 11n AP	String
WLAN_TAG	Used to differentiate between multiple BSSs on an AP. For example: WLAN_TAG value 1 is for the first BSS. WLAN_TAG 2 is for the second BSS. If not present, WLAN_TAG should be set to 1. An AP returns the MAC address corresponding to the WLAN_TAG value.	Integer

Return Values

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

Example

```
UCC: ap_get_mac_address,interface,5G
CA: status,RUNNING
CA: status,COMPLETE,mac,11:22:33:44:55:66


UCC: ap_get_mac_address,WLAN_TAG,1,interface,24G
CA: status,RUNNING
CA: status,COMPLETE,mac,11:22:33:44:55:66

UCC: ap_get_mac_address,WLAN_TAG,2,interface,5G
CA: status,RUNNING
CA: status,COMPLETE,mac,00:12:23:34:45:56
```

8.6 AP_GET_PARAMETER

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

Parameters

Name	Description	Value
Name	A name such as ABC 11n AP	String
Parameter	The parameter name whose value should be returned by a return value.	String SSID PSK

Return Values

Name	Description	Value
PSK	Passphrase	String
SSID	A SSID to the security profile to distinguish it from others if needed	String

Example

```
UCC: ap_get_parameter,NAME,abc,parameter,SSID
AP: status,RUNNING
AP: status,COMPLETE,SSID,wifi

UCC: ap_get_parameter,NAME,abc,parameter,PSK
AP: status,RUNNING
AP: status,COMPLETE,PSK,1234567890
```

8.7 AP_PRESET_TESTPARAMETERS

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

Parameters

Name	Description	Value
Name	Name of the AP	String
Oper_Chn	Operating channel, any valid Wi-Fi channel number (channel in which an AP beacons with a default SSID, before any DPP protocol exchange) Applicability?	Integer
Conf_Chn	Configuration channel, any valid Wi-Fi channel number (channel in which an AP beacons after being configured with DPP) Applicable only to test bed	Integer
Program	Program name	String DPP

Return Values

None.

Example

```
UCC: ap_preset_testparameters,program,DPP,NAME,xyz,oper_chn,6
AP: status,RUNNING
AP: status,COMPLETE

UCC: ap_preset_testparameters,program,DPP,NAME,xyz,conf_chn,11
AP: status,RUNNING
AP: status,COMPLETE
```

8.8 AP_RESET_DEFAULT

This command is used to reset the AP to its default program specific configuration, as well as remove any cached profiles, keys and credentials.

Parameters

Name	Description	Value
Name	Name such as ABC 11n AP	String
Program	Program name Applicable only to DPP: 1. A device shall delete all its configuration objects/connectors that it received as an Enrollee. 2. A Configurator may only change its Configurator signing key when executing this command.	String WPA2 WMM HS2 HS2-R2 VHT 11n 60GHz LOC[1] IoTLP TM DPP WPA3
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
Notes: 1. Refer to the Location Test Plan for the AP default configuration. As part of ap_reset_default with program=LOC and type=testbed, an AP is required to configure the LOC specific default settings and turn ON its radio.		

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

CCG: ap_reset_default,NAME,xyz-AP,program,IoTLP
```

```
AP: status,RUNNING
AP: status,COMPLETE


CCG: ap_reset_default,NAME,xyz-AP,program,TM
AP: status,RUNNING
AP: status,COMPLETE
```

8.9 AP_SEND_ADDBA_REQ

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

Parameters

<table>
<tr><th>Name</th><th>Description</th><th>Value</th></tr>
<tr><td>Name</td><td>A name such as ABC 11n AP</td><td>String</td></tr>
<tr><td>STA_MAC_Address</td><td>MAC address of station device</td><td>String
Example:
00:11:22:33:44:55</td></tr>
<tr><td>TID</td><td>Traffic identifier
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.
<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></td><td>Integer
0-15</td></tr>
</table>

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
```

8.10 AP_SEND_BCNRPT_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,2
0,MEAMODE,ACTIVE,RPTCOND,0,RPTDET,0,MEADURMAND,1,APCHANRPT,(???),REQINFO,0/48/54,
AP: status,RUNNING
AP: status,COMPLETE
```

8.11 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_MO
DE,1,DIS_ASSOC_TIME,0,VALIDITY_INT,200,CANDIDATE_LIST,WiFi1
AP: status,RUNNING
AP: status,COMPLETE
```

8.12 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
```

8.13 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
Repeatition	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
```

8.14 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
```

8.15 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
```

8.16 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,2
SS,width,20
AP:status,RUNNING
AP:status,COMPLETE
```

8.17 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
```

8.18 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
```

8.19 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
```

8.20 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
```

8.21 AP_SET_RFEATURE

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

Parameters

Name	Description	Value
AllocID	Indicates the Allocation ID used.	Integer 1-n
AllocType	Indicates the type of allocation used.	String CBAP SP
Assoc_Delay	Set the association retry delay, in seconds	Integer
Assoc_Disallow	Association disallowed	String Enable Disable
BSS_Term_TSF	To set BSS Termination TSF field duration value in seconds	Integer
BSS_Term_Duration	To set BSS Termination Duration field for which the AP is not present in minutes	Integer
BTMReq_DisAssoc_Imnt	To set Disassoc Imminent Bit	Integer 1 0
BTMReq_Term_Bit	To set Request Termination Bit	Integer 1 0
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 of the DTI portion of the beacon interval is provided through the Extended Schedule element.	Integer 1 0
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
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
Disassoc_Timer	Disassociation Timer number of tbtt	Integer
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
Name	Name of the AP	String
Nebor_BSSID	BSSID of neighbour AP	String
Nebor_Op_Class	Operating Class of the neighbor AP	Integer
Nebor_Op_Ch	Channel Number the neighbor AP is running on	Integer
Nebor_Pref	Preference value of Neigbor AP for BSS transition candidate list (Nebor_BSSID)	Integer
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

Name	Description	Value
		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)
PercentBI	Percentage of beacon interval used.	Integer 0-100
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
SrcAID	The Source AID field is set to the AID of the STA that initiates channel access during the SP or CBAP allocation.	Hex
TxBandwidth	Sets the AP transmission bandwidth	Integer 20 40 80 160
Type	Program name	String VHT MBO

Return Values

None.

Example

```
UCC: ap_set_rfeature,NAME,xyz,type,VHT,nss_mcs_opt,1;9
AP: status,RUNNING
AP: status,COMPLETE

UCC: ap_set_rfeature,NAME,xyz,type,VHT,nss_mcs_opt,1;9
AP: status,RUNNING
AP: status,COMPLETE


Example
UCC: ap_set_rfeature,NAME,xyz,type,MBO,BTMReq_DisAssoc_imnt,1
AP: status,RUNNING
AP: status,COMPLETE
```

8.22 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
```

8.23 AP_SET_SECURITY

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

Parameters

Name	Description	Value
AntiCloggingThreshold	Threshold value for Anti-Clogging token generation (For CTT devices only)	Integer
ECGroupID	Specifies the (EC)DH group ID A list of group IDs may be specifcified as a space separated list	Integer Examples: 0 (to set invalid groupID on CTT/TB only) 19 20 19 20
Encrypt	Encryption cipher configuration	String WEP TKIP AES AES-GCMP
GroupCipher	Group Cipher type A list of multiple ciphers may be specifcified as a space separated list.	String AES-GCMP-256 AES-CCMP-256 AES-GCMP-128 AES-CCMP-128
GroupMgntCipher	Group Management Cipher type A list of multiple ciphers may be specifcified as a space separated list.	String BIP-GMAC-256 BIP-CMAC-256 BIP-GMAC-128 BIP-CMAC-128
Interface	Radio hardware interface	String
InvalidSAEElement	Instructs the test bed to send an SAE Commit message with an invalid element. This parameter should only be used for negative tests. (hex dump)	String 00xxxxxxxx….
KeyMgnt	WPA2-PSK-SAE is a transition compatibility mode that supports both PSK and SAE. When WPA3(SuiteB/OWE/SAE) is set as key management type, PMF shall be set to Required as default.	String NONE WPA-PSK WPA2-PSK WPA-ENT WPA2-ENT WPA2-PSK-Mixed WPA2-Mixed WPA2-PSK-SAE SuiteB OWE SAE OSEN
Name	A name such as ABC 11n AP	String
OWE	Enables/disables OWE capability on the test bed device. This is for a CTT device only.	String Enable Disable
PairwiseCipher	Specifies the pairwise cipher. A list of multiple ciphers may be specifcified as a space separated list	String AES-GCMP-256

Name	Description	Value
		AES-CCMP-256 AES-GCMP-128 AES-CCMP-128
PMF	Robust management frame protection configuration	String Required Optional Disabled
PreAuthentication	Enables/disables Preauthentication in the AP	String Enabled Disabled
PSK	An ASCII passphrase (8..63 characters).	String
PSKHEX	A hex encoded PSK (64 characters).	String
Reflection	To reflect an SAE commit message from the DUT (For CTT devices only)	String SAE
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
WEPKey	WEP key	String
WLAN_TAG	To link KEYMGNT (security profile) to specific WLAN. For Location, the WLAN_TAG differentiates between multiple BSSs on an AP. For example: WLAN_TAG value 1 is for the first BSS. WLAN_TAG 2 is for the second BSS. If not present, WLAN_TAG should be set to 1. The AP returns the MAC address corresponding to the WLAN_TAG value.	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
```

```
CCG: ap_set_security,NAME,abc,KEYMGNT,SAE,PSK,12345678,AntiCloggingThreshold,0
AP: status,RUNNING
AP: status,COMPLETE

CCG: ap_set_security,NAME,abc,KEYMGNT,OWE,ECGroupID,19
AP: status,RUNNING
AP: status,COMPLETE

CCG: ap_set_security,NAME,abc,KEYMGNT,SAE,PSK,12345678,reflection,SAE
AP: status,RUNNING
AP: status,COMPLETE
```

8.24 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_staqos,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
```

8.25 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
ABFTLRang	Specifies the minimum value of the range. Does not specify the maxium value of the range.	String Gt1 = Greater than or equal to 1
ADDBA_Reject	Enables/disables rejecting any ADDBA request by sending ADDBA response with status “decline”	Enable Disable
AdvCoLocBSSIDs	Enable or disable advertisement of all actively beaconing BSSIDs in the Co-Located BSSID list subelement	Integer 0 = Disable 1 = Enable
AMPDU	Enables/disables AMPDU Aggregation	Enable Disable
AMSDU	Enables/disables AMSDU Aggregation	Enable Disable
ANQP	Enable or disable an ANQP query	Integer 0 = Disable 1 = Enable
BAckRcvBuf	BlockAck receive buffer size in terms of number of MPDUs.	Integer 2 4
BCNInt	Beacon Interval	String
BSS_Max_Idle	Enable or disable the BSS_max_idle feature.	String Enable Disable
BSS_Max_Idle_Period	Configures the BSS_max_idle_period in units of 1000 TUs (One TU = 1024 µS).	Integer
BW_Sgnl	Enables/disables the ability to send out RTS with bandwidth signaling	String Enable Disable
Cellular_Cap_Pref	Cellular Capability preference	Integer
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. For Location, sets the ISO 3166 country code for use with the Location Civic address. Refer to RFC 4776, Section 3.1.	String For example, ‘US’
DMS	Enables/disables directed multicast service (DMS).	Integer 0 = Disabled 1 = Enabled

Name	Description	Value
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_BSS_LIST	Set BSSID of APs belongs to same mobility domain for Fast BSS Transition test (Vendor specific)	String Space seperated BSSIDs of APs
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
FTMinBSSIDInfo	FTM subfield in BSSID Information field (bit 13)	Integer 0 1
GAS_CB_Delay	GAS Comeback Delay in Tus. This parameter only applies a test bed AP that supports 4-frame GAS exchanges.	Integer
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
Greenfield	Enables/disables HT Greenfield	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
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
InfoZ	Configures the Z subelement on the AP in a space seperated list that contains: Subelement ID Length STA floor information Height above floor Height above floor uncertainity	Integer **Example**: 0x04 06 00 40 00 38 00 0e

Name	Description	Value
Interface	A radio interface for multiple radio AP	String
Interworking	Enable or disable the interworking element	Integer 0 = Disable 1 = Enable
LCI	LCI to be configured on the AP. Value can be either 23 bytes or 1 byte in length. If the value is 23 bytes in length, set the LCI as follows: Latitude uncertainty (1 byte) Latitude (5 bytes) Longitude uncertainty (1 byte) Longitude (5 bytes) Altitude type (1 byte) Altitude uncertainty (1 byte) Altitude (4 bytes) Datum (1 byte) RegLocAgreement (1 byte) RegLocDSE (1 byte) Dependent state (1 byte) Version (1 byte) If the value is 0x00 (1 byte in length), set LCI to “Unknown”	Hex value
LDPC	Enables/disables the use of LDPC code at the physical layer for both Tx and Rx side	String Enable Disable
LocCivicAddr	Value of the Location Civic Address	Hex value
LocCivicAddrLength	Length of the Location Civic Address. A Length of zero tells the DUT to set the Location Civic to “Unknown”.	Integer
LocCivicAddrType	Type of the Location Civic Address	Integer
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

Name	Description	Value
		11na 11ng 11ac 11ad
MSDUSize	Size of each MSDU in bytes.	Integer
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 Refer to IEEE 11ac 8.3.1.20 and 9.7.6.6 for implementation.	Integer 0 1 2
MU_TxBF	Enables or disables Multi User (MU) TxBF beamformer capability with explicit feedback	String Enable Disable
MultiBurstRequest	Configures the AP to accept multiburst requests	Integer 0 = Reject 1 = Accept
Name	A name such as ABC 11n AP	String
NEIBRPT		String Enable Disable
NeighAPBSSID	BSSID of the neighbor AP. This parameter is used to identify a neighbor AP. This command also includes the neighbor AP's parameters such as PHYType, ChannelWidth etc.	String
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
OpChannel	Operating channel	Integer
P2PCoexistence	Checks for the P2PCoexistence bit	Integer 1 0
P2PMgmtBit	Set P2P Management bit	Boolean
PHYType	PHY Type	Integer 7 = HT 9 = VHT
PreAuthentication	Enables/disables Preauthentication in AP	String Enabled Disabled
Prog	Program name	String
Program	A certification program currently supported	String VHT HS2 HS2-R2

Name	Description	Value
		60GHz LOC IoTLP MBO
Proxy_ARP	Enables/disables the Proxy ARP feature.	Integer 0 = Disabled 1 = Enabled
PublicIdentifierURI-FQDN	Configures the URI/FQDN of the Location server	String held.example.com supl.example.com
Pwr_Const	AP Power Contstrain (in dBm)	Integer
Radio	Turn On/Off the radio of given interface	Boolean
RadioMsnt	Enable/Disable radio measurement report functtionality	Integer 0 = Disable LCI and CivicLOC 1 = Enable LCI 2 = Enable CivicLOC 3 = Enable LCI and CivicLOC
Reg_Domain	Sets Regulatory Domain	String Global
RIFS_Test	Enables/disables the Set Up (Tear Down) RIFS Transmission Test as instructed within Appendix H of the 11n Test Plan	Enable Disable
RMEnabledCapBitmap	Provides a bitmap containing a semicolon separated list of colon separated bit position:value in the RM enabled capabilities field that indicates the corresponding capability is enabled or disabled. **Example**: 12:0;35:1 represents: LCI measurement capability enabled bit is set to disabled Civic Location Measurement Capability Enabled bit is set to enabled	String 12 = LCI measurement capability enabled 35 = Civic Location Measurement Capability Enabled
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. For VHT, If 'MODE' = 11na, sets the Rx spacial streams of the AP which relates to the Rx MCS Rates capability.	String For 11n: 1SS – MCS 0-7 2SS – MCS 0-15 3SS – MCS 0-23

Name	Description	Value
	If 'MODE' = 11ac, sets the Rx spacial streams of the AP. No inter-dependency of number of spatial streams and MCS rates.	For 11ac: 1SS 2SS 3SS 4SS
Spatial_TX_Stream	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.	String For 11n: 1SS – MCS 0-7 2SS – MCS 0-15 3SS – MCS 0-23 For 11ac: 1SS 2SS 3SS 4SS
SpectrumMgt	Enables/disables the spectrum management feature with minimum number of beacons with switch announcement IE = 10, channel switch mode = 1	String Enable Disable
SSID	SSID	String
STBC_Rx	Enables/disables Rx STBC in the AP	Integer 0 1
STBC_TX	STBC Transmit Streams Comma separated list of the number of spatial streams and the number of space time streams. For example – 1spatial stream and 2 space time streams = 1;2	String
TDLSChswitchProhibit		String Enable Disable
TDLSProhibit		String Enable Disable
TM_Support	Sets or resets Extendend capability (bit 23) in the Timing Measurement Action frame.	String 1= Enable 0 = Disable
TxBF	Enables/disables the SU TxBF beamformer capability with explicit feedback	String Enable Disable
URI-FQDNdescriptor	Configures the URI/FQDN descriptor field	String HELD SUPL
VHT_TKIP	Enables/disables TKIP in VHT mode	String Enable Disable
VHT_WEP	Enables/disables WEP in VHT mode	String Enable Disable
WideBwChnl	Wide bandwidth channel subelement. Values (as per Draft P802.11REVmc_D4.3) supplied in a colon separated list of Channel Width, Frequency Segment 0, Frequency Segment 1 values.	String Channel Width values 0 = 20 MHz

Name	Description	Value
	In the example, 2:106:0 Channel Width = 80 MHz Frequency Segment 0 = 106 Frequency Segment 1 = 0	1 = 40 MHz 2 = 80 MHz
Width	Channel width for 11n channel bonding Used to configure the AP channel width to 160 MHz	20 40 60 80 160 Auto
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 Note: For WLAN_TAG 2, if any of SSID, Channel, Mode, Width parameters are not specified, then WLAN_TAG 2 inherit these parameter values from WLAN_TAG 1.	Integer
WME	Enables/disables WME	String On Off
WMMPS	Disable/enable APSD	String On Off
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
WscIEFragment	If enabled, test bed devices will enable to fragmentation of the WSC IE.	String Enable Disable
WscEapFragment	If enabled, test bed devices will enable fragmentation of the EAP message.	String Enable Disable
WscState	WSC state of the device set to unconfigure.	String Unconfigure
Zero_CRC	Clears the CRC field	String Enable Disable

Return Values

Name	Description	Return Value
ErrMsg	Message describing the reason for the configuration error. The message should not exceed 200 bytes in length.	String

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
```

Location examples: Configuring AP with Neighbor AP information

```
UCC: ap_set_wireless,NAME,ABC-
LOC,program,LOC,NeighAPBSSID,AA:BB:CC:DD:EE:FF,PHYType,9,FTMinBSSIDInfo,1,OpChannel,36,WideBwChnl,
2:42:0, LCI,52834d12efd2b08b9b4bf1a2b2c2d2e2,InfoZ,040600400038000e,RMEnabledCapBitmap,12:1;35:0
,CountryCode,US,LocCivicAddr,
0102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f30313
2333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f60616263
6465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f90919293949
5969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadae,LocCivicAddrType,2,LocCivicAddrLength,174
AP:status,RUNNING
AP:status,COMPLETE
UCC: ap_set_wireless,NAME,xyzLOC,CountryCode,US,LocCivicAddr,
0102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f30313
2333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f60616263
6465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f90919293949
5969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadae,LocCivicAddrType,2,LocCivicAddrLength,174,lci,
120d3649bc0312c32e6e2e01010f330b00000100000001,InfoZ,040600400038000e
AP: status,RUNNING
AP: status,COMPLETE
```

Example for error condition

```
UCC: ap_set_wireless,NAME,xyzLOC,LocCivicAddr,ffb2a6f3cdf5,
LocCivicAddrType,2,LocCivicAddrLength,174,lci,994d7f011067, InfoZ, 040640000038000e
AP: status,RUNNING
AP: status,ERROR,errmsg,Unable to configure Location Civic Address for FTM_1
```

BSS Max Idle example

```
CCG: ap_set_wireless,NAME,abc,Program,IoTLP,BSS_max_idle,Enable, BSS_max_Idle_period 5,
BSS_Idle_Protection_options,0
AP: status,RUNNING
AP: status,COMPLETE
```

8.26 AP_WPS_ENTER_PIN

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

This command selects the PIN as the configuration method as part of the WPS registration process. This command does not start the WPS registration process.

Parameters

Name	Description	Value
Band	The band in which the device should operate.	String 60GHz
Name	A name such as AP1	String
Pin	WPS PIN value	String

Return Values

None.

Example

```
UCC: ap_wps_enter_pin,name,AP1,pin,1234ABCD
CA: status,RUNNING
CA: 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_CALC_PARTIAL_TSF

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

Parameters

Name	Description	Value
TSFts	TSF time stamp	Integer
OffPTSF	Partial TSF timer to calculate offset	Integer

Return Value

Name	Description	Value
PTSF	Calcualted partial TSF timer value	Integer
Offset	Offset to burst start	Integer

Example

```
UCC: sniffer_calc_partial_tsf,tsfTs,56115423,offPtsf,56241152
Sniffer: status,RUNNING
Sniffer: status,COMPLETE,ptsf,54800,offset,125729
```

9.3 SNIFFER_CHECK_DSCV_WINDOW

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

Parameters

Name	Description	Value
Band		
Duration	Duration in seconds for the Wi-Fi Test Suite sniffer capture	Integer
FieldName	Field that is to be checked. AMI = Anchor Master Information 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 SnifferTimeStamp = Relative Tx time of the frame from the beginning of the trace	String AMI bcn_int RandFactor TSF 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 widows 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
Role		
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,dura
tion,300
Sniffer: status,RUNNING
Sniffer: status,COMPLETE,CheckResult,SUCCESS
```

9.4 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
```

9.5 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 retrigge	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,NULL,TrigDataLen,1044
Sniffer:status,RUNNING
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.6 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
```

9.7 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
```

9.8 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 transmition	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
```

9.9 SNIFFER_CHECK_P2P_OPPS_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-seoncds	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
```

9.10 SNIFFER_CHECK_P2P_OPPS_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-seoncds	Integer
DataLen	Data length of ping traffic	Integer
DestMAC	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
```

9.11 SNIFFER_CHECK_PMK_ID

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
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
```

9.12 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,d
atalen,1444
Sniffer:status,RUNNING
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.13 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:f
f,datalen,1444
Sniffer:status,RUNNING
Sniffer:status,COMPLETE, FilterStatus,SUCCESS
```

9.14 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
```

9.15 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
```

9.16 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

Name	Description	Value
3GPP_GUD	Confirms if the GUD value in 3GPP information matches the given value	Integer
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 Provdisresp P2PPresenceResp GasInitReq GasinitResp GasComebackResp DLSReq
ActionStatus	Confirms the Status code in the Action frame	Integer 0 = Disabled 1 = Enabled

Name	Description	Value
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
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 LCI-Z = LCI and Z information for Location CivicAddr = Location Civic address LocPubId = Location public identifier
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 LCI-Z = LCI and Z information for Location CivicAddr = Location Civic address LocPubId = Location public identifier
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

Name	Description	Value
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
AssocStatus	Association status code	Integer 0 = Successful association 1 = Failed association
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
BSS_Max_Idle_IE	Confirms the existence of the Max Idle IE.	Integer 0 = Should not exist 1 = Should exist
BSS_Max_Idle_Option	Checks the Max Idle option field.	Integer 0 = Any data, management or PS-poll frame may serve as keep-alive frame 1 = Only protected management or data frames may serve as keep-alive frame
BSS_Max_Idle_Period	Checks that the BSS Max Idle Period value matches the given value.	Integer (unit: 1000 TUs)
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

Name	Description	Value
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
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_Format	Confirms if the cluster ID in NAN beacon is in the correct format	String

Name	Description	Value
CoLocBSSID	Colocated BSSID measurmenet report	Integer 0 = Should not exist 1 = Should exist
Conn_CapElem	Confirms if the Connection capability element is present	Integer 1 = Should be present 0 = Should not be present
Conn_CapElem_ProtoInfo	Confirms if the Protocol infofield 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
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

Name	Description	Value
Data_Rate	Confirms the data transmission rate of the 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
DMG_cap_field	Confirm the DMG capability element is present	Integer 0 = Not present 1 = present
DMS_Req_TCLAS	Checks if the DMS Request has a TCLAS element with tag number 14.	Integer 1 = Should match 0 = Should not match
DMS_Req_Type	Checks the DMS Request Type field	Integer
DMS_Resp_TCLAS	Confirms if DMS Response has a TCLAS element with tag number 14	Integer 1 = Should match 0 = Should not match
DMS_Resp_Type	Checks that the DMS Response Type Field 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
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

Name	Description	Value
EC_IE	Confirms the Extended Capabilities element	Integer 0 – Should not exist 1 – Should exist
EC_IE_Bit12, EC_IE_Bit14, EC_IE_Bit15, EC_IE_Bit26, EC_IE_Bit31, EC_IE_Bit32, EC_IE_Bit46, EC_IE_Bit70, EC_IE_Bit71	Confirms the corresponding bit of extended capabilities field 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
ExistMaxAge	Confirms max age subelement in LCI request	Integer 0 = Should not exist 1 = Should exist
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
ExtChnHt20	Confirms PPDU of type HT20	Integer 0 = Not present 1 = Should be present
ExtChnHt40d	Confirms PPDU of type HT40d	Integer 0 = Not present 1 = Should be present
ExtChnHt40u	Confirms PPDU of type HT40u	Integer 0 = Not present 1 = Should be present
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 = Acknowlegment 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 frmae RTSP = Real Time Streaming Protocol GasinitResp GasComebackResp eapol icmp FTM = Fine Timing Measurement frame FTMreq = FTM request frame
FrameTxRate	Confirms the Transmitted DataRate of the frame	String Not11bRate
Friendly_Name	Checks for the friendly name	String

Name	Description	Value
FrmReTx	Confirms retry bit setting	Integer 0 = Reset 1 = Set
FTMasap	Confirms FTM ASAP bit	Integer
FTMasapCapable	Confirms FTM ASAP capable bit	Integer
FTMburstPeriod	Confirms FTM session burst period	Integer
FTMburstsDuration	Confirms FTM session burst duration value	Integer
FTMburstsExponent	Confirms the FTM session burst exponent	Integer
FTMdialogToken	Confirms FTM dialog token	Integer
FTMfollowupDialogToken	Confirms FTM followup dialog token	Integer
FTMformatBw	Confirms FTM format and bandwidth value	Integer
FTMmaxToaErr	Confirms FTM max ToA error	Integer
FTMmaxTodErr	Confirms FTM max ToD error	Integer
FTMparams	FTM parameters element	Integer 0 = Should not exist 1 = Should exist
FTMrangeReport	FTM range report element	Integer 0 = Should not exist 1 = Should exist
FTMrepIncapable	Confirms FTM report Incapable bit	Integer
FTMrepLate	Confirms FTM report Late bit	Integer
FTMrepRangeErrBssid	Confirms FTM report range error BSSID	String
FTMrepRangeRngBssid	Confirms FTM report ranging BSSID	String
FTMrepRefused	Confirms FTM report Refused bit	Integer
FTMreqTrigger	Confirms the FTM request frame trigger	Integer
FTMsPerBurst	Confirms FTMs per burst value	Integer
FTMstatusInd	Confirms the FTM status indication bit	Integer
FTMtoaErr	Confirms FTM ToA error	Integer
FTMtoaNotCont	Confirms FTM ToA Not Continuous value	Integer
FTMtodErr	Confirms FTM ToD error	Integer
FTMtodNotCont	Confirms FTM ToD Not Continuous value	Integer
FTMvalue	Confirms FTM value bit	Integer
Further_NAN_ServiceDiscAttr	Confirms the Further NAN Service Discovery attribute	Short integer
FurtherAvailMapAttr	Confirms the Further Availability Map attribute	Short integer
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

Name	Description	Value
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
HT_Info_IE_Ch_Width	Confirms the channel width that is set in HT Information element	Integer 20 40
HT_Opt_Prim_Chn	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

Name	Description	Value
HtGreenField	Confirms HT greenfield capabilities	Integer 0= Not supported 1= Supported
HT-SIG2_STBC	Confirms support for STBC	Integer 0= Not supported 1= Supported
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
LCIreport	LCI measurement report	Integer 0 = Should not exist 1 = Should exist

Name	Description	Value
LCIrequest	LCI measurement request	Integer 0 = Should not exist 1 = Should exist
Legacy_OSU_SSID	Legacy OSU SSID	String SSID_2
LL_TargetAddr	Link layer target address	String aa:bb:cc:dd:ee:ff
LocCivicReport	Location Civic measurement report	Integer 0 = Should not exist 1 = Should exist
LocCivicReport_Sub	LOC civic report subelement	Integer 0 = Should not exist 1 = Should exist
LocCivicRequest	Location civic measurement request	Integer 0 = Should not exist 1 = Should exist
LocConfigReport_Sub	LOC configuration report subelement	Integer 0 = Should not exist 1 = Should exist
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
MaxAgeSubElem	Confirms the value of max age subelement	Integer 0 - 65535
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
MinAPcnt	Confirms minimum AP count in FTM	Integer
minDeltaFTM	Confirms min delta FTM value	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

Name	Description	Value
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.		
NrrBSSID	BSSID in neighbor report	String
NrrChan	Confirms channel number in neighbor report request	Integer
NrrFTMbssidInfoNegative	Confirms FTMbssidInfo in neighbor report	String
NrrPhyTypeNegative	Confirms PhyType in neighbor report	Integer 1 = Should be present 0 = Should not be present
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
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

Name	Description	Value
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
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

Name	Description	Value
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	Integerr 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		
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

Name	Description	Value
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 Attirbute 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
PartialTsfTimer	Confirms partial TSF timer	Integer 0 - 65535
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
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

Name	Description	Value
PtsfNoPref	Confirms partial TSF timer no preference bit set	Integer 0 = Disabled 1 = Enabled
PubIdUFDesc	Confirms public identifier URI/FQDN descriptor	String
PubIdUFName	Confirms public identifier URI/FQDN name	String
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
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

Name	Description	Value
RandInterval	Confirms Randomization interval in FTM range request	Integer
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
ReservedBit7	Confirms reserved bit 7	Integer 0 = Reset 1 = Set
ReservedBits_48-49	Confirms reserved bit 48 & 49	Integer 0 = Reset 1 = Set
RetransAllowed	Confirms retransmission allowed bit is set	Integer 0 = Disabled 1 = Enabled
RMdialogToken	Confirms RM dialog token	Integer
RMEC_IE	Confirms the RM Enabled Capabilities element	Integer 0 = Should not exist 1 = Should exist
RMEC_IE_Bit3, RMEC_IE_Bit12, RMEC_IE_Bit34, RMEC_IE_Bit35	Confirms the corresponding bit of RM Enabled capabilities is set	Integer 0 = Disabled 1 = Enabled
RMlocSubject	Confirms Location subject field in radio measurement	Integer 0 = local 1 = remote
RMmodeEnable	Confirms enable mode in radio measurement	Integer 0 = Disabled 1 = Enabled
RMrepetitions	Confirms number of Repetions in in radio measurement	Integer
RMreqMode	Confirms request mode in radio measurement	Integer 0 = Disabled 1 = Enabled
RMtoken	Confirms Radio measurement token value	Integer
RrmCap_B0, RrmCap_B1, RrmCap_B14, RrmCap_B15, RrmCap_B32, RrmCap_B4, RrmCap_B5	Confirms the corresponding Rrm capabilities IE	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_AKMS_Count	Confirms that the given AKM Suite Count	Integer Ex. 1
RSN_AKMS_OUI	Confirms that the given AKM Suite OUI	Integer Ex. 0x000FAC

Name	Description	Value
RSN_AKMS_Type	Confirms that the given AKM Suite type	Integer Ex. 1
RSN_IE	Confirms the RSN IE is present	Integer 0 = Should not exist 1 = Should exist
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_Count	Confirms that the given Pairwise Cipher Suite Count	Integer Ex. 1
RSN_PCS_OUI	Confirms that the given Pairwise Cipher Suite OUI	Hex value
RSN_PCS_Type	Confirms that the given Pairwise Cipher Suite type	Integer Ex. 4
RSN_PCS_Type_NE	Confirms if the PCS type field in RSN information is not equal to the given value	Integer
RSN_PMKID_Count	Confirms that the given PMKID Count	Integer Ex. 0
RSN_PMKID_List	Confirms the presence of the PMKID list	Integer 0 = Should not exist 1 = Should exist
RSN_QCS_OUI	Confirms that the given Group Cipher Suite OUI	Hex value
RSN_QCS_Type	Confirms that the give Group Cipher Suite type	Integer Ex. 4
RSN_Version	Confirms that the given RSN version	Integer 1 = RSN version 0001
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

Name	Description	Value
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
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

Name	Description	Value
TSF_Init	Confirms if the TSF of the first beacon transmitted is less than the specified value as per test plan	Hex value
TSFSyncInfo	Confirms TSF synchronization info	Integer 0 = Not present 1 = Should be present
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_ServiceId	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
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

Name	Description	Value
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
VHT_Tx_Pwr_Env_IE	Confirms the VHT Transmit Power Envelope element	Integer 0 = Should not exist 1 = Should exist
VHTCmpBmForm_ FirFeedSeg	Confirms the First Feedback Segment in the VHT compressed beam forming frame	Integer
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_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

Name	Description	Value
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
WMMIE	Confirms if the IE exists or is not in the frame	Integer 0 = Should not exist 1 = Exists
WMMmmIe	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
WPA_IE	Confirms the WPA Information Element in EAPOL KEY packets	Integer 0 = Not present 1 = Should be present
WPS_Auth_Open	Checks if security mode is open security	Integer 0 = No 1 = Yes
WPS_Auth_WPA2PSK	Checks if secruity mode is WPA2 PSK	Integer 0 = No 1 = Yes

Name	Description	Value
WPS_ConfMethod	Confirms the Config method in the Config Method Attribute of WPS IE	String WPS_Display WPS_Keypad WPS_Label WPS_PushButton
WPS_Conn_Type_ESS	Checks if connection type is ESS	Integer 0 = No 1 = Yes
WPS_DevPass_IDAttr	Confirms the Existence of Device Password ID attribute in WPS IE	Integer 0 = Should not exist 1 = Should exist
WPS_DevPassIDAttr	Confirms the Device Password ID Attirbute is present	Integer 0 = Disabled 1 = Enabled
WPS_DevPwId	Confirms the Device Password ID matches the given mode	String Registrar User PIN PushButton
WPS_Encr_AES	Checks if encryption type is AES	Integer 0 = No 1 = Yes
WPS_Encr_None	Checks if encryption type is none	Integer 0 = No 1 = Yes
WPS_IE	Confirms the WPS IE is present	Integer 0 = Should not exist 1 = Should exist
WPS_IE_AssocStateAttr	Checks if the association state attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_ConfigErrorAttr	Checks if the configuration error attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_configMethodsAttr	Check if the configuration method attribute is present	Integer 0 = Not present 1 = Present
WPS_IE_DevName	Confirms the Existence of Device name in WPS IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_DevNameAttr	Checks if the device name attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_ManufacturerAttr	Checks if the manufacturer attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_ModNameAttr	Checks if the model name attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist

Name	Description	Value
WPS_IE_ModNumberAttr	Checks if the model number attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_New	Confirmes the new attribute is added to WPS IE	Integer 0 = Not added 1 = Added
WPS_IE_PrimaryDevTypeAttr	Confirms the Existence of Primary device type attribute in WPS IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_ReqTypeAttr	Checks if the request type attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_RfBandsAttr	Checks if the RF band state attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_SerialAttr	Checks if the serial number attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_UUIDAttr	Checks if the UUID attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_IE_Ver2Attr	Checks if the version 2 subelement is in WFA Vendor Extension	Integer 0 = Should not exist 1 = Should exist
WPS_IE_VerAttr	Checks if the version number attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_SelReg_Flag	Checks if the selected registrar flag attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist
WPS_SelRegConfigMethodAttr	Checks if the selected registrar configuration methods attribute is in WSC IE	Integer 0 = Should not exist 1 = Should exist

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,framename,probereq,
wmmie,1
Sniffer:status,RUNNING
Sniffer:status,COMPLETE,CheckResult,SUCCESS
```

9.17 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 = Acknoledge Assoresp = Association response EapolKey = EAPOL Key frame VhtAction = VHT Action frame FTM = Fine Timing Measurement frame
FTMdialogToken	Confirms FTM dialog token	Integer
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
```

9.18 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 DMS_Response DMS_Request
ANQP_InfoID	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
bidiMAC	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
EthsSrcIP	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

Name	Description	Value
		ProbeResp = Probe Response frame Ack = Acknoledge Assoresp = Association response Assocreassocreq = Associtation Re-Association Req Assocreassocresp = Associtation 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 = Gratituous 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 eaprespiden anqpresp data_mgmt_pspoll data_mgmt
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[1]	MAC address of receiver in aa:bb:cc:dd:ee:ff format	String
SnifferTimeGTE[1]	Epoch time greater than the specified time in seconds	Integer
SnifferTimeLTE[1]	Epoch time lesser than the specified time in seconds	Integer
SrcMAC[1]	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

Name	Description	Value
udpdstport		
Notes: 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
```

9.19 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
```

9.20 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
```

9.21 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
```

9.22 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
```

9.23 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
```

9.24 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_4
6
Sniffer:status,RUNNING
Sniffer:status,COMPLETE
```

9.25 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
```

9.26 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
```

9.27 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
```

9.28 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 CrossConBit = returns the Cross Connection Bit value P2P_ActionStatus, P2P_Status = returns the P2P Action Status value SnifferTimeStamp = returns the relative of the fram from the biginig of the trace in seconds WPS_ConfMethod = Returns the WPS Config methods as WPS_Display WPS_Label WPS_Keypad WPS_PushButton WSCState = Out of box WSC state as Configured or NotConfigured DialogToken = 802.11 Action frames dialog token P2P_DialogToken P2PAction_DialogToken RMdialogToken FTMdialogToken FTMfollowupdialogToken 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 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_deauth_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 SdaId = 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 CoLocMaxBssidInd = Co-located max BSSID indicator PartialTsfTimer = Partial TSF timer field value FTMburstsDuration = Duration of burst in FTM session FTMburstPeriod = Burst period in FTM session TsfTimeStamp = TSF time stamp MCSindex = MCS index MindeltaFTM = min delta FTM value FTMsPerBurst = FTMs per burst FTMformatBw = FTM format bandwidth value FTMburstPeriod = FTM burst period

Name	Description	Value
		SequenceNum = Sequence number RangeEntryCnt = Range Entry Count ErorrEntryCnt = Error entry count DMS_ID AssocID MCSRate
Filename	Wi-Fi Test Suite sniffer trace file	String
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 = Acknoledge Assoresp = Association response EapolKey = EAPOL Key frame VhtAction = VHT Action frame HDCP_AKE_TxInfo HDCP_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

Name	Description	Value
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,bssi
d, aa:bb:cc:dd:ee:ff,frameName,Acton,ActionCode,GoNegResp,FieldName,P2P_ActionStatus
Sniffer:status,RUNNING
Sniffer:status,COMPLETE, CheckResult,SUCCESS,ReturnValue,0
```

9.29 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
```

9.30 SNIFFER_MEDIA_CHECK

This command is used to rigger 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
MediaYype	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,LPC
M_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 headerfrom 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_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,filena
me,511_Step25_Video.mp4
Sniffer:status,RUNNING
Sniffer: status,COMPLETE,CheckResult,SUCCESS
```

9.33 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
```

9.34 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
```

9.35 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
```

9.36 SNIFFER_CHECK_CHNl_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,bs
sid, aa:bb:cc:dd:ee:ff,BcnFrames,6
Sniffer:status,RUNNING
Sniffer:status,COMPLETE, CheckResult,SUCCESS
```

9.37 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_wmmps_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
```

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

10.1 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
```

10.2 DEV_CONFIGURE_IE

This command is used to (re)-configure the IE contents. The device should also allow invalid IE values and force the original value to be overwriten.

Note: For RSN IE (RSNE), this CAPI will be used after the AP_SET_SECURITY and STA_SET_SECURITY commands.

Name	Description	Value
IE_Name	Name of the Information Element (IE) to be configured.	String Ex. RSNE
Contents	Hex string representing the contents of the Information Element (IE), including element ID, length, and other optional and required field values.	String Ex. 30020100
Name	Name of device. This parameter is required for an AP Control agent implementation.	String
Interface	Interface ID	String For APs: 24G 5G For STAs (example): wlan0

Return Values

None.

Examples

For CTT:

1. None of optional fields are included in the RSNE

```
UCC:dev_configure_ie,Name,<AP_Name>,interface,wlan0,IE_Name,RSNE,contents,30020100
Sniffer: status,RUNNING
Sniffer: status,COMPLETE
```

2. Group Data Cipher Suite / Pairwise Sipher Suite count and List, AKM Suite Count and List / RSN Capabilities

```
UCC:dev_configure_ie,Name,CTT,interface,wlan0,IE_Name,RSNE,contents,
30140100000fac040100000fac040100000fac018000
Sniffer: status,RUNNING
Sniffer: status,COMPLETE
```

3. RSN protocol Version field set to value 2 which is invalid

```
UCC:dev_configure_ie,Name,CTT interface,wlan0,IE_Name,RSNE,contents,
30160200000fac040100000fac040100000fac0280000000
Sniffer: status,RUNNING
Sniffer: status,COMPLETE
```

4. Reserved bits (bit-14 and bit 15) in RSN Capabilities field are set to 1

```
UCC:dev_configure_ie,Name,CTT,interface,wlan0,IE_Name,RSNE,contents,
30160100000fac040100000fac040100000fac0280C00000
Sniffer: status,RUNNING
Sniffer: status,COMPLETE
```

10.3 DEV_GET_FRAME_INFO

This command is used to retrieve frame information from a device unless specified otherwise in the command description. The device may be an AP or STA or other controlled device hosted by the Wi-Fi Alliance Test Suite. If any of operation or execution is not supported or allowed, the command will return ERROR with an error code if possible.

Parameters

Name	Description	Value
Interface	Interface ID	String
Program	Program Name	String TM
Name	Device name. This parameter is required for an AP Control agent implementation only.	String AP Name Example: XYZ-AP
FrameName	This parameter indicates that the received frame is a Timing Measurement action frame.	String TM_Action
TM_Frame_Count	This parameter indicates the number of TM Action frames that should be sent if OTA_op is set to TX and received if OTA_op is set to RX.	Integer Range 1 to 20
OTA_op	Over the Air operation. Rx: TM action frame details of the received action frame. Tx: TM action frame details of the transmitted action frame.	String Tx Rx
Src_MAC	Source MAC address	String 6-byte Hex sting seprated by periods ':'

Return Values

The return values depend on frame the type.

FrameName	Description	Value
TM_Action	Returns elements from frame info aray. TM_Frame_Count specifies the number of elements returned. Format for one element given below. Additional elements are seperated by an apostraphe. dt,<DT.>,fdt,<FDT>,tod,<ToD>,toa,<ToA>,todmerr,<ToDMaxErr>,todmerr,<ToAMaxErr>,tod'<TOD'>,toa',<ToA'>,tod'merr,<ToD'MaxErr'>,toa'merr<ToA'MaxErr'> Where: DT. = Dialog token value from the Timing Measurement Action frame. If set to zero, no indication is returned. FDT = Followup dialog token value (can be zero) ToD and ToA = Time of depatrure are arrival in decimal. If OTA_op set to Rx then ToD and ToA shall be fetched from the Timing Measurement frame.	Integer Range 1 to 20

Examples:

```
UCC: dev_get_frame_info, program, TM, interface, wlan0,Src_MAC, 00:11:22:33:44:55, FrameName,
TMAction,OTA_op,Rx, TM_Frame_Cnt,3
CA:status,RUNNING
CA:status,COMPLETE,dt,1,fdt,0,tod,418987296,toa, 418997001,todmerr,0,toamerr, 255,
tod',418987296,toa', 418997001,tod'merr,0,toa'merr, 255, dt,2,fdt,1,tod,418987296,toa,
418997001,todmerr,0,toamerr, 255, tod',418987296,toa', 418997001,tod'merr,0,toa'merr, 255,
```

10.4 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
```

10.5 DEV_GET_PARAMETER

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

Parameters

Name	Description	Value
Parameter		String MacAddr BSSID ALID
Program	Program name	String MAP
RUID	A radio unique ID.	String
SSID	The SSID that fronthaul AP is configured. When SSID is not specified for the parameter "macaddr", it implies that device should return backhaul STA's MAC address. SSID is also not required for the parameter "ALid".	String For Ex.: Multi-AP-24GC-1

Return Values

Name	Description	Value
ALID	Returns the AL MAC address.	aLid,11:22:33:44:55:66
BSSID	Returns the BSSID currently oeprating on the given RUID and SSID.	bssid,11:22:33:44:55:66
MacAddr	Returns the MAC address operating on the given RUID and SSID	macaddr,11:22:33:44:55:66

Examples

```
dev_get_parameter,program,map,ruid,WTS_REPLACE_RUID,ssid,Multi-AP-24GC-1,parameter,macaddr
status,COMPLETE,macaddr,11:22:33:44:55:66

dev_get_parameter,program,map,ruid,WTS_REPLACE_RUID,ssid,Multi-AP-24GC-1,parameter,bssid
status,COMPLETE,bssid,11:22:33:44:55:66
```

10.6 DEV_EXEC_ACTION

This command is used to notify or indicate a device to execute aa specific command to initiate operation or to start a specific operation unless specified otherwise in the command description. The device can be an AP or STA or other controlled device hosted by the Wi-Fi Alliance Test Suite. If any of operation or execution is not supported or allowed, the command will return ERROR with an error code if possible.

Parameters

Name	Description	Value
CodecProfile	For a source, this parameter sets or enables direct streaming of the specified codec and profile. The format is H264-<profile> or H265-<profile>. Profile index is as defined in the specification.	String
Dest_MAC	Destination /peer MAC address	String
DirectStreaming	Trigger direct streaming	String Start Stop
DPPActionType	GetLocalBootstrap instructs the device to send a hex dump of its bootstrapping data (see return value BootstrappingData for details) SetPeerBootstrap instructs the device to set supplied bootstrapping data (excluding PKEX) AP - AutomaticDPP/ManualDPP will enable new configuration and return a response once the AP is ready to accept new associations STA - AutomaticDPP will enable a configuration and return a response once connected to the network QR: AutomaticDPPstarts at Auth using keys from GetLocalBootstrap/SetPeerBootstrap procedure QR: ManualDPP starts with manual BootStrapping PKEX: AutomaticDPP starts with PKEX Bootstrapping PKEX: ManualDPP is not applicable The AutomaticDPP command is always preceded by either GetLocalBootstrap or SetPeerBootstrap for one-sided authentication with QR codes. The AutomaticDPP command is always preceded by both GetLocalBootstrap or SetPeerBootstrapfor mutual authentication with QR codes. The AutomaticDPP command is never preceded by GetLocalBootstrap or SetPeerBootstrap for PKEX. The ManualDPP command is never preceded by GetLocalBootstrap or SetPeerBootstrap for PKEX. A device that is able to change its bootstrapping key shall only change its bootstrapping key when it recevies: • A GetLocalBootstrap command, • An AutomaticDPP command with argument DPPBS=PKEX, • A ManualDPP command. A device shall start a new run of the DPP bootstrapping protocol (if applicable), DPP Authentication protocol, DPP Configuration protocol and DPP Introduction protocol (if applicable) when it recevies: • An AutomaticDPP command, • A ManualDPP command	String GetLocalBootstrap SetPeerBootstrap ManualDPP AutomaticDPP

Name	Description	Value
DPPBootstrappingdata	The Hex dump of bootstrapping info URI. Applicable to DPPActionTypes: SetPeerBootstrap	Hex
DPPCryptoIdentifier	Indentifies the ECC curve to use for both DPP bootstrapping and authentication. Applicable to DPPActionTypes: GetLocalBootstrap, AutomaticDPP and ManualDPP	String P-256 P-384 P-521 BP-256R1 BP-384R1 BP-512R1
DPPBS	DPP Bootstrapping mechanism. Applicable to all DPPActionTypes	String QR BTLE NAN NFC PKEX
DPPAuthRole	DPP Authentication Role. Applicable to DPPActionTypes: AutomaticDPP and ManualDPP. The AutomaticDPP command or ManualDPP command with argument DPPAuthRole=Responder always precedes the corresponding AutomaticDPP command or ManualDPP command with argument DPPAuthRole=Initiator. There shall be a one second wait between these calls to allow the Responder to be in reception state before the Initiator starts transmitting. The following items are specifically applicable to authentication with the QR Code Bootstrapping method: If a device receives a ManualDPP command with argument DPPAuthRole=Responder, the device shall immediately display its QR-code and may stop displaying its QR-code only after the reception of a DPP Authentication Request frame. If a device receives a ManualDPP command with argument DPPAuthRole=Initiator, the device shall start to display its QR-code upon receiving DPP Authentication Response with DPP_STATUS equal to STATUS_RESPONSE_PENDING is the latest when QR-code shall be displayed. Device may stop displaying its QR-code only after the reception of a DPP Authentication Response frame with DPP_STATUS equal to STATUS_OK. (The Initiator never knows up front whether the Responder wants to do mutual authentication.) If a device receives a ManualDPP command with argument DPPAuthRole=Responder and with argument DPPAuthDirection=Mutual, it shall start its scanner to scan the QR-code of the Initiator. It shall do so in such a way that the QR scanning does not interfere with the display of its own QR-code. For example, this can be done by starting its QR-code scanner only after the reception of the DPP Authentication Request frame. If a device receives a ManualDPP command with argument DPPAuthRole=Initiator, the device shall start its QR-code scanner for obtaining the Responder QR code before initiating the DPP Authentication protocol.	String Initiator Responder
DPPProvisioningRole	DPP Provisioning role. Both means that the device can be both a Configurator and an Enrollee. Both is only applicable to testbed as an Authentication Initiator Applicable to DPPActionTypes: AutomaticDPP and ManualDPP	String Configurator Enrollee Both

Name	Description	Value
DPPStep	Applicable to DPPActionTypes: AutomaticDPP and ManualDPP Use with DPPFrameType and DPPIEAttribute to indicate which frame and what type of attribute this step pertains to. When InvalidValue is used with DPPIEAttribute: WrappedData, additional DPP status attribute is added after the wrapped data. When Timeout is used with DPPFrameType, CTT will stop the protocol exchange completely upon receiving the particular frame to cause a Timeout for the DUT. DPPStep, Timeout can also be used with DPPFrameType,AuthenticationConfirm to instruct the CTT (acting as an Enrollee) to stop the protocol exchange after having transmitted Authentication Confirmation. In that case, CTT is expected to not send a Configuration Request frame and report with the LastFrameReceived which was the last received frame (AuthenticationResponse). Whenever the Timeout argument is used, DPPIEAttribute is not included (Optional, Used by CTT to introduce errors by test bed. Indicates where and how to deliberately fail certain DPP steps.)	String Timeout InvalidValue MissingAttribute
DPPFrameType	This is used jointly with DPPStep and DPPIEAttribute to specify which frame should have the error or attribute.	String PKEXExchangeRequest PKEXExchangeResponse PKEXCRRequest PKEXCRResponse AuthenticationRequest AuthenticationResponse AuthenticationConfirm ConfigurationRequest ConfigurationResponse PeerDiscoveryRequest PeerDiscoveryResponse
DPPIEAttribute	This is used jointly with DPPStep and DPPFrameType to specifically indicate the type of error or attribute. String StatusRespPending is only applicable to DPPStep set to the DPPState value. FiniteCyclicGroup may be used with: PKEXExchangeRequest EncryptedKey may be used with: PKEXExchangeRequest DPPStatus may be used with: PKEXExchangeResponse, AuthenticationResponse[1], AuthenticationConfirm[2], ConfigurationResponse WrappedData may be used with: PKEXCRRequest, PKEXCRResponse, AuthenticationRequest, AuthenticationResponse[1], AuthenticationConfirm[2], ConfigurationRequest, ConfigurationResponse BSKey may be used with: PKEXCRRequest, PKEXCRResponse InitAuthTag may be used with: PKEXCRRequest, AuthenticationConfirm[2] RespAuthTag may be used with: PKEXCRResponse, AuthenticationResponse[1] RespBSKeyHash may be used with: AuthenticationRequest, AuthenticationResponse[1], AuthenticationConfirm[2]	String FiniteCyclicGroup EncryptedKey DPPStatus WrappedData BSKey InitAuthTag RespAuthTag RespBSKeyHash InitBSKeyHash InitProtocolKey InitNonce InitCapabilities PrimaryWrappedData RespCapabilities EnrolleeNonce ConfigAttr TransactionID Connector StatusRespPending RespProtocolKey RespNonce

Name	Description	Value
	InitBSKeyHash may be used with: AuthenticationRequest, AuthenticationResponse[1], AuthenticationConfirm[2] InitProtocolKey may be used with: AuthenticationRequest, AuthenticationResponse[1] InitNonce may be used with: AuthenticationRequest, AuthenticationResponse[1] InitCapabilities may be used with: AuthenticationRequest PrimaryWrappedData may be used with: AuthenticationResponse[1] RespCapabilities may be used with: AuthenticationResponse[1] EnrolleeNonce may be used with: ConfigurationRequest, ConfigurationResponse ConfigAttr may be used with: ConfigurationRequest TransactionID may be used with: PeerDiscoveryRequest, PeerDiscoveryResponse Connector may be used with: PeerDiscoveryRequest, PeerDiscoveryResponse StatusRespPending may be used with: AuthenticationResponse[1] RespProtocolKey may be used with: AuthenticationResponse[1] RespNonce may be used with: AuthenticationResponse[1]	
DPPAuthDirection	Applicable to DPPActionTypes: AutomaticDPP and ManualDPP Indicates single-sided or mutual authentication. When using QR codes, this is only applicable for DPPAuthRole = Responder. Applicable to QR Code as follows: • DPPAuthRole = Responder • CTT as Initiator in order to validate DUT's mutual authentication behavior N/A for PKEX.	String Single Mutual
DPPConfIndex	Applicable to Configurators and to DPPActionTypes: AutomaticDPP and ManualDPP DPP configuration index which points to a test plan table entry for the configuration object.	Integer
DPPConfEnrolleeRole	Applicable to Configurators and to DPPActionTypes: AutomaticDPP and ManualDPP To instruct Configurator to allow Enrollee to be AP or STA (one at a time)	String AP STA
DPPPKEXCodeIdentifier	Applicable to DPPActionTypes: AutomaticDPP and ManualDPP Optional DPP PKEX Code Identifier. Present only when Code Identifier is used.	String Example: dpp device
DPPPKEXCode	Applicable to DPPActionTypes: AutomaticDPP and ManualDPP Mandatory DPP Shared Code for PKEX	String Example: password
DPPTimeout	Applicable to DPPActionTypes: AutomaticDPP and ManualDPP	Integer Example. 120

Name	Description	Value
	Optional. How long (in seconds) the DPP process may proceed prior to causing a timeout. If not set, the default timeout of 120 seconds is used.	
DPPSigningKeyECC	Applicable to Configurator and to DPPActionTypes: AutomaticDPP and ManualDPP Curve to use for the Configurator signing key	String P-256 P-384 P-521 BP-256R1 BP-384R1 BP-512R1
DPPWaitForConnect	Applicable to DPPActionTypes: AutomaticDPP and ManualDPP Optional. If set to Yes, command returns after association or timeout occurs. If set to No, command returns after DPP configuration completes or timeout occurs. If not included, default "No" behavior is used.	String Yes No
DPPSelfConfigure	Applicable to Configurator and to DPPActionTypes: AutomaticDPP and ManualDPP Optional. If set to Yes, the Configurator issues itself with a Connector and a Config Object that matches its own netrole (AP/STA) and the network defined by DPPConfigIndex. If set to No, the Configurator does not configure itself for the network. If not included, default "No" behavior is used.	String Yes No
DPPChannelList	Applicable to QR Code and GetLocalBootstrap for a test bed that serves as an Authentication Responder in subsequent handshakes via AutomaticDPP. The value is defined to indicate the channel list that test bed shall produce in its QR Code	String Space separated tuple(s) of the following format: OperatingClass/Channel Example: 81/6 116/44
DPPListenChannel	Applicable to DPPActionTypes: AutomaticDPP This value indicates the channel will listen on for subsequent handshakes for a test bed that serves as an Authentication Responder	Integer
DPPDelayQRResponse	Applicable to DPPActionTypes: AutomaticDPP, and to bootstrapping method: QR Code Mutual Authentication This argument indicates the minimum delay in unit of seconds for sending STATUS_RESPONSE_PENDING with STATUS_OK by a test bed device serving as an Authentication Responder.	Integer
DPPSubsequentChannel	Specifies the value of the channel TLV in the DPP Authentication Request. The Initiator must insert this TLV when this parameter is present in the WTS command. Mandatory for test bed. Optional for DUT pending its support for such TLV.	String OperatingClass/Channel Example: 81/6
FrameName	Indicates that the received frame is a Timing Measurement action frame	String TM_Action
Get_Correlated_Time	If enabled, the command returns with a 64-bit system clock value and a 64-bit timestamp counter value in decimal. The format is: <64bit system clock>, <64bit time stamp>	Integer 1 = Enable 0 = Disable
Interface	Interface ID	String
Name	A name such as ABC AP	String
Program	Program Name	String

Name	Description	Value
		DisplayR2 TM DPP
ServiceType	Service type used to register a service Triggers the device to register a service using mDNS protocol with type of service as specified in argument ServiceType	String display._tcp
TM_1AS_Process	Start/stops the TM 1AS process.	String Start Stop
TM_1AS_Role	Role of a TM 1AS device. It can either be a master or slave clock.	String Master Slave
TM_Frame_Count	Indicates the number of TM action frames that should be sent if OTA_op is TX and received if OTA_op is RX.	Integer Range 1 to 20
TMS_Sync_Verification	This parameter is used in conjunction with the parameter TM_1AS_Process. On: return Delta with TM_1AS_Process set to Stop Off: return Offset with TM_1AS_Process set to stop	String On Off
VideoFormat	For a source, this parameter sets or enables direct streaming of the specified video format. The format is <prog>_CEA-<Index> or <prog>_VESA-<Index> or <prog>_HH-<Index>. Index is as defined in the specification.	String WFD_CEA-<index> WFD_VESA-<index> WFD_HH-<index> WFD_ALL WFD2_CEA-<index> WFD2_VESA-<index> WFD2_HH-<index> WFD2_ALL NONE
Notes: 1. InitBSKeyHash is only applicable to Mutual Authentication. 2. InitAuthTag is only applicable to Mutual Authentication.		

Return values

Name	Description	Value
AuthResult	This value indicates the status of DPP Authentication when DPPActionType is set to AutomaticDPP or ManualDPP. OK is used by DUT/CTT to indicate successful completion of Authentication phase. Failed is used by DUT/Testbed to report a unsucessful Authentication phase frame exchange Timeout is used by DUT/Testbed to report non-reception of Authentication phase frames. Errorsent is used by CTT to send an intentional error. In the context of authentication negative test cases: 1. For invalid value test cases: The sender of malformed frame shall return Errorsent value. Return value for subsequent stages may be any.	String ROLES_NOT_COMPATIBLE OK Failed Timeout Errorsent

Name	Description	Value
	The receiver of such error frame shall provide Failed as return value for the authentication stage. Return value for subsequent stages may be any. 2. Timeout cases are done by withholding Authentication Response or Authentication Confirm The originator of such timeout shall provide Errorsent as return value for the Authentication stage. Return values for subsequent stages may be any. The receiving peer which expects this frame to proceed in protocol exchange shall provide Timeout as return value for the Authentication stage. Return values for subsequent stages may be any.	
BootstrappingData	This value is returned if DPPActionType is set to GetLocalBootstrap and contains a hex dump of the bootstrapping info URI.	Hex
BootstrapResult	This value indicates the status of the DPP bootstrapping phase when DPPActionType is set to AutomaticDPP or ManualDPP. Errorsent is used by the test bed to send an intentional error.	String OK Failed Timeout Errorsent
ConfResult	This value indicates the status of DPP Configuration when DPPActionType is set to AutomaticDPP or ManualDPP. OK is used by DUT/CTT to indicate successful completion of Configuration phase. Failed is used by DUT/Testbed to report a unsucessful Configuration phase frame exchange Timeout is used by DUT/Testbed to report non-reception of Configuration phase frames. Errorsent is used by CTT to send an intentional error. In the context of configuration negative test cases: 1. For invalid value test cases: The sender of malformed frame shall return Errorsent value. Return value for subsequent stages may be any. The receiver of such error frame shall provide Failed as return value for the Configuration stage. Return value for subsequent stage may be any. 2. Timeout cases are done by withholding Configuration Request or Configuration Response frame. The originator of such timeout shall provide Errorsent as return value for the Configuration stage. Return values for subsequent stage may be any. The receiving peer which expects this frame to proceed in protocol exchange shall provide Timeout as return value for the Configuration stage. Return values for subsequent stage may be any.	String Ok Failed Timeout Errorsent
Delta	This value is returned if TMS_Sync_Verification is set to 'On'. The format is: D#,<value>, D#,<value>, D#,<value>, D#,<value>,… For example: D1,123456, D2,1234567, D3,53423534, D4,32543453	String
InstanceName	Only returned when the supplied argument is ServiceType. Returns the instance name of the service registered for the specified ServiceType.	String
LastFrameReceived	This inidicates the last frame received by a CTT for DPP protocol exchange. Is only applicable in negative cases for CTT (specifically when CTT is acting as a Responder).	String AuthenticationRequest AuthenticationResponse

Name	Description	Value
		AuthenticationResponseWithStatusPending AuthenticationConfirm ConfigurationRequest None
NetworkIntroResult	This value indicates the status of DPP Network Introduction when DPPActionType is set to AutomaticDPP or ManualDPP and DPP AKM is provisioned. When a legacy AKM is provisioned, this argument is not returned. Errorsent is used by the test bed to send an intentional error. In the context of negative test cases: 1. For invalid value test cases: The testbed sender of malformed frame shall return Errorsent value. Return value for subsequent stages may be any. The receiver of such error frame shall provide Failed as return value for the Network Introduction stage. Return value for subsequent stages may be any. 2. Timeout cases are done by withholding Peer Discovery Response frame. The originator of such timeout shall provide Errorsent as return value for the Network Introduction stage. Return values for subsequent stages may be any. The receiving peer which expects this frame to proceed in protocol exchange shall provide Timeout as return value for the Network Introduction stage. Return values for subsequent stage may be any.	String OK Failed Timeout Errorsent
NetworkConnectResult	This value indicates the status of DPP Network Connection when DPPActionType is set to AutomaticDPP or ManualDPP. Errorsent is used by the test bed to send an intentional error.	String OK Failed Timeout Errorsent
Offset	This value is returned if TMS_Sync_Verification is set to 'Off'. The format is: O#,<value>, O#,<value>, O#,<value>, O#,<value>,… For example: O1,123456, O2,1234567, O3,53423534, O4,32543453	String

Examples

```
UCC: dev_exec_action,program,DisplayR2,interface,wlan0,DirectStreaming,Start
CA:status,RUNNING
CA:status,COMPLETE

UCC: dev_exec_action,program,DisplayR2,interface,wlan0,ServiceType,_display._tcp
CA:status,RUNNING
CA:status,COMPLETE,InstanceName,XYZ-Sink
```

eNPD TM examples:

```
UCC: dev_exec_action, program, TM, interface, wlan0, Dest_MAC, 00:11:22:33:44:55,
TM_1AS_Process,Start, TM_1AS_Periodicity,125
CA:status,RUNNING
CA:status,COMPLETE

UCC: dev_exec_action,Name,<Ap name>,program,TM,interface,wlan0, Dest_MAC, 00:11:22:33:44:55,
TM_1AS_Process,Start, TM_1AS_Periodicity,125
CA:status,RUNNING
```

```
CA:status,COMPLETE

UCC: dev_exec_action,Name,<Ap name>,program,TM,interface,wlan0, Dest_MAC, 00:11:22:33:44:55,
TM_1AS_Process,Start, TMS_Synchronization,On
CA:status,RUNNING
CA:status,COMPLETE

UCC: dev_exec_action,Name,<Ap name>,program,TM,interface,wlan0, Dest_MAC, 00:11:22:33:44:55,
TM_1AS_Process,Start, TMS_Synchronization,On, TMS_Sync_Verification,On
CA:status,RUNNING
CA:status,COMPLETE
```

For offset return

```
UCC: dev_exec_action, program, TM, interface, wlan0, Dest_MAC, 00:11:22:33:44:55,
TM_1AS_Process,Stop
CA:status,RUNNING
CA:status,COMPLETE,O#, 123456, O#,1234567., O#,53423534, O#,32543453,…….
```

For delta return

```
UCC: dev_exec_action, program, TM, interface, wlan0, Dest_MAC, 00:11:22:33:44:55,
TM_1AS_Process,Stop
CA:status,RUNNING
CA:status,COMPLETE,D#,123456, D#,1234567, D#,53423534, D#,32543453,…….

UCC: dev_exec_action, program, TM, interface, wlan0, Dest_MAC, 00:11:22:33:44:55,
Get_Correlated_Time,1
CA:status,RUNNING
CA:status,COMPLETE, 1234561234567, 5342353432543453

DPP Example 1 (step 1 to first peer - get local bootstrapping data from Authentication Responder):
UCC: dev_exec_action,program,DPP,DPPActionType,GetLocalBootstrap,DPPCryptoIdentifier,P-256,DPPBS,QR
CA:status,RUNNING
CA:status,COMPLETE,BootstrappingData,4450503a493a534e3d343737344c483262343034343b4d3a30313032303330
34303530363a4b3a4d446b77457759484b6f5a497a6a3043415159494b6f5a497a6a3044415163444967414455527a786d7
4745a6f4952495057476f514d5630305848574341514968587275567 4f7a304e6a6c6b49413d3b3b

DPP Example 2 (step 2 to second peer - set peer bootstrapping data into the Authentication
Initiator):
UCC:
dev_exec_action,program,DPP,DPPActionType,SetPeerBootstrap,DPPBS,QR,DPPBootstrappingData,4450503a49
3a534e3d343737344c483262343034343b4d3a303130323033303430353036 3a4b3a4d446b77457759484b6f5a497a6a304
3415159494b6f5a497a6a3044415163444967414455527a786d74745a6f4952495057476f514d5630305848574341514968
58727556574f7a304e6a6c6b49413d3b3b
CA:status,RUNNING
CA:status,COMPLETE

DPP example 3 (protocol initiation - Configurator to configure an enrollee):
UCC:
dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPAuthRole,Responder,DPPConfIndex,1,DPPAuth
Direction,Single,DPPProvisioningRole,Configurator,DPPSigningKeyECC,P-
256,DPPConfEnrolleeRole,STA,DPPBS,QR
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK

DPP example 4 (protocol initiation - STA enrollee that initiates authentication):
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPProvisioningRole,Enrollee,DPPTimeout,120
```

```
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK

DPP example 5 (protocol initiation - STA enrollee that initiates authentication and connects to AP
with DPP AKM):
UCC:
dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPAuthRole,Responder,DPPConfIndex,1,DPPAuth
Direction,Single,DPPProvisioningRole,Configurator,DPPSigningKeyECC,P-
256,DPPConfEnrolleeRole,STA,DPPBS,QR,DPPWaitForConnect,Yes
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK,NetworkIntroResult,OK,NetworkConn
ectResult,OK

DPP example 6 (protocol initiation - STA enrollee that initiates authentication and connects to AP
with legacy AKM):
UCC:
dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPAuthRole,Responder,DPPConfIndex,1,DPPAuth
Direction,Single,DPPProvisioningRole,Configurator,DPPSigningKeyECC,P-
256,DPPConfEnrolleeRole,STA,DPPBS,QR,DPPWaitForConnect,Yes
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK,NetworkConnectResult,OK


DPP example 7 (configurator self configuration - Configurator to act as an AP/STA without network
connection)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPProvisioningRole,Configurator,DPPAuthDirection,Single,DPPConf
Index,1,DPPTimeout,120,DPPSigningKeyECC,P-256,DPPSelfConfigure,Yes
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK

DPP example 8 (configurator self configuration - Configurator to act as an AP/STA with network
connection)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPProvisioningRole,Configurator,DPPAuthDirection,Single,DPPConf
Index,1,DPPTimeout,120,DPPSigningKeyECC,P-256,DPPWaitForConnect,Yes,DPPSelfConfigure,Yes
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK,NetworkIntroResult,OK,NetworkConn
ectResult,OK

DPP example 9 (negative case - configure Configurator to send malformed PKEX frames resulting in
Bootstrapping to fail)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,PKEX,DPPAuthRole,Initiator,DPPProvisioningRole,Configurator,DPPConfIndex,1,DPPConfEnrolle
eRole,STA,DPPPKEXCode,password,DPPTimeout,120,DPPSigningKeyECC,P-
256,DPPStep,InvalidValue,DPPFrameType,PKEXExchangeRequest,DPPIEAttribute,EncryptedKey
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,Errorsent

DPP example 10 (negative case - configure Enrollee to timeout during DPP Authentication phase in
order to test CoUT's handling of timeouts)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPAuthDirection,Single,DPPProvisioningRole,Enrollee,DPPTimeout,
120,DPPStep,Timeout,DPPFrameType,AuthenticationResponse
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,Errorsent

DPP example 11 (negative case - configure Enrollee to send malformed frames during DPP
Authentication phase in order to test CoUT's handling of malformed frames)
```

```
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPListenChannel,11,DPPAuthRole,Responder,DPPProvisioningRole,Enrollee,DPPAuthDirectio
n,Single,DPPTimeout,120,DPPStep,InvalidValue,DPPFrameType,AuthenticationResponse,DPPIEAttribute,DPP
Status
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,Errorsent

DPP example 12 (negative case - CTT-Enrollee returns status LastFrameReceived when sending either
malformed frame or timed out in DPP Authentication phase, while DUT is acting as Responder)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPAuthDirection,Single,DPPProvisioningRole,Enrollee,DPPTimeout,
120,DPPStep,InvalidValue,DPPFrameType,AuthenticationRequest,DPPIEAttribute,RespBSKeyHash
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,Errorsent,LastFrameReceived,None

DPP example 13 (negative case - An Enrollee connecting to a configurator which has been configured
to do malformed authentication)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPAuthDirection,Mutual,DPPProvisioningRole,Enrollee,DPPTimeout,
20,DPPWaitForConnect,NO
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,Errorsent

DPP example 14 (negative case - CTT-Configurator returns status LastFrameReceived when sending
either malformed frame or timed out in DPP Authentication phase, while DUT is acting as Responder)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPAuthDirection,Single,DPPProvisioningRole,Configurator,DPPConf
Index,1,DPPConfEnrolleeRole,STA,DPPTimeout,120,DPPSigningKeyECC,P-
256,DPPStep,InvalidValue,DPPFrameType,AuthenticationRequest,DPPIEAttribute,RespBSKeyHash
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,Errorsent,LastFrameReceived,None

DPP example 15 (negative case - configure Enrollee to timeout during DPP Configuration phase in
order to test CoUT's handling of timeouts)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPListenChannel,11,DPPAuthRole,Responder,DPPProvisioningRole,Enrollee,DPPAuthDirectio
n,Single,DPPTimeout,120,DPPStep,Timeout,DPPFrameType,AuthenticationConfirm
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,Errorsent

DPP example 16 (negative case - configure Enrollee to send malformed frames during DPP
Configuration phase in order to test CoUT's handling of malformed frames)
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPListenChannel,11,DPPAuthRole,Responder,DPPProvisioningRole,Enrollee,DPPAuthDirectio
n,Single,DPPTimeout,120,DPPStep,InvalidValue,DPPFrameType,ConfigurationRequest,DPPIEAttribute,Wrapp
edData
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,Errorsent

DPP example 16 (negative case - configure Enrollee to send malformed frames during DPP
Configuration phase in order to test CoUT's handling of malformed frames))
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPListenChannel,11,DPPAuthRole,Responder,DPPProvisioningRole,Enrollee,DPPAuthDirectio
n,Single,DPPTimeout,120,DPPStep,InvalidValue,DPPFrameType,ConfigurationRequest,DPPIEAttribute,Wrapp
edData
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,Errorsent
```

```
DPP example 17 (manual bootstrapping):
UCC: dev_exec_action,program,DPP,DPPActionType,ManualDPP,DPPCryptoIdentifier,P-
256,DPPBS,QR,DPPAuthRole,Initiator,DPPProvisioningRole,Enrollee,DPPTimeout,120,DPPWaitForConnect,Ye
s
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK,NetworkIntroResult,OK,NetworkConn
ectResult,OK

DPP example 18 (PKEX bootstrapping):
UCC: dev_exec_action,program,DPP,DPPActionType,AutomaticDPP,DPPCryptoIdentifier,P-
256,DPPBS,PKEX,DPPListenChannel,6,DPPAuthRole,Responder,DPPProvisioningRole,Configurator,DPPConfInd
ex,1,DPPConfEnrolleeRole,STA,DPPPKEXCode,password,DPPTimeout,120,DPPSigningKeyECC,P-256
CA:status,RUNNING
CA:status,COMPLETE,BootstrapResult,OK,AuthResult,OK,ConfResult,OK,NetworkIntroResult,OK,NetworkConn
ectResult,OK
```

10.7 DEV_RESET_DEFAULT

This command is used to reset the device to its default program specific configuration, as well as remove any cached profiles, keys and credentials.

Dependencies

None.

Parameters

Name	Description	Value
DevRole	Device role.	String Controller Agent
Name	Device name	String
Program	Program name	String MAP
Type	Type of the device – Test bed or DUT	String Test bed DUT

Return Values

None.

Example

```
UCC: dev_reset_default,name,agt1,program,map,devrole,agent,type,testbed
CA:status,RUNNING
CA:status,COMPLETE

UCC: dev_reset_default, name,cntlr1,program,map,devrole,controller,type,testbed
CA:status,RUNNING
CA:status,COMPLETE
```

10.8 DEV_SEND_1905

This command is used to send a 1905 message with the relevant TLVs between the devices (triggered by Wi-Fi Test Suie). If multiple TLVs, tlv_type, tlv_length, and tlv_value parameters will be indexed (e.g. tlv_type1, tlv_length1, tlv_value1).

Dependencies

None.

Parameters

Name	Description	Value
DestALID	The destination ALID.	String Ex. 11:22:33:44:55:66
MessageTypeValue	The Multi-AP message type value in hex.	Hex value: Ex.: 0x8009
TLV_Type		Hex value: Ex.: 0x90
TLV_Length		Hex value: Ex.: 0x000C
TLV_Value	TLV value is separated by curly bracket and space for each field and subfield. The field value should be in hex format.	Hex value: Ex.: {11:22:33:44:55:66 aa:bb:cc:dd:ee:f0}

Return Values

Name	Description	Value
MID	Returns 1905 message ID	Hex value: Ex.: 0xb242

Example

```
UCC:
dev_send_1905,DestALid,00:90:4C:2A:21:C2,MessageTypeValue,0x8006,tlv_type1,0x8B,tlv_length1,0x004C,
tlv_value1,{0x00904C2A11C2 0x14 {0x51 {0x00 0x00}} {0x52 {0x00 0x00}} {0x53 {0x00 0x00}} {0x54
{0x00 0x00}} {0x73 0x03 {0x28 0x2C 0x30} 0x00} {0x74 0x01 {0x2C} 0x00} {0x75 {0x00 0x00}} {0x76
{0x00 0x00}} {0x77 {0x00 0x00}} {0x78 {0x00 0x00}} {0x79 {0x00 0x00}} {0x7A {0x00 0x00}} {0x7B
{0x00 0x00}} {0x7C {0x00 0x00}} {0x7D {0x00 0x00}} {0x7E {0x00 0x00}} {0x7F {0x00 0x00}} {0x80 0x05
{0x3A 0x6A 0x7A 0x8A 0x9B} 0x00} {0x81 {0x00 0x00}} {0x82 {0x00
0x00}}},tlv_type2,0x8D,tlv_length2,0x0007,tlv_value2,{0x00904C2A11C2 0x14}
CA:status,RUNNING
CA: status,COMPLETE,MID,0xb242

UCC:
dev_send_1905,DestALid,00:90:4C:2A:21:B7,MessageTypeValue,0x0002
CA:status,RUNNING
CA: status,COMPLETE,MID,0xb3c9
```

10.9 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
Address3	Configure the Address 3 field with either a wild card MAC address or AP BSSID	String Eg. FF:FF:FF:FF:FF:FF Eg. A1:B1:C1:D1:E1:F1
Add_DMS_filter	TCLAS element values ID number. Refer to the eNPD-IoTLP test plan for TCLAS element values. Returns a DMS_filter ID.	String
ANQP_Cap_List	GAS ANQP for the Capability list.	Integer 0 = Don't include (Default) 1 = Include
ANQPQuery_ID	If FrameName parameter has ANQPQuery, then this parameter will set ANQP element definition per IEEE	String NeighborReportReq QueryListWithCellPref
APChanRpt	AP Channel Report	String Format is "#-#". For example, to set the channels to 36 and 48, 36_48
AskForLCI	Request for LCI of the AP in ANQP query	Integer 0 = Do not include 1 = Include
AskForLocCivic	Request for Civic Location of the AP in ANQP query	Integer 0 = Do not include 1 = Include
AskForPublicIdentifierURI-FQDN	Requests Location Public Identifier URI/FQDN of AP in ANQP query	Integer 0 = Do not include 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
BTMQuery_Reason_Code	BSS Transition Query Reason Used for MBO. 0 to 20 are valid	Integer

Name	Description	Value
Cand_List	Used to set Preferred Candidate List included bit in Request mode field. (From IEEE 9.6.14.9) candidate into the candidate list	Integeger 0 = Don't include 1 = Include
CapReNegotiateCodecProfile	Renegotiate with the requested codec and profile. The format is H264-<profile> and H265-<profile>	String Example: H264-0 means CBP
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
Channel	Channel number	Short Integer 0- 255
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.
CTT_Dialog_Token	CTT only parameter. Dialog token number in frame.	Integer
CTT_FU_Dialog_Token	CTT only parameter. Follow Up Dialog Token.	Integer
CTT_ToD	CTT only parameter. ToD value to be configured.	Integer 32-bit value
CTT_ToA	CTT only parameter. ToA value to be configured.	Integer 32-bit value
CTT_ToD_Max_Err	CTT only parameter. CTT_ToD_Max_Err value to be configured.	Integer 8-bit
CTT_ToA_Max_Err	CTT only parameter. CTT_ToA_Max_Err value to be configured.	Integer 8-bit
Dest	Destination MAC address (BSSID or STA MAC address)	String
Dest_MAC. DestMAC	Destination MAC address (BSSID or STA MAC address) For 60GHz and Location, 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
DevRole	Device role.	String AP STA P2P
DevType	Specifiies the device type.	String P-sink, Dual Source
Disassoc_Timer	Time for the disassoc, number of beacon count	Integer

Name	Description	Value
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 ARPProbe ARPAnnounce NeighSolicitReq
	For WFD: Frame template to be used for injector or transmission that contains the necessary frame data	String WFD_ProbeReq WFD_ProbeReqTdls 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
	For Location: Frame information transmitted from the device	String ANQPQuery NeighReportReq RadioMsntReq
	For Timing measurement TM Request: Timing Measurement Request frame TM_Action: Timing Measurement Action frame	String TM Request TM_Action
	For MBO: ANQPQuery: ANQP query request BTMQuery: BTM action frame Query BTMReq: BTM action frame Request BcnRptReq: Beacon Report Request WNM_Notify: WNM Notification Request disassoc: Disassociation Frame	String ANQPQuery BTMQuery BTMReq BcnRptReq WNM_Notify disassoc
	For WPS: Probe request	String Probereq
HS_Cap_List	HS Capability list	Integer 0 = Don't include (Default)

Name	Description	Value
		1 = Include
Icon_Request	Icon Request includes the filename to be requested	String Ex: 1234_wifi.png
Interface	Interface ID	String For APs: 24G 5G For STAs (example): wlan0
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
LocationSubject	Location Subject field value in Measurement Request element	Integer 0 = Location subject Local 1 = Location subject Remote 2 = Location subject Third party
MaxAgeSubelem	Maximum Age Subelement in LCI Request	Integer 1 = Included 0 = Not included 2 = Included and Maximum Age value set to 65535
MeaDur	Measurement Duration in milliseconds. The minimum is 20 ms.	Integer
MeaDurMand	Measurement Duration Mandatory	Boolean
MeaMode	Measurement Mode	String ACTIVE TABLE PASSIVE
Message	Type of RTSP message	String RTSP_M14_Req RTSP_M15_Req
MinAPcount	Configures the Minimum AP count	Short Integer
MsntType	Configure Measurement Type of the Measurement Request element. For example: a value of 4 means that two Measurement Request elements shall be included in the transmitted frame for Measurement Types LCI and Location Civic.	Integer 0 = Reserved 1 = FTMRangeReq (16) 2 = LCI (8) 3 = Location Civic (11) 4 = LCI and Location Civic
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

Name	Description	Value
		0 = Don't include (Default) 1 = Include
Name	A device name. This parameter is required for an AP Control agent implementation.	String Ex: MarvellLOC/CiscoLOC
NSS	Indicates the number of spatial streams the sending STA is able to receive	Short integer
NumOfRep	Number of Repetitions fields in Radio Measurement Request frame	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
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 LOC IoTLP TM MBO WPS
RandInterval	Configures the Randomization Interval	Integer 0 - Any 20
Reason	Reason code for TDLS teardown frame. This option is used with frame type 'Teardown'.	Integer
Remove_DMS_filter	Removes the specified DMS_filter	String
Request_Mode	Mode of request code	String DisAssoc
RegClass	Operating Class	Short Integer 8 115
ReqInfo	Information Codes Number separated by '_'	String For example: 0_48_54…
RptDet	Reporting Detail	Boolean 0 1
RTSPMsgType	Type of RTSP message	String

Name	Description	Value
		GET_PARAMETER PAUSE PLAY TEARDOWN TRIGGER-PAUSE TRIGGER-PLAY TRIGGER-TEARDOWN SET_PARAMETER SETUP
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
SSID	SSID	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
TM_Frame_Cnt	Indicates the number of TM action frames to be transmitted.	Integer Range 1 to 20
TM_Trigger	Trigger value for TM Request frame	Integer 0 = Reciver to stop TM operation 1 = Receiver to start TM operation
TransportType	Type of transport	String TCP UDP
Type	Frame type to be sent	String discovery setup teardown

Name	Description	Value
		chswitchreq2
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
WNM_Notify_Element	If FrameName parameter has WNM_Notify, then this parameter will set sub-element of WNM Notification request frame	String NonPrefChanReport CellularCapabilities
WSC_IE_Fmt_Code	The code defines the number of WSC IEs to be created within the frame as specified by FrameName, e.g. Probe Request, and how TLV data is partitioned into each IE. 1 - Create two IEs. The first one includes only the first octet of the TLV data and the rest of it is in the second IE.	Integer 1

Return Values

Name	Description	Value
DMS_filter	Only returned when performing the add_dms_filter action	String

Examples

For Wi-Fi Test Suite Sniffer Injector:

```
UCC:dev_send_frame,Name,<AP_Name>,interface,wlan0,Program,PMF,FrameName,saquery,Protected,CorrectKe
y,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,desti
nation,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
```

Location examples:

```
UCC: dev_send_frame,program,LOC,NAME,ap1loc,interface,5G,DestMac,
AA:BB:CC:DD:EE:FF,FrameName,RadioMsntReq,MsntType,1,RandInterval,0,MinAPcount,3,MaxAgeSubelem,0,Num
OfRep,0
CA:status,RUNNING
CA:status,COMPLETE
UCC:
dev_send_frame,program,LOC,NAME,abcLOC,interface,wlan0,DestMacAA:BB:CC:DD:EE:FF,FrameName,RadioMsnt
Req,MsntType,1,RandInterval,0,MinAPcount,2
CA:status,RUNNING
CA:status,COMPLETE
UCC: dev_send_frame,program,LOC,interface,wlan0,DestMac, AA:BB:CC:DD:EE:FF,
FrameName,NeighRepReq,MsntType,4
CA:status,RUNNING
CA:status,COMPLETE
UCC: dev_send_frame,program,LOC,interface,wlan0,DestMac,AA:BB:CC:DD:EE:FF,
FrameName,ANQPQuery,AskForPublicIdentifierURI-FQDN,1
CA:status,RUNNING
CA:status,COMPLETE
UCC: dev_send_frame,program,LOC,interface,wlan0,DestMac,AA:BB:CC:DD:EE:FF,
FrameName,ANQPQuery,address3,ff:ff:ff:ff:ff:ff,AskForPublicIdentifierURI-FQDN,1
CA:status,RUNNING
CA:status,COMPLETE
```

IoTLP example 1:

```
UCC: dev_send_frame,program,IoTLP,interface,wlan0,add_DMS_filter,1
CA:status,RUNNING
CA:status,COMPLETE,DMS_filter,42
```

IoTLP example 2:

```
UCC: dev_send_frame,program,IoTLP,interface,wlan0,remove_DMS_filter,42
CA:status,RUNNING
CA:status,COMPLETE
```

TM example:

```
UCC: dev_send_frame, program, TM, interface, wlan0, Dest_MAC, 00:11:22:33:44:55, FrameName,
TM_Action, TM_Frame_Cnt,10, TM_Frame_Periodicity,125
CA:status,RUNNING
CA:status,COMPLETE
```

MBO example:

```
UCC: dev_send_frame,interface,$Interface,Program,MBO,Framename,BcnRptReq,NAME,xyz,Dest_MAC,
00:11:22:33:44:55,RegClass,115,Channel,0,RandInt,50,MeaDur,20,MeaMode,1,BSSID,FF:FF:FF:FF:FF:FF,SSI
D,wi-fi,RptCond,0,RptDet,1,MeaDurMand,0,APChanRpt,36_48,ReqInfo,0_48_70_54_221!DEFAULT
```

```
CA:status,RUNNING
CA:status,COMPLETE
```

10.10 DEV_SET_CONFIG

This command is used to configure the device with its configuration parameters.

Dependencies

None.

Parameters

Name	Description	Value
Backhaul	Set a single backhual interface to Ethernet or Wi-Fi. For Wi-Fi, a RUID associated with its frequency band (2.4GHz, 5GHz Low, 5GHz High) is provided. For Ethernet, the device should return status,COMPLETE until it completes Ethernet onboarding.	String Eth RUID
BSS_Info1	BSS initialization data used by controller to configure agent's fronthaul radio. The value is a list of fields separated by space that stands for WTS_REPLACE_DEST_ALID, operating class, SSID, authentication type (WPA2-PSK), encryption type(AES-CCMP), network key, bit 6 of Multi-AP IE's extension attirbute, bit 5 of Multi-AP IE's extension attirbute. To clear the BSS info stored for a specific operating class, only the device ALID and operating class are retained.	String WTS_REPLACE_DEST_ALID 8x Multi-AP-2G-1 0x0020 0x0008 maprocks1 0 1 WTS_REPLACE_DEST_ALID 8x
BSS_Info2		String WTS_REPLACE_DEST_ALID 8x Multi-AP-2G-2 0x0020 0x0008 maprocks2 0 0
BSS_InfoN		String WTS_REPLACE_DEST_ALID 8x Multi-AP-2G-n 0x0020 0x0008 maprocksn 0 0
Name	Device name.	String For Ex.: dev1
Program	Program name	String MAP

Return values

None.

Example

```
UCC: dev_set_config,name,agt1,program,map,backhaul,0x00904C2A11B7
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.

NoteThe 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_intolerent /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
```

11.13 SET_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
```

11.14 SEND_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
```

11.15 SEND_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 40Mz 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
```

11.16 SET_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
```

11.17 SET_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
```

11.18 SET_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_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
```

14.2 SERVER_EXEC_ACTION

This command is used to execute various action commands for the server.

Parameters

Name	Description	Value
Action	Specifies the desired server action	String Start Stop Restart
Device	Device type	String AAAServer
Name	Server name	String hostapd
ConfigFile	Specify configuration file for server	String Config_file_name
EAPType	EAP type	String EAPTLS EAPTTLS EAPPEAP EAPSIM EAPFAST EAPAKA EAPTLS-SuiteB
<ParameterX>	Sets additional parameters in the server configuration if needed. Multiple parameters may be specified. The parameter names are provided by the user.	String <ValueX>

Return Values

None.

Examples

```
CCG:
server_exec_action,action,start,name,hostapd,device,AAAserver,configfile,all_enabled.conf,EAPType,t
ls,parameter1,value1
CA: status,RUNNING
CA: status,COMPLETE
```

14.3 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
```

14.4 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**:

Name	Description	Value
		SUCCESS Unauthorized
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,1234567812345678123456781234567812345678123456781234567812345678123456781
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 Example 5: - 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,OSUServer,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,OSUServer,OCSP,fetch
CA:status,RUNNING
CA:status,COMPLETE
```

14.5 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 confiugred 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
```

14.6 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 theOSU 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 mumber 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
```

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	NFC action. 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. WRITE_CONFIG writes WPS configuration token onto NFC Tag when the user put the tag on the NFC interface. 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. 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. 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. 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. 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. 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.	String WRITE_SELECT WRITE_CONFIG WRITE_PASSWD READ_TAG WPS_READ_CONFIG WPS_READ_PASSWD CONN_HNDOVR WPS_CONN_HNDOVR

Return Values

5. If Operation is Write then all **return values** will be NULL/bank (zero length string).
6. If Operation is READ_TAG, then the return value must have a valid Result and GroupID.
7. 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. Otherwiese wait for the connection completion and return the valid Result and GroupID.

8. 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.
9. 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.
10. If Operation is WPS_READ_PASSWD and WPSStaAction is Enroll, then the station returns immediately with return parameters as NULL/blank (zero length string).
11. 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 clinet 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	NFC action. WRITE_CONFIG writes WPS configuration token onto NFC Tag when the user puts the tag on the NFC interface. 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. 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. 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. 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.	String WRITE_CONFIG WRITE_PASSWD WPS_READ_CONFIG WPS_READ_PASSWD WPS_CONN_HNDOVR

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 interenal registrar, but should not start the registration process (the Selected Registrar bit shoult 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.

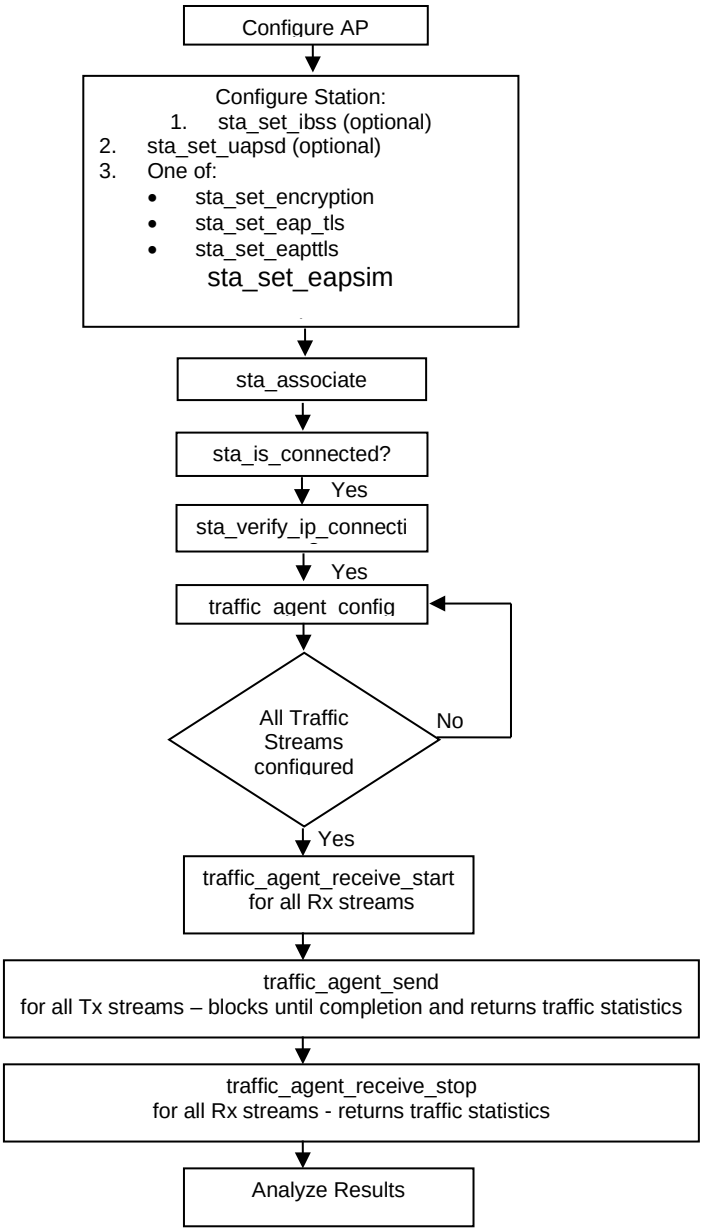

Figure 3. UCC command sequence flow diagram

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	Changed 'ASD' to 'DUT' throughout the document. Specified that the maximum command or response length is 2048 characters. Specified that the 'ssid' parameter is case-sensitive. Specified that the 'String' data type is represented as a String. Removed command '*noop*' and added '*ca_get_version*' Added *sta_get_driver_version, sta_set_eapttls, sta_set_eapsim, sta_set_peap, sta_set_ibss.* Removed the *'ping'* profile in *traffic_agent_config.*
0.4	03/02/2006	Removed *traffic_agent_start* and added *traffic_agent_send* and *traffic_agent_receive_start.* Removed *traffic_agent_stop* and added *traffic_agent_receive_stop.* Removed '*protocol*' parameter from *traffic_agent_config.* Added *traffic_agent_reset.* Added Appendix B that has a flow chart to show the sequence in which the traffic commands are used. Added a note that communication between the Control Agent and DUT should be out-of-band rather than in-band.
0.5	03/03/2006	Removed *traffic_agent_status* Removed *traffic_agent_get_results*. Modified *traffic_agent_receive_start* and *traffic_agent_receive_stop* to accept multiple streamIDs using a space-separated list. Modified *traffic_agent_send* to return results for multiple streamIDs using a space-separated list. Renamed *sta_get_driver_version* to *sta_get_device_info*. Modified it to return vendor, model and software version.
0.6	03/08/2006	Added optional parameter '*startDelay*' in *traffic_agent_config* for '*IPTV send'*. Added command table in Appendix A for PC Traffic Agent.
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 *traffic_agent_config* for '*send*' profiles '*File_Transfer*', '*Multicast*' and '*Transaction*' too.
0.9	03/22/2006	Added mandatory parameter '*source*' in *traffic_agent_config* for '*receive*' profiles '*File_Transfer*', '*Multicast*', '*IPTV*' and '*Transaction*'.
1.0	03/23/2006	Renamed mandatory parameter '*port*' to '*destinationPort*' and added '*sourcePort*' in *traffic_agent_config* for both '*send*' and '*receive*' of all profiles.
1.1	04/04/2006	Renamed *sta_get_device_info* to *device_get_info*. Added *device_list_interfaces.* Added mandatory parameter '*interface'* parameter to all the sta_* functions. Added *sta_get_info* to get vendor specific information about a specified interface. Added *sta_set_ip_config*.
1.2	04/04/2006	Changed column heading from 'Interface Status' to 'Device Status' in table of API commands. In *sta_set_ip_config* changed parameters '*ip*', '*mask*' and '*dns'* from mandatory to optional. Added optional parameter '*defaultGateway'*
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 *sta_set_ip_config* 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 reinvoke 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 reinvoke 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_EAPTTLS 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. Aslo 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
		21. Added WLAN_TAG parameter and examples to AP_SET_RADIUS. 22. Added WMM and HS2-R2 examples to AP_RESET_DEFAULT. 23. Added Minor_Code parameter to AP_DEAUTH_STA. 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. 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. 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. 28. Added optional parameters BSSID, DestMAC, Type, AfterTimeStamp, ClientMAC, DataLen to SNIFFER_FRAME_CHECK. 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, Framecount, WNM_DisassocTimer, BSSLoad_CU, SeqNo to FieldName. Added optional parameters Message, StationID, Name. 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_InfoID, ANQP_SubType, bidiMAC, DSValue. 31. Added parameter TransFile to SINFFER_FETCH_FILE. 32. Added optional parameters CertReEnroll, SerialNo, ProvisioningProto, TrustRootCACert, InterCACert, OSUServerCert, Name, Issuing_Arch, Timeout to SERVER_SET_PARAMETER. 33. Added Server APIs for HS2. 34. Replaced command matrix with Command Matrix spreadsheet. 35. Removed obsolete commands: AP_SEND_FRAME, AP_SET_FEATURE, STA_SET_EAPAKAPRIME, STA_GET_KEY

Revision Number	Date	Changes
9.1.0	02/24/2016	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	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		3. Corrected time units for NoA parameters in STA-SET_P2P. 4. For STA_SET_WMM, added the value 60Hz to Group, and DMGADDTS, PTADDTS to Action. Added new parameters. 5. For TRAFFIC_AGENT_CONFIG, added new parameter transProtoType. 6. For TRAFFIC_AGENT_SEND, added new parameter txActFrames.Corrected txPayloadBytes to long integer. 7. For TRAFFIC_AGENT_RECEIVE_STOP, added new parameter txActFrames. 8. For STA_PRESET_TESTPARAMETERS, added values 60GHz, WPS to Program. Added new parameters WPSVersion, WPSTestAttributes, WSCIEFragment, WSCEapFragment, WSCState, WPS. Added value "on" for parameter Powersave. 9. 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. 10. For STA_SEND_ADDBA, added more detail for parameter TID. 11. For STA_SET_SECURITY, added OPEN as a value for Type, added value AES-GCMP for EncpType. 12. For STA_RESET_DEFAULT, added values 60GHz, WPS to Prog. Added new parameters Type, DevRole, Band and new examples. 13. For STA_SET_RF_FEATURE, added value 60GHz to Prog. Added new parameters CBAPOnly, ExtSchIE, AllocType, AllocID, PercentBI, SrcAID, DestAID and new examples. 14. For STA_GET_PARAMETERS, added value 60GHz to Program. Added value AID to the parameters Parameter, added new return value AID. 15. For STA_SET_POWERSAVE, added new value uncscheduled to parameter Powersave, added new parameter Program, and new example. 16. Added new commands TRAFFIC_AGENT_VERSION. 17. Added depreciated command STA_SEND_FRAME for VE. 18. Added commands AP_SEND_BCNRPT_REQ, AP_SEND_TSMRPT_REQ, AP_SEND_LINK_MEA_REQ, AP_SEND_BSSTRANS_MGMT_REQ for VE. 19. 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. 20. For AP_SET_SECURITY, added AES-GCMP as a value for Encrypt. 21. For AP_RESET_DEFAULT, added value 60GHz to Program. Added 60GHz text to parameters Type and new examples.

Revision Number	Date	Changes
		22. For AP_GET_MAC_ADDRESS, added values to parameter Interface, and new examples. 23. For AP_SEND_ADDBA, added more detail for parameter TID. 24. Added new text and examples to AP_WPS_READ_PIN. 25. For SINFFER_CONTROL_START and SINFFER_CONTROL_STOP and SNIFFER_GET_FIELD_VALUE, added value 60GHz to Program. 26. 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. 27. Added new command START_WPS_REGISTRATION. 28. For STA_SET_11n, changed values for TXSP_Stream and RXST_Stream from integer to string. 29. Removed unsed 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. 30. 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. 31. Moved WPS-NFC commands to a separate chapter.
9.3.0	01/25/2017	1. Moved DEV_GET_INFO to section 10. 2. Added new parameters to STA_EXEC_ACTION for Location: ASAP, AskForLCI, AskForLocCivic, BurstDuration, BurstsExponent, DestMac, FormatBwFTM, FTMsperBurst, Trigger, ErrMsg. Added usage examples for Location. 3. Added values for parameter Network_Role in the command STA_MANAGE_SERVICE> 4. Added new parameters to STA_PRESET_TESTPARAMETERS for Location: ANPQ, Interworking, RadioMsnt, RMEnabledCapBitmap. Added usage examples for Location. 5. Removed parameters Interface and MinorCode from AP_DEAUTH_STA. 6. Added new parameters to AP_GET_MAC_ADDRESS for Location: WLAN_TAG. Added usage examples for Location. 7. Added new parameters to AP_SET_WIRELESS for Location: AdvCoLocBSSIDs, ANPQ, FTMinBSSIDInfo, GAS_CB_Delay, InfoZ, Interworking, LCI, LocCivicAddr, LocCivicAddrLength, LocCivicAddrType, MultiBurstRequest, NeighAPBSSID, OpChannel, PHYType, PublicIdentifierURI-FQDN, RadioMsnt, RMEnabledCapBitmap, URI-FQDNdescriptor, WideBwChnl. Added return value ErrMsg. Added usage examples for Location. 8. Added new command SNIFFER_CALC_PARTIAL_TSF. 9. Removed parameter FieldName from SNIFFER_CHECK_DSCV_WINDOW and added ROLE. 10. Added new parameters to SNIFFER_CONTROL_FIELD_CHECK for Location: AssocStatus, ExistMaxAge, ExtChnHt20, ExtChnHt40u, ExtChnHt40d, FrmReTx, FTMparams, FTMreqTrigger, FTMstatusInd, FTMvalue, FTMburstsExponent, FTMburstsDuration, FTMasapCapable, FTMasap, FTMsPerBurst, FTMformatBw, FTMburstPeriod, FTMdialogToken, FTMfollowupDialogToken, FTMtodErr, FTMtoaErr, FTMmaxTodErr, FTMmaxToaErr, FTMtodNotCont, FTMtoaNotCont, FTMrepLate, FTMrepIncapable, FTMrepRefused, FTMrepRangeErrBssid, FTMrepRangeRngBssid, FTMrangeReport, HtGreenField, HT-SIG2_STBC, LocCivicReport, LCIreport, LocCivicRequest, LCIrequest, LocConfigReport_Sub, LocCivicReport_Sub, MaxAgeSubElem, MinAPcnt, minDeltaFTM, NrrBSSID, NrrChan, NrrFTMbssidInfoNegative, NrrPhyTypeNegative, PartialTsfTimer, PtsfNoPref, PubIdUFDesc, PubIdUFName, RandInterval, RMlocSubject, RMmodeEnable, RMrepetitions, RMreqMode, RMtoken, ReservedBit7, ReservedBits_48-49, RetransAllowed, RMdialogToken, RMEC_IE, RMEC_IE_Bit3, RMEC_IE_Bit12,, RMEC_IE_Bit34, RMEC_IE_Bit35, TSFSyncInfo. 11. Added new parameters to SNIFFER_CONTROL_FIELD_CHECK_ALL for Location: FTMdialogToken. 12. Added new parameters to DEV_SEND_FRAME for Location: Address3, AskForLCI, AskForLocCivic, AskForPublicIdentiferURI-FQDN, LocationSubject, MaxAgeSubelem, MinAPcount, MsntType, Name, NumOfRep, RandInterval. Added usage examples for Location.
10.0.0		Supporting eNPD Display R2: 1. Added DisplayR2 as supported program for STA_GET_PARAMETER.

Revision Number	Date	Changes
		2. Added new parameter Oper_Chn to STA_MANAGE_SERVICE. 3. Added new parameters CodecProfile, mDNS_Disc, mDNS_Role, SessionState, CTT_Sequence<#> to STA_PRESET_TESTPARAMETERS. Added examples for CTT and DisplayR2. 4. Added new parameter MaxTCPSegmentSize to STA_SET_P2P. 5. Added new parameters R2ConnectionType, ServiceType, InstanceName to START_WFD_CONNECTION. Also added new examples using these parameters. 6. Added new command DEV_EXEC_ACTION. 7. Added new parameters CapReNegotiateCodecProfile, TransportType to DEV_SEND_FRAME. Supporting eNPD IoT Low Power: 1. Added IoTLP as supported program for STA_PRESET_TESTPARAMETERS and added new parameters DMS, RSN. 2. Added IoTLP as supported program for STA_RESET_DEFAULT and IoTLP examples. 3. Added IoTLP as supported program for STA_SET_POWER_SAVE, AP_RESET_DEFAULT. 4. Added IoTLP as supported program for STA_SET_RFEATURE and added new parameters Mgmt_Data_TX_Resp_Frame, KeepAlive, FrameName and examples. 5. Added IoTLP as supported program for AP_SET_WIRELESS and added new parameters BSS_max_idle, BSS_max_Idle_period, BSS_Idle_Protection_options, Proxy_ARP, DMS. 6. Added new parameters to SNIFFER_CONTROL_FIELD_CHECK: BSS_Max_Idle_IE, BSS_Max_Idle_period, BSS_Max_Idle_option, EC_IE_Bit26, DMS_Resp_TCLAS, DMS_Resp_type, DMS_Req_TCLAS, DMS_Req_type. 7. Added DMS_ID, AssocID values to parameter FieldName in SNIFFER_GET_FIELD_VALUE. 8. Added DMS_Response, DMS_Request values to parameter ActionCode, and data_mgmt_pspoll, data_mgmt values to parameter FrameName in SNIFFER_CONTROL_FILTER_CAPTURE. 9. Added IoTLP as supported program for DEV_SEND_FRAME and added new parameters Add_DMS_filter, Remove_DMS_filter. Added return value DMS_filter. Supporting eNPD Timing Measurement: 1. Added TM as supported program and P2P as DevRole for STA_RESET_DEFAULT. 2. Added TM as supported program and P2P as DevRole for STA_SET_WIRELESS. 3. Added TM as supported program for AP_RESET_DEFAULT. 4. Added TM values to parameter FrameName for DEV_SEND_FRAME. Also added new parameters TM_Frame_Cnt, TM_Frame_Periodicity, TM_Trigger, CTT_Dialog_token, CTT_FU_Dialog_token, CTT_ToD, CTT_ToA, CTT_ToD_Max_Err, CTT_ToA_Max_Err.
10.1.0	2017-08-28	Added support for Wi-Fi Agile Multiband. 1. Added parameters Assoc_Disallow, BSS_Transition, Cellular_Data_Cap, Ch_Op_Class, Ch_Pref, Ch_Pref_Num, Ch_Reason_Code, Roaming to STA_PRESET_TESTPARAMETERS. 2. Added parameters Ch_Op_Class, Ch_Pref_Num, Ch_Pref, Ch_Reason_Code, Nebor_BSSID, Nebor_Op_Class, Nebor_Op_Ch, Nebor_Pref to STA_SET_RFEATURE. 3. Added parameters Assoc_Delay, Assoc_Disallow, BSS_Term_TSF, BSS_Term_Duration, BTMReq_DisAssoc_Imnt, BTMReq_Term_Bit, Disassoc_Timer, Nebor_BSSID, Nebor_Op_Class, Nebor_Op_Ch, Nebor_Pref to AP_SET_RFEATURE. 4. Added parameters Cellular_Cap_Pref, FT_BSS_LIST, Reg_Domain to AP_SET_WIRELESS. 5. Moved CA_GET_VERSION to chapter 10. 6. Added parameters ANQPQuery_ID, APChanRpt, BTMQuery_Reason_Code, Cand_List, Channel, Disassoc_Timer, MeaDur, MeaDurMand, MeaMode, Request_Mode, RegClass, ReqInfo. RptDet, SSID, WNM_Notify_Element to DEV_SEND_FRAME.
10.2.0	2017-09-	Added CAPI command STA_ACCEPT_P2P_INVITATION_REQ. Added support for Wi-Fi Aware R2. 1. Added new methods to STA_EXEC_ACTION. 2. Added new values to Parameter in STA_GET_PARAMETER.
10.4.0	2017-12-06	1. Added notes to AP_RESET_DEFAULT, STA_RESET_DEFAULT.

Revision Number	Date	Changes
		2. Modified STA_SET_SECURITY to include enterprise parameters and configure multiple wireless profiles. 3. Modified AP_SET_RFEATURE to include WPS related, AP transmit power and uplink/downlink available capacity related parameters. 4. Modified START_WPS_REGISTRATION and add CAPI AP_SET_WPS_PBC. Remove WPS config from AP_SET_RFEATURE. 5. Update START_WPS_REGISTRATION and add Auto channel to AP_SET_WIRELESS. 6. Added STA-CFON parameters to STA_RESET_DEFAULT and STA_SET_WIRELESS. 7. Added CAPI AP_DISCONNECT. 8. Update AP_SET_WIRELESS, STA_PRESET_TESTPARAMETERS, AP_SET_SECURITY with paramters for FILS SK. 9. Added support for 80+80 bandwidth in AP_SET_WIRELESS. 10. Added DataPPDUduration and AirTimeFraction for AP's ESP element configuration to AP_SET_WIRELESS. 11. Added Station Count, Channel Utilization and AAC for AP's BSS Load element configuration to AP_SET_WIRELESS. 12. Any stateful proprietary AP selection algorithms on STAUT are reset (not disabled). Updated STA_PRESET_TESTPARAMETERS. 13. Test case 5.7.2 step 4 CAPI to not send disassoc or deauth frame upon turning off the radio using AP_SET_WIRELESS. 14. DHCPv4RapidCommit argument removed from AP_SET_WIRELESS. 15. BTM Request frame for FILS roaming added to AP_SET_RFEATURE and DEV_SEND_FRAME. 16. Added parameters to AP_SET_WIRELESS for ESP test cases.
10.5.0	04-09-2018	Added new commands and parameters for Wi-Fi Easy Connect and WPA3.
10.6.0	06-13-2018	Added new commands and parameters for Multi-AP. New commands: DEV_RESET_DEFAULT, DEV_SET_CONFIG, DEV_GET_PARAMETER, DEV_SEND_1905
10.7.0	07-06-2018	Added new command AP_GET_PARAMETER and parameters for Wi-Fi CERTIFIED WiGig Release 2.